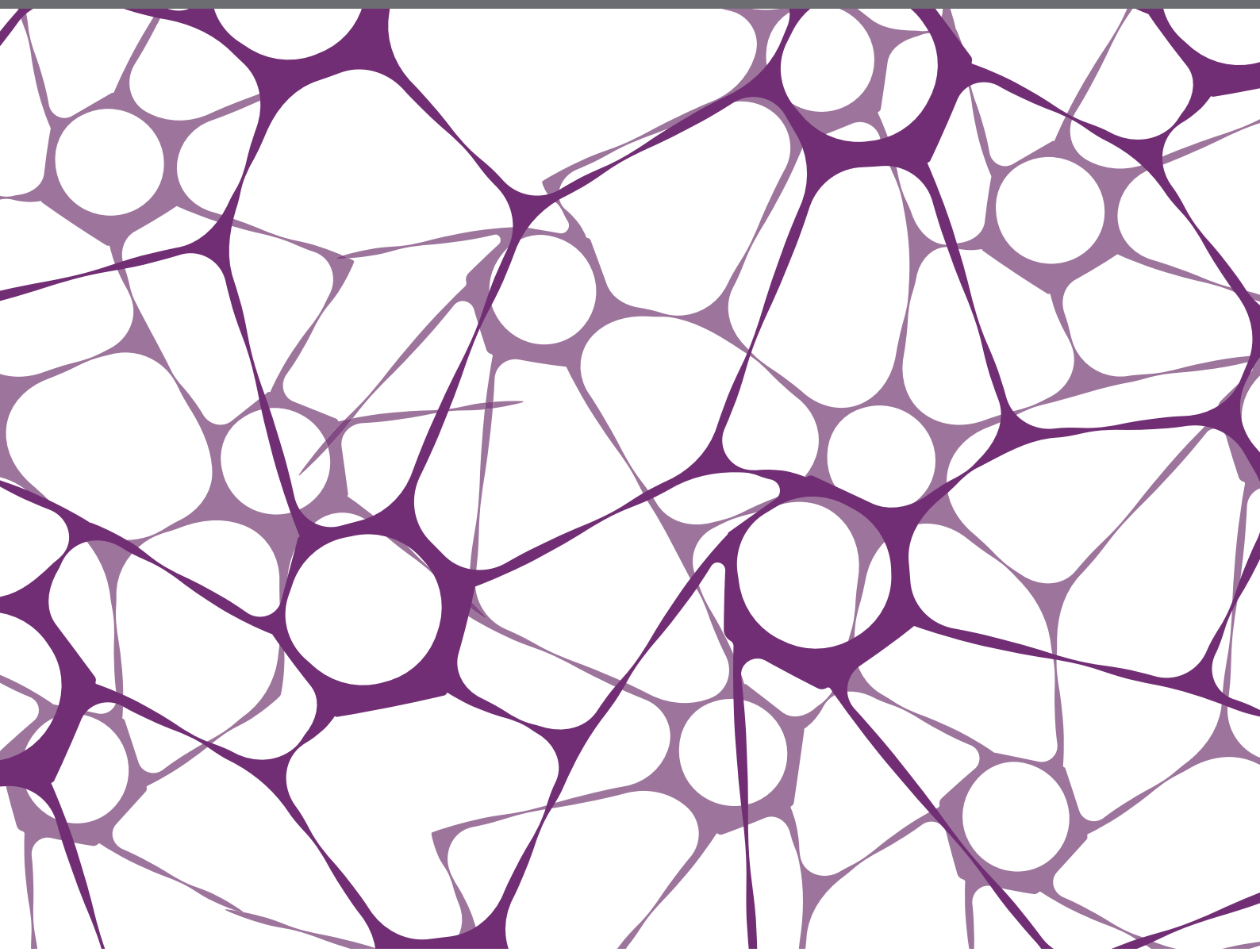
VIRTUAL MATERIALS DESIGN

EDITED BY: Norbert Huber, Surya R. Kalidindi, Stefan Blügel and Wolfgang Wenzel

PUBLISHED IN: Frontiers in Materials



frontiers Research Topics

ISSN 1664-8714
ISBN 978-2-88976-677-2
DOI 10.3389/978-2-88976-677-2

About Frontiers

Frontiers is more than just an open-access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers Journal Series

The Frontiers Journal Series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the Frontiers Journal Series operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to Quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews.
Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the Frontiers Journals Series: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area! Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers Editorial Office: frontiersin.org/about/contact

VIRTUAL MATERIALS DESIGN

Topic Editors:
Norbert Huber, Helmholtz-Zentrum Hereon, Germany
Surya R. Kalidindi, Georgia Institute of Technology, United States
Stefan Blügel, Quantum Theory of Materials (PGI-1 / IAS-1), Peter Grünberg Institute, Julich Research Center, Helmholtz Association of German Research Centres (HZ), Germany
Wolfgang Wenzel, Karlsruhe Institute of Technology (KIT), Germany

Citation: Huber, N., Kalidindi, S. R., Blügel, S., Wenzel, W., eds. (2022). Virtual Materials Design. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-88976-677-2

Table of Contents

frontiers in Materials

ORIGINAL RESEARCH
published: 13 December 2021
doi: 10.3389/fmats.2021.760602

Efficient Fitting of 3D Tessellations to Curved Polycrystalline Grain Boundaries

Lukas Petrich[1]*, *Orkun Furat*[1], *Mingyan Wang*[2], *Carl E. Krill III*[2] and *Volker Schmidt*[1]

[1]Institute of Stochastics, Faculty of Mathematics and Economics, Ulm University, Ulm, Germany, [2]Institute of Functional Nanosystems, Faculty of Engineering, Computer Science and Psychology, Ulm University, Ulm, Germany

The curvature of grain boundaries in polycrystalline materials is an important characteristic, since it plays a key role in phenomena like grain growth. However, most traditional tessellation models that are used for modeling the microstructure morphology of these materials, e.g., Voronoi or Laguerre tessellations, have flat faces and thus fail to incorporate the curvature of the latter. For this reason, we consider generalizations of Laguerre tessellations—variations of so-called generalized balanced power diagrams (GBPDs)—that exhibit non-convex cells. With as many as ten parameters for each cell, it is computationally demanding to fit GBPDs to three-dimensional image data containing hundreds of grains. We therefore propose a modification of the traditional definition of GBDPs that allows gradient-based optimization methods to be employed. The resulting reduction in runtime makes it feasible to find approximations to real experimental datasets. We demonstrate this on a three-dimensional x-ray diffraction (3DXRD) mapping of an AlCu alloy, but we also evaluate the modeling errors for simulated data. Furthermore, we investigate the effect of noisy image data and whether the smoothing of image data prior to the fitting step is advantageous.

Keywords: polycrystalline material, tessellation, generalized balanced power diagram, gradient-based optimization, image noise

OPEN ACCESS

Edited by:
Norbert Huber,
Helmholtz-Zentrum Hereon, Germany

Reviewed by:
Napat Vajragupta,
Ruhr University Bochum, Germany
Fabrice Barbe,
Insa Rouen Normandie, France

***Correspondence:**
Lukas Petrich
lukas.petrich@uni-ulm.de

Specialty section:
This article was submitted to Computational Materials Science, a section of the journal Frontiers in Materials

Received: *18 August 2021*
Accepted: *02 November 2021*
Published: *13 December 2021*

Citation:
Petrich L, Furat O, Wang M, Krill III CE and Schmidt V (2021) Efficient Fitting of 3D Tessellations to Curved Polycrystalline Grain Boundaries. Front. Mater. 8:760602. doi: 10.3389/fmats.2021.760602

1 INTRODUCTION

The grain boundaries of polycrystalline materials play an important role in many different phenomena, ranging from fundamental processes like grain growth and extending to applied scenarios like the degradation of electrodes in lithium-ion batteries. In many such cases, the investigation and modeling of grain boundaries presupposes that their locations can be represented precisely. For this purpose, tessellations have proven to be a powerful tool, as they provide a partitioning of space into disjoint subsets called cells. For example, the representation of a material's microstructure by means of tessellations can be utilized for the analysis of microstructure-property relationships (Raabe, 1998; Westhoff et al., 2018). For the latter, realistic "virtual polycrystals" generated by parametric stochastic models for these tessellations are particularly helpful (see, e.g., Allen et al., 2021). A prominent tessellation type in materials science is the Laguerre tessellation (Lautensack and Zuyev, 2008), which is a generalization of the well-known Voronoi tessellation (Møller, 1994; Okabe et al., 2000). It is therefore not surprising that the fitting of Laguerre tessellations to experimental data has already received much attention. For example, in Bourne et al. (2020); Petrich et al. (2019); Quey and Renversade (2018) the problem of finding good representations for statistical data, such as grain volumes and centroids, is discussed. Of

particular interest is the description of 3D image data, e.g., from 3D electron backscatter diffraction (EBSD) or 3D x-ray diffraction (3DXRD) microscopy, which was studied in Liebscher (2015); Quey and Renversade (2018); Spettl et al. (2016). Additional details regarding the method proposed in Spettl et al. (2016) are given in **Section 2.4.1**. A major drawback of the Laguerre tessellation, however, is the fact that its facets are planar and therefore apply only to grains having nearly flat boundaries. This is unacceptable when it comes to the investigation of curvature-related phenomena like grain growth. In this case, other tessellation models—often generalizations of the Voronoi/Laguerre tessellations—have been proposed; we refer to Altendorf et al. (2014); Šedivý et al. (2018) for an overview. Heuristics for fitting some of these tessellation models are described in Altendorf et al. (2014); Teferra and Graham-Brady (2015). A quite general tessellation model, the so-called *generalized balanced power diagram* (GBPD), is introduced in Alpers et al. (2015), in which a fitting procedure based on a (very high-dimensional) linear optimization is also proposed. A different fitting method, again relying on optimization, is described by Šedivý et al. (2016). Moreover, a completely different approach is taken in Teferra and Rowenhorst (2018), where closed formulas for approximating GBPDs are presented. The latter two methods are discussed in detail in **Sections 2.4.2** and **4.2**.

The major goal of the present paper is to propose a fitting method that works well for GBPDs and other distance-based tessellations. Taking advantage of efficient gradient-descent optimization, the new approach aims to achieve a goodness of fit similar or better than that of other techniques—but with much shorter computational runtime. This is investigated on 3DXRD mapping data obtained from a sample of an AlCu alloy, but we also evaluate the modeling errors for simulated data. Note that the fitting method presented here is also applicable to image data obtained by techniques other than 3DXRD, such as 3D EBSD (Zaefferer et al., 2008; Schwartz et al., 2009; Burnett et al., 2016). Furthermore, the robustness with respect to noisy image data is studied, and the question is posed whether the smoothing of grain boundaries prior to tessellation fitting—as is routinely carried out—actually improves the fit. Even though different tessellation models were fitted to the datasets, the topic of model selection is not discussed; for the latter, we refer to Šedivý et al. (2018). The present paper extends a previous version of the fitting algorithm originally described in Furat et al. (2021) by considering more general types of tessellations and a thorough analysis of the goodness of fit for different datasets.

2 MATERIALS AND METHODS

In this section we describe the materials and methods used in the present paper. These topics include the 3DXRD image data described in **Section 2.1**, the definitions of various tessellation models in **Section 2.2**, a procedure for gradient descent-based tessellation fitting in **Section 2.3** (originally introduced in Furat et al. (2021)), and two further methods from the literature for the gradient-free fitting of tessellations to image data (**Section 2.4**).

2.1 Description of 3DXRD Image Data

One of the main goals of the present paper is to describe a procedure for finding accurate parametric representations of real, experimental image data. To that end, a 3D microstructural mapping was carried out on a 1.4 mm-diameter cylinder of Al-5 wt%Cu, which was cut out of a cold-rolled plate (50% thickness reduction) that had been subsequently homogenized at 500°C for 24 h in air. The shape of individual grains in the specimen and the location of internal grain boundaries were revealed by 3DXRD microscopy measurements, performed at beamline BL20XU of the Japanese synchrotron radiation facility SPring-8 using a monochromatic beam of 32 keV x-rays (Poulsen, 2004). For 10 min prior to this room-temperature mapping, the specimen was subjected to a heat treatment at 575°C in air, which results in a liquid AlCu phase of approximately 2 vol% wetting the boundaries between the solid, aluminum-rich grains. Owing to the simultaneous presence of two phases, the resulting evolution of the sample's microstructure is classified as *Ostwald ripening* (Wang and Glicksman, 2007). Once the sample is removed from the furnace, however, the liquid layer crystallizes and the growth/shrinkage of individual grains ceases.

Reconstruction of the 3DXRD data followed the protocol described in Dake et al. (2016), relying on the data processing routines of Schmidt (2005, 2014). To each voxel in the reconstructed volume, the software assigns the crystal lattice orientation that generates the most complete diffraction signal, whereby "completeness" is defined as the ratio between the number of experimentally detected diffraction spots associated with the voxel in question and the number of diffraction spots that are simulated to arise from this particular voxel if it were to have the assumed orientation. The grain labels were then assigned voxel-by-voxel to the orientation having the greatest completeness value. Formally, we describe the resulting image dataset as a mapping

$$I_{\mathrm{E,raw}} : \mathcal{W}_{\mathrm{E,raw}} = \{1, \ldots, 531\} \times \{1, \ldots, 321\} \times \{1, \ldots, 321\} \rightarrow \{0, \ldots, 943\},$$

where each voxel coordinate is mapped to the corresponding grain label. Here, the label 0 is assigned to the background (i.e., voxels located outside the specimen). Each of the remaining labels is associated with one of the 943 grains.

However, with this reconstruction procedure the grain boundaries may manifest irregularities, such as local roughness, "island" voxels, zigzag shapes, or regions of fluctuating curvature (see **Figure 1A**) as a result of measurement uncertainties. These artifacts can be eliminated by treating the raw reconstruction as the initial configuration of a computational simulation of curvature-driven grain growth. If the duration of such a simulation is kept short enough, any boundary location manifesting severe curvature will tend to smoothen out, and any island voxels will be consumed by the surrounding grain, but no long-range translation of boundaries will occur—see **Figure 1B**. In the present paper, we employed 25 iterations of a 3D phase field algorithm (Krill and Chen, 2002) to reduce the roughness of grain boundaries in the raw 3DXRD

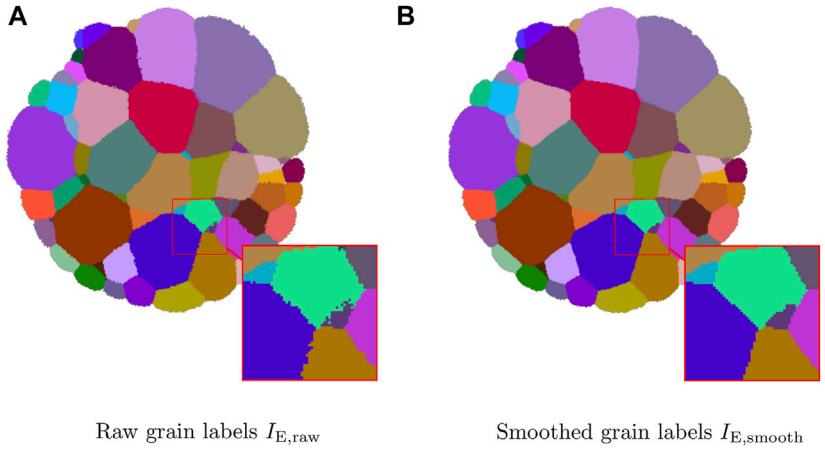


FIGURE 1 | Two-dimensional slice through the raw **(A)** and smoothed **(B)** experimental 3D image data and a magnified region showing the grain boundaries.

reconstructions. The resulting smoothed experimental image is referred to as

$$I_{\mathrm{E,smooth}} : \mathcal{W}_{\mathrm{E,smooth}} \to \{0, \ldots, 938\},$$

where $\mathcal{W}_{\mathrm{E,smooth}} = \mathcal{W}_{\mathrm{E,raw}}$. Note that some smaller grains vanished during the data smoothing procedure; consequently, $I_{\mathrm{E,smooth}}$ had fewer grains than $I_{\mathrm{E,raw}}$ (938 grains instead of 943).

2.2 Tessellation Models

In order to represent the 3D grain architecture of (measured and simulated) image data in an efficient way, we apply an optimization method to decompose the volume of interest into subvolumes using tessellations. Thus, to begin with, we briefly describe the tessellation models considered in the present paper. For additional details on tessellations in general, we refer, e.g., to Chiu et al. (2013).

Roughly speaking, a tessellation is a partitioning of space into pairwise disjoint sets, so-called cells. More precisely, a tessellation $\mathcal{T}$ in a sampling window $\mathcal{W} \subset \mathbb{R}^3$ is a countable collection of sets (cells), $\mathcal{T} = \{C_i^{\mathcal{T}} \subset \mathcal{W} : i = 1, 2, \ldots\}$, such that

1) $\mathrm{int}(C_i^{\mathcal{T}}) \cap \mathrm{int}(C_j^{\mathcal{T}}) = \emptyset$ for $i \neq j$,
2) $\cup_{i=1}^{\infty} C_i^{\mathcal{T}} = \mathcal{W}$,
3) and $\mathcal{T}$ is locally finite—i.e., $\#\{C^{\mathcal{T}} \in \mathcal{T} : C^{\mathcal{T}} \cap B \neq \emptyset\} < \infty$ for all bounded $B \subset \mathcal{W}$,

where int(·) denotes the interior of a set. Note that in this paper we consider tessellations only in a *bounded* sampling window $\mathcal{W} \subset \mathbb{R}^3$. In this case, the number of (non-empty) cells is finite and is denoted by $n_{\mathcal{T}}$.

For practical purposes, such as finding simplified representations of experimental image data, parametric tessellation models are probably the most suitable class of tessellations. The tessellation models considered in the present paper have in common that their cells are defined in terms of a distance function $d_{\mathcal{T}} : \mathbb{R}^3 \times \mathbb{G} \to \{\mathbb{R}\}$, where $\{\mathbb{G}\}$ denotes the domain of generators (i.e., a set of admissible parameters of a single tessellation cell). For a (finite) set of generators $\mathcal{G} = \{g_i\}_{i=1}^{n_{\mathcal{T}}} \subset \mathbb{G}^{n_{\mathcal{T}}}$, the i-th cell $C_i^{\mathcal{T}}$ of a distance-based tessellation $\mathcal{T}$ is given by

$$C_i^{\mathcal{T}} = \left\{x \in \mathcal{W} : d_{\mathcal{T}}(x, g_i) \leq d_{\mathcal{T}}\left(x, g_j\right) \text{ for each } j = 1, \ldots, n_{\mathcal{T}}\right\}. \tag{1}$$

For brevity, we use the notation

$$C_i^{\mathcal{T}}(x) = \begin{cases} 1 & \text{if } x \in C_i^{\mathcal{T}}, \\ 0 & \text{otherwise,} \end{cases}$$

to indicate whether a point $x \in \mathcal{W}$ belongs to the i-th cell, where $i = 1, \ldots, n_{\mathcal{T}}$.

The simplest model of a distance-based tessellation is the Voronoi tessellation, where $d_{\mathcal{T}}(x, s) = \|x - s\|$ for $x \in \mathcal{W}$ with a generator $s \in \mathbb{R}^3 = \mathbb{G}$, and $\|x - s\|$ denotes the Euclidean norm of $x - s$. While widely studied in literature, see e.g. Aurenhammer et al. (2013); Møller (1994); Okabe et al. (2000), the Voronoi tessellation is often found to be insufficiently flexible to fit experimental maps of polycrystalline materials (Šedivý et al., 2018); thus, more sophisticated tessellation models are needed. The fitting procedure considered in the present paper is able to handle many tessellations of the form given by **Eq. 1** for which the distance function $d_{\mathcal{T}}$ is differentiable. However, we focus on tessellation models that are special cases of generalized balanced power diagrams (GBPDs), listed here in order from simplest to most complex:

1) The Laguerre tessellation (Lautensack and Zuyev, 2008) is obtained if $d_{\mathcal{T}}(x, (s, w)) = \|x - s\|^2 - w$ for $x \in \mathcal{W}$, with a generator consisting of a seed point $s \in \mathbb{R}^3$ and an additive weight $w \in \mathbb{R}$.
2) The multiplicatively weighted Laguerre tessellation is obtained if $d_{\mathcal{T}}(x, (s, m, w)) = m\|x - s\|^2 - w$ for $x \in \mathcal{W}$, with a generator consisting of a seed point $s \in \mathbb{R}^3$, a multiplicative weight $m > 0$ and an additive weight $w \in \mathbb{R}$.
3) The diagonal *GBPD* is obtained if $d_{\mathcal{T}}(x, (s, M, w)) = (x - s)^{\top} M (x - s) - w$ for $x \in \mathcal{W}$, with a generator

consisting of a seed point $s \in \mathbb{R}^3$, a diagonal distance matrix $M \in \mathbb{R}^{3\times3}$ where every (diagonal) entry is positive and an additive weight $w \in \mathbb{R}$.

4) The general GBPD (Alpers et al., 2015) is obtained if $d_{\mathcal{T}}(x, (s, M, w)) = (x-s)^{\top} M (x-s) - w$ for $x \in \mathcal{W}$, with a generator consisting of a seed point $s \in \mathbb{R}^3$, a positive definite distance matrix $M \in \mathbb{R}^{3\times3}$ and an additive weight $w \in \mathbb{R}$.

Note that all of these models, except the Laguerre tessellation, can exhibit curved cell boundaries and thus non-convex cells. With this property comes the possibility, however, that cells are no longer connected, which might be undesirable when seeking parametric representations of polycrystalline materials. To rectify this issue, modifications of the original tessellation models, such as the one given by Šedivý et al. (2018), can be applied to fitted generators as a post-processing step. Another problem that affects all of the tessellation models described above is the possibility for a generator not to produce a corresponding cell. This can be mitigated by considering a volume-based cost function during the fitting that penalizes missing cells—see **Section 2.3**.

2.3 Gradient Descent-Based Tessellation Fitting

In this section we describe an efficient, gradient descent-based fitting procedure for GBPD-type tessellations. This procedure was originally introduced in Furat et al. (2021), but in **Section 3** it will be applied to a broader class of tessellation models than in Furat et al. (2021).

Note that the fitting of a tessellation $\mathcal{T} = \{C_i^{\mathcal{T}}\}_{i=1}^{n_{\mathcal{T}}}$ can be achieved by finding generators such that the similarity between the tessellation $\mathcal{T}$ and the ground truth image data is maximized. Formally, we consider the i-th grain of the ground truth image data as a map $C_i^{\mathrm{GT}}\colon \mathbb{Z}^3 \to \{0,1\}$ given by

$$C_i^{\mathrm{GT}}(x) = \begin{cases} 1 & \text{if } x \text{ belongs to the } i-\text{th grain,} \\ 0 & \text{otherwise,} \end{cases}$$

with $\mathbb{Z}$ the set of all integers, $i = 1, \ldots, n_{\mathrm{GT}}$, and n_{GT} the number of grains in the sampling window $\mathcal{W} \subset \mathbb{R}^3$. Nearest-neighbor interpolation can be used to extend the domain of C_i^{GT} from the integer lattice $\mathbb{Z}^3$ to the continuous Euclidean space $\mathbb{R}^3$. Furthermore, let $\mathcal{X}^{\mathrm{F}} = \{x_j^{\mathrm{F}}\}_{j=1}^{n_{\mathrm{F}}} \in \mathbb{Z}^{3\times n_{\mathrm{F}}}$ be the set of all coordinates of voxels that belong to one of the grains, which we call the foreground voxels of the image data. If n_{F} denotes the number of foreground voxels in $\mathcal{W}$, then for each $j = 1, \ldots, n_{\mathrm{F}}$ there is an integer $i = 1, \ldots, n_{\mathrm{GT}}$ such that $C_i^{\mathrm{GT}}(x_j^{\mathrm{F}}) = 1$. In **Section 3** we will consider the smoothed experimental image data from **Section 2.1** (among others) and set

$$C_i^{\mathrm{GT}}(x) = \begin{cases} 1 & \text{if } x \in \mathcal{W}_{\mathrm{E,smooth}} \text{ and } I_{\mathrm{E,smooth}}(x) = i, \\ 0 & \text{otherwise,} \end{cases}$$

and $\mathcal{X}^{\mathrm{F}} = \{x \in \mathcal{W}_{\mathrm{E,smooth}}\colon I_{\mathrm{E,smooth}}(x) > 0\}$.

Probably the most natural way to define the similarity between a tessellation $\mathcal{T}$ and the ground truth image data is to count the voxels at which each cell of the tessellation and the corresponding grain of the ground truth dataset overlap. To be more precise, the value of the objective function $E\colon \mathbb{G}^{n_{\mathrm{GT}}} \to [0,\infty)$ for a set of generators $\mathcal{G} = \{g_i\}_{i=1}^{n_{\mathrm{GT}}}$ with $g_i \in \mathbb{G}$ is given by

$$E(\mathcal{G}) = \frac{1}{n_{\mathrm{F}}} \sum_{j=1}^{n_{\mathrm{F}}} \sum_{i=1}^{n_{\mathrm{GT}}} C_i^{\mathcal{T}}\left(x_j^{\mathrm{F}}\right) C_i^{\mathrm{GT}}\left(x_j^{\mathrm{F}}\right), \tag{2}$$

where the cells $C_1^{\mathcal{T}}, \ldots, C_{n_{\mathrm{GT}}}^{\mathcal{T}}$ of the tessellation $\mathcal{T}$ depend on the choice of the generators in $\mathcal{G}$ subject to $n_{\mathcal{T}} = n_{\mathrm{GT}}$. The corresponding fitting problem is thus to determine an optimal set of generators $\mathcal{G}_{\mathrm{opt}}$ defined as

$$\mathcal{G}_{\mathrm{opt}} = \operatorname*{argmax}_{\mathcal{G}} E(\mathcal{G}). \tag{3}$$

It is easy to see that $C_j^{\mathcal{T}}(x^{\mathrm{F}})$ with $x^{\mathrm{F}} \in \mathcal{X}^{\mathrm{F}}$ can be reformulated as

$$C_j^{\mathcal{T}}(x^{\mathrm{F}}) = \operatorname{argmin}_j^*\left(d_{\mathcal{T}}(x^{\mathrm{F}}, g_1), \ldots, d_{\mathcal{T}}(x^{\mathrm{F}}, g_{n_{\mathcal{T}}})\right), \tag{4}$$

where $\operatorname{argmin}_j^*$ is the j-th component of the $n_{\mathcal{T}}$-dimensional vector-valued argmin function, i.e.,

$$\operatorname{argmin}^*\colon \mathbb{R}^{n_{\mathcal{T}}} \to \{0,1\}^{n_{\mathcal{T}}},$$
$$\operatorname{argmin}_j^* z = \begin{cases} 1 & \text{if } z_j \le z_i \text{ for all } i = 1, \ldots, n_{\mathcal{T}}, \\ 0 & \text{otherwise,} \end{cases}$$

with $z = (z_1, \ldots, z_{n_{\mathcal{T}}}) \in \mathbb{R}^{n_{\mathcal{T}}}$. In cases where the minimum is not unique, i.e., there are indices $j_1, j_2 \in \{1, \ldots, n_{\mathcal{T}}\}$ with $j_1 \neq j_2$ and $z_{j_1} = z_{j_2}$, only the component with the smallest index is set equal to 1. The function argmax^* is defined analogously.

Even though the distance function $d_{\mathcal{T}}$ is differentiable (with respect to the generators), the fact that argmin^* in **Eq. 4** does not have a derivative makes the objective function E defined in **Eq. 2** non-differentiable. This leaves us having to resort to derivative-free optimization algorithms to solve **Eq. 3**, which in most cases converge slower than gradient descent methods (Audet and Hare, 2017). In order to increase efficiency, we slightly deviate from the original tessellation formulation by replacing the argmin^* function in **Eq. 4** with a "$\operatorname{softmin}^*$" function—i.e., a $\operatorname{softmax}^*$ function with a negative argument, $\tilde{C}_j^{\mathcal{T}}(x^{\mathrm{F}}) = \operatorname{softmax}_j^*(-d_{\mathcal{T}}(x^{\mathrm{F}}, g_1), \ldots, -d_{\mathcal{T}}(x^{\mathrm{F}}, g_{n_{\mathcal{T}}}))$ for $x^{\mathrm{F}} \in \mathcal{X}^{\mathrm{F}}$. Here, the $n_{\mathcal{T}}$-dimensional function

$$\operatorname{softmax}^*\colon \mathbb{R}^{n_{\mathcal{T}}} \to [0,1]^{n_{\mathcal{T}}},$$
$$\operatorname{softmax}_j^* z = \frac{\exp\left(z_j\right)}{\sum_{k=1}^{n_{\mathcal{T}}} \exp(z_k)}$$

with $z = (z_1, \ldots, z_{n_{\mathcal{T}}}) \in \mathbb{R}^{n_{\mathcal{T}}}$ is a smooth version of the argmax^* function. So, instead of returning a vector the components of which are either 0 or 1, the $\operatorname{softmax}^*$ function is a vector-valued map, the components of which are continuous functions with values between 0 and 1. In fact, the output vector $\operatorname{softmax}^* z$ for some argument $z \in \mathbb{R}^{n_{\mathcal{T}}}$ defines a discrete probability measure (i.e., the values of all components are between 0 and 1 and their sum is equal to 1), which assigns the highest probability to the index j if $z_j \ge z_i$ for all $i \in \{1, \ldots, n_{\mathcal{T}}\}$. Here, the last property can be understood in the sense that the $\operatorname{softmax}^*$ function preserves the maximum of the input vector. The largest value of a component of the vector $(\tilde{C}_1^{\mathcal{T}}(x^{\mathrm{F}}), \ldots, \tilde{C}_{n_{\mathcal{T}}}^{\mathcal{T}}(x^{\mathrm{F}}))$ is therefore

the one whose corresponding generator has the shortest tessellation distance to the given evaluation point x^{F}.

The benefit of applying the softmax* function instead of using the tessellation distances directly is found in the fact that the output vector of the softmax* function is normalized, and thus for each evaluation point x^{F} only the relative changes in the tessellation distances to the generators are considered, providing the same scale (i.e., values in [0, 1]) for all evaluation points. This trick is often used for multi-class classification problems in machine learning (Goodfellow et al., 2016). Furthermore, note that since softmax* is a composition of differentiable functions and is itself therefore differentiable, the function $\tilde{C}_j^{\mathcal{T}}$ is differentiable, as well, for each $j \in \{1, \ldots, n_{\mathcal{T}}\}$.

Because—in contrast to $C_j^{\mathcal{T}}$, which is either 0 or 1—$\tilde{C}_j^{\mathcal{T}}$ assumes continuous values, it is necessary to adapt the objective function. Consequently, instead of E defined as in **Eq. 2**, we consider the function $\tilde{E}\colon \mathbb{G}^{n_{\mathrm{GT}}} \to (-\infty, 0]$, where

$$\tilde{E}(\mathcal{G}) = \frac{1}{n_{\mathrm{F}}} \sum_{j=1}^{n_{\mathrm{F}}} \sum_{i=1}^{n_{\mathrm{GT}}} \lambda_{\mathrm{NCE}}\left(\tilde{C}_i^{\mathcal{T}}\left(x_j^{\mathrm{F}}\right), C_i^{\mathrm{GT}}\left(x_j^{\mathrm{F}}\right)\right) \tag{5}$$

with the *negative* (binary) cross-entropy loss function $\lambda_{\mathrm{NCE}}\colon [0,1]^2 \to (-\infty, 0]$ given by $\lambda_{\mathrm{NCE}}(\tilde{y}, y) = y\log\tilde{y} + (1-y)\log(1-\tilde{y})$. Note that the cross-entropy loss is often used in machine learning (Goodfellow et al., 2016) to compare the output of a classifier to ground truth data, which is basically the same purpose it serves here: If an evaluation point x^{F} belongs to the i-th grain (i.e., $C_i^{\mathrm{GT}}(x^{\mathrm{F}}) = 1$), $\tilde{C}_i^{\mathcal{T}}(x^{\mathrm{F}})$ also needs to be close to 1 in order to maximize λ_{NCE}, and vice versa. Note that the modified objective function $\tilde{E}$ is differentiable with respect to the generators. The fitted set of generators $\tilde{\mathcal{G}}_{\mathrm{opt}}$ can then be obtained by solving

$$\tilde{\mathcal{G}}_{\mathrm{opt}} = \operatorname*{argmax}_{\mathcal{G}} \tilde{E}(\mathcal{G}). \tag{6}$$

In summary, we reformulated the original fitting problem, **Eq. 3**, into the differentiable version given in **Eq. 6**. This allows us to employ fast, gradient-based optimization algorithms, such as the one used in the present work: the stochastic gradient descent algorithm ADAM (Kingma and Ba, 2015) (applied to the negative objective function). The optimization is stopped after a maximum of 25 iterations through (random permutations of) the dataset or if the objective function $\tilde{E}$ defined in **Eq. 5** does not increase by more than 10^{-4} in 3 iterations. Note that for the discretization of a fitted GBPD-type tessellation, we compute the (unique) cell labels using the classical definition in **Eq. 4**. The software implementation for the fitting and the discretization is based on TENSORFLOW (Abadi et al., 2015), which allows for highly parallel and even GPU-accelerated computations.

2.4 Gradient-free Tessellation Fitting

In addition to the procedure described in **Section 2.3**, we mention two additional methods from the literature for fitting tessellations to image data, to which we will refer below. We start with the procedure for Laguerre tessellations introduced in Spettl et al. (2016), which was used to acquire the initial parameter configuration in **Section 3**. Furthermore, in order to compare the results of our method described in **Section 2.3**, we also employed a different method for the fast fitting of GBPDs that was originally developed in Teferra and Rowenhorst (2018).

2.4.1 Laguerre Tessellation Fitting with the Cross-Entropy Method

In Spettl et al. (2016), approximations of polycrystalline image data were sought in the form of Laguerre tessellations. Just like in **Section 2.3** of the present paper, an optimization problem was formulated. However, instead of considering a volume-based objective function, an interface-based discrepancy measure was minimized. More precisely, the quality of fit for a given set of Laguerre generators $\mathcal{G}$ was judged by looking at each boundary between two grains. Let their grain labels be denoted by $i \neq \ell = 1, \ldots, n_{\mathrm{GT}}$. Then, a plane $P_{i,\ell}^{\mathrm{or}}$ was determined by orthogonal regression of the boundary voxel coordinates, and ten test points $x_{i,\ell,1}^{\mathrm{T}}, \ldots x_{i,\ell,10}^{\mathrm{T}} \in P_{i,\ell}^{\mathrm{or}}$ on this plane were considered. Furthermore, the plane $P_{i,\ell}^{\mathrm{eq}}$ that is equidistant (with respect to the Laguerre distance $d_{\mathcal{T}}$) to the two corresponding generators g_i and g_ℓ was computed. Note that if the cells $C_i^{\mathcal{T}}$ and $C_\ell^{\mathcal{T}}$ are neighboring, the plane $P_{i,\ell}^{\mathrm{eq}}$ covers their shared facet, but otherwise—e.g., when one of the cells is empty—the plane $P_{i,\ell}^{\mathrm{eq}}$ does not have a correspondence in the tessellation. The total discrepancy $D\colon \mathbb{G} \to [0, \infty)$ was then obtained as the average of squared distances between the test points $\{x_{i,\ell,1}^{\mathrm{T}}, \ldots, x_{i,\ell,10}^{\mathrm{T}}\}$ to the plane $P_{i,\ell}^{\mathrm{eq}}$ for all neighboring grains; more precisely,

$$D(\mathcal{G}) = \frac{1}{n_{\mathrm{T}}} \sum_{\substack{i,\ell=1,\ldots,n_{\mathrm{GT}} \\ \text{grains } i,\ell \text{ neighboring}}} \sum_{k=1}^{10} \operatorname{dist}\left(x_{i,\ell,k}^{\mathrm{T}}, P_{i,\ell}^{\mathrm{eq}}\right)^2,$$

where n_{T} is the total number of test points for all neighboring grains, and $\operatorname{dist}(x, P)$ is the shortest Euclidean distance of the point x to a point on the plane P. The resulting minimization problem is rather high-dimensional and non-convex. For this reason, in Spettl et al. (2016) a global stochastic optimization technique was employed—namely, the cross-entropy method (Rubinstein and Kroese, 2004)—to escape local minima of the objective function. In the present paper, the same values for the parameters of the algorithm as proposed by Spettl et al. were used (see Spettl et al. (2016) for a full list).

2.4.2 GBPD Fitting Using a Direct Approach

A quite different approach, this time for fitting GBPDs, was proposed in Teferra and Rowenhorst (2018) (which is referred to as the direct approach in the following). In **Section 3**, we will employ this method as a baseline comparison to our gradient-based fitting method. With the direct approach no optimization was performed, but rather formulas for directly estimating the tessellation parameters were presented, which leads to a very fast heuristic to fit GBPDs. This was achieved by determining the generators (s_i, M_i, w_i) of the i-th cell only by considering the i-th grain and without knowledge of the other grains/cells: The seed point s_i was set to the center of mass of the grain, and the distance matrix M_i was computed from the covariance matrix of its voxel coordinates. Since the isosurface of the GBPD distance function

$\{x \in \mathcal{W}\colon d_{\mathcal{T}}(x, (s_i, M_i, w_i)) = (x - s_i)^{\top} M_i (x - s_i) - w_i = 0\}$ can be considered as an ellipsoid, the additive weights were computed by equating the volume of this ellipsoid to that of the i-th grain. Note that this approach is similar to the one proposed in Lyckegaard et al. (2011) for Laguerre tessellations.

3 RESULTS

In order to evaluate the gradient descent-based fitting method described in **Section 2.3**, we applied it to several different image datasets using the following procedure. First, initial Laguerre generators were determined by the cross-entropy approach developed in Spettl et al. (2016) (see **Section 2.4.1**). Then, tessellation models with increasing complexity were successively fitted, using the generators of a simpler tessellation model as the initial parameter configuration. Here, any parameters that were not part of the simpler model were initialized with default values: For example, when optimizing the fit of a multiplicatively weighted Laguerre tessellation based on the generators of a fitted Laguerre tessellation, the multiplicative weights were set equal to 1; in the case of a diagonal GBPD, each diagonal matrix was filled with a value equal to the corresponding multiplicative weight. For purposes of comparison, we independently applied the direct approach of Teferra and Rowenhorst (2018) to each image dataset (see **Section 2.4.2**).

3.1 Performance Measures

To evaluate the goodness of fit of the tessellation $\{C_i^{\mathcal{T}}\}_{i=1}^{n_{\mathcal{T}}}$ fitted to the foreground voxels $\mathcal{X}^{\mathrm{F}} = \{x_j^{\mathrm{F}}\}_{j=1}^{n_{\mathrm{F}}}$ of the ground truth image data $\{C_i^{\mathrm{GT}}\}_{i=1}^{n_{\mathrm{GT}}}$ (with $n_{\mathrm{GT}} = n_{\mathcal{T}}$), we consider various performance measures. The fraction of correctly assigned voxels is given by

$$F_{\mathrm{c}} = \frac{1}{n_{\mathrm{F}}} \sum_{j=1}^{n_{\mathrm{F}}} \sum_{i=1}^{n_{\mathrm{GT}}} C_i^{\mathcal{T}}\left(x_j^{\mathrm{F}}\right) C_i^{\mathrm{GT}}\left(x_j^{\mathrm{F}}\right).$$

Similarly, the fraction of correctly assigned boundary voxels is defined as

$$F_{\mathrm{c}}^{\mathrm{B}} = \frac{1}{n_{\mathrm{B}}} \sum_{j=1}^{n_{\mathrm{B}}} \sum_{i=1}^{n_{\mathrm{GT}}} C_i^{\mathcal{T}}\left(x_j^{\mathrm{B}}\right) C_i^{\mathrm{GT}}\left(x_j^{\mathrm{B}}\right),$$

where

$$\mathcal{X}^{\mathrm{B}} = \left\{x_j^{\mathrm{B}}\right\}_{j=1}^{n_{\mathrm{B}}} = \{x_1^{\mathrm{B}} \in \mathcal{X}^{\mathrm{F}}\colon \|x_1^{\mathrm{B}} - x_2^{\mathrm{B}}\| \le \sqrt{3} \text{ and } C_i^{\mathrm{GT}}(x_1^{\mathrm{B}}) = C_\ell^{\mathrm{GT}}(x_2^{\mathrm{B}}) = 1 \text{ for some } x_2^{\mathrm{B}} \in \mathcal{X}^{\mathrm{F}} \text{ and } i, \ell = 1, \ldots, n_{\mathrm{GT}}, i \neq \ell\}$$

are the coordinates of the n_{B} grain boundary voxels (with respect to the 26-neighborhood, where the voxels x_1, x_2 are neighbors if $\|x_1 - x_2\| \le \sqrt{3}$). Moreover, note that the fraction of empty cells F_0 can be written as

$$F_0 = \frac{1}{n_{\mathcal{T}}} \sum_{i=1}^{n_{\mathcal{T}}} \mathbb{1}\{C_i^{\mathcal{T}}(x^{\mathrm{F}}) = 0 \text{ for all } x^{\mathrm{F}} \in \mathcal{X}^{\mathrm{F}}\},$$

where $\mathbb{1}\{\cdot\}$ denotes the indicator function. Furthermore, consider the set of all grains that are a neighbor of the i-th grain in the ground truth image data (with respect to the 6-neighborhood of each voxel, where the voxels x_1, x_2 are neighbors if $\|x_1 - x_2\| \le 1$),

$$\mathcal{N}_i^{\mathrm{GT}} = \{\ell \in \{1, \ldots, n_{\mathrm{GT}}\}\colon C_i^{\mathrm{GT}}(x_1^{\mathrm{F}}) = C_\ell^{\mathrm{GT}}(x_2^{\mathrm{F}}) = 1 \text{ for any } x_1^{\mathrm{F}}, x_2^{\mathrm{F}} \in \mathcal{X}^{\mathrm{F}} \text{ with } \|x_1^{\mathrm{F}} - x_2^{\mathrm{F}}\| \le 1,\ i \neq \ell\},$$

the set of all cells that are a neighbor of the i-th tessellation cell, $\mathcal{N}_i^{\mathcal{T}}$ (defined analogously), and the resulting set of correctly assigned neighbors, $\mathcal{N}_i = \mathcal{N}_i^{\mathrm{GT}} \cap \mathcal{N}_i^{\mathcal{T}}$. Then, the fraction of cells for which all cell neighbors are correct can be written as

$$N_0 = \frac{1}{n_{\mathrm{GT}}} \sum_{i=1}^{n_{\mathrm{GT}}} \mathbb{1}\left\{\mathcal{N}_i^{\mathrm{GT}} = \mathcal{N}_i^{\mathcal{T}}\right\},$$

and the mean number of incorrect cell neighbors is

$$\bar{N} = \frac{1}{n_{\mathrm{GT}}} \sum_{i=1}^{n_{\mathrm{GT}}} \left(\#\mathcal{N}_i^{\mathrm{GT}} - \#\mathcal{N}_i\right),$$

where # denotes cardinality. These performance measures were calculated for simulated and experimental image data with both smooth and rough grain boundaries (see **Sections 3.2, 3.3**).

3.2 Simulated Data

In this section, the fitting of tessellations to simulated image data is investigated. This allows us to study scenarios in which, in principle, the tessellations can perfectly describe the image data, which is usually not the case for experimental data. Apart from that, it is also possible to simulate the effect of noisy image data while still having access to the true grain boundaries.

3.2.1 Smooth Grain Boundaries

As a first step, the performance of the fitting procedure described in **Section 2.3** is evaluated for simulated data having smooth grain boundaries—more precisely, for a (discretized) realization of a random multiplicatively weighted Laguerre tessellation. Since a tessellation of the same type (among others) was fitted to the simulated image dataset, it is clear that theoretically a perfect match could have been achieved. However, whether or not this global optimum is actually found depends strongly on the initial generators. In the present investigation, we made sure that no information leaked from the generation of the simulated image data to the choice of initial generators (apart from the image data, of course).

The tessellation underlying the simulated data was created as follows in the cubic sampling window $\mathcal{W} = [0, 299]^3$. The seed points $\{s_i\}_{i=1}^{n_{\mathcal{T}}}$ were a realization of a Matérn hardcore process with (overall) intensity $\lambda_{\mathrm{sim}} > 0$ and hardcore radius $r_{\mathrm{sim}} > 0$. We refer to Chiu et al. (2013) for additional details. The weights were then independently drawn: in the case of additive weights $\{w_i\}_{i=1}^{n_{\mathcal{T}}}$, from a $(0, \infty)$-truncated normal distribution with (untruncated) mean $\mu_{\mathrm{sim}} > 0$ and variance $\sigma_{\mathrm{sim}}^2 > 0$, and, in the case of multiplicative weights $\{m_i\}_{i=1}^{n_{\mathcal{T}}}$, from an inverse gamma distribution with shape parameter $\alpha_{\mathrm{sim}} > 0$ and scale parameter $\beta_{\mathrm{sim}} > 0$. To mitigate boundary effects, the seed point process was simulated in a larger window, and only those generators whose cell was located at least partly within the actual simulation window were retained (this

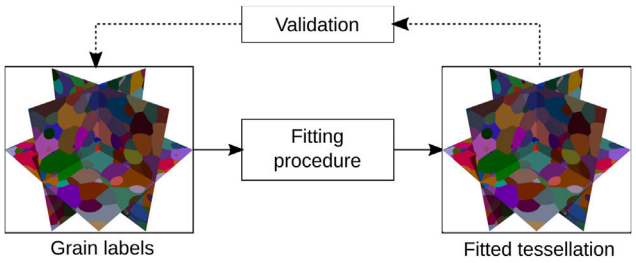


FIGURE 2 | Schematic overview of the fitting of tessellations to simulated image data having smooth grain boundaries and validation of the resulting fits. Orthogonal 2D slices through 3D datasets are shown.

procedure is called plus-sampling in the literature—see e.g. Chiu et al. (2013)).

In our case, the parameters of the random tessellation model were set as follows: $\lambda_{\text{sim}} = 0.0000205$, $r_{\text{sim}} = 14.1$, $\mu_{\text{sim}} = 27.4$, $\sigma^2_{\text{sim}} = 9.25$, $\alpha_{\text{sim}} = 1.5$ and $\beta_{\text{sim}} = 0.0939$. In total, the resulting realization had $n_{\text{GT}} = 1073$ cells and was discretized in the window $\mathcal{W}_{\text{sim}} = \{0, \ldots, 299\}^3$ by assigning each voxel a label associated with the simulated cell in which the corresponding voxel coordinate is located. This is described by the mapping $I_{\text{sim}}: \mathcal{W}_{\text{sim}} \to \{1, \ldots, n_{\text{GT}}\}$, the values of which are hereafter referred to as *smooth simulated image data*. All voxels were considered during the fitting (i.e., $\mathcal{X}^{\text{F}} = \mathcal{W}_{\text{sim}}$).

Once the simulated image dataset was obtained, different tessellation models were successively fitted to it, and their goodness of fit was evaluated using the performance measures from **Section 3.1**. A schematic overview of this procedure is depicted in **Figure 2**. A visual comparison of the ground truth image data and the fits is given in **Figure 3**, whereas numerical fitting results are presented in **Table 1**.

3.2.2 Perturbed Grain Labels

One of the main goals of the present paper is to investigate the fitting of tessellations to image data containing rough grain boundaries—which may result, for example, from measurement uncertainties. For an in-depth analysis of this scenario, the grain labels of the simulated image data from **Section 3.2.1** were perturbed such that the originally smooth boundaries exhibited a similar degree of roughness as in the experimental image data. With this approach, it was possible to vary the intensity of the perturbation and to study the robustness of the fitting even for degrees of boundary roughness well beyond that observed in experiment. Another benefit was the ability to evaluate the goodness of fit with respect to the (true) smooth grain boundaries instead of with respect to the perturbed grain boundaries that were input to the fitting procedure (**Figure 4**).

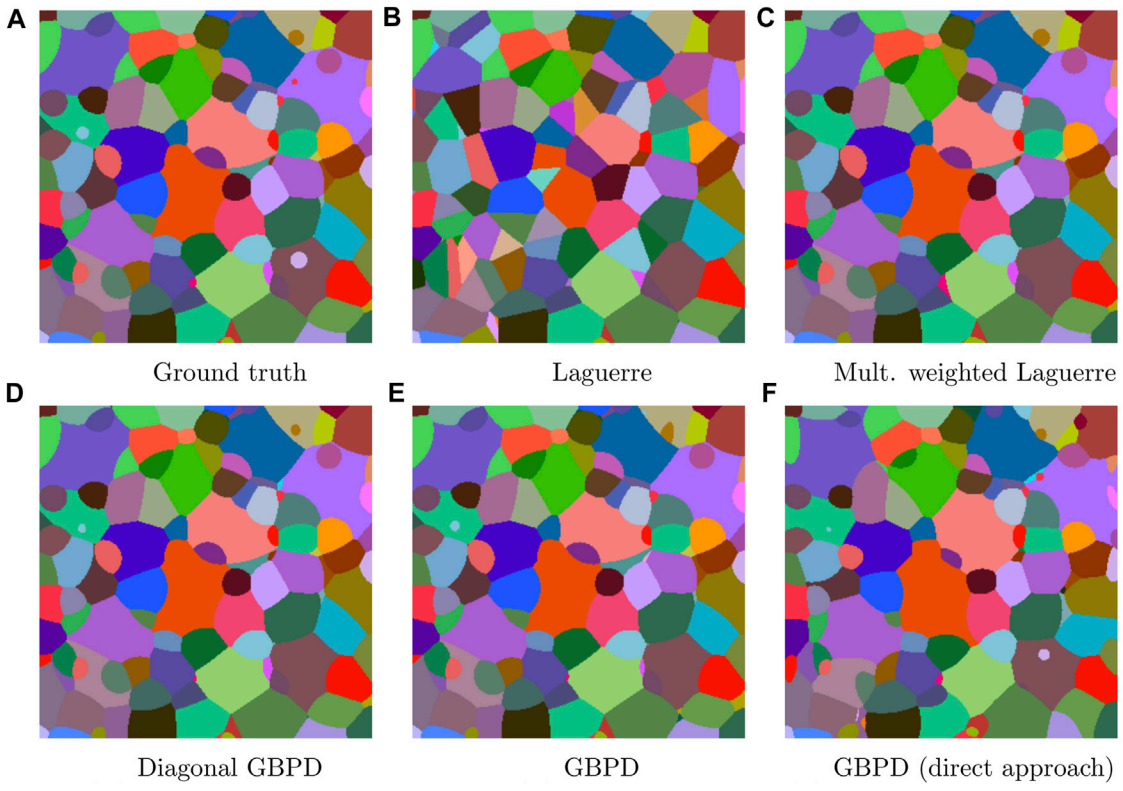


FIGURE 3 | Two-dimensional slices through **(A)** the simulated ground truth image data and the corresponding fitted tessellations: **(B)**–**(E)** were obtained by the gradient descent-based method described in **Section 2.3**, whereas **(F)** followed from the direct approach described in **Section 2.4.2**.

TABLE 1 | Values of performance measures for various tessellation models fitted to the smooth simulated image data, considering the fraction of correctly assigned voxels F_c, the fraction of correctly assigned boundary voxels F_c^B, the fraction of empty cells F_0, the fraction of cells for which all cell neighbors are correct N_0, and the mean number of incorrect cell neighbors $\bar{N}$. The fits were obtained by the cross-entropy approach ("initial configuration") described in **Section 2.4.1**, the gradient descent-based approach described in **Section 2.3**, and the "direct approach" described in **Section 2.4.2**.

	F_c	F_c^B	F_0	N_0	$\bar{N}$
Laguerre (initial configuration)	0.516	0.324	0.005	0.064	6.289
Laguerre	0.802	0.562	0.036	0.114	3.290
Multiplicatively weighted Laguerre	0.987	0.948	0.058	0.587	0.634
Diagonal GBPD	0.988	0.952	0.056	0.644	0.529
GBPD	0.979	0.909	0.055	0.579	0.624
GBPD (direct approach)	0.824	0.551	0.004	0.271	1.392

Intuitively, the perturbed simulated image data was generated by computing a binary image of the grain boundaries from the smooth simulated image I_{sim} considered in **Section 3.2.1**. Then, the grain boundaries were blurred. The resulting grayscale image was used to define probabilities with which the grain labels of voxels in I_{sim} were reassigned to labels drawn from each voxel's near vicinity. This way, a grain label is most likely to be changed when a voxel is located near a grain boundary, while the labels of voxels closer to a grain center will usually remain unchanged. A similar tendency is evident in the raw experimental data (**Figure 1**).

More precisely, the perturbation probabilities were obtained as follows. First, a 'subvoxel' boundary image $I_b\colon \mathcal{W}_{\text{pert}}^* \to \{0,1\}$ of the smooth simulated image I_{sim} was computed according to

$$I_b(y) = \begin{cases} 1 & \text{if } \#\mathcal{N}^b(y) > 1, \\ 0 & \text{otherwise,} \end{cases}$$

with

$$\mathcal{N}^b(y) = \Big\{ I_{\text{sim}}(x_1,x_2,x_3)\colon\ x_k = \frac{y_k \pm 1}{2} \text{ if } y_k \text{ odd, and } x_k = \frac{y_k}{2} \text{ if } y_k \text{ even}, k = 1,2,3 \Big\}$$

for $y = (y_1, y_2, y_3) \in \mathcal{W}_{\text{pert}}^* = \{0,\ldots,598\}^3$. This means $I_b(y)$ is labeled as a boundary voxel whenever the set $\mathcal{N}^b(y)$ contains at least two distinct grain labels. Effectively, this amounts to assigning boundary locations by considering an upsampled version of I_{sim} with nearly twice the number of voxels in each spatial dimension. So, if two neighboring voxels in I_{sim} have different labels, a boundary is drawn between these voxels in I_b—see **Figure 5**. Next, a Gaussian blur (Russ and Neal, 2017) with standard deviation $\sigma \geq 0$ (where $\sigma = 0$ implies no blurring) was applied to the boundary image I_b to obtain the blurred boundary image $I_{\text{blur}}\colon \mathcal{W}_{\text{pert}}^* \to [0,1]$. Here, the values of voxels in $\mathcal{W}_{\text{pert}}^*$ are scaled such that their minimum and maximum are equal to 0 and 1, respectively. Since σ determines how far into a grain the perturbations occur, we call it the perturbation spread. Finally, the perturbation probability image $I_{\text{prob}}\colon \mathcal{W}_{\text{pert}} \to [0,1]$ was acquired by a subsequent downsampling to the original resolution of I_{sim}:

$$I_{\text{prob}}(x) = \frac{1}{8} \sum_{z_1,z_2,z_3 \in \{0,1\}} I_{\text{blur}}(2x_1 + z_1, 2x_2 + z_2, 2x_3 + z_3)$$

for $x = (x_1, x_2, x_3) \in \mathcal{W}_{\text{pert}} = \mathcal{W}_{\text{sim}}$. During this calculation, values outside the domain of I_{blur} were set equal to 0.

The smooth simulated image I_{sim} was perturbed by considering the random variables Z_x with values in $\{1, \ldots, n_{\text{GT}}\}$ such that

$$Z_x = \begin{cases} I_{\text{sim}}(x) & \text{with probability } I_{\text{prob}}(x), \\ I_{\text{sim}}(y) & \text{with probability } 1 - I_{\text{prob}}(x), \end{cases}$$

for each voxel $x \in \mathcal{W}_{\text{pert}}$, where y is the closest voxel in I_{sim} to x for which $I_{\text{sim}}(y) \neq I_{\text{sim}}(x)$ (ties are broken by choosing a grain label uniformly at random). We assume that all Z_x are stochastically independent of each other. The perturbation $I_{\text{pert}}\colon \mathcal{W}_{\text{pert}} \to \{1,\ldots,n_{\text{GT}}\}$ of the smooth simulated image I_{sim} was then obtained as a realization of the random variables $\{Z_x\}_{x \in \mathcal{W}_{\text{pert}}}$. Note that I_{pert} is a function of the perturbation spread $\sigma \geq 0$ and that, even for the case $\sigma = 0$, perturbations can still occur within a voxel of the grain boundaries, as I_{prob} can

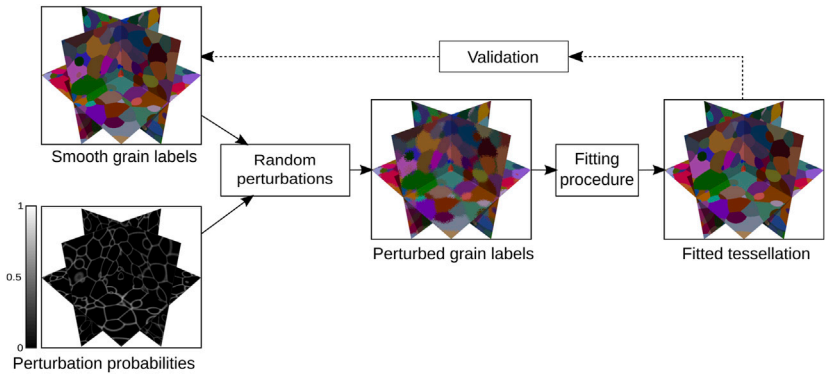


FIGURE 4 | Schematic overview of the generation of perturbed simulated grain labels and validation of their fits. Orthogonal 2D slices through 3D datasets are shown.

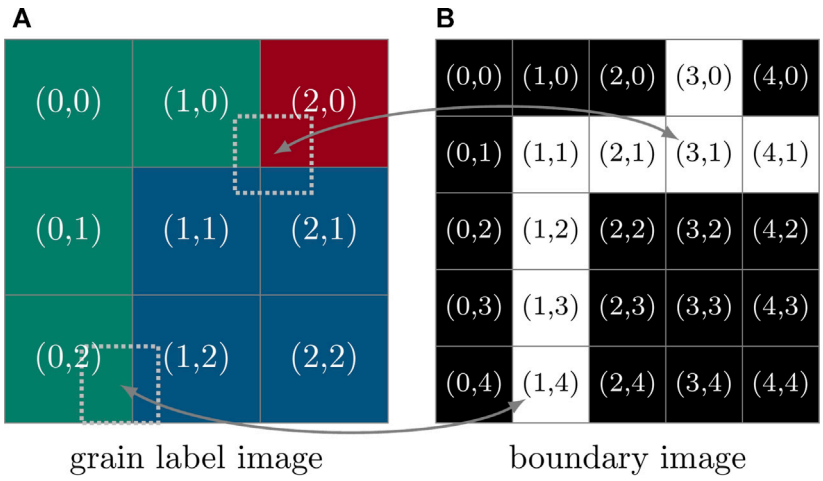


FIGURE 5 | Two-dimensional example of **(A)** a 3 × 3-pixel grain label image I_{sim} with three grain labels (green, red and blue) and **(B)** its "subvoxel" boundary image I_b. Two pixels of the boundary image (one at location (3, 1) and another at (1, 4)) are superimposed on their corresponding locations in the grain label image (dotted squares). Effectively, the set $\mathcal{N}^b(\cdot)$ contains all grain labels covered by these shifted pixels. The cardinality of the set $\mathcal{N}^b((1,4))$ is therefore 2, whereas $\#\mathcal{N}^b((3,1)) = 3$. If the indices specifying the location of a pixel in the boundary image are all even, this pixel lies entirely within the bounds of a single pixel in the grain label image; consequently, the cardinality of $\mathcal{N}^b(\cdot)$ is always 1, and the pixel in I_b will always be assigned the value of zero (shaded black in the boundary image).

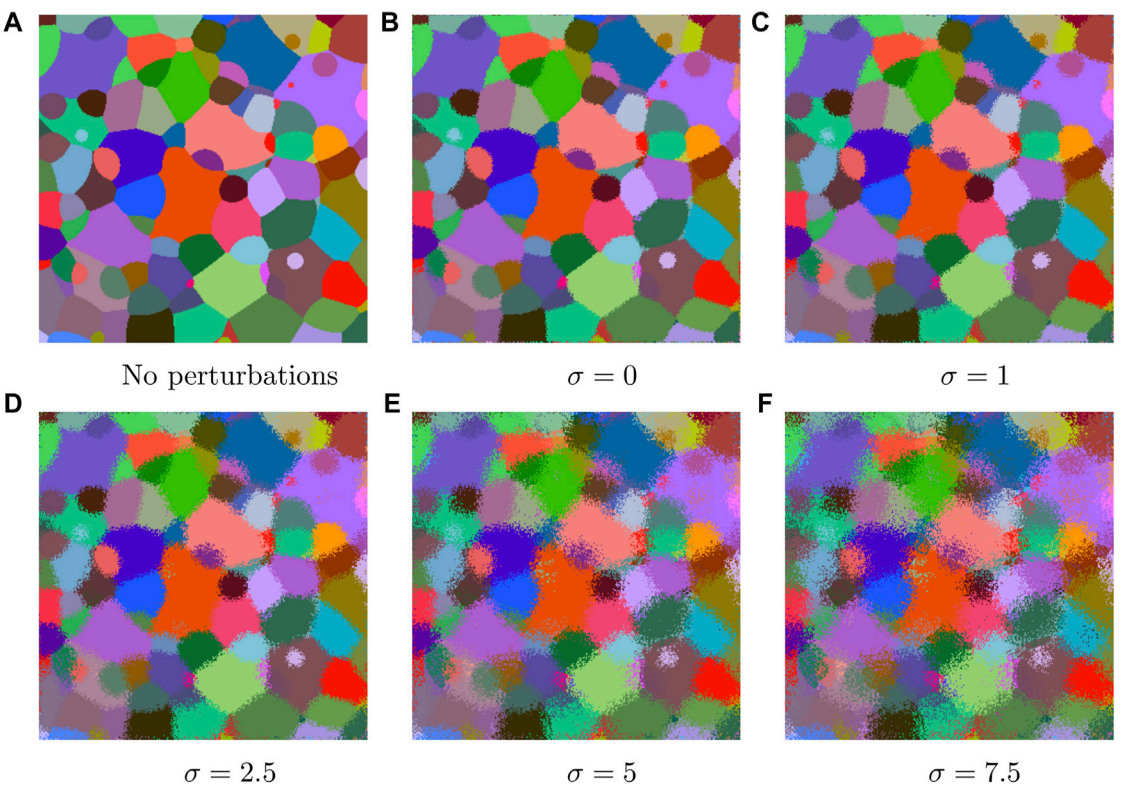


FIGURE 6 | Two-dimensional slices through the simulated image data I_{sim} following perturbation of the latter with I_{pert}, shown in **(A–F)** for different values of the perturbation spread $\sigma \geq 0$.

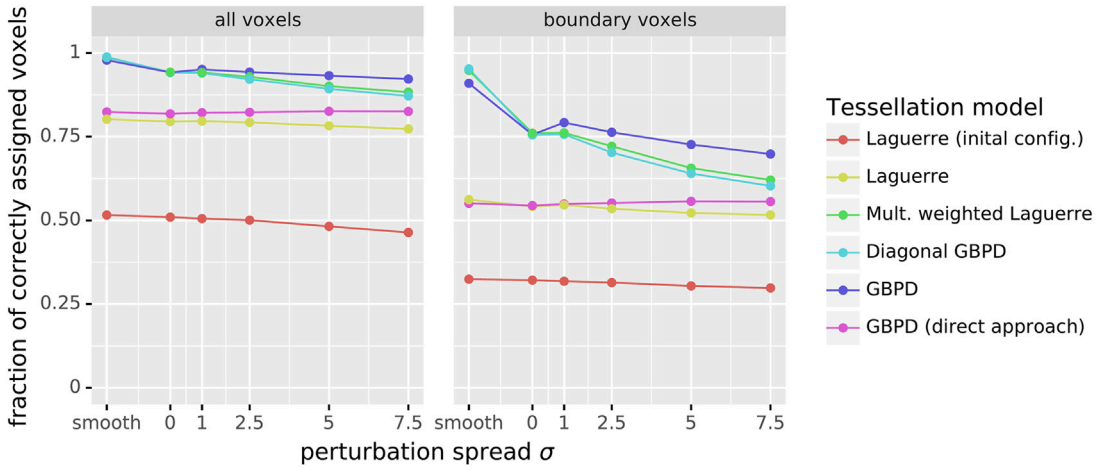


FIGURE 7 | Fraction of correctly assigned voxels plotted against the perturbation spread σ, evaluated for all voxels in the image dataset and for the boundary voxels.

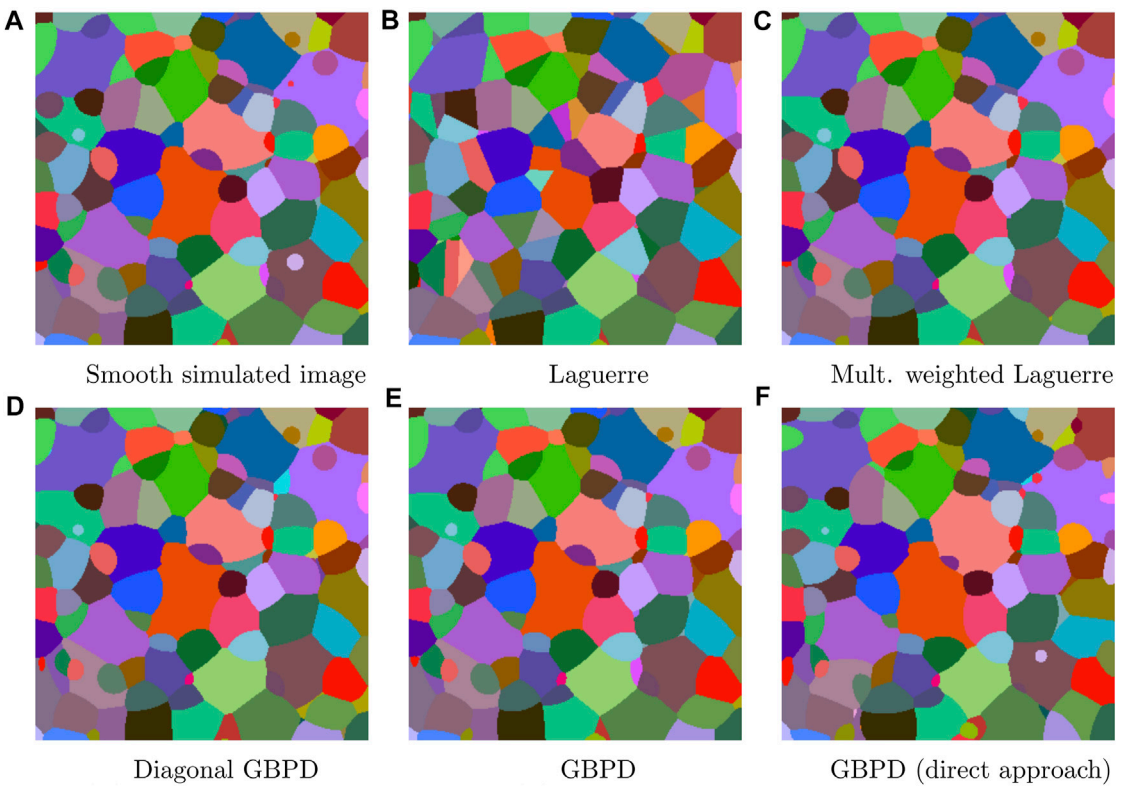


FIGURE 8 | Two-dimensional slices through **(A)** the smooth simulated image I_{sim} and **(B)**–**(F)** tessellations fitted to the perturbed simulated image data I_{pert} with σ = 1. The fits in **(B)**–**(E)** were obtained by the gradient descent-based method described in **Section 2.3**, whereas (**F**) followed from the direct approach described in **Section 2.4.2**.

take non-zero values. Since there were no background voxels in I_{pert}, we set $\mathcal{X}^F = \mathcal{W}_{pert}$.

Several different values of the perturbation spread σ are considered in the present paper. In **Figures 6**, **2D** slices through the resulting perturbed simulated image data are shown. For each of these datasets the same fittings as in the previous section were performed. As the perturbations are assumed to have originated from measurement errors,

TABLE 2 | Values of performance measures for various tessellation models fitted to the perturbed simulated image I_{pert} with $\sigma = 1$ and evaluated with respect to the smooth simulated image I_{sim}.

	F_c	F_c^B	F_0	N_0	$\bar{N}$
Laguerre (initial configuration)	0.505	0.318	0.006	0.062	6.222
Laguerre	0.796	0.546	0.036	0.130	3.236
Multiplicatively weighted Laguerre	0.941	0.761	0.083	0.376	1.118
Diagonal GBPD	0.940	0.757	0.063	0.427	0.980
GBPD	0.951	0.792	0.050	0.486	0.781
GBPD (direct approach)	0.822	0.549	0.002	0.286	1.349

TABLE 3 | Values of performance measures for various tessellation models fitted to the smoothed experimental image data.

	F_c	F_c^B	F_0	N_0	$\bar{N}$
Laguerre (initial configuration)	0.595	0.373	0.000	0.089	3.219
Laguerre	0.845	0.566	0.021	0.293	1.266
Multiplicatively weighted Laguerre	0.936	0.688	0.050	0.446	0.993
Diagonal GBPD	0.955	0.758	0.031	0.543	0.779
GBPD	0.970	0.831	0.047	0.624	0.679
GBPD (direct approach)	0.903	0.619	0.000	0.324	1.163

the fits were compared to the smooth image data instead of evaluating the goodness of fit with respect to the perturbed image data (cf. **Figure 4**). The dependence of the quality of fit on the perturbation spread σ is visualized in **Figure 7**. For $\sigma = 1$, a visual comparison of the fitted tessellations to the smooth simulated image data is shown in **Figure 8**, whereas in **Table 2** numerical fitting results are given.

3.3 Experimental Data

In this section, the fitting method described in **Section 2.3** is tested with realistic grain boundaries. For this purpose, experimental image data obtained from a 3DXRD mapping of an AlCu sample (**Section 2.1**) was used. As with the simulated data of **Section 3.2**, we first consider image data with smooth grain boundaries before tackling a dataset with rougher boundaries (**Figure 1**). The goal is to assess the robustness of the fitting procedure with respect to real-world grain boundary perturbations—originating, e.g., from measurement uncertainties—and also to determine whether the custom of preprocessing raw experimental image data to obtain smoother

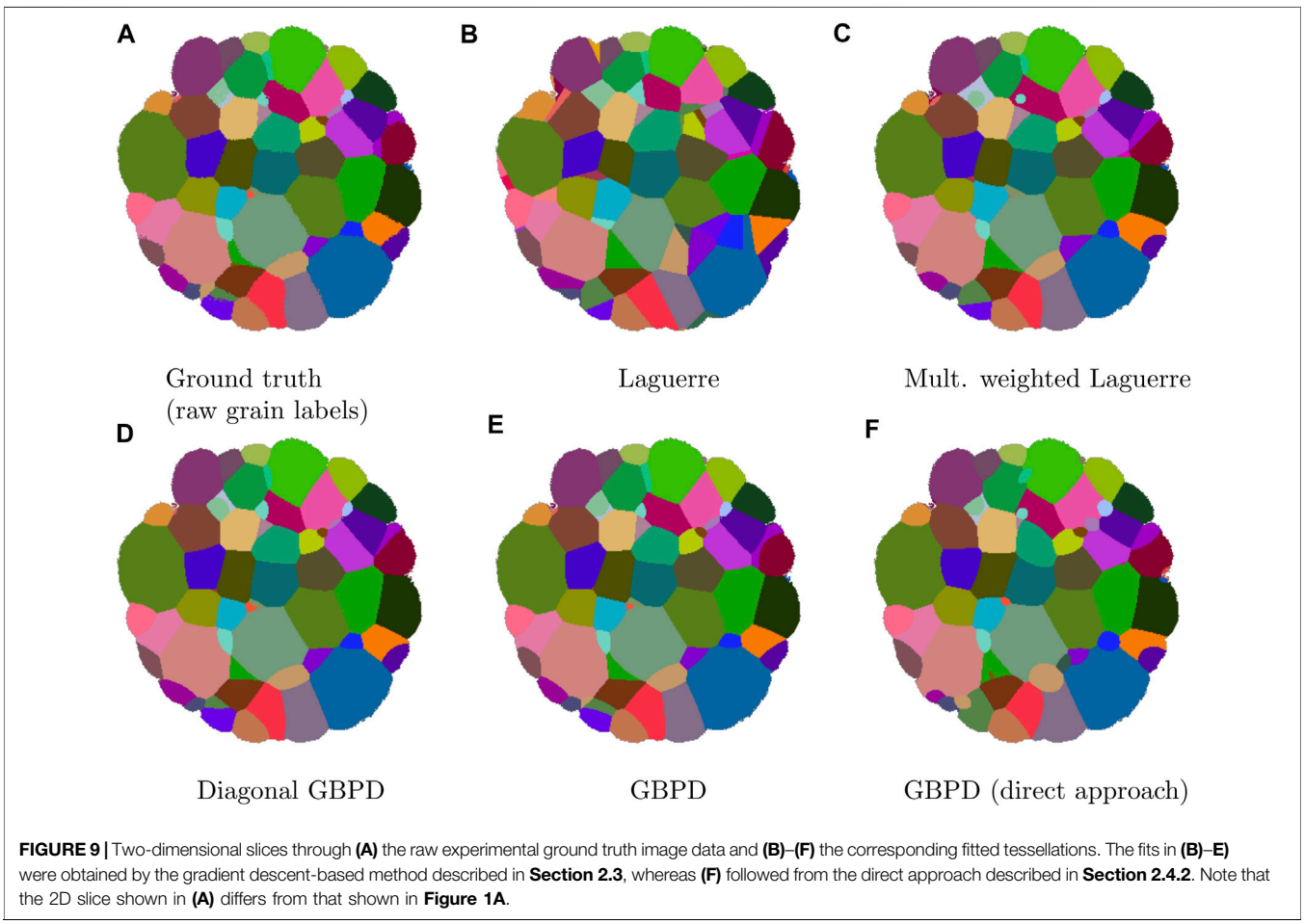


FIGURE 9 | Two-dimensional slices through **(A)** the raw experimental ground truth image data and **(B)**–**(F)** the corresponding fitted tessellations. The fits in **(B)**–**(E)** were obtained by the gradient descent-based method described in **Section 2.3**, whereas **(F)** followed from the direct approach described in **Section 2.4.2**. Note that the 2D slice shown in **(A)** differs from that shown in **Figure 1A**.

TABLE 4 | Values of performance measures for various tessellation models fitted to the raw experimental image data.

	F_c	F_c^B	F_0	N_0	$\bar{N}$
Laguerre (initial configuration)	0.595	0.378	0.000	0.101	3.221
Laguerre	0.846	0.576	0.021	0.288	1.316
Multiplicatively weighted Laguerre	0.935	0.696	0.053	0.443	0.987
Diagonal GBPD	0.951	0.754	0.044	0.510	0.847
GBPD	0.966	0.820	0.045	0.622	0.652
GBPD (direct approach)	0.901	0.627	0.000	0.329	1.146

(and, thus, physically more sensible) grain boundaries prior to fitting is actually necessary.

3.3.1 Smoothed Grain Boundaries

The image $I_{\mathrm{E,smooth}}$ gained from an AlCu sample exhibits smooth grain boundaries after being preprocessed with a phase field algorithm (**Section 2.1**). Tessellation fitting was performed with respect to all foreground voxels $\mathcal{X}^{\mathrm{F}} = \{x \in \mathcal{W}_{\mathrm{E,smooth}} \colon I_{\mathrm{E,smooth}}(x) > 0\}$. The numerical results of the fitting are presented in **Table 3**. Because 2D slices through these fitted tessellations were found to be qualitatively similar to those shown in **Figure 9**—which were obtained from fits to the raw experimental map—we omit the images of tessellations fitted to the smoothed experimental data.

3.3.2 Raw Grain Boundaries

The raw experimental image data, described by the mapping $I_{\mathrm{E,raw}}$ (see **Section 2.1**), was not subjected to the phase field smoothing step following tomographic reconstruction of the 3DXRD measurement. For this reason, artifacts attributable to measurement uncertainties are visible at and near the grain boundaries (cf. **Figure 1A**). In contrast to the fitting of tessellations to the perturbed simulated image data of **Section 3.2.2**, in the present case we evaluate the quality of fit with respect to the same data that was used as the input dataset (i.e., the raw experimental map). One might consider treating $I_{\mathrm{E,smooth}}$ as a good approximation of the true, unobserved grain boundaries; however, since some grains were removed by the smoothing step, the number of grains in the 'reference' dataset would differ from the number of cells in the fitted tessellations. As a result, the performance measures of **Section 3.1** would no longer be applicable. A visual comparison of the fits to the raw experimental image data is shown in **Figure 9**, and the numerical results are presented in **Table 4**. Here, all foreground voxels $\mathcal{X}^{\mathrm{F}} = \{x \in \mathcal{W}_{\mathrm{E,smooth}} \colon I_{\mathrm{E,smooth}}(x) > 0\}$ were considered during the fitting.

4 DISCUSSION

4.1 Fitting Results

As seen in **Section 3.2.1**, the fitted multiplicatively weighted Laguerre tessellation, the diagonal GBPD and the GBPD matched the smooth simulated image data very well, and a near perfect voxelwise accuracy was obtained. Between these tessellation models, there were only minor differences in the goodness of fit. Most notably, the GBPD was slightly worse at reconstructing the boundary voxels (see **Table 1**). Nevertheless, no significant decline in goodness of fit was observed even for the tessellation models that employ more parameters than necessary to describe the ground truth image data (which was generated from a multiplicatively weighted Laguerre tessellation). Furthermore, except for the number of empty cells, the gradient descent-based fitting procedure described in **Section 2.3** achieved a notably better fit than the good results obtained by the direct approach of Teferra and Rowenhorst (2018) (see **Section 2.4.2**). On the other hand, in light of the results given in **Table 1** for the initial Laguerre tessellation and the improved generators that resulted from the fitting procedure of **Section 2.3**, it is clear that the Laguerre tessellations with their flat boundaries (see **Figure 3B**) lacked sufficient flexibility for an accurate reconstruction of the ground truth data. However, the gradient descent-based fitting procedure still managed to bring about a significant improvement compared to the initial generators from the cross-entropy approach. This can likely be traced to the fact that the gradient descent-based approach considers a volume-based objective function (see **Section 2.3**) rather than an interface-based one (see **Section 2.4.1**). In general, we cannot expect to solve such a high-dimensional optimization problem by finding its global optimum; this would be equivalent to finding a perfect reconstruction of the multiplicatively weighted Laguerre tessellation that underlays the ground truth image data. Nevertheless, the fitted tessellations were quite close to the optimum, despite having been obtained by a local optimization method.

When it comes to the perturbed simulated image data investigated in **Section 3.2.2**, it is somewhat surprising how well the voxels of the smooth simulated image data could be reconstructed even from very noisy input image data (see **Figure 7**). As the same effect is observed for all three fitting methods considered in the present paper—i.e., the cross-entropy approach (**Section 2.4.1**), the direct approach (**Section 2.4.2**), and the gradient descent-based method (**Section 2.3**)—we attribute this robustness against perturbations to the inherent smoothing property of tessellations. Another observation that might surprise is the finding that the results for the direct approach were practically independent of the perturbation spread σ, but, just as in the case of the smooth simulated dataset, the gradient descent-based fitting procedure of **Section 2.3** was able to surpass the direct method. The proposed method also achieved a better accuracy of the boundary voxels. In the latter case, however, some degradation could be observed with increasing σ. Some part of this degradation can be explained by the fact that a procedure producing more accurate approximations of the grain boundaries in the first place is going to be more sensitive to noise in the grain boundaries. Naturally, the deterioration in the fit of the boundary voxels also influences the considered grain neighborhood characteristics N_0 and $\bar{N}$—compare **Tables 1**, **2**. Nevertheless, the decline in goodness of fit with σ is still well within reason, given the high noise level of the input image data (see **Figure 6**). In fact, from a visual comparison of the raw experimental image data in **Figure 1B** to the perturbed simulated image data in

TABLE 5 | Runtimes for the fitting of tessellation models to the smoothed experimental image dataset. System A employed only a CPU (Intel Core i7-4770K with four 3.50 GHz cores), whereas System B performed some of the computations on a GPU (CPU: AMD Ryzen 5 3600 with six 3.6 GHz cores; GPU: NVIDIA GeForce RTX 3060).

Tessellation model	System A	System B
Laguerre	6:07 h	1:12 h
Multiplicatively weighted Laguerre	6:27 h	1:13 h
Diagonal GBPD	8:26 h	1:19 h
GBPD	16:42 h	3:58 h

Figure 6, it can be seen that the case with the lowest level of noise (i.e., $\sigma = 0$) comes closest to the considered experimental data. This indicates that the gradient descent-based fitting procedure described in **Section 2.3** is not only able to deal with the present level of measurement artifacts but also with much noisier scenarios.

For the experimental image data investigated in **Sections 3.3.1, 3.3.2**, quite high voxelwise accuracies F_c and F_c^B were observed, particularly for the more sophisticated tessellation models, as they are better at describing non-convex grain morphologies. Despite this fact, with respect to F_c and F_c^B the simpler Laguerre tessellation does a better job of fitting the experimental datasets than the simulated image—compare **Tables 3**, **4**; **Table 1**. The same is true when considering the grain neighborhood characteristics N_0 and $\bar{N}$. As could already be anticipated from our analysis of the perturbed simulated image data, there was no significant difference in goodness of fit to the smoothed versus to the raw experimental datasets (for all performance measures). This leads us to conclude that there is no benefit to smoothing the grain boundaries in experimental image data prior to tessellation fitting.

One of the main advantages of the gradient descent-based fitting procedure described in **Section 2.3** compared to other optimization approaches lies in its runtime performance. This comes from the reformulation of the objective function and the resulting ability to employ efficient gradient descent optimization algorithms. Another reason is the fact that the objective function and its gradient can both be computed on multiple CPU cores or even on GPUs. This opens up the possibility of reducing the (wall clock) time for the fitting procedure by employing hardware having a higher degree of parallelism (such as GPUs), which would not be so readily feasible for sequential fitting procedures. As runtime benchmarks are notorious for their dependence on a multitude of factors (in this case, for example, on the number of generators/grains, the number of voxels, the computer hardware, etc.), the runtimes quoted in **Table 5** for the fitting of tessellations to the smoothed experimental image data should be taken with a grain of salt. In this particular case, the slower fitting of the GBPD model than the other tessellations can be attributed not only to the increased number of parameters but, more importantly, to the fact that a less efficient software implementation had to be used to compute the GBPD distance function. That being said, to achieve such a high goodness of fit for a dataset with $531 \times 321 \times 321$ voxels and 938 grains, the runtimes are quite competitive, especially for the other tessellation models.

4.2 Other Fitting Approaches in the Literature

The fitting results for the direct approach of Teferra and Rowenhorst (2018) (**Section 2.4.2**) were discussed in the previous section. For all datasets, the direct approach delivered reasonably good fits of GBPDs, but they were consistently worse than those obtained by the gradient descent-based fitting procedure of **Section 2.3**. The only exception here is the fact that the direct method produced very few empty cells. Its main benefit, however, is the very short runtime of only a couple of seconds. Therefore, the direct approach is a good choice if a tessellation must be found quickly, but if the focus lies on the quality of fit, the gradient descent-based method developed in the present paper may be more suitable.

In Šedivý et al. (2016), still another method for fitting GBPDs was proposed, in which—similar to the present paper—a volume-based objective function is optimized. However, instead of a gradient descent method, Šedivý et al. employed a global stochastic optimization technique, the simulated annealing algorithm. As the name implies, this technique is inspired by the annealing (heat treatment) of metals. Specifically, during each iteration of the optimization, a random modification of the previous generators is proposed. These changes are accepted depending on whether the objective function is improved as well as on the current value of the 'temperature.' Here, the parameter 'temperature' governs the likelihood that a change in generators is accepted even though it leads to a worse value of the objective function. As the temperature is decreased over the course of the optimization, the probability increases that only improvements in the fit are accepted. In Šedivý et al. (2016), this fitting procedure was applied to both simulated and experimental image data. For the former case, which was a realization of a random GBPD, the quality of fit of the simulated annealing approach ($F_c = 0.975$, $N_0 = 0.604$, $\bar{N} = 0.57$) was similar to the results obtained in **Section 3.2.1** of the present paper. However, since the simulated datasets were drawn from two different models (GBPD vs. multiplicatively weighted Laguerre tessellation), direct comparison warrants caution. Regarding the runtime of the fitting routine, it is mentioned in Šedivý et al. (2016) that roughly 19 h were needed to carry out 10 million optimization iterations on an Intel Xeon E3-1240 CPU with four 3.4 GHz cores (slightly slower than the Intel Core i7-4770K used in the present paper). The procedure stopped after 13 million iterations, which corresponds to a total runtime of about 24 h. When comparing this to the 16:42 h that the same fitting took in the present paper (see **Table 5**), we note that the dataset considered by Šedivý et al. had only about a 10th the size (180^3 vs. $531 \times 321 \times 321$ voxels) but more than twice as many grains (1894 *vs*. 938 grains). A direct comparison to their experimental dataset is omitted, as the two datasets are quite different. In summary, the goodness of fit achieved by the simulated annealing approach applied to simulated data was similar to that achieved by the gradient descent-based fitting

procedure of the present paper, but the runtime reported in Šedivý et al. (2016) was significantly longer.

5 CONCLUSION

In this paper, a novel method for fitting distance-based tessellations, such as the Laguerre tessellation or generalized balanced power diagrams, to 3D image data was developed. With this approach, it is possible to obtain parametric representations of the curved grain boundaries of real polycrystalline materials. The method employs efficient gradient descent optimization, with the technique proving to be capable of reconstructing a tessellation from its discretized image. Nearly identical fits were obtained when the procedure was applied to smoothed versus raw experimental data. From the observed robustness against noise in the input image data, we conclude that there is no benefit to smoothing an experimental image dataset prior to fitting it with a tessellation model.

The proposed method could facilitate the study of physical phenomena like curvature-driven grain growth, in which smooth representations of grain boundaries—such as those provided by tessellation models—are required for accurate calculations. Furthermore, the fitted tessellations could serve as the basis for stochastic models of polycrystalline microstructures. These models could potentially enable researchers to investigate mechanical properties of material samples *in silico* instead of through resource-intensive laboratory experimentation.

DATA AVAILABILITY STATEMENT

The datasets presented in this article are not readily available because they are part of ongoing research. Requests to access the datasets should be directed to LP, lukas.petrich@uni-ulm.de.

AUTHOR CONTRIBUTIONS

Tomographic image data of the AlCu specimen were provided by MW and CK. The basic idea of the gradient-based fitting approach came from OF. The software implementation and computer experiments were performed by LP. Main parts of the paper were written by LP and CK. All authors discussed the results and contributed to writing of the manuscript. CK and VS designed and supervised the research.

FUNDING

The authors gratefully acknowledge partial financial support provided by the Deutsche Forschungsgemeinschaft (DFG) through projects KR 1658/9-1 and SCHM 997/41-1. In addtion, The authors are grateful to the Japan Synchrotron Radiation Research Institute for the allotment of beamtime on beamline BL20XU of SPring-8 (Proposal 2015A1580).

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org.

Allen, J. M., Weddle, P. J., Verma, A., Mallarapu, A., Usseglio-Viretta, F., Finegan, D. P., et al. (2021). Quantifying the Influence of Charge Rate and Cathode-Particle Architectures on Degradation of Li-Ion Cells through 3D Continuum-Level Damage Models. *J. Power Sourc.* 512, 230415. doi:10.1016/j.jpowsour.2021.230415

Alpers, A., Brieden, A., Gritzmann, P., Lyckegaard, A., and Poulsen, H. F. (2015). Generalized Balanced Power Diagrams for 3D Representations of Polycrystals. *Phil. Mag.* 95, 1016–1028. doi:10.1080/14786435.2015.1015469

Altendorf, H., Latourte, F., Jeulin, D., Faessel, M., and Saintoyant, L. (2014). 3D Reconstruction of a Multiscale Microstructure by Anisotropic Tessellation Models. *Image Anal. Stereol* 33, 121–130. doi:10.5566/ias.v33.p121-130

Audet, C., and Hare, W. (2017). *Derivative-Free and Blackbox Optimization*. Springer.

Aurenhammer, F., Klein, R., and Lee, D.-T. (2013). *Voronoi Diagrams And Delaunay Triangulations*. World Scientific.

Bourne, D. P., Kok, P. J. J., Roper, S. M., and Spanjer, W. D. T. (2020). Laguerre Tessellations and Polycrystalline Microstructures: a Fast Algorithm for Generating Grains of Given Volumes. *Phil. Mag.* 100, 2677–2707. doi:10.1080/14786435.2020.1790053

Burnett, T. L., Kelley, R., Winiarski, B., Contreras, L., Daly, M., Gholinia, A., et al. (2016). Large Volume Serial Section Tomography by Xe Plasma FIB Dual Beam Microscopy. *Ultramicroscopy* 161, 119–129. doi:10.1016/j.ultramic.2015.11.001

Chiu, S. N., Stoyan, D., Kendall, W. S., and Mecke, J. (2013). *Stochastic Geometry and its Applications*. 3 edn. J. Wiley & Sons.

Dake, J. M., Oddershede, J., Sørensen, H. O., Werz, T., Shatto, J. C., Uesugi, K., et al. (2016). Direct Observation of Grain Rotations during Coarsening of a Semisolid Al-Cu alloy. *Proc. Natl. Acad. Sci. USA* 113, E5998–E6006. doi:10.1073/pnas.1602293113

Furat, O., Petrich, L., Finegan, D. P., Diercks, D., Usseglio-Viretta, F., Smith, K., et al. (2021). Artificial Generation of Representative Single Li-Ion Electrode Particle Architectures from Microscopy Data. *Npj Comput. Mater.* 7, 105. doi:10.1038/s41524-021-00567-9

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

Kingma, D. P., and Ba, J. (2015). "Adam: a Method for Stochastic Optimization," in 3rd International Conference On Learning Representations, ICLR 2015. Editors Y. Bengio and Y. LeCun.

Krill III, C. E., III, and Chen, L.-Q. (2002). Computer Simulation of 3-D Grain Growth Using a Phase-Field Model. *Acta Materialia* 50, 3059–3075. doi:10.1016/s1359-6454(02)00084-8

Lautensack, C., and Zuyev, S. (2008). Random Laguerre Tessellations. *Adv. Appl. Probab.* 40, 630–650. doi:10.1239/aap/1222868179

Liebscher, A. (2015). Laguerre Approximation of Random Foams. *Phil. Mag.* 95, 2777–2792. doi:10.1080/14786435.2015.1078511

Lyckegaard, A., Lauridsen, E. M., Ludwig, W., Fonda, R. W., and Poulsen, H. F. (2011). On the Use of Laguerre Tessellations for Representations of 3D Grain Structures. *Adv. Eng. Mater.* 13, 165–170. doi:10.1002/adem.201000258

Møller, J. (1994). *Lectures on Random Voronoi Tessellations*. Springer.

Okabe, A., Boots, B., Sugihara, K., and Chiu, S. N. (2000). *Spatial Tessellations Concepts and Applications of Voronoi Diagrams*. 2 edn. J. Wiley & Sons.

Petrich, L., Staněk, J., Wang, M., Westhoff, D., Heller, L., Šittner, P., et al. (2019). Reconstruction of Grains in Polycrystalline Materials from Incomplete Data Using Laguerre Tessellations. *Microsc. Microanal* 25, 743–752. doi:10.1017/s1431927619000485

Poulsen, H. F. (2004). *Three-Dimensional X-Ray Diffraction Microscopy: Mapping Polycrystals and Their Dynamics*. Springer.

Quey, R., and Renversade, L. (2018). Optimal Polyhedral Description of 3D Polycrystals: Method and Application to Statistical and Synchrotron X-ray

Diffraction Data. *Comp. Methods Appl. Mech. Eng.* 330, 308–333. doi:10.1016/j.cma.2017.10.029

Raabe, D. (1998). *Computational Materials Science*. Wiley VCH.

Rubinstein, R. Y., and Kroese, D. P. (2004). *The Cross-Entropy Method*. Springer.

Russ, J. C., and Neal, F. B. (2017). *The Image Processing Handbook*. 7 edn. Boca Raton: CRC Press.

Schmidt, S. (2014). GrainSpotter: a Fast and Robust Polycrystalline Indexing Algorithm. *J. Appl. Cryst.* 47, 276–284. doi:10.1107/s1600576713030185

Schmidt, S. (2005). GrainSweeper. Available at: https://svn.code.sf.net/p/fable/code/GrainSweeper. [Accessed August 17, 2021].

Schwartz, A., Kumar, M., Adams, B., and Field, D. (2009). *Electron Backscatter Diffraction in Materials Science*. Springer.

Šedivý, O., Brereton, T., Westhoff, D., Polívka, L., Beneš, V., Schmidt, V., et al. (2016). 3D Reconstruction of Grains in Polycrystalline Materials Using a Tessellation Model with Curved Grain Boundaries. *Phil. Mag.* 96, 1926–1949.

Šedivý, O., Westhoff, D., Kopeček, J., Krill, C. E., III, and Schmidt, V. (2018). Data-driven Selection of Tessellation Models Describing Polycrystalline Microstructures. *J. Stat. Phys.* 172, 1223–1246.

Spettl, A., Brereton, T., Duan, Q., Werz, T., Krill, C. E., III, Kroese, D. P., et al. (2016). Fitting Laguerre Tessellation Approximations to Tomographic Image Data. *Phil. Mag.* 96, 166–189. doi:10.1080/14786435.2015.1125540

Teferra, K., and Graham-Brady, L. (2015). Tessellation Growth Models for Polycrystalline Microstructures. *Comput. Mater. Sci.* 102, 57–67. doi:10.1016/j.commatsci.2015.02.006

Teferra, K., and Rowenhorst, D. J. (2018). Direct Parameter Estimation for Generalised Balanced Power Diagrams. *Phil. Mag. Lett.* 98, 79–87. doi:10.1080/09500839.2018.1472399

Wang, K. G., and Glicksman, M. (2007). "Ostwald Ripening in Materials Processing," in *Materials Processing Handbook*. Editors J. R. Groza, J. F. Shackelford, E. J. Lavernia, and M. T. Powers (Boca Raton: CRC Press), 75–94. doi:10.1201/9781420004823.ch5

Westhoff, D., Skibinski, J., Šedivý, O., Wysocki, B., Wejrzanowski, T., and Schmidt, V. (2018). Investigation of the Relationship between Morphology and Permeability for Open-Cell Foams Using Virtual Materials Testing. *Mater. Des.* 147, 1–10. doi:10.1016/j.matdes.2018.03.022

Zaefferer, S., Wright, S. I., and Raabe, D. (2008). Three-dimensional Orientation Microscopy in a Focused Ion Beam-Scanning Electron Microscope: a New Dimension of Microstructure Characterization. *Metall. Mater. Trans. A.* 39, 374–389. doi:10.1007/s11661-007-9418-9

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers in Materials

ORIGINAL RESEARCH
published: 23 December 2021
doi: 10.3389/fmats.2021.786502

Large-Scale Statistical Learning for Mass Transport Prediction in Porous Materials Using 90,000 Artificially Generated Microstructures

Benedikt Prifling[1*†], *Magnus Röding*[2,3†], *Philip Townsend*[2,3], *Matthias Neumann*[1] *and Volker Schmidt*[1]

[1]Institute of Stochastics, Ulm University, Ulm, Germany, [2]Agriculture and Food, Bioeconomy and Health, Research Institutes of Sweden, Gothenburg, Sweden, [3]Department of Mathematical Sciences, Chalmers University of Technology and University of Gothenburg, Gothenburg, Sweden

Effective properties of functional materials crucially depend on their 3D microstructure. In this paper, we investigate quantitative relationships between descriptors of two-phase microstructures, consisting of solid and pores and their mass transport properties. To that end, we generate a vast database comprising 90,000 microstructures drawn from nine different stochastic models, and compute their effective diffusivity and permeability as well as various microstructural descriptors. To the best of our knowledge, this is the largest and most diverse dataset created for studying the influence of 3D microstructure on mass transport. In particular, we establish microstructure-property relationships using analytical prediction formulas, artificial (fully-connected) neural networks, and convolutional neural networks. Again, to the best of our knowledge, this is the first time that these three statistical learning approaches are quantitatively compared on the same dataset. The diversity of the dataset increases the generality of the determined relationships, and its size is vital for robust training of convolutional neural networks. We make the 3D microstructures, their structural descriptors and effective properties, as well as the code used to study the relationships between them available open access.

Keywords: diffusivity, permeability, virtual materials testing, deep learning, porous materials, mass transport, structure-property relationship

OPEN ACCESS

Edited by:
Norbert Huber,
Helmholtz-Zentrum Hereon, Germany

Reviewed by:
Yongxing Shen,
Shanghai Jiao Tong University, China
Wenxiang Xu,
Hohai University, China

****Correspondence:***
Benedikt Prifling
benedikt.prifling@uni-ulm.de

†These authors have contributed equally to this work

Specialty section:
This article was submitted to Computational Materials Science, a section of the journal Frontiers in Materials

Received: *30 September 2021*
Accepted: *27 October 2021*
Published: *23 December 2021*

Citation:
Prifling B, Röding M, Townsend P, Neumann M and Schmidt V (2021) Large-Scale Statistical Learning for Mass Transport Prediction in Porous Materials Using 90,000 Artificially Generated Microstructures. Front. Mater. 8:786502. doi: 10.3389/fmats.2021.786502

1 INTRODUCTION

The performance of functional materials is significantly influenced by the underlying 3D structure and morphology (Torquato, 2002; Willot and Forest, 2018). Thus, optimizing 3D microstructures for high performance in particular applications is one of the main goals in many branches of materials research. Typically, the amount of candidate materials structures is enormous and beyond the reach of conventional experimental screening (Dunn et al., 2020; Saunders et al., 2021) (for example, the number of potential pharmacologically active molecules is at least 10^{20} and theoretically estimated at 10^{60}) (Hoffmann and Gastreich, 2019). To overcome this limitation, virtual materials testing is an approach of increasing importance, where mathematical models are used for both the analysis of artificially generated materials structures, as well as for the investigation of their effective properties. By systematically varying the model parameters, a large number of virtual but realistic 3D microstructures can be drawn from stochastic models just at the cost of computer simulations.

These structures serve as geometry input for spatially-resolved numerical simulations of effective properties. In this way, together with the computation of various descriptors of the 3D microstructures, quantitative microstructure-property relationships can be established.

Typically, the mathematical models of random porous microstructures with two phases, solid and pores, are located in the (continuous) Euclidean space. If we restrict ourselves to discrete representations on a computational grid, say a cube of N^3 lattice points, the number of theoretically possible structures is 2^{N^3}. Although many of those structures are not suitable to describe real materials, still a huge number of structural scenarios remains for which this is the case. In virtual materials testing, 3D microstructures are typically explored by means of parametric models that span some region of interest, using tools from mathematical morphology and stochastic geometry (Lantuéjoul, 2002; Chiu et al., 2013; Jeulin, 2021). These microstructure models can be calibrated with experimental data gained, e.g., by tomographic imaging or simply be inspired by experimentally observed structures. There are numerous examples for artificial generation and virtual testing of functional materials, including applications for lithium ion batteries (Feinauer et al., 2015; Hein et al., 2016; Westhoff et al., 2018a; Prifling et al., 2019; Hein et al., 2020; Allen et al., 2021; Prifling et al., 2021a; Birkholz et al., 2021; Furat et al., 2021), solid oxide fuel cells (Abdallah et al., 2016; Neumann et al., 2016; Moussaoui et al., 2018), amorphous silica (Prifling et al., 2021b), gas diffusion electrodes (Neumann et al., 2019a), open-cell foams (Westhoff et al., 2018b), organic semiconductors (Westhoff et al., 2015), mesoporous alumina (Wang et al., 2015), solar cells (Stenzel et al., 2011), electric double-layer capacitors (Prill et al., 2017), platelet-filled composites (Röding et al., 2018), fiber-based materials (Röding et al., 2016; Townsend et al., 2021), and pharmaceutical coatings for controlled drug release (Barman et al., 2019).

The focus of the present paper is on quantifying the influence of 3D microstructure on mass transport properties of porous materials and, specifically, in effective diffusivity and fluid permeability. There are numerous microstructural descriptors that are useful for the prediction of those properties. The most fundamental one is porosity, followed by specific surface area i.e., the pore-solid interface area per unit volume. However, there are many more sophisticated structural descriptors considered in literature, e.g., various measures of tortuosity, pore size distributions, constrictivity, and two-point correlation functions. Various combinations of such descriptors have been used to establish microstructure-property relationships of varying complexity. The most well-known relationship for permeability is perhaps the Kozeny-Carman equation which in its basic form only uses porosity and specific surface area to describe the underlying 3D microstructure (Kozeny, 1927; Carman, 1937). For effective diffusivity, there exist equally simple formulas, in some cases involving only porosity (Masaro and Zhu, 1999). Later on, analytical expressions of this type have been developed where more complex structural descriptors have been taken into account, such as constrictivity and tortuosity (Barman et al., 2019; Neumann. et al., 2020), as well as two-point correlation functions (Berryman, 1985; Torquato, 1991; Jiao and Torquato, 2012; Liasneuski et al., 2014; Hlushkou et al., 2015; Ma and Torquato, 2018). Some of these formulas still are sufficiently simple to allow a certain physical interpretation. On the other hand, numerous attempts have been made to use high-dimensional regression methods and machine learning in order to obtain more accurate prediction models, where the descriptors of 3D microstructures and mass transport properties, as input and output variables, still have physical underpinnings. But the relationships derived between them are, in a sense, more data-driven and less determined by the underlying physics, where this effect amplifies with increasing dimension and complexity of the prediction model. For example, in van der Linden et al. (2016), the permeability and 27 different microstructural descriptors were computed for 536 granular materials structures. This information was then used to develop (log-)linear relationships and find relevant subsets of descriptors through variable selection procedures. In Stenzel et al. (2017), effective conductivity (mathematically equivalent to effective diffusivity) and numerous structural descriptors including constrictivity and tortuosity were computed for 8,119 microstructures, where conventional regression, random forests, and artificial neural networks (ANNs) were used for prediction. In Röding et al. (2020), permeability, tortuosity and two-point correlation functions were computed for 30,000 structures, where log-linear regression and ANNs were used for prediction. Although machine learning regression using ANNs is less transparent compared to analytical prediction formulas and hence less interpretable, the benefit of this approach is that arbitrarily complex relationships can be represented by a feed-forward network due to the universal approximation theorem (Cybenko, 1989; Hornik et al., 1989). Hence, machine learning regression can be considered a data-science approach that leads to insight into new relationships and into which descriptors are most useful for prediction (Umehara et al., 2019; Röding et al., 2020). A third option is the prediction of effective properties using convolutional neural networks (CNNs). Note that conventional ANNs learn to perform nonlinear regression using predefined descriptors, whereas CNNs perform their own descriptor extraction directly from the microstructure, expressed as nonlinear compositions of convolution filters. These are then used as input to a conventional ANN that performs the regression (Kawaguchi et al., 2021). CNNs have been used for predicting both permeability (Srisutthiyakorn, 2016; Wu et al., 2018; Araya-Polo et al., 2019; Sudakov et al., 2019; Graczyk and Matyka, 2020; Kamrava et al., 2020) and effective diffusivity (Wu et al., 2019; Wang et al., 2020), although in many cases with small datasets and/or only with 2D structures. Generally, CNNs have the tendency to be even less transparent than ANNs in terms of understanding how the prediction works.

In the present paper, we investigate an extremely broad range of virtual two-phase microstructures which are drawn from nine different stochastic models. For each model type, 10,000 microstructures are generated for different specifications of model parameters leading, e.g., to varying porosities and length scales. Hence, in total, our study comprises 90,000 microstructures. For each structure, both effective diffusivity

and permeability are computed. Furthermore, a multitude of structural descriptors is determined, like porosity, specific surface area, median pore radius, radius of the characteristic bottleneck, constrictivity, tortuosity (and its distribution), chord lengths (and their distribution), spherical contact distribution, and two-point correlation functions. The dimension of the largest possible descriptor space is equal to 236. As already mentioned above, we utilize analytical prediction formulas as well as ANNs and CNNs to establish microstructure-property relationships. Due to the large diversity of the dataset considered in this paper, the determined relationships are not specific to any particular morphology, but rather quite generally applicable. For CNNs, in particular, large amounts of data are needed to ensure robust training and good generalization to new data. Thus, to the best of our knowledge, the data considered in this paper is the largest and most diverse dataset for the study of effective diffusivity and permeability, which has been reported so far in the literature. To facilitate further development of microstructure-property relationships and their predictive power, the microstructures, their corresponding morphological and effective properties, and the code used to study microstructure-property relationships are available open access (Prifling et al., 2021c). Note that besides the comprehensive database for porous materials considered in the present paper, there are several further research activities of this kind for other types of materials, e.g., for composite particulate materials occurring in crushed ores (Ditscherlein et al., 2021).

The rest of this paper is organized as follows. In **Section 2** the definitions of various structural descriptors are explained as well as their estimation from 3D image data. They are used as input variables for establishing microstructure-property relationships. Then, in **Section 3**, two descriptors of effective properties, namely diffusivity and permeability of the pore space, are presented, which serve as output variables. **Section 4** introduces the stochastic models used for the artificial generation of 3D microstructures, whereas in **Section 5** three different approaches are explained which are applied to establish the microstructure-property relationships. Finally, a discussion of the results obtained in this paper is provided in **Section 6**.

2 STRUCTURAL DESCRIPTORS AND THEIR ESTIMATION

The goal of this section is to explain the definitions of various structural descriptors considered in this paper as well as methods for their estimation from simulated 3D image data, where the underlying stochastic 3D microstructure models described in **Section 4** are stationary and isotropic. This implies that the pore space is a stationary and isotropic random set as well, which will be denoted by Ξ in the following, i.e., its distribution is invariant with respect to translations of and rotations around the origin (Lantuéjoul, 2002; Chiu et al., 2013; Jeulin, 2021). Note that all microstructures that are drawn from these stochastic models fulfill periodic boundary conditions in x-, y- and z-directions. This is taken into account when estimating the structural descriptors as presented below. In addition to simple scalar descriptors of Ξ such as volume fraction, specific surface area or constrictivity, we also consider more complex descriptors like the chord length distribution, the spherical contact distribution and the distribution of geodesic tortuosity. In practice, we represent these distributions by their quantiles, starting from 5%- up to 95%-quantiles in 5% steps.

2.1 Porosity

To begin with, we consider one of the simplest but most important structural descriptors, namely the porosity $\varepsilon \in [0, 1]$, i.e., the volume fraction of the random pore space $\Xi \subset \mathbb{R}^3$, where $\varepsilon = \mathbb{E}(\nu_3(\Xi \cap [0,1]^3))$ and ν_3 denotes the three-dimensional Lebesgue measure. This characteristic can be easily estimated from 3D image data by the point-count method (Chiu et al., 2013). Note that this estimation method has obviously not to be adapted further to account for periodic boundary conditions.

2.2 Specific Surface Area

A further relevant scalar descriptor is the specific surface area of the interface between solid and pores, denoted by $S > 0$. It is defined as the expected surface area of the pore space per unit volume. In order to estimate this characteristic from voxelized binary images, we compute weighted sums by considering local $2 \times 2 \times 2$ voxel configurations, where we use the weights proposed by Schladitz et al. (2007). Periodic boundary conditions are taken into account by a circular padding of size one in each direction using the Matlab command "padarray" (MATLAB, 2021).

2.3 Geodesic Tortuosity

Next, we consider the geodesic tortuosity of the pore space, which is a purely geometric quantity that significantly influences effective properties (Stenzel et al., 2016; Barman et al., 2019; Neumann. et al., 2020; Holzer et al., 2021). It is important to point out that different concepts of tortuosity exist in the literature (Clennell, 1997; Ghanbarian et al., 2013; Tjaden et al., 2018). The general idea is to consider shortest paths from a predefined starting plane to a target plane, which have to be completely contained in the pore phase. This means, that we consider the shortest path with respect to the geodesic metric of the pore space (Lantuéjoul and Maisonneuve, 1984). Then, the distribution of the lengths τ_{geo} of those shortest paths, divided by the distance of the two planes, is denoted by $d(\tau_{\text{geo}})$. Recall that, in this paper, the distribution $d(\tau_{\text{geo}})$ is represented by 19 quantiles, starting from 5%- up to 95%-quantiles in 5% steps. Furthermore, mean geodesic tortuosity, denoted by $m(\tau_{\text{geo}}) \geq 1$, is defined as the mean value of the random variable τ_{geo}, whereas its standard deviation is denoted by $\sigma(\tau_{\text{geo}})$. A more formal definition of these quantities within the framework of random sets can be found in Neumann et al. (2019b), whereas a slightly different definition of geodesic tortuosity is presented in Barman et al. (2019). Regarding the estimation of the distribution of geodesic tortuosity from 3D image data, we compute the shortest paths with the Dijkstra algorithm (Jungnickel, 2007), where the transport direction is from low x-values to high x-values. Note that the transport paths are allowed to "leave" the sampling window in the y- and z-directions in order to account for periodic boundary conditions.

2.4 Constrictivity

In order to define the constrictivity of the pore space, denoted by $\beta \in [0, 1]$, we first describe the continuous pore size distribution (CPSD) as well as the concept of simulated mercury intrusion porosimetry (MIP). The continuous pore size distribution CPSD: $[0, \infty) \to [0, 1]$ has been introduced in Münch and Holzer (2008) and is based on a morphological opening (Serra, 1982; Soille, 2003). More precisely, for each radius $r \geq 0$, the value $\mathrm{CPSD}(r)$ is given by the volume fraction of the pore space that can be covered by a potentially overlapping union of spheres with radius r, which have to be completely contained in the pore space. The radius $r_{\max}$ is now defined as the radius for which $\mathrm{CPSD}(r_{\max})$ equals half of the porosity, i.e., $\mathrm{CPSD}(r_{\max}) = \mathrm{CPSD}(0)/2$. The estimation of $\mathrm{CPSD}(r)$ from image data relies on the computation of the so-called Euclidean distance transform (Maurer et al., 2003; Soille, 2003), which allows to perform an erosion followed by a dilation using a ball with radius r as structuring element. In order to take periodic boundary conditions into account, the computation of the Euclidean distance transform has to be adapted as follows. At first, a circular padding is applied to the binary image leading to a 3D image, which is twice as large in each of the three directions. In a second step, the Euclidean distance transform is computed on the enlarged image. Afterwards, a 3D cutout with the same size as the original image, whose centroid corresponds to the centroid of the enlarged Euclidean distance transform, is cut out of the enlarged Euclidean distance transform. Finally, a circular shift is applied to this cutout using the "circshift" command in Matlab in order to restore the original spatial arrangement of the voxels (MATLAB, 2021).

The concept of simulated mercury intrusion porosimetry (MIP) is similar to that of the CPSD, except that MIP depends in general on a predefined direction. More precisely, the union of potentially overlapping balls of radius r considered above has to form an intrusion from the given direction, when computing the correspondingly modified volume fraction $MIP(r)$. This accounts for bottlenecks in the pore space such that the values of the function MIP: $[0, \infty) \to [0, 1]$ are always less or equal than the corresponding values of CPSD: $[0, \infty) \to [0, 1]$, i.e., $\mathrm{MIP}(r) \leq \mathrm{CPSD}(r)$ for all $r \geq 0$. In order to simulate the intrusion from low x-values to high x-values, the Hoshen-Kopelman algorithm is used (Hoshen and Kopelman, 1976), where it is straightforward to apply periodic boundary conditions in y- and z-directions. Note that due to our stationarity and isotropy assumptions, the values of MIP do not depend on the choice of the predefined direction. However, within the present paper, this quantity is computed in x-direction, where the intrusion starts at low x-values. Analogously to $r_{\max}$, the radius $r_{\min}$ is defined as the radius for which $\mathrm{MIP}(r_{\min})$ equals half the porosity, i.e., $\mathrm{MIP}(r_{\min}) = \mathrm{CPSD}(0)/2$.

The constrictivity of the pore space is now given by $\beta = (\frac{r_{\min}}{r_{\max}})^2 \in [0, 1]$ (Münch and Holzer, 2008), where a constrictivity of one occurs if there are no constrictions within the pore space at all. The lower the constrictivity, the more the transport within the pore space is hindered by bottlenecks. This geometric characteristic has turned out to significantly influence effective properties of functional materials (Holzer et al., 2013). A formal definition of constrictivity within the framework of random sets can be found in Neumann et al. (2019b).

2.5 Chord Length Distribution

The chord length distribution of the pore space, which is modelled by a stationary random set $\Xi \subset \mathbb{R}^3$, is defined as follows (Ohser and Mücklich, 2000; Ohser and Schladitz, 2009; Chiu et al., 2013). Given a predefined direction $\varphi \in [0, \frac{\pi}{2}] \times [0, 2\pi)$, the chord length distribution of the random set Ξ in direction φ is the distribution of the length L of the so-called typical line segment (selected at random) in $\Xi \cap \ell$, where ℓ denotes the line passing through the origin in direction φ. The distribution of L is denoted by $d(L)$ and, again, represented by 19 quantiles, starting from 5%- up to 95%-quantiles in 5% steps. Note that $d(L)$ does not depend on the particular choice of φ, when considering stationary and isotropic random sets. The mean and the standard deviation of the chord length distribution are denoted by $m(L)$ and $\sigma(L)$, respectively. In the present paper, observing $\Xi \cap \ell \cap W$ within some sampling window $W \subset \mathbb{R}^3$, the chord length distribution is estimated by counting subsequent voxels belonging to the pore space along the x-axis and computing the empirical distribution function of the lengths of these voxel sequences. Note that periodic boundary conditions can be simply accounted for by merging the first and the last chord in $\Xi \cap \ell \cap W$, provided that both chords belong to the pore space.

2.6 Spherical Contact Distribution

Consider the (random) distance H from the typical point of $\Xi^c = \mathbb{R}^3 \backslash \Xi$ to the nearest point within Ξ. The function F_H: $[0, \infty) \to [0, 1]$ with $F_H(r) = \mathbb{P}(H \leq r)$ for each $r \geq 0$ is called the spherical contact distribution function of Ξ. Note that the values of F_H are given by

$$F_H(r) = 1 - \frac{\mathbb{P}(\Xi \cap B(o, r) = \varnothing)}{1 - \varepsilon} \qquad \text{for each } r \geq 0,$$

where $B(o, r) \subset \mathbb{R}^3$ denotes the closed ball with radius r centered at the origin (Chiu et al., 2013). The mean, standard deviation and distribution of H are denoted by $m(H)$, $\sigma(H)$ and $d(H)$, respectively. These quantities can be estimated from voxelized 3D image data using the algorithm proposed by Mayer (2004), which relies on the computation of the Euclidean distance transform. Thus, periodic boundary conditions are taken into account by computing the Euclidean distance transform with respect to periodic boundary conditions as described above in **Section 2.4**. As in case of the distribution of geodesic tortuosity and the chord length distribution, the distribution $d(H)$ is represented by 19 quantiles.

2.7 Two-Point Correlation Function

For a stationary and isotropic random set $\Xi \subset \mathbb{R}^3$ describing the pore space of a porous material, the two-point (pore-pore) correlation function C: $[0, \infty) \to [0, 1]$, which is also called covariance function (Serra, 1982; Ohser and Schladitz, 2009), is defined as

$$C(r) = \mathbb{P}(o \in \Xi, x \in \Xi) \qquad \text{for each } r \geq 0, \tag{1}$$

where $x \in \mathbb{R}^3$ is an arbitrary point with distance r to the origin (Matheron, 1975; Torquato, 2002; Chiu et al., 2013). This quantity can be estimated from voxelized image data by the Fourier method described in Ohser and Schladitz (Ohser and Schladitz, 2009), where no further step is required to account for periodic boundary conditions. In practice, we represent the two-point correlation function by $C(0), C(1), \ldots, C(167)$, where $167 = \left\lfloor \sqrt{3 \cdot (0.5 \cdot 192)^2} \right\rfloor$ is the maximal distance of two points within the sampling window $W = [0,192]^3$ with respect to periodic boundaries. Note that the pore-solid and solid-solid correlation functions are uniquely determined by the pore-pore correlation function.

3 EFFECTIVE TRANSPORT PROPERTIES

In this section we briefly explain two effective transport properties, namely diffusivity and permeability of the pore space, for which numerical simulations are carried out to estimate these quantities from 3D image data.

3.1 Diffusivity and M-Factor

Effective tortuosity of the pore space is usually defined by

$$\tau_{\text{eff}} = \varepsilon \cdot \frac{D_0}{D_{\text{eff}}},$$

where D_0 denotes the intrinsic diffusivity and D_{eff} the effective diffusivity of the pore phase (Cooper et al., 2016). Note that $D_{\text{eff}} \leq D_0$, because the solid phase acts as obstacle hindering the diffusion process. The characteristic D_{eff} plays a major role in a broad spectrum of applications including water flow (Sahimi, 2011; Bear, 2018), battery electrodes (Newman and Thomas-Alyea, 2004; Thorat et al., 2009; Kehrwald et al., 2011; Nguyen et al., 2020), solid oxide fuel cells (Cooper et al., 2013), biology (Jiao and Torquato, 2012) and heat transfer (Nellis and Klein, 2009; Kaviany, 2012). This quantity is estimated from voxelized image data using the TauFactor app for Matlab (Cooper et al., 2016). More precisely, effective diffusivity is obtained by numerically solving Laplace's equation on Ξ, i.e., the following second-order differential equation is solved:

$$\nabla^2 c = 0 \text{ on } \Xi, \tag{2}$$

where c denotes the concentration of the diffusing species. Apart from mass conservation within the pore space, one has to ensure that the diffusing species can not intrude into the solid phase, which is formally described the following equation at the interface:

$$\nabla c \circ n = 0 \text{ on } \partial\Xi, \tag{3}$$

where the outward pointing unit normal is denoted by n and $\circ$ denotes the scalar product. Finally, the following equations are the driving force for the flux in x-direction:

$$c = 0 \text{ on } \Xi \cap x_0 \text{ and } c = 1 \text{ on } \Xi \cap x_{max}, \tag{4}$$

where x_0 and x_{max} denote the two parallel planes described by $x = 0$ and $x = 192$, respectively. Note that periodic boundary conditions are applied in y- and z-direction. Further technical details regarding the implementation of the equations above can be found in (Cooper et al., 2016).

The M-factor, defined as $M = D_{\text{eff}}/D_0$, is now given by $M = \varepsilon/\tau_{\text{eff}}$, where it holds that $M \in [0, \varepsilon]$ and, equivalently, $\tau_{\text{eff}} \geq 1$, according to Eq. 21.14 in the book of Torquato (2002). In particular, lower values of M correspond to more pronounced transport limitations, whereas a high value of M indicates nearly no hindrance of diffusion processes.

3.2 Permeability

The lattice Boltzmann method is a numerical framework for solving partial differential equations based on kinetic theory, and is used to simulate fluid flow through porous microstructures (Gebäck and Heintz, 2014; Gebäck et al., 2015). The Navier-Stokes equations for pressure-driven flow are solved for the steady state. No-slip, bounce-back boundary conditions are used on the interface between the two phases and periodic boundary conditions are applied orthogonal to the flow direction. We use the two relaxation time collision model with the free parameter $\lambda_{\text{eo}} = \frac{3}{16}$. This ensures that the computed permeability is independent of the relaxation time (and thus the viscosity) (Ginzburg et al., 2008). The relaxation time $t_{\text{rel}} = -\frac{1}{\lambda_e}$ is kept at 1.25. The flow is driven by a constant pressure difference across the structure in the transport direction along the x-axis (Zou and He, 1997), and a linear gradient is used as initial condition. The computational grid has the same resolution as the binary structure arrays, i.e., 192^3. After convergence to steady-state flow, the permeability κ is extracted from Darcy's law (Torquato, 2002),

$$\bar{u} = -\frac{\kappa \Delta p}{\mu d}. \tag{5}$$

Here, $\bar{u}$ is the average velocity, Δp is the applied pressure difference, μ is the dynamic viscosity, and d is the length of the microstructure in the flow direction. The permeability is independent of the fluid and the pressure difference and is hence a property solely of the microstructure, provided that the Reynolds number is sufficiently small (< 0.01). This also ensures that the velocity is proportional to the pressure difference. Note that, since we are dealing with simulated microstructure data on the voxel grid, computed permeabilities are given in (voxel unit)^2.

4 MICROSTRUCTURE GENERATORS

In order to determine microstructure-property relationships which are as general as possible, we generate a large set of periodic microstructures with different types of geometries. More precisely, from each of nine different types of spatial stochastic models we draw 10,000 microstructures such that 1) their porosities are (approximately) uniformly distributed in the interval [0.3, 0.95], and 2) the values of further transport-relevant

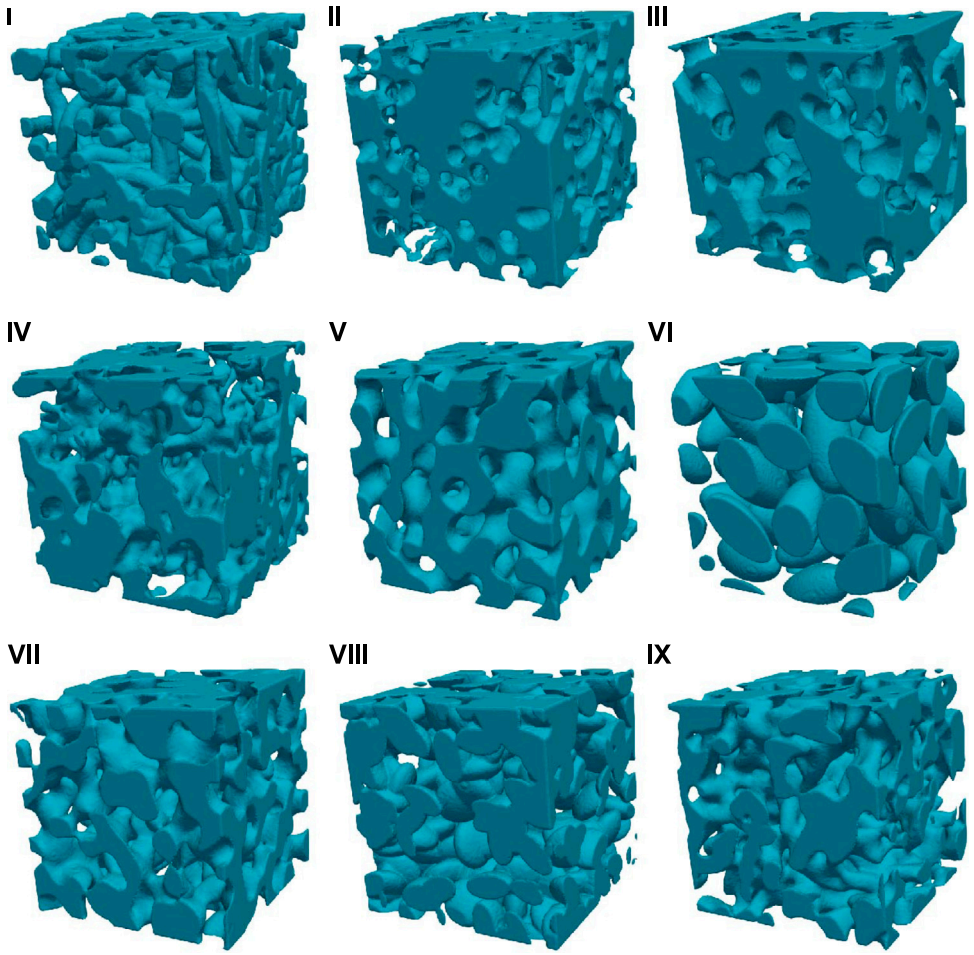


FIGURE 1 | Examples of the different types of microstructures, showing an artificially generated fiber system **(I)**, channel system **(II)**, spatial graph **(III)**, level set of a Gaussian random field **(IV)**, level set of a spinodal decomposition **(V)**, as well as systems of hard ellipsoids **(VI)**, smoothed hard ellipsoids, **(VII)**, soft ellipsoids **(VIII)**, and smoothed soft ellipsoids **(IX)**. Note that the solid phase is always depicted in blue, whereas mass transport takes place in the transparent porous phase.

microstructure characteristics (like specific surface area, $r_{\min}$ and $r_{\max}$) are located in the same ranges. As a consequence, also the values of effective diffusivity and permeability are located in the same ranges for the nine different types of stochastic microstructure models. Some of the microstructures considered in the present paper are defined in the (continuous) Euclidean space, whereas others are generated directly on a computational grid. Finally, all structures are converted into 192^3 binary arrays with periodic boundary conditions. The different types of microstructures, described below in detail, are (I) fiber systems, (II) channel systems, (III) spatial stochastic graphs, (IV) level sets of Gaussian random fields, (V) level sets of spinodal decompositions, (VI) hard ellipsoids, (VII) smoothed hard ellipsoids (VIII) soft ellipsoids, and (IX) smoothed soft ellipsoids. Examples of microstructures drawn from the nine different model types are shown in **Figure 1**. Note that, as can be seen in **Figure 2**, the microstructure models considered in the present paper are designed in such a way that the resulting sets of artificially generated microstructures are disperse in the sense that their microstructure descriptors cover a wide spectrum of values. In particular, keeping the value of a certain microstructure descriptor fixed, the values of other characteristics can still be varied "independently" (to a certain extent). However, on the other hand, note that due to inherent correlations between some pairs of geometric microstructure descriptors, the space of values that can be covered is naturally limited. For example, porosity values close to one typically go along with a low mean geodesic tortuosity. To summarize, the 90,000 microstructures drawn from the nine different stochastic models lead to an extensive dataset representing a broad range of morphologies, which allows us to attribute a certain generality of the microstructure-property relationships determined in the present paper.

4.1 Fiber Systems

The fiber systems considered in this paper are generated using essentially the method described in Townsend et al. (2021), with modifications to allow for periodic and isotropic structures. Individual fibers are first represented as a set of

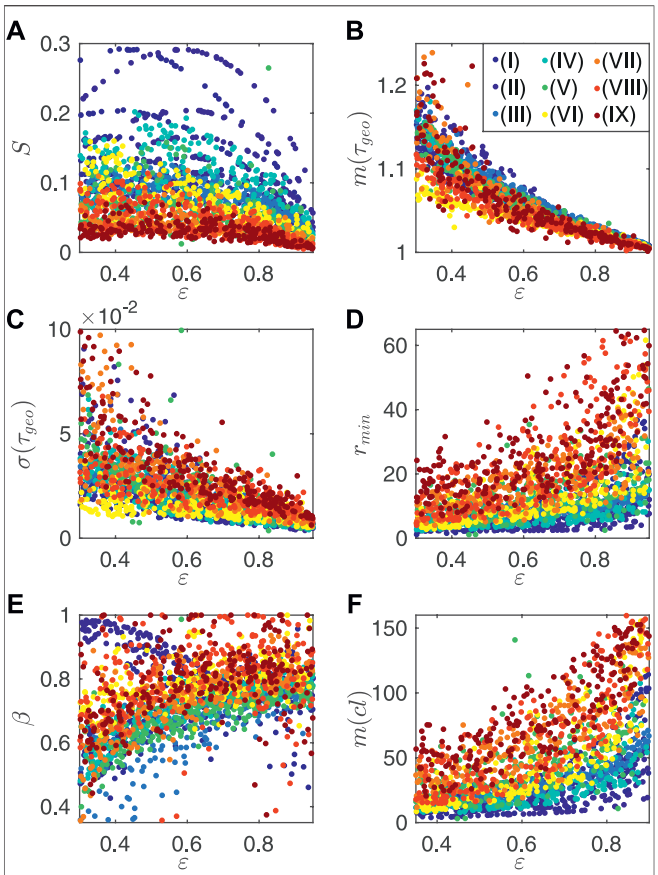


FIGURE 2 | Specific surface area (in (voxelunit)$^{-1}$) **(A)**, mean geodesic tortuosity **(B)**, standard deviation of geodesic tortuosity **(C)**, characteristic bottleneck radius r_{min} (in voxelunit) **(D)**, constrictivity **(E)** and mean chord length (in voxelunit) **(F)** as a function of porosity, where 250 structures have been randomly selected for each of the nine different microstructure models, depicted by nine distinct colors. Note that the roman numbers in the legend correspond to the numbering of the nine stochastic model types mentioned at the beginning of **Section 4**.

nodes generated by a random walk. The angle between the directions of two consecutive steps of the random walk are drawn from a spherical von Mises distribution (Mardia and Jupp, 1999). The parameters of the von Mises distribution are the mean direction μ_{dir}, which is computed using the angle between the previous two nodes, and the concentration θ_{dir}, which specifies the global bending tendency of the fibers. The random walks have a fixed length of 8 nodes. The step length is drawn from a normal distribution with mean 16 voxels and standard deviation 2 voxels. To generate the skeleton of a fiber, we plot a smooth Bézier curve (Gallier, 2000) through its node points. A curvature parameter ρ for the Bézier curves controls the local bending tendency of the fibers. This skeleton is discretized on the 192^3 voxel grid and then dilated to obtain the final cylindrical fibers with radius r, subject to periodic boundary conditions. The concentration parameter θ_{dir} and the curvature ρ are drawn from the uniform distribution on the intervals (Torquato, 2002; Furat et al., 2021) and [0.25, 0.75], respectively. Furthermore, the fiber radius r is drawn at random from the set of integers {2, 3, 4, 5, 6, 7, 8}. Starting from a fully porous structure (without any fibers), new fibers are added until the desired porosity is reached. The method is implemented in MATLAB (2021).

4.2 Channel Systems

Systems of channels are generated using essentially the same code as for the fiber systems considered in **Section 4.1**, treating the fibers as porous channels instead. The parameters of the random walks, the Bezier curves, and the fiber radii are sampled in the same fashion as in **Section 4.1**. Starting from a fully solid structure (without any channels), new channels are added until the desired porosity is reached.

4.3 Spatial Stochastic Graphs

Spatial stochastic graph structures are generated using a model that has been introduced in Gaiselmann et al. (2014), where it has been applied to mimic the 3D microstructure of solid oxid fuel cells. The spatial graph model is based on a homogeneous Poisson point process $X = \{X_n\}_{n\in\mathbb{N}}$ with parameter $\lambda > 0$ (Last and Penrose, 2017), where $\lambda = \mathbb{E}\#\{n \in \mathbb{N}\colon X_n \in [0,1]^3\}$ is the expected number of points per unit volume, with $\#A$ denoting the cardinality of a set A. Note that by this frequently used point-process model the situation of complete spatial randomness is depicted, i.e., there is no interaction between the points X_n. For a more formal introduction to the concept of point processes, the reader is referred, e.g., to the monographs of Daley and Vere-Jones (2005) and Daley and Vere-Jones (2008). Next, the relative neighborhood graph $G_X = (X, E_X)$ (Toussaint, 1980) is constructed, where the set X of vertices is given by the atoms X_n of the underlying homogeneous Poisson process. In order to define the edge set E_X, a so-called critical region $R(p_1, p_2) = B(p_1, \|p_1 - p_2\|) \cap B(p_2, \|p_1 - p_2\|)$ for $p_1, p_2 \in \mathbb{R}^3$ is considered, where $\|p\|$ denotes the Euclidean norm of $p \in \mathbb{R}^3$. The edge set E_X is now given by $E_X = \{(X_i, X_j) \in X \times X\colon \#(R(X_i, X_j) \cap X) = 2\}$, This means that there is an edge between two points X_i and X_j if and only if there is no further point of X within the critical region $R(X_i, X_j)$ except for X_i and X_j themselves. Finally, each edge $e \in E_X$ is dilated using a sphere as structuring element (Soille, 2003). The radius of the sphere follows a Gamma distribution with expectation μ_Γ and variance σ_Γ^2, where each edge is dilated independently from the other edges. In order to obtain an approximately uniformly distributed porosity between 0.3 and 0.95, we draw the model parameters λ, μ_Γ and σ_Γ^2 from the uniform distribution on the intervals $[10^{-5}, 5 \cdot 10^{-4}]$, [1, 10] and [1, 6], respectively. Then, after generating 25,000 samples of such microstructures, it was possible to select 10,000 of them leading to the desired uniform distribution of porosities on the interval [0.3, 0.95]. The generation of 3D microstructures by the spatial stochastic graph model described above is carried out by in-house Java software using the Geostoch framework (Mayer et al., 2004).

4.4 Level Sets of Gaussian Random Fields

Level sets of Gaussian random fields are a well-known concept of stochastic geometry (Lantuéjoul, 2002; Chiu et al., 2013) which, among others, is used for modeling the 3D microstructure of

anodes in lithium-ion batteries (Kremer et al., 2020) and electrodes in solid oxide cells (Moussaoui et al., 2018). In the present paper, periodic Gaussian random fields are generated using an approach based on the fast Fourier transform (FFT) (Lang and Potthoff, 2011). It utilizes the spectral density of the covariance function to generate a random field with the desired covariance structure. In particular, zero-centered white noise (independent and normal distributed) is generated in the spatial domain, transformed to the Fourier domain, and multiplied by the square root of the spectral density, where four types of spectral densities (power-law, exponential, Gaussian, and circular top-hat) are used to generate 2,500 structures of each kind (Röding et al., 2020). After linear rescaling and applying inverse FFT, a Gaussian random field with mean zero and the specified covariance function is obtained. The parameters of the different spectral densities are chosen to ensure a suitable range of length scales. Finally, binary microstructures are obtained as level sets of the random fields, i.e., the desired porosities are obtained by thresholding at appropriate quantiles of the Gaussian random field intensity distributions. The method is implemented in MATLAB (2021).

4.5 Level Sets of Spinodal Decompositions

Phase separation dynamics through the spinodal decomposition mechanism are simulated by solving a system of Navier-Stokes and Cahn-Hilliard equations (Miranville, 2019). A field of spatially resolved concentrations, denoted by $\psi(\mathbf{x}, t) \in [0, 1]$, is evolved in time using the lattice Boltzmann method (Krüger et al., 2017). Initially, the values of ψ are uniformly distributed in the interval [0, 1], independently for all grid points on a 96^3 grid. The spatiotemporal evolution of the concentration profile is determined by several factors such as surface tension, density and viscosity ratio between the two phases. These three characteristics are chosen log-uniformly distributed in their respective ranges to yield a slightly different behaviour of the interfacial geometry. Furthermore, the numbers of time steps, which control the degree of coarsening of the structures, are chosen such that a suitable range of length scales is obtained. Note that the Lifshitz-Slyozov law states that the typical length scale in the structure is proportional to the cubic root of the simulation time (Lifshitz and Slyozov, 1961). After terminating the simulations, the solutions are upscaled to 192^3 voxels. Finally, in the same way as described in **Section 4.4** for Gaussian random fields, the desired porosities are obtained by thresholding at appropriate quantiles of the concentration intensity distributions. The spinodal decomposition simulations are implemented using in-house software based on the lattice Boltzmann method (Gebäck and Heintz, 2014; Gebäck et al., 2015).

4.6 Hard Ellipsoids

Configurations of hard (i.e., non-overlapping) ellipsoids have been used as models for e.g., separation columns (Bertei et al., 2014) and are simulated using a hard-particle Markov Chain Monte Carlo (MCMC) algorithm (Brooks et al., 2011). Initially, the ellipsoids are placed at random locations and with random orientations. Then, the configurations are relaxed by performing random translations and rotations of all particles until no pairs of particles overlap. If the desired porosity is larger than 0.5, non-overlapping configurations can be generated easily at constant porosity as described above. Otherwise, as a preliminary procedure, the steps described above are performed for a porosity of 0.5 and then the resulting configuration is compressed in small steps, until the target porosity is reached. The magnitudes of proposed translations and rotations are adaptively selected such that the acceptance probability is held at 0.25. The number of ellipsoids is between 8 and 512, yielding a wide range of length scales. In addition, the ellipsoids are given by vectors of semi-axes (1, 1, α) where the random variable α is uniformly distributed in the interval [0.25, 1] (oblate) with probability 0.5 and, otherwise, uniformly distributed in [1, 4] (prolate). The microstructures drawn from this model are generated using in-house developed software (Röding, 2017; Röding, 2018) implemented in Julia (Bezanson et al., 2017).

4.7 Smoothed Hard Ellipsoids

Configurations of smoothed hard ellipsoids are generated in the same manner as the hard ellipsoid systems described in **Section 4.6**, with the only difference that the final discretized structure is smoothed with a Gaussian filter, the standard deviation of which is randomly sampled in the range of [2, 16] voxels (Gonzalez and Woods, 2008; Russ, 2007). This yields structures with a semi-continuous solid phase in contrast to the systems of hard ellipsoids described in **Section 4.6**, which consist of discrete particles. Note that similar structures have been used in Prill et al. (2017) as a model for porous electrodes, where spheres instead of random ellipsoids are considered.

4.8 Soft Ellipsoids

Configurations of soft (i.e., overlapping) ellipsoids are created in a similar fashion as the systems of hard ellipsoids considered in **Section 4.6**, but without implementing a specific overlap criterion. Instead, ellipsoids with random locations and orientations are sequentially added until the desired porosity has been obtained. The ellipsoid sizes relative to the simulation window are selected at random to yield an appropriate range of length scales. As in the case of hard ellipsoid systems considered in **Section 4.6**, soft ellipsoids are given by vectors of semi-axes (1, 1, α), where α is uniformly distributed in [0.25, 1] (oblate) with probability 0.5 and, otherwise, uniformly distributed in [1, 4] (prolate). The method is implemented in Bezanson et al. (2017).

4.9 Smoothed Soft Ellipsoids

Configurations of smoothed soft ellipsoids are generated in the same manner as the systems of soft ellipsoids considered in **Section 4.8**, with the only difference that the final discretized structure is smoothed with a Gaussian filter, the standard deviation of which is randomly sampled in the range of [2, 16] voxels (Russ, 2007; Gonzalez and Woods, 2008). This yields structures with a smoother, more continuous solid phase.

5 MICROSTRUCTURE-PROPERTY RELATIONSHIPS

For the 90,000 samples of 3D microstructures generated by the stochastic models described in **Section 4**, the structural descriptors explained in **Section 2** as well as the effective transport properties stated in **Section 3** are computed. This data is then used to establish microstructure-property relationships by means of three different approaches, namely analytical prediction formulas, (artificial) fully-connected neural networks (ANNs) and convolutional neural networks (CNNs). For this, the data is randomly shuffled and split into three subsets of training, validation, and test data, respectively. This is done in a stratified manner such that an equal number of microstructures of each type is included in each of the three datasets. The split, which is the same for each of the three types of prediction models, is 70% training data (7,000 per type of microstructure and 63,000 in total) and 15% each for the validation and test data (1,500 per type of microstructure and 13,500 in total).

Several error measures are considered to assess predictive performance. For fitting the prediction models, the mean squared error (MSE) loss is used, where

$$\mathrm{MSE} = \frac{1}{k}\sum_{j=1}^{k}(\hat{y}_j - y_j)^2. \tag{6}$$

Here $y_1, \ldots, y_k$ is the ground truth data of a given output variable and $\hat{y}_1, \ldots, \hat{y}_k$ are the corresponding estimates predicted by the model. Depending on context, the MSE loss is used either on the output variables themselves or on transformed outputs. Further details can be found below in **Sections 5.1–5.3** for the three types of microstructure-property relationships considered in this paper. Note that the MSE loss is an appropriate loss function for optimization because it is differentiable. In contrast, for final assessment, the mean absolute percentage error (MAPE) loss is used, where

$$\mathrm{MAPE} = \frac{100}{k}\sum_{j=1}^{k}\left|\frac{\hat{y}_j - y_j}{y_j}\right|. \tag{7}$$

However, because the MAPE loss is not everywhere differentiable, it is less well-behaved as an optimization target, but on the other hand it is more interpretable.

In addition, we consider the coefficient of determination $R^2 \in [0, 1]$, which is given by

$$R^2 = 1 - \frac{\sum_{j=1}^{k}(y_j - \hat{y}_j)^2}{\sum_{j=1}^{k}(y_j - \bar{y})^2}, \tag{8}$$

where $\bar{y}$ denotes the empirical mean of the ground truth data $y_1, \ldots, y_k$. Since the values for permeability cover several orders of magnitude, the value of R^2 would be dominated by those terms that involve large values. For this reason, we also consider the coefficient of determination on the log scale, denoted by $R^2_{\log} \in [0, 1]$. In particular, it holds

$$R^2_{log} = 1 - \frac{\sum_{j=1}^{k}(\log(y_j) - \log(\hat{y}_j))^2}{\sum_{j=1}^{k}(\log(y_j) - \bar{y}_{\log})^2}, \tag{9}$$

where $\bar{y}_{\log}$ is defined as the empirical mean of the sample $\log(y_1), \ldots, \log(y_k)$.

Before the derivation of microstructure-property relationships is discussed, we first quantify the "amount of information" that is contained in the different structural descriptors with regard to permeability, effective tortuosity and the M-factor. For this purpose, we make use of the measure of general functional dependence, which has been introduced in (Xu et al., 2017). In contrast to several widely used quantities such as Pearson's ρ, Kendall's τ or Spearman's ρ, this characteristic, denoted by $0 \le \delta(x, y) \le 1$, relies neither on a linear nor on a monotone relationship between the quantities x and y. Thus, it can be used to quantify the importance of a single (scalar) structural descriptor x on an effective property y, where higher values correspond to a higher "amount of information". The upper limit of 1 is reached if and only if x is a function of y or vice versa since in general it holds $\delta(x, y) = \delta(y, x)$. The corresponding results are shown in **Table 1**. Note that the values for the effective tortuosity are always larger than for the M-factor, which is probably caused by the fact that τ_{eff} already contains the porosity ε. However, it is worth mentioning that a low value of the functional dependence measure does not automatically imply that this quantity should not be considered for predicting a certain effective property. This is due to the fact that $\delta(x, y)$ only contains some information on the predictive power of x with regard to y, but not regarding the usefulness of x in combination with other structural descriptors. This also explains the fact that the normalized quantity M turned out to be predicted with higher accuracy than τ_{eff}, and therefore we stick to M for the rest of this work. It is also interesting to point out that $m(L)$ and $\sigma(L)$ seem to be closely related to permeability and effective tortuosity, considering that the chord length distribution is rarely used in the literature for establishing microstructure-property relationships.

5.1 Analytical Prediction Formulas

In this section, microstructure-property relationships are derived by analytical prediction formulas, whose parameters are fitted by least-squares regression (Sen and Srivastava, 2012). More precisely, we use the trust-region-reflective algorithm (Coleman and Li, 1994; Coleman and Li, 1996) for unconstrained least-squares problems and the interior point algorithm (Byrd et al., 2000; Waltz et al., 2006) for constrained ones. Since there are no hyperparameters in case of analytical prediction formulas, we merge the training set with the validation set for computing the fitting parameters and use the test set for assessing performance. Altogether, we state nine analytical prediction formulas and their fitting parameters. Afterwards, a comparison of these formulas is carried out, including an interpretation of the results.

To begin with, we consider several relationships between geometric microstructure descriptors given in **Section 2** and the M-factor stated in **Section 3**. Recall that according to

TABLE 1 | Functional dependence measure $\delta(x, y)$ between a scalar structural descriptor x and an effective property y.

x/y	κ	τ_{eff}	M
ε	0.674	0.927	0.916
S	0.726	0.833	0.467
β	0.631	0.864	0.638
$r_{\min}$	0.685	0.856	0.242
$r_{\max}$	0.798	0.847	0.565
$m(\tau_{\text{geo}})$	0.474	0.964	0.851
$\sigma(\tau_{\text{geo}})$	0.632	0.866	0.742
$m(L)$	0.900	0.872	0.614
$\sigma(L)$	0.902	0.883	0.677

Equation 21.14 in the book of S. Torquato (Torquato, 2002), it holds that $M \in [0, 1]$, which is also ensured for the predictions $\hat{M}$ of M presented below. At first, we consider a type of a parametric formula, which has been originally proposed in Stenzel et al., 2016) and later reconsidered in **Eq. 4** of Neumann. et al. (2020). More precisely, the M-factor is predicted by

$$\widehat{M} = \varepsilon^{c_1}\beta^{c_2} m(\tau_{\text{geo}})^{c_3}. \tag{10}$$

To ensure that $\hat{M} \in [0, 1]$, we use the constraints $c_1 \geq 1$ and $c_2 \geq 0$, which leads to the following results: $c_1 = 1, c_2 = 0$ and $c_3 = -8.45$. More precisely, the best fit in a least-squares sense does not contain the constrictivity β, whereas the exponent of the porosity equals the lower limit of one. Note that Formula (**Eq. 10**) for predicting the M-factor has been modified in Neumann. et al. (2020) to ensure that $\hat{M} \in [0, 1]$ holds also for the dilute limit by using the constrictivity β within the exponent of porosity. More precisely, in Neumann. et al. (2020) the following parametric prediction formula is proposed:

$$\widehat{M} = \varepsilon^{c_1+c_2\beta} m(\tau_{\text{geo}})^{c_3}, \tag{11}$$

where the additional constraint $c_1 + c_2 \geq 0$ is used to ensure that $\widehat{M} \in [0, 1]$. This leads to $c_1 = 1.25, c_2 = -1.25$ and $c_3 = -7.82$. Finally, we consider a formula for predicting the M-factor by porosity as well as the mean and standard deviation of geodesic tortuosity of the pore space, see Barman et al. (2019):

$$\widehat{M} = c_1 m(\tau_{\text{geo}})^{c_2} \sigma(\tau_{\text{geo}})^{c_3} \varepsilon^{c_4}, \tag{12}$$

where least-squares fitting gives that $c_1 = 1.18, c_2 = -9.17, c_3 = 0.03$ and $c_4 = 1.02$.

Having discussed parametric formulas for predicting the M-factor, we now predict the permeability κ using geometric microstructure descriptors given in **Section 2**. Since the values of κ can cover several orders of magnitude, the fitting of parameters is carried out on the log scale. First, we consider the prediction formula

$$\hat{\kappa} = c_1 \varepsilon^{c_2} \beta^{c_3} S^{-2} m(\tau_{\text{geo}})^{c_4}, \tag{13}$$

which has been introduced in Neumann. et al. (2020). By least-squares regression, we obtain that $c_1 = 0.16, c_2 = 2.05, c_3 = 0.64$ and $c_4 = -7.31$. Moreover, we consider still another type of a parametric prediction formula for κ, proposed in Neumann. et al. (2020). Namely,

$$\hat{\kappa} = c_1 (c_2 r_{\min} + c_3 r_{\max})^2 \varepsilon^{c_4} m(\tau_{\text{geo}})^{c_5}, \tag{14}$$

where least-squares fitting gives that $c_1 = 0.24, c_2 = 0.92, c_3 = 0.08, c_4 = 1.6$ and $c_5 = -6.82$. Note that for fitting the parameters in **Eq. 14** we use the additional constraint $c_2, c_3 \in [0, 1]$ with $c_2 + c_3 = 1$. Thus, we use a convex combination of $r_{\min}$ and $r_{\max}$, which is subsequently squared to ensure the right unit of permeability. Last not least, a further parametric prediction formula for κ, which has been discussed in the literature, is given by

$$\hat{\kappa} = c_1 \varepsilon^{c_2} S^{-2} m(\tau_{\text{geo}})^{c_3}, \tag{15}$$

see Röding et al. (2020), where least-squares regression leads to $c_1 = 0.14, c_2 = 2.07$ and $c_3 = -8.57$.

In addition to the results mentioned above, we consider the following prediction formula for κ:

$$\hat{\kappa} = c_1 \varepsilon^{c_2+c_3\beta} S^{-2} m(\tau_{\text{geo}})^{c_4}, \tag{16}$$

which uses the constrictivity β within the exponent of the porosity ε, similar to **Eq. 11**, where the fitting parameters are given by $c_1 = 0.14, c_2 = 3.07, c_3 = -1.38$ and $c_4 = -7.37$. Furthermore, we predict κ by the porosity ε, the mean geodesic tortuosity $m(\tau_{\text{geo}})$ and the median $r_{\min}$ via

$$\hat{\kappa} = c_1 \varepsilon^{c_2} m(\tau_{geo})^{c_3} r_{min}^2, \tag{17}$$

where least-squares fitting on the log scale gives that $c_1 = 0.25, c_2 = 1.6$ and $c_3 = -6.6$. Finally, we consider a prediction formula for κ which involves the mean chord length $m(L)$ of the pore space:

$$\hat{\kappa} = c_1 m(L)^{c_2} \varepsilon^{c_3+c_4\beta} r_{\min}^{c_5} m(\tau_{\text{geo}})^{c_6}, \tag{18}$$

where we use the constraint $c_2 + c_5 = 2$ in order to obtain the right unit (voxels^2), and the fitting parameters are given by $c_1 = 0.1, c_2 = 0.63, c_3 = 1.38, c_4 = -0.2, c_5 = 1.37$ and $c_6 = -6.74$.

Regarding the prediction of effective diffusivity, the data visualized in the top row **Figure 3** lead to the following results. It is interesting to observe that the problem of **Eq. 10** described in Neumann. et al. (2020), namely not fulfilling that $\hat{M} \approx 1$ in the dilute limit ($\varepsilon \to 1$), does not occur, see **Figure 3A**, since fitting the coefficients leads to $c_2 = 0$, i.e., the constrictivity β is not used for predicting effective diffusivity. Furthermore, note that the performance of **Eq. 10** is similar to the performance of **Eqs. 11**, **12** even though the latter one does not involve constrictivity, see **Figures 3A–C**.

In contrast to the prediction of effective diffusivity, the performance of the analytical prediction formulas for permeability significantly varies case by case, see **Figures 3D–I**. In particular, **Eq. 15** leads to the lowest predictive power in terms of MAPE, where the low value of the coefficient of determination R^2 is attributed to an overestimation of permeability in the regime of large permeability values, see **Table 2**. This also holds with regard to **Eqs. 13**, **16**, which both lead to a similar MAPE of approximately 18%. On the other hand, a remarkable improvement can be observed by disregarding constrictivity and including instead $r_{\min}$, as in **Eq. 17**, or a convex

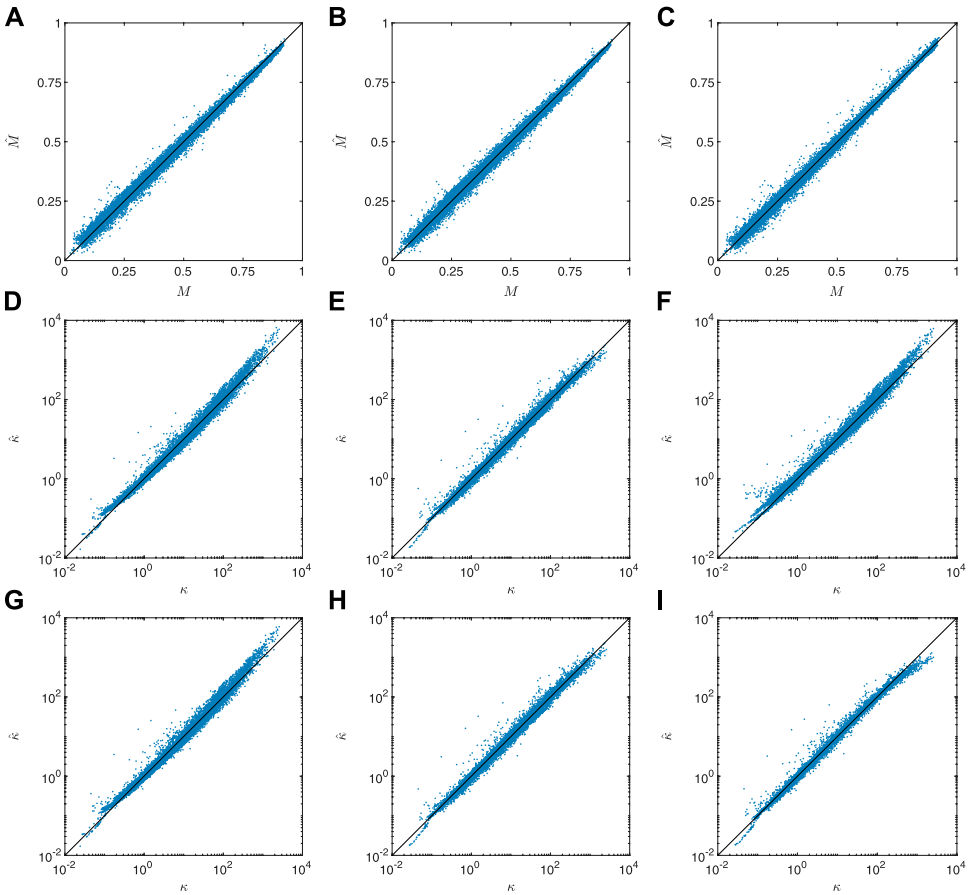


FIGURE 3 | Prediction of effective transport properties using analytical formulas. Top row **(from left to right):** Prediction of M-factor via **Eqs. 10–12**. Middle row **(from left to right):** Prediction of permeability via **Eqs. 13–15**. Bottom row **(from left to right):** Prediction of permeability via **Eqs. 16–18**. Note that the scatter plots show results based on the test data.

combination of $r_{\min}$ and $r_{\max}$, as in **Eq. 14**, where both prediction formulas lead to a MAPE of approximately 15%. However, it is worth mentioning that **Eq. 17** only uses three adjustable parameters and three geometric descriptors of the underlying 3D microstructure. In addition, the small value $c_3 = 0.08$ of the parameter corresponding to $r_{\max}$ in **Eq. 14** highlights the importance of the direction-dependent characteristic $r_{\min}$, which in contrast to $r_{\max}$ also contains information about bottleneck effects. The best performance with regard to MAPE is obtained when additionally using the mean chord length as in **Eq. 18**, where one has to note that this prediction formula also contains the highest number of parameters as well as microstructure descriptors. However, it is interesting to note that such a simple quantity as the mean chord length turns out to be beneficial in terms of further improving the predictive power see **Table 2**.

TABLE 2 | Error measures computed on the test set, corresponding to the analytical prediction formulas **Eqs. 10–18** for M-factor and permeability, respectively. Note that in case of predicting permeability, the quantity $R^2_{\log}$ denotes the coefficient of determination on the log scale.

Equation	R^2	MAPE	$R^2_{\log}$
(10)	0.993	5.76	-
(11)	0.994	5.44	-
(12)	0.994	5.15	-
(13)	0.218	18.94	0.985
(14)	0.920	14.84	0.990
(15)	0.219	21.33	0.982
(16)	0.331	18.34	0.986
(17)	0.915	14.81	0.990
(18)	0.820	11.63	0.992

5.2 Artificial Neural Networks

In addition to the analytical prediction formulas **Eqs. 10–18** presented above, we investigate artificial neural networks (ANNs) for the prediction of mass transport properties. Note that a conventional, fully-connected ANN is a composition of linear and nonlinear operations. The building blocks are fully-connected (also called dense) layers, each of which consists of a certain number of nodes. The input to each node is a weighted sum of the outputs from the nodes in the previous layer, to which a nonlinear so-called activation function $f\colon \mathbb{R} \to \mathbb{R}$ is applied. As

TABLE 3 | Descriptor sets used as input for the ANNs, together with the dimensions of the corresponding input vectors. Note that Models 8, 9 and 10 involve four scalar quanities and one distributional characteristic described by 19 quantiles, whereas the two-point correlation function in Model 1 is evaluated for 168 different radii.

Model	Descriptors	Dimension
1	ε, S, $m(\tau_{geo})$	3
2	ε, S, $m(\tau_{geo})$, β	4
3	ε, $m(\tau_{geo})$, $\sigma(\tau_{geo})$	3
4	ε, $m(\tau_{geo})$, β	3
5	ε, $m(\tau_{geo})$, r_{min}	3
6	ε, $m(\tau_{geo})$, r_{min}, r_{max}	4
7	ε, $m(\tau_{geo})$, r_{min}, β, $m(L)$	5
8	ε, S, $m(\tau_{geo})$, $\sigma(\tau_{geo})$, $d(\tau_{geo})$	23
9	ε, S, $m(L)$, $\sigma(L)$, $d(L)$	23
10	ε, S, $m(H)$, $\sigma(H)$, $d(H)$	23
11	ε, S, C	170
12	all	236

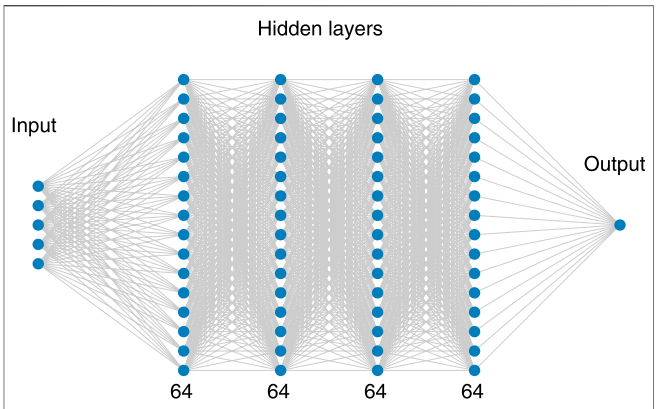


FIGURE 4 | Illustration of the ANN architecture with 4 hidden layers, each with 64 nodes, where only a smaller number of nodes is shown in this figure for clarity. Furthermore, the input in this figure is 5-dimensional, but in the present paper the input dimension varies from 3 to 236, see **Table 3**.

a whole, the network describes a nonlinear mapping from some input to some output that can be arbitrarily complex. During training of an ANN, the parameters (weights of the sums) are optimized with respect to a loss function that penalizes deviations from the target output (Schmidhuber, 2015).

The first layer is, in a sense, the input layer, the dimension of which depends on the set of microstructural descriptors used. The first layer is followed by a certain number of fully-connected layers, so-called hidden layers, which are described below in more detail. The final layer is the output layer, which in our case of a scalar output consists of a single node only. For the hidden layers, the exponential linear unit (Elu) activation function is used (Clevert et al., 2016), which is given by

$$f(x) = \begin{cases} x, & \text{if } x > 0, \\ \gamma(e^x - 1), & \text{if } x \leq 0, \end{cases} \tag{19}$$

where we put $\gamma = 1$.

We investigate 12 different sets of geometric microstructure descriptors, re-using all sets of descriptors considered in the analytical prediction formulas **Eqs. 10–18** and also adding new ones involving more complex descriptors like distributions/quantiles of e.g., geodesic tortuosity, chord length distributions, spherical contact distribution and the two-point correlation function. The descriptor sets are summarized in **Table 3**. The inputs, i.e. the microstructure descriptors, are linearly rescaled such that in each case, the input data provided by the training set has zero mean and unit variance, which often improves convergence during training (LeCun et al., 2012). Furthermore, the outputs are transformed in the following fashion. The values of $M \in (0, 1)$ and $\kappa \in (0, \infty)$ cover some orders of magnitude. To simplify training, the logit-transformed M-factor $y = \log(M/(1 - M))$ and the log-transformed permeability $y = \log(\kappa)$ are used as the target outputs. This yields the benefit that the inverse-transformed predictions belong to (0, 1) and (0, ∞), respectively.

The networks are implemented in Tensorflow 2.4.1 (Abadi et al., 2015) and optimized with respect to MSE loss (on the logit and log scales). Glorot/Xavier uniform initialization is used for all weights (Glorot et al., 2010), as well as stochastic gradient descent (SGD) with momentum (Qian, 1999; Bottou, 2010) for optimization. A random search hyperparameter optimization (Bergstra and Bengio, 2012) is performed to quantify the importance of the number of hidden layers, the number of nodes per layer, the batch size (in each update in the SGD), the momentum and the learning rate of the SGD. It turns out that 4 hidden layers, each with 64 nodes, is a reasonable choice, where the number of weights varies from 12,801 to 27,713 depending on the dimension of the descriptor input. The network architecture is illustrated in **Figure 4**. Furthermore, a batch size of 128 and a momentum of 0.9 is chosen, but those particular values turned out to be not critical with regard to the performance. Also, we consider different kinds of regularization such as weight decay, i.e., l_2 regularization (Krogh and Hertz, 1992) and dropout regularization (Srivastava et al., 2014), where both did not lead to any improvement. It is also worth mentioning that batch normalization (Ioffe and Szegedy, 2015), another common regularization method, is not investigated here because combining it with the Elu activation function (Clevert et al., 2016) has been found not useful. Finally, it turned out that the learning rate (LR) has considerable impact on the results. Therefore, we design an LR scheme with a step-wise increasing and then step-wise decreasing learning rate. More precisely, let LR $\in \{10^{-4.5}, 10^{-4}, 10^{-3.5}, 10^{-3}, 10^{-2.5}\}$ for 1,000 epochs (iterations over the whole training set) each, and then, we choose LR $\in \{10^{-2}, 10^{-2.5}, 10^{-3}, 10^{-3.5}, 10^{-4}\}$ for 4,000 epochs each. In total, the training procedure comprises 25,000 epochs. For each set of inputs, i.e., microstructural descriptors, 100 networks are trained using different random seeds. Note that the random seed controls the weight initializations in the network as well as the shuffling of data in the SGD. The model yielding the minimal validation loss (over all epochs and all runs) for each set of microstructural descriptors is selected. On a single NVIDIA T4 GPU, the average execution time for each run is 3.7 h.

Scatter plots visualizing the prediction results for M-factor and permeability are shown in **Figures 5**, **6**, respectively. Furthermore, error measures for the prediction of M-factor and permeability are shown in **Tables 4** and **5**, respectively. It

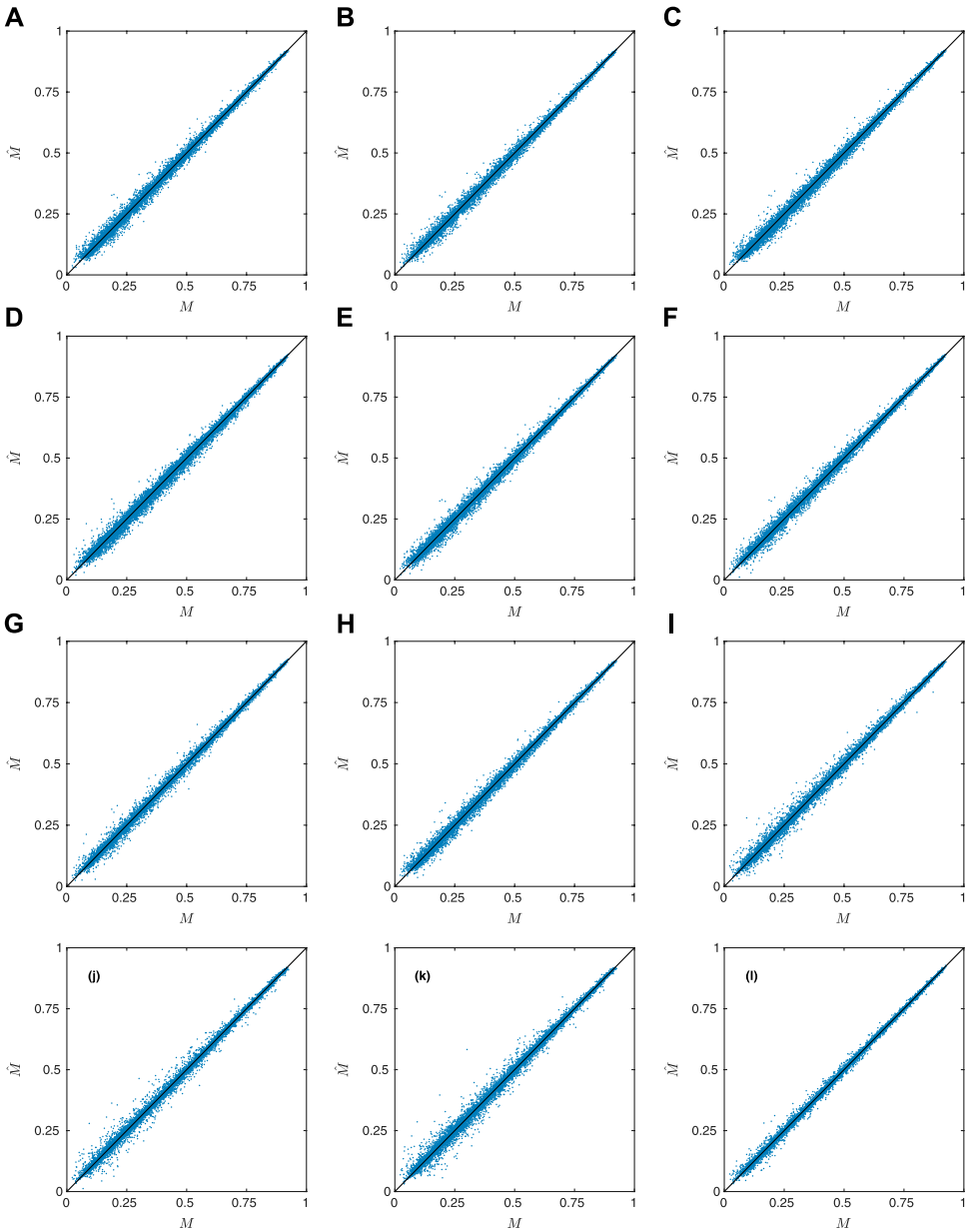


FIGURE 5 | Prediction of M-factor using ANNs. Top row **(from left to right):** Prediction of M via Models 1–3. Second row **(from left to right):** Prediction of M via Models 4–6. Third row **(from left to right):** Prediction of M via Models 7–9. Bottom row **(from left to right):** Prediction of M via Models 10–12. Note that the scatter plots show results based on the test data.

turns out that the predictions obtained by ANNs are consistently better than those obtained by the analytical prediction formulas considered in **Section 5.1**, when the same descriptors are used as input. In addition, adding more complex descriptors like quantiles (of tortuosity, etc.) improves the results even further. Unsurprisingly, the best results are obtained using all the computed descriptors.

Note that some of the ANN models (Models 1–7 in **Table 3**) involve exactly the same sets of descriptors as input which are used in the analytical models for the prediction of M and/or κ. Not surprisingly, the ANNs outperform the corresponding analytical prediction formulas (with the same descriptor sets) in all cases in terms of MAPE evaluated on the test set. Furthermore, in the case of geodesic tortuosity τ_{geo}, there are clear improvements when adding more detailed information on the distribution of τ_{geo}, i.e., starting with ϵ, $m(\tau_{\text{geo}})$, $\sigma(\tau_{\text{geo}})$ and then adding 19 quantiles of $d(\tau_{\text{geo}})$, the MAPE is reduced from 3.84 to 3.38% for the M-factor, and from 29.25 to 10.01% for permeability. Also, descriptor sets involving detailed information on the distributions of chord

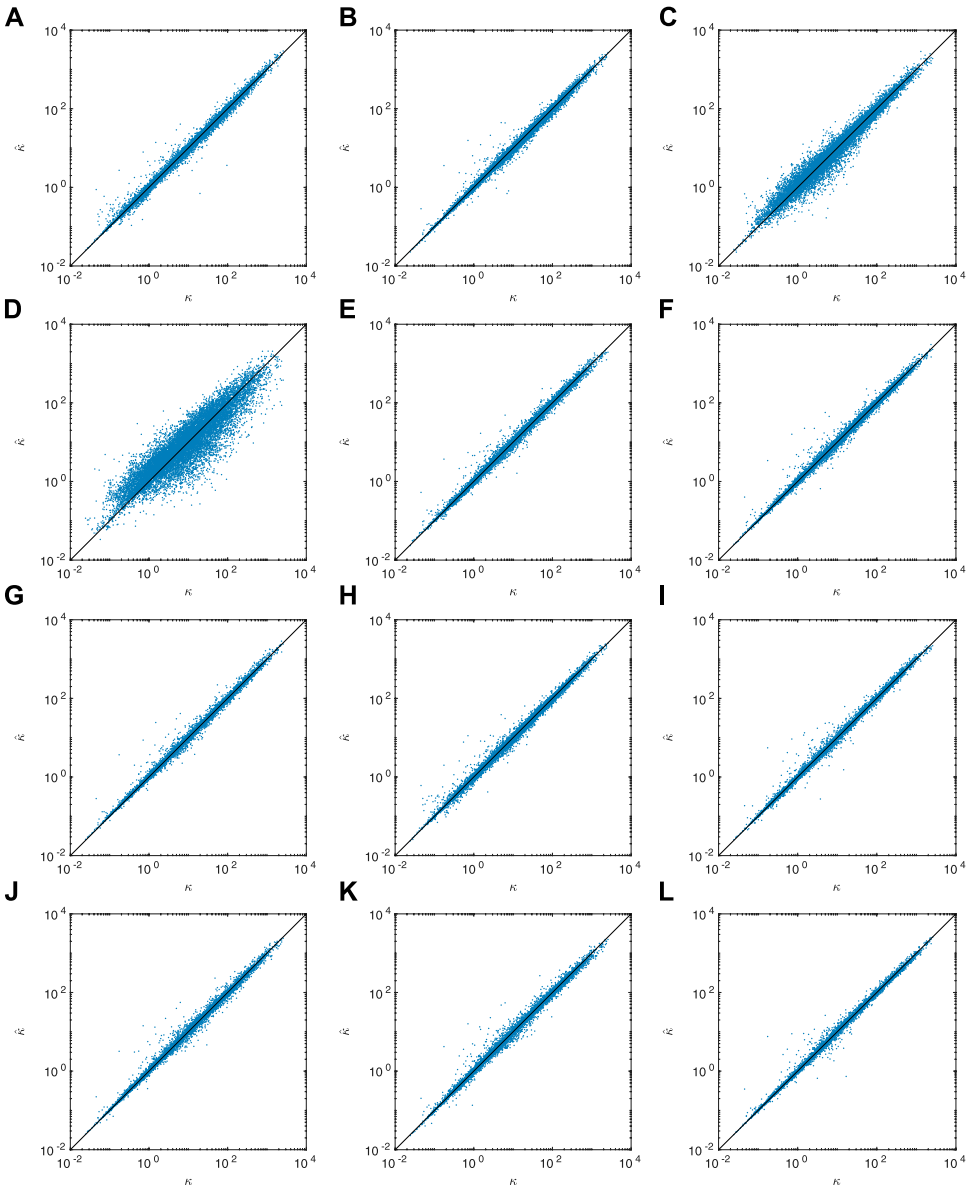


FIGURE 6 | Prediction of permeability using ANNs. Top row **(from left to right):** Prediction of κ via Models 1–3. Second row **(from left to right):** Prediction of κ via Models 4–6. Third row **(from left to right):** Prediction of κ via Models 7–9. Bottom row **(from left to right):** Prediction of κ via Models 10–12. Note that the scatter plots show results based on the test data.

lengths and spherical contact distances perform well. With regard to the prediction of the M-factor, these high-dimensional descriptors perform better than the ANN model (no. 10 in **Table 3**) involving the two-point correlation function. However, for the prediction of permeability, the ANN model involving the two-point correlation function performs better than the model (no. 8) involving the distribution of tortuosity.

Note that the Models 3 and 4 which do not use neither $r_{\min}$ nor the specific surface area S, perform substantially worse than all other models with regard to permeability. Not surprisingly, including all descriptors (Model 12) yields the best performance for predicting both M and κ. However, comparing models with low-dimensional (Models 1–7) and high-dimensional (Models 8–12) descriptor sets, the best low-dimensional descriptor (Model 7) gives very good performance. In particular, going from 5 to 236 input dimensions reduces the MAPE from 2.74 to 2.03% for M and from 7.47 to 6.51% with regard to κ. Considering that this reduction in error requires a massive reduction in interpretability of the model, it is not obvious how to make this trade-off between model complexity and performance.

TABLE 4 | Error measures for the prediction of M via ANNs where MSE is given for the training, validation and test sets, and MAPE for the test set. Note that MSE is evaluated on the logit scale and MAPE on the linear scale.

	MSE			MAPE (in %)
Model	**Training**	**Validation**	**Test**	**Test**
1	0.0073	0.0077	0.0075	3.55
2	0.0055	0.0064	0.0062	3.07
3	0.0081	0.0084	0.0084	3.84
4	0.0075	0.0089	0.0089	4.03
5	0.0067	0.0069	0.0076	3.59
6	0.0053	0.0061	0.0065	3.10
7	0.0043	0.0055	0.0054	2.74
8	0.0062	0.0068	0.0071	3.38
9	0.0061	0.0073	0.0078	3.25
10	0.0063	0.0080	0.0077	3.03
11	0.0077	0.0102	0.0103	3.66
12	0.0021	0.0037	0.0034	2.03

5.3 Convolutional Neural Networks

In a CNN, as opposed to a conventional ANN, the main building blocks are convolutional layers. A typical CNN architecture comprises convolutional layers, pooling layers, and fully-connected layers. In the convolutional layers, the input is convolved with several convolution kernels, whose entries are trainable parameters. The convolutions themselves are linear operations, but a nonlinear activation function $f\colon \mathbb{R}\to\mathbb{R}$ is applied to the result to produce the outputs, called feature maps. In the pooling layers, the feature maps are downsampled by computing, e.g., the mean or maximum on small (typically non-overlapping) patches of the feature maps from the preceding layer, reducing the resolution such that the next convolutional layer can extract information from another spatial scale. After the convolutional and pooling layers, fully-connected layers are typically used to obtain a scalar output. The first part of the CNN can be thought of as a feature extractor that produces geometrical features which are qualitatively similar to the ones used for the analytical prediction formulas and ANNs considered in the previous sections of this paper, whereas the second part corresponds directly to the ANNs themselves.

The size of the microstructures themselves is too large to be practically feasible from a computational point of view. Therefore, we downsample to half the size in all directions by averaging. The result is a 96^3 array for each structure with values in $\{0, \ldots, 8\}$ stored as 8-bit unsigned integers. Apart from a linear scaling factor, this preprocessing step is equivalent to having an average pooling filter with a $2 \times 2 \times 2$ window as the first layer in the CNN. Preprocessing the structures in this manner allows for a larger batch size without running out of GPU memory. During training, the input arrays are batch-wise converted to 32-bit floating point precision and rescaled to $[-1/2, 1/2]$ prior to the gradient update. The outputs are transformed in the same way as for the ANNs. The first part of the CNN consists of three convolutional blocks, each with two convolutional layers with Elu activations and one average pooling layer. The convolutional layers use $3 \times 3 \times 3$ kernels, and the number of filters, i.e. kernels, used is 16, 32, and 64 in the respective blocks. The average pooling layers use $2 \times 2 \times 2$ windows. The second part of the CNN consists of 4 fully-connected layers with Elu activations, each with 64 nodes, i.e. the same as for the ANNs considered in the previous section. The total number of weights is 2,324,689. The network architecture is illustrated in **Figure 7**.

The weight initialization procedure, optimizer, and momentum are the same as for ANNs. However, the batch size is 16 and the learning rate (LR) schedule is also different: Let $\mathrm{LR} \in \{10^{-4}, 10^{-3.75}, 10^{-3.5}, 10^{-3.25}\}$ for 25 epochs each, then $\mathrm{LR} = 10^{-3}$ for 100 epochs, and finally $\mathrm{LR} \in \{10^{-3.25}, 10^{-3.5}, 10^{-3.75}, 10^{-4}\}$ for 50 epochs each, in total comprising 400 epochs. In contrast to ANNs, we now introduce a data augmentation scheme for the training data, involving random flips, rotations, and circular shifts of the structure array in both dimensions orthogonal to the mass transport direction. The mass transport is invariant with respect to these transformations such that a data augmentation scheme like this will act as a regularizer that increases the generalization performance of the network (Hernández-García et al., 2018). On a single NVIDIA V100 GPU, the average execution time is 140 h. The model yielding the minimal validation loss over all epochs is selected. Because of the large computational workload for CNNs, we do only one run for M-factor and one for permeability.

In addition to this ordinary CNN, we train the same architecture with a different kind of input data. Instead of using the structure arrays, we compute the Euclidean distance transform in the pore space which effectively comprises a spatial map of local pore sizes (Russ, 2007). To the best of our knowledge, this approach of using the distance transform as a representation of the pore space has not been used before as inputs to a CNN, where these inputs are again rescaled to 96^3 arrays. The only differences compared to the ordinary CNN are that the input data has to be stored in 32-bit floating point precision (leading to high demands in storage space, ~ 320 GB for training, validation, and test) and that the data are

TABLE 5 | Error measures for the prediction of κ via ANNs, where MSE is given for the training, validation and test sets, and MAPE for the test set. Note that MSE is evaluated on the logit scale and MAPE on the linear scale.

	MSE			MAPE (in %)
Model	Training	Validation	Test	Test
1	0.0241	0.0296	0.0298	10.99
2	0.0176	0.0227	0.0225	9.44
3	0.1387	0.1473	0.1491	29.25
4	0.6256	0.6815	0.6830	76.63
5	0.0206	0.0236	0.0240	10.35
6	0.0164	0.0211	0.0214	9.31
7	0.0132	0.0176	0.0163	7.47
8	0.0181	0.0247	0.0236	10.01
9	0.0114	0.0174	0.0170	7.39
10	0.0180	0.0228	0.0208	8.49
11	0.0203	0.0254	0.0243	9.51
12	0.0069	0.0131	0.0140	6.51

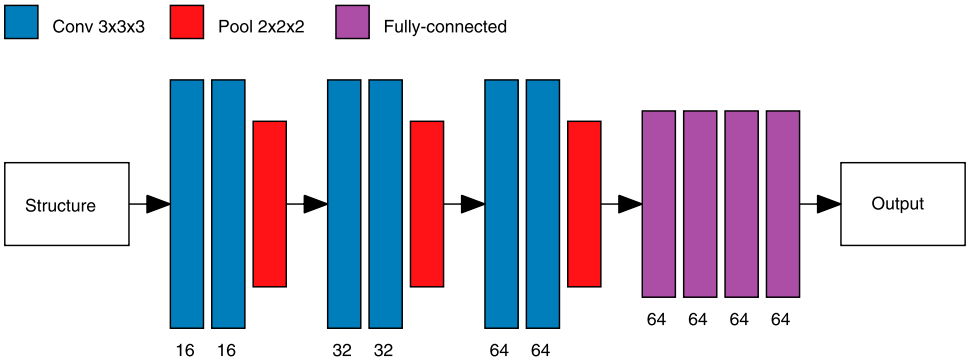


FIGURE 7 | Illustration of the CNN architecture. The inputs, arrays of size 96^3, are fed into the convolutional part of the network, consisting of three convolutional blocks, each in turn consisting of two convolutional layers (the numbers of filters are indicated in the figure) followed by an average pooling layer. The feature maps produced as output from the convolutional part are passed to 4 fully-connected layers with 64 nodes each.

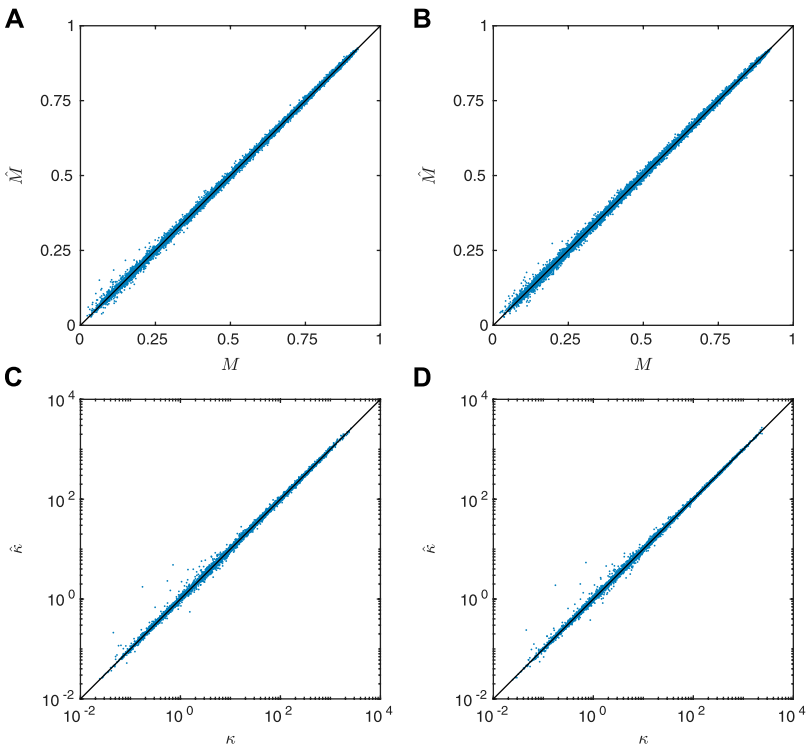


FIGURE 8 | Prediction results using different CNNs, showing **(A)** prediction of M using an ordinary CNN, **(B)** prediction of M using a DT-CNN, **(C)** prediction of κ using an ordinary CNN, and **(D)** prediction of κ using a DT-CNN. Note that the scatter plots show results based on the test data.

batch-wise rescaled by a factor 1/24 (the 95%-quantile of the distance transform values is approximately 24).

Scatter plots visualizing the prediction results for M-factor and permeability are shown in **Figure 8**. Furthermore, error measures are shown in **Table 6** for both M and κ and for both the ordinary CNN (briefly denoted by CNN) and the distance-transform CNN (denoted by DT-CNN). As can be seen, all CNNs perform better than their best ANN counterparts, the best models attaining 1.65% (M-factor) and 3.78% (permeability) MAPE, although this is at the expense of even less interpretability than for the highest-dimensional descriptor used for the ANNs. Also, the differences between the ordinary CNN and the DT-CNN are not substantial and neither one of them is consistently better. This is possibly because the ordinary CNN learns similar information as that already supplied to the DT-

CNN. Hence, we conclude that given the increased computational workload of the distance transform, there are at least no substantial benefits of using the DT-CNN over the ordinary CNN.

6 CONCLUSION

We investigate microstructure-property relationships for artificially generated porous materials, developing prediction models for diffusivity and permeability based on the geometry of the pore space. The basis for this is a comprehensive dataset of 90,000 structures with size of 192^3 voxels, which are generated by 9 different stochastic 3D microstructure models. To the best of our knowledge, this is the largest and most diverse dataset for studying diffusivity and permeability published so far in the literature. Microstructural descriptors like porosity, specific surface area, tortuosity and its distribution, constrictivity, spherical contact distributions, chord length distribution and two-point correlation functions are used in various combinations as input to both analytical prediction formulas and artificial neural networks (ANNs). Furthermore, the structure itself as well as a distance transform in the pore space, capturing the shortest distance to the solid phase, is used as input to convolutional neural networks (CNNs). In terms of mean absolute percentage error (MAPE), the best analytical models attain 5.15% (diffusivity) and 11.63% (permeability) error. The ANNs outperform the analytical prediction formulas with the same inputs, which indicates that the microstructure-property relationships are more complex and nonlinear than can be expressed through simple analytical models. In addition, ANNs can naturally incorporate high-dimensional descriptors like distributions (which haven been characterized by quantiles), and correlation functions. The best ANN models attain a MAPE of 2.03% (diffusivity) and 6.51% (permeability). However, one downside of ANNs is that their results are not as interpretable as those based on analytical prediction formulas. Furthermore, the CNNs outperform the best-performing ANNs, where the best CNN models attain 1.65% (diffusivity) and 3.78% (permeability) MAPE. This comes at the price of a significant increase in training time and use of computational resources, and yet another decrease in interpretability of the results. The fact that the prediction quality of best-performing ANNs comes at least reasonably close to that of CNNs indicates that the microstructural descriptors considered in this paper strongly influence mass transport and are thus suitable for predicting diffusivity and permeability. To our knowledge, analytical prediction formulas, ANNs, and CNNs have not been compared quantitatively on the same dataset before, and in particular not on such a large and diverse dataset. To facilitate further development of microstructure-property relationships, we make the artificially generated microstructures, their descriptors, and the code used to study the relationships between them available open access (Prifling et al., 2021c).

TABLE 6 | Error measures for the prediction of M and κ via ordinary CNN and DT-CNN where MSE is given for the training, validation and test sets, and MAPE for the test set. Note that MSE is evaluated on the logit scale for M and log scale for κ, and MAPE on the linear scale.

		MSE			MAPE (in %)
Model	**Property**	**Training**	**Validation**	**Test**	**Test**
CNN	M	0.0016	0.0023	0.0020	1.65
DT-CNN	M	0.0020	0.0028	0.0022	1.87
CNN	κ	0.0033	0.0059	0.0050	4.33
DT-CNN	κ	0.0027	0.0053	0.0043	3.78

DATA AVAILABILITY STATEMENT

The microstructures, descriptors, and the code used to study microstructure-property relationships are available open access via the following Zenodo repository: https://zenodo.org/record/4047774, see Prifling et al. (2021c).

AUTHOR CONTRIBUTIONS

BP, MR and MN generated the data and performed the analysis of microstructure-property relationships. PT developed the microstructure model for fiber and channel systems. All authors discussed the results and contributed to writing the paper. VS designed and supervised the research.

FUNDING

The financial support of the Swedish Research Council for Sustainable Development (grant number 2019-01295) and the Swedish Research Council (grant number 2016–03809) is acknowledged. Furthermore, the presented work was financially supported by the Bundesministerium für Bildung und Forschung (BMBF) within the project HiStructures under the grant number 03XP0243D as well as by the Deutsche Forschungsgemeinschaft (DFG) under the grant number SCHM997/39-1. The computations were in part performed on resources at Chalmers Centre for Computational Science and Engineering (C3SE) provided by the Swedish National Infrastructure for Computing (SNIC). A GPU used for part of this research was donated by the NVIDIA Corporation.

ACKNOWLEDGMENTS

Victor Wåhlstrand Skärström is acknowledged for assistance with implementing the convolutional neural networks.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Available at: tensorlow.org (Accessed September 29, 2021).

Abdallah, B., Willot, F., and Jeulin, D. (2016). Morphological Modelling of Three-phase Microstructures of Anode Layers Using SEM Images. *J. Microsc.* 263, 51–63. doi:10.1111/jmi.12374

Allen, J., Weddle, P. J., Verma, A., Mallarapu, A., Usseglio-Viretta, F., Finegan, D. P., et al. (2021). Quantifying the Influence of Charge Rate and Cathode-Particle Architectures on Degradation of Li-Ion Cells through 3D Continuum-Level Damage Models. *J. Power Sourc.* 512, 230415. doi:10.1016/j.jpowsour.2021.230415

Araya-Polo, M., Alpak, F. O., Hunter, S., Hofmann, R., and Saxena, N. (2019). Deep Learning-Driven Permeability Estimation from 2D Images. *Comput. Geosci.* 24, 571–580. doi:10.1007/s10596-019-09886-9

Barman, S., Rootzén, H., and Bolin, D. (2019). Prediction of Diffusive Transport through Polymer Films from Characteristics of the Pore Geometry. *AIChE J.* 65, 446–457. doi:10.1002/aic.16391

Bear, J. (2018). *Modeling Phenomena of Flow and Transport in Porous Media. Theory and Applications of Transport in Porous Media*. Cham: Springer.

Bergstra, J., and Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *J. Mach. Learn. Res.* 13, 281–305.

Berryman, J. G. (1985). Bounds on Fluid Permeability for Viscous Flow through Porous media. *J. Chem. Phys.* 82, 1459–1467. doi:10.1063/1.448420

Bertei, A., Chueh, C.-C., Pharoah, J. G., and Nicolella, C. (2014). Modified Collective Rearrangement Sphere-Assembly Algorithm for Random Packings of Nonspherical Particles: Towards Engineering Applications. *Powder Tech.* 253, 311–324. doi:10.1016/j.powtec.2013.11.034

Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. (2017). Julia: A Fresh Approach to Numerical Computing. *SIAM Rev.* 59, 65–98. doi:10.1137/141000671

Birkholz, O., Neumann, M., Schmidt, V., and Kamlah, M. (2021). Statistical Investigation of Structural and Transport Properties of Densely-Packed Assemblies of Overlapping Spheres Using the Resistor Network Method. *Powder Tech.* 378, 659–666. doi:10.1016/j.powtec.2020.09.056

Bottou, L. (2010). "Large-scale Machine Learning with Stochastic Gradient Descent," in Proceedings of COMPSTAT'2010, Paris, France, August 22–27, 2021 (Paris: Springer), 177–186. doi:10.1007/978-3-7908-2604-3_16

Brooks, S., Gelman, A., Jones, G., and Meng, X. (2011). *Handbook of Markov Chain Monte Carlo. Chapman & Hall/CRC Handbooks of Modern Statistical Methods*. Boca Raton: CRC Press.

Byrd, R. H., Gilbert, J. C., and Nocedal, J. (2000). A Trust Region Method Based on Interior Point Techniques for Nonlinear Programming. *Math. Program.* 89, 149–185. doi:10.1007/pl00011391

Carman, P. (1937). Fluid Flow through Granular Beds. *Trans. Inst. Chem. Eng.* 15, 150–166.

Chiu, S. N., Stoyan, D., Kendall, W. S., and Mecke, J. (2013). *Stochastic Geometry and its Applications*. 3rd ed. Chichester: J. Wiley & Sons.

Clennell, M. B. (1997). Tortuosity: A Guide through the Maze. *Geol. Soc. Lond. Spec. Publications* 122, 299–344. doi:10.1144/gsl.sp.1997.122.01.18

Clevert, D.-A., Unterthiner, T., and Hochreiter, S. (2016). "Fast and Accurate Deep Network Learning by Exponential Linear Units (Elus)," in 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016. Editors Y. Bengio and Y. LeCun. Conference Track Proceedings.

Coleman, T. F., and Li, Y. (1996). An Interior Trust Region Approach for Nonlinear Minimization Subject to Bounds. *SIAM J. Optim.* 6, 418–445. doi:10.1137/0806023

Coleman, T. F., and Li, Y. (1994). On the Convergence of Interior-Reflective Newton Methods for Nonlinear Minimization Subject to Bounds. *Math. Programming* 67, 189–224. doi:10.1007/bf01582221

Cooper, S. J., Bertei, A., Shearing, P. R., Kilner, J. A., and Brandon, N. P. (2016). TauFactor: An Open-Source Application for Calculating Tortuosity Factors from Tomographic Data. *SoftwareX* 5, 203–210. doi:10.1016/j.softx.2016.09.002

Cooper, S. J., Kishimoto, M., Tariq, F., Bradley, R. S., Marquis, A. J., Brandon, N. P., et al. (2013). Microstructural Analysis of an LSCF Cathode Using *In Situ* Tomography and Simulation. *ECS Trans.* 57, 2671–2678. doi:10.1149/05701.2671ecst

Cybenko, G. (1989). Approximation by Superpositions of a Sigmoidal Function. *Math. Control. Signal. Syst.* 2, 303–314. doi:10.1007/bf02551274

Daley, D. J., and Vere-Jones, D. (2005). *An Introduction to the Theory of Point Processes I, Elementary Theory and Methods*. 2nd ed. New York: Springer.

Daley, D. J., and Vere-Jones, D. (2008). *An Introduction to the Theory of Point Processes II, General Theory and Structure*. 2nd ed. New York: Springer.

Ditscherlein, R., Furat, O., Löwer, E., Mehnert, R., Trunk, R., Leißner, T., et al. (2021). PARROT - A Pilot Study on the Open Access Provision of Particle Discrete Tomographic Datasets. *Microsc. Microanal.* (under revision).

Dunn, A., Wang, Q., Ganose, A., Dopp, D., and Jain, A. (2020). "Benchmarking Materials Property Prediction Methods: The Matbench Test Set and Automatminer Reference Algorithm. *Npj Comput. Mater.* 6, 138. doi:10.1038/s41524-020-00406-3

Feinauer, J., Brereton, T., Spettl, A., Weber, M., Manke, I., and Schmidt, V. (2015). Stochastic 3D Modeling of the Microstructure of Lithium-Ion Battery Anodes via Gaussian Random fields on the Sphere. *Comput. Mater. Sci.* 109, 137–146. doi:10.1016/j.commatsci.2015.06.025

Furat, O., Petrich, L., Finegan, D. P., Diercks, D., Usseglio-Viretta, F., Smith, K., et al. (2021). Artificial Generation of Representative Single Li-Ion Electrode Particle Architectures from Microscopy Data. *Npj Comput. Mater.* 7, 105. doi:10.1038/s41524-021-00567-9

Gaiselmann, G., Neumann, M., Schmidt, V., Pecho, O., Hocker, T., and Holzer, L. (2014). Quantitative Relationships between Microstructure and Effective Transport Properties Based on Virtual Materials Testing. *AIChE J.* 60, 1983–1999. doi:10.1002/aic.14416

Gallier, J. (2000). *Curves and Surfaces in Geometric Modelling: Theory and Algorithms*. San Francisco: Morgan Kaufmann.

Gebäck, T., and Heintz, A. (2014). A Lattice Boltzmann Method for the Advection-Diffusion Equation with Neumann Boundary Conditions. *Commun. Comput. Phys.* 15, 487–505. doi:10.4208/cicp.161112.230713a

Gebäck, T., Marucci, M., Boissier, C., Arnehed, J., and Heintz, A. (2015). Investigation of the Effect of the Tortuous Pore Structure on Water Diffusion through a Polymer Film Using Lattice Boltzmann Simulations. *J. Phys. Chem. B.* 119, 5220–5227. doi:10.1021/acs.jpcb.5b01953

Ghanbarian, B., Hunt, A. G., Ewing, R. P., and Sahimi, M. (2013). Tortuosity in Porous Media: A Critical Review. *Soil Sci. Soc. America J.* 77, 1461–1477. doi:10.2136/sssaj2012.0435

Ginzburg, I., Verhaeghe, F., and Humières, D. d. (2008). Study of Simple Hydrodynamic Solutions with the Two-Relaxation-Times Lattice Boltzmann Scheme. *Commun. Comput. Phys.* 3, 519–581.

Glorot, X., and Bengio, Y. (2010). "Understanding the Difficulty of Training Deep Feedforward Neural Networks," in Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, May 13–15, 2010. Editors Y. W. Teh and M. Titterington, 249–256.

Gonzalez, R. C., and Woods, R. E. (2008). *Digital Image Processing*. 3rd ed. New Jersey: Prentice-Hall.

Graczyk, K. M., and Matyka, M. (2020). Predicting Porosity, Permeability, and Tortuosity of Porous media from Images by Deep Learning. *Sci. Rep.* 10, 21488. doi:10.1038/s41598-020-78415-x

Hein, S., Danner, T., Westhoff, D., Prifling, B., Scurtu, R., Kremer, L., et al. (2020). "Influence of Conductive Additives and Binder on the Impedance of Lithium-Ion Battery Electrodes: Effect of Morphology. *J. Electrochem. Soc.* 176, 013546. doi:10.1149/1945-7111/ab6b1d

Hein, S., Feinauer, J., Westhoff, D., Manke, I., Schmidt, V., and Latz, A. (2016). Stochastic Microstructure Modeling and Electrochemical Simulation of Lithium-Ion Cell Anodes in 3D. *J. Power Sourc.* 336, 161–171. doi:10.1016/j.jpowsour.2016.10.057

Hernández-García, A., and König, P. (2018). *Further Advantages of Data Augmentation on Convolutional Neural Networks*. Editors V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis (Cham:

Springer International Publishing), 95–103. doi:10.1007/978-3-030-01418-6_10

Hlushkou, D., Liasneuski, H., Tallarek, U., and Torquato, S. (2015). Effective Diffusion Coefficients in Random Packings of Polydisperse Hard Spheres from Two-point and Three-point Correlation Functions. *J. Appl. Phys.* 118, 124901. doi:10.1063/1.4931153

Holzer, L., Marmet, P., Keller, L., Fingerle, M., Wiegmann, A., Neumann, M., et al. (2021), Review of Tortuosity: Classical Theories and Modern Methods. Working paper (under preparation).

Hoffmann, T., and Gastreich, M. (2019). The Next Level in Chemical Space Navigation: Going Far beyond Enumerable Compound Libraries. *Drug Discov. Today* 24, 1148–1156. doi:10.1016/j.drudis.2019.02.013

Holzer, L., Wiedenmann, D., Münch, B., Keller, L., Prestat, M., Gasser, P., et al. (2013). The Influence of Constrictivity on the Effective Transport Properties of Porous Layers in Electrolysis and Fuel Cells. *J. Mater. Sci.* 48, 2934–2952. doi:10.1007/s10853-012-6968-z

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer Feedforward Networks Are Universal Approximators. *Neural Networks* 2, 359–366. doi:10.1016/0893-6080(89)90020-8

Hoshen, J., and Kopelman, R. (1976). Percolation and Cluster Distribution. I. Cluster Multiple Labeling Technique and Critical Concentration Algorithm. *Phys. Rev. B.* 14, 3438–3445. doi:10.1103/physrevb.14.3438

Ioffe, S., and Szegedy, C. (2015). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in Proceedings of the 32nd International Conference on Machine Learning, vol. 37 of ICML15, Lille, France, July 7–9, 2015, 448–456.

Jeulin, D. (2021). *Morphological Models of Random Structures. Interdisciplinary Applied Mathematics*. Cham: Springer.

Jiao, Y., and Torquato, S. (2012). Quantitative Characterization of the Microstructure and Transport Properties of Biopolymer Networks. *Phys. Biol.* 9, 036009. doi:10.1088/1478-3975/9/3/036009

Jungnickel, D. (2007). *Graphs, Networks and Algorithms. (Algorithms and Computation in Mathematics)*. 3rd ed. Berlin: Springer.

Kamrava, S., Tahmasebi, P., and Sahimi, M. (2020). Linking Morphology of Porous Media to Their Macroscopic Permeability by Deep Learning. *Transp Porous Med.* 131, 427–448. doi:10.1007/s11242-019-01352-5

Kaviany, M. (2012). *Principles of Heat Transfer in Porous Media*. 2nd ed. New York: Springer.

Kawaguchi, M., Tanabe, K., Yamada, K., Sawa, T., Hasegawa, S., Hayashi, M., et al. (2021). Determination of the Dzyaloshinskii-Moriya Interaction Using Pattern Recognition and Machine Learning. *Npj Comput. Mater.* 7, 20. doi:10.1038/s41524-020-00485-2

Kehrwald, D., Shearing, P. R., Brandon, N. P., Sinha, P. K., and Harris, S. J. (2011). Local Tortuosity Inhomogeneities in a Lithium Battery Composite Electrode. *J. Electrochem. Soc.* 158, A1393–A1399. doi:10.1149/2.079112jes

Kozeny, J. (1927). Über kapillare Leitung des Wassers im Boden: Aufstieg, Versickerung und Anwendung auf die Bewässerung. *Sitz. Ber. Akad. Wiss, Wien, Math. Nat.* 136, 271–306.

Kremer, L. S., Danner, T., Hein, S., Hoffmann, A., Prifling, B., Schmidt, V., et al. (2020). Influence of the Electrolyte Salt Concentration on the Rate Capability of Ultra-Thick NCM 622 Electrodes. *Batteries. Supercaps* 3, 1172–1182. doi:10.1002/batt.202000098

Krogh, A., and Hertz, J. (1992). "A Simple Weight Decay Can Improve Generalization," in Proceedings of the 4th International Conference on Neural Information Processing Systems, Denver, Colorado, December 2–5, 1991, 950–957.

Krüger, T., Kusumaatmaja, H., Kuzmin, A., Shardt, O., Silva, G., and Viggen, E. M. (2017). *The Lattice Boltzmann Method – Principles and Practice*. Cham: Springer.

Lang, A., and Potthoff, J. (2011). Fast Simulation of Gaussian Random Fields. *Monte Carlo Methods Appl.* 17, 195–214. doi:10.1515/mcma.2011.009

Lantuéjoul, C. (2002). *Geostatistical Simulation: Models and Algorithms*. Berlin: Springer.

Lantuéjoul, C., and Maisonneuve, F. (1984). Geodesic Methods in Quantitative Image Analysis. *Pattern Recognition* 17, 177–187. doi:10.1016/0031-3203(84)90057-8

Last, G., and Penrose, M. (2017). *Lectures on the Poisson Process*. Cambridge: Cambridge University Press.

LeCun, Y. A., Bottou, L., Orr, G. B., and Müller, K.-R. (2012), "Efficient BackProp," in *Neural Networks: Tricks of the Trade*, 2nd ed. Editors G. Montavon, G. B. Orr, and K.-R. Müller (Berlin: Springer), 9–48. doi:10.1007/978-3-642-35289-8_3

Liasneuski, H., Hlushkou, D., Khirevich, S., Höltzel, A., Tallarek, U., and Torquato, S. (2014). Impact of Microstructure on the Effective Diffusivity in Random Packings of Hard Spheres. *J. Appl. Phys.* 116, 034904. doi:10.1063/1.4889821

Lifshitz, I. M., and Slyozov, V. V. (1961). The Kinetics of Precipitation from Supersaturated Solid Solutions. *J. Phys. Chem. Sol.* 19, 35–50. doi:10.1016/0022-3697(61)90054-3

Ma, Z., and Torquato, S. (2018). Precise Algorithms to Compute Surface Correlation Functions of Two-phase Heterogeneous Media and Their Applications. *Phys. Rev. E.* 98, 013307. doi:10.1103/PhysRevE.98.013307

Mardia, K., and Jupp, P. (1999). *Directional Statistics*. Chichester: J. Wiley & Sons.

Masaro, L., and Zhu, X. X. (1999). Physical Models of Diffusion for Polymer Solutions, Gels and Solids. *Prog. Polym. Sci.* 24, 731–775. doi:10.1016/s0079-6700(99)00016-7

Matheron, G. (1975). *Random Sets and Integral Geometry*. New York: J. Wiley & Sons.

Matlab (2021). *Matlab, Version R2021a*. Natick, Massachusetts: The MathWorks Inc.

Maurer, C. R., Rensheng Qi, R., and Raghavan, V. (2003). A Linear Time Algorithm for Computing Exact Euclidean Distance Transforms of Binary Images in Arbitrary Dimensions. *IEEE Trans. Pattern Anal. Machine Intell.* 25, 265–270. doi:10.1109/tpami.2003.1177156

Mayer, J. (2004). A Time-Optimal Algorithm for the Estimation of Contact Distribution Functions of Random Closed Sets. *Image Anal. Stereol.* 23, 177–183. doi:10.5566/ias.v23.p177-183

Mayer, J., Schmidt, V., and Schweiggert, F. (2004). A Unified Simulation Framework for Spatial Stochastic Models. *Simulation Model. Pract. Theor.* 12, 307–326. doi:10.1016/j.simpat.2004.02.001

Miranville, A. (2019). *The Cahn–Hilliard Equation: Recent Advances and Applications*. Philadelphia: SIAM.

Moussaoui, H., Laurencin, J., Gavet, Y., Delette, G., Hubert, M., Cloetens, P., et al. (2018). Stochastic Geometrical Modeling of Solid Oxide Cells Electrodes Validated on 3D Reconstructions. *Comput. Mater. Sci.* 143, 262–276. doi:10.1016/j.commatsci.2017.11.015

Münch, B., and Holzer, L. (2008). Contradicting Geometrical Concepts in Pore Size Analysis Attained with Electron Microscopy and Mercury Intrusion. *Society* 91, 4059–4067. doi:10.1111/j.1551-2916.2008.02736.x

Nellis, G., and Klein, S. (2009). *Heat Transfer*. New York: Cambridge University Press.

Neumann, M., Hirsch, C., Staněk, J., Beneš, V., and Schmidt, V. (2019). Estimation of Geodesic Tortuosity and Constrictivity in Stationary Random Closed Sets. *Scand. J. Statist.* 46, 848–884. doi:10.1111/sjos.12375

Neumann, M., Osenberg, M., Hilger, A., Franzen, D., Turek, T., Manke, I., et al. (2019). On a Pluri-Gaussian Model for Three-phase Microstructures, with Applications to 3D Image Data of Gas-Diffusion Electrodes. *Comput. Mater. Sci.* 156, 325–331. doi:10.1016/j.commatsci.2018.09.033

Neumann, M., Staněk, J., Pecho, O. M., Holzer, L., Beneš, V., and Schmidt, V. (2016). Stochastic 3D Modeling of Complex Three-phase Microstructures in SOFC-Electrodes with Completely Connected Phases. *Comput. Mater. Sci.* 118, 353–364. doi:10.1016/j.commatsci.2016.03.013

Neumann., M., Stenzel, O., Willot, F., Holzer, L., and Schmidt, V. (2020). Quantifying the Influence of Microstructure on Effective Conductivity and Permeability: Virtual Materials Testing. *Int. J. Sol. Structures* 184, 211–220. doi:10.1016/j.ijsolstr.2019.03.028

Newman, J., and Thomas-Alyea, K. (2004). *Electrochemical Systems. The ECS Series of Texts and Monographs*. 3rd ed. Hoboken: J. Wiley & Sons.

Nguyen, T.-T., Demortière, A., Fleutot, B., Delobel, B., Delacourt, C., and Cooper, S. J. (2020). The Electrode Tortuosity Factor: Why the Conventional Tortuosity Factor Is Not Well Suited for Quantifying Transport in Porous Li-Ion Battery Electrodes and What to Use Instead. *Npj Comput. Mater.* 6, 123. doi:10.1038/s41524-020-00386-4

Ohser, J., and Mücklich, F. (2000). *Statistical Analysis of Microstructures in Materials Science*. Chichester: J. Wiley & Sons.

Ohser, J., and Schladitz, K. (2009). *3D Images of Materials Structures: Processing and Analysis*. Weinheim: Wiley VCH.

Prifling, B., Ademmer, M., Single, F., Benevolenski, O., Hilger, A., Osenberg, M., et al. (2021). Stochastic 3D Microstructure Modeling of Anodes in Lithium-Ion Batteries with a Particular Focus on Local Heterogeneity. *Comput. Mater. Sci.* 192, 110354. doi:10.1016/j.commatsci.2021.110354

Prifling, B., Neumann, M., Hlushkou, D., Kübel, C., Tallarek, U., and Schmidt, V. (2021). "Generating Digital Twins of Mesoporous Silica by Graph-Based Stochastic Microstructure Modeling. *Comput. Mater. Sci.* 187, 109934. doi:10.1016/j.commatsci.2020.109934

Prifling, B., Röding, M., Townsend, P., Neumann, M., and Schmidt, V. (2021). Data from: Large-scale Statistical Learning for Mass Transport Prediction in Porous Materials Using 90,000 Artificially Generated Microstructures [Data set] Zenodo. doi:10.5281/zenodo.4047774

Prifling, B., Westhoff, D., Schmidt, D., Markötter, H., Manke, I., Knoblauch, V., et al. (2019). "Parametric Microstructure Modeling of Compressed Cathode Materials for Li-Ion Batteries. *Comput. Mater. Sci.* 169, 109083. doi:10.1016/j.commatsci.2019.109083

Prill, T., Jeulin, D., Willot, F., Balach, J., and Soldera, F. (2017). Prediction of Effective Properties of Porous Carbon Electrodes from a Parametric 3D Random Morphological Model. *Transp Porous Med.* 120, 141–165. doi:10.1007/s11242-017-0913-1

Qian, N. (1999). On the Momentum Term in Gradient Descent Learning Algorithms. *Neural Networks* 12, 145–151. doi:10.1016/s0893-6080(98)00116-6

Röding, M., Ma, Z., and Torquato, S. (2020). Predicting Permeability via Statistical Learning on Higher-Order Microstructural Information. *Sci. Rep.* 10, 15239. doi:10.1038/s41598-020-72085-5

Röding, M., Gaska, K., Kádár, R., and Lorén, N. (2018). Computational Screening of Diffusive Transport in Nanoplatelet-Filled Composites: Use of Graphene to Enhance Polymer Barrier Properties. *ACS Appl. Nano Mater.* 1, 160–167. doi:10.1021/acsanm.7b00067

Röding, M., Schuster, E., Logg, K., Lundman, M., Bergström, P., Hanson, C., et al. (2016). Computational High-Throughput Screening of Fluid Permeability in Heterogeneous Fiber Materials. *Soft Matter* 12, 6293–6299. doi:10.1039/c6sm01213b

Röding, M. (2017). Shape-dependent Effective Diffusivity in Packings of Hard Cubes and Cuboids Compared with Spheres and Ellipsoids. *Soft Matter* 13, 8864–8870. doi:10.1039/c7sm01910f

Röding, M. (2018). "Effective Diffusivity in Lattices of Impermeable Superballs. *Phys. Rev. E.* 98, 052908. doi:10.1103/physreve.98.052908

Russ, J. C. (2007). *The Image Processing Handbook*. 5th ed. Boca Raton: CRC Press.

Sahimi, M. (2011). *Flow and Transport in Porous Media and Fractured Rock: From Classical Methods to Modern Approaches*. 2nd ed. Weinheim: Wiley VCH.

Saunders, R., Butler, C., Michopoulos, J., Lagoudas, D., Elwany, A., and Bagchi, A. (2021). "Mechanical Behavior Predictions of Additively Manufactured Microstructures Using Functional Gaussian Process Surrogates. *Npj Comput. Mater.* 7, 81. doi:10.1038/s41524-021-00548-y

Schladitz, K., Ohser, J., and Nagel, W. (2007). "Measuring Intrinsic Volumes in Digital 3D Images," in 13th International Conference Discrete Geometry for Computer Imagery, Szeged, Hungary, October 25–27, 2006. Editors A. Kuba, L. Nyúl, and K. Palágyi (Berlin: Springer), 247–258.

Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. *Neural Networks* 61, 85–117. doi:10.1016/j.neunet.2014.09.003

Sen, A., and Srivastava, M. (2012). *Regression Analysis: Theory, Methods, and Applications*. Springer Texts in Statistics. New York: Springer.

Serra, J. (1982). *Image Analysis and Mathematical Morphology*. London: Academic Press.

Soille, P. (2003). *Morphological Image Analysis: Principles and Applications*. 2nd ed. New York: Springer.

Srisutthiyakorn, N. (2016), "Deep-learning Methods for Predicting Permeability from 2D/3D Binary-Segmented Images," in 2016 SEG International Exposition and Annual Meeting, 3042–3046. SEG Technical Program Expanded Abstracts. doi:10.1190/segam2016-13972613.1

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* 15, 1929–1958.

Stenzel, O., Hassfeld, H., Thiedmann, R., Koster, L. J. A., Oosterhout, S. D., van Bavel, S. S., et al. (2011). Spatial Modeling of the 3D Morphology of Hybrid Polymer-ZnO Solar Cells, Based on Electron Tomography Data. *Ann. Appl. Stat.* 5, 1920–1947. doi:10.1214/11-aoas468

Stenzel, O., Pecho, O., Holzer, L., Neumann, M., and Schmidt, V. (2017). Big Data for Microstructure-property Relationships: A Case Study of Predicting Effective Conductivities. *AIChE J.* 63, 4224–4232. doi:10.1002/aic.15757

Stenzel, O., Pecho, O., Holzer, L., Neumann, M., and Schmidt, V. (2016). Predicting Effective Conductivities Based on Geometric Microstructure Characteristics. *AIChE J.* 62, 1834–1843. doi:10.1002/aic.15160

Sudakov, O., Burnaev, E., and Koroteev, D. (2019). Driving Digital Rock towards Machine Learning: Predicting Permeability with Gradient Boosting and Deep Neural Networks. *Comput. Geosciences* 127, 91–98. doi:10.1016/j.cageo.2019.02.002

Thorat, I. V., Stephenson, D. E., Zacharias, N. A., Zaghib, K., Harb, J. N., and Wheeler, D. R. (2009). Quantifying Tortuosity in Porous Li-Ion Battery Materials. *J. Power Sourc.* 188, 592–600. doi:10.1016/j.jpowsour.2008.12.032

Tjaden, B., Brett, D. J. L., and Shearing, P. R. (2018). Tortuosity in Electrochemical Devices: A Review of Calculation Approaches. *Int. Mater. Rev.* 63, 47–67. doi:10.1080/09506608.2016.1249995

Torquato, S. (2002). *Random Heterogeneous Materials - Microstructure and Macroscopic Properties*. New-York: Springer.

Torquato, S. (1991). Random Heterogeneous Media: Microstructure and Improved Bounds on Effective Properties. *Appl. Mech. Rev.* 44, 37–76. doi:10.1115/1.3119494

Toussaint, G. T. (1980). The Relative Neighbourhood Graph of a Finite Planar Set. *Pattern Recognition* 12, 261–268. doi:10.1016/0031-3203(80)90066-7

Townsend, P., Larsson, E., Karlson, T., Hall, S. A., Lundman, M., Bergström, P., et al. (2021). Stochastic Modelling of 3D Fiber Structures Imaged with X-ray Microtomography. *Comput. Mater. Sci.* 194, 110433. doi:10.1016/j.commatsci.2021.110433

Umehara, M., Stein, H., Guevarra, D., Newhouse, P., Boyd, D., and Gregoire, J. (2019). Analyzing Machine Learning Models to Accelerate Generation of Fundamental Materials Insights. *Npj Comput. Mater.* 5, 34. doi:10.1038/s41524-019-0172-5

van der Linden, J., Narsilio, G., and Tordesillas, A. (2016). Machine Learning Framework for Analysis of Transport through Complex Networks in Porous, Granular media: A Focus on Permeability. *Phys. Rev. E.* 94, 022904. doi:10.1103/PhysRevE.94.022904

Waltz, R. A., Morales, J. L., Nocedal, J., and Orban, D. (2006). An interior Algorithm for Nonlinear Optimization that Combines Line Search and Trust Region Steps. *Math. Program* 107, 391–408. doi:10.1007/s10107-004-0560-5

Wang, H., Pietrasanta, A., Jeulin, D., Willot, F., Faessel, M., Sorbier, L., et al. (2015). Modelling Mesoporous Alumina Microstructure with 3D Random Models of Platelets. *J. Microsc.* 260, 287–301. doi:10.1111/jmi.12295

Wang, H., Yin, Y., Hui, X., Bai, J., and Qu, Z. (2020). Prediction of Effective Diffusivity of Porous media Using Deep Learning Method Based on Sample Structure Information Self-Amplification. *Energy and AI.* 2, 100035. doi:10.1016/j.egyai.2020.100035

Westhoff, D., Manke, I., and Schmidt, V. (2018). Generation of Virtual Lithium-Ion Battery Electrode Microstructures Based on Spatial Stochastic Modeling. *Comput. Mater. Sci.* 151, 53–64. doi:10.1016/j.commatsci.2018.04.060

Westhoff, D., Skibinski, J., Šedivý, O., Wysocki, B., Wejrzanowski, T., and Schmidt, V. (2018). Investigation of the Relationship between Morphology and Permeability for Open-Cell Foams Using Virtual Materials Testing. *Mater. Des.* 147, 1–10. doi:10.1016/j.matdes.2018.03.022

Westhoff, D., van Franeker, J. J., Brereton, T., Kroese, D. P., Janssen, R. A. J., and Schmidt, V. (2015). Stochastic Modeling and Predictive Simulations for the Microstructure of Organic Semiconductor Films Processed with Different Spin Coating Velocities. *Model. Simul. Mat. Sci. Eng.* 23, 045003. doi:10.1088/0965-0393/23/4/045003

Willot, F., and Forest, S. (2018). *Physics and Mechanics of Random Media: From Morphology to Material Properties* (Paris: Presses des MINES).

Wu, H., Fang, W. Z., Kang, Q., Tao, W. Q., and Qiao, R. (2019). Predicting Effective Diffusivity of Porous Media from Images by Deep Learning. *Sci. Rep.* 9, 20387. doi:10.1038/s41598-019-56309-x

Wu, J., Yin, X., and Xiao, H. (2018). Seeing Permeability from Images: Fast Prediction with Convolutional Neural Networks. *Sci. Bull.* 63, 1215–1222. doi:10.1016/j.scib.2018.08.006

Xu, N., Huang, X., and Huang, S. (2017). A Measure of General Functional Dependence between Two Continuous Variables. *Commun. Stat. - Theor. Methods* 46, 4327–4352. doi:10.1080/03610926.2015.1081951

Zou, Q., and He, X. (1997). On Pressure and Velocity Boundary Conditions for the Lattice Boltzmann BGK Model. *Phys. Fluids* 9, 1591–1598. doi:10.1063/1.869307

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers in Materials

ORIGINAL RESEARCH
published: 02 February 2022
doi: 10.3389/fmats.2021.824958

Explainable Artificial Intelligence for Mechanics: Physics-Explaining Neural Networks for Constitutive Models

Arnd Koeppe[1]*, *Franz Bamer*[2], *Michael Selzer*[1,3], *Britta Nestler*[1,3] and *Bernd Markert*[2]

[1]Institute for Applied Materials - Computational Materials Science (IAM-CMS), Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, [2]Institute of General Mechanics (IAM), RWTH Aachen University, Aachen, Germany, [3]Institute for Digital Materials Science (IDM), Karlsruhe University of Applied Sciences (HSKA), Karlsruhe, Germany

(Artificial) neural networks have become increasingly popular in mechanics and materials sciences to accelerate computations with model order reduction techniques and as universal models for a wide variety of materials. However, the major disadvantage of neural networks remains: their numerous parameters are challenging to interpret and explain. Thus, neural networks are often labeled as black boxes, and their results often elude human interpretation. The new and active field of physics-informed neural networks attempts to mitigate this disadvantage by designing deep neural networks on the basis of mechanical knowledge. By using this a priori knowledge, deeper and more complex neural networks became feasible, since the mechanical assumptions can be explained. However, the internal reasoning and explanation of neural network parameters remain mysterious. Complementary to the physics-informed approach, we propose a first step towards a physics-explaining approach, which interprets neural networks trained on mechanical data a posteriori. This proof-of-concept explainable artificial intelligence approach aims at elucidating the black box of neural networks and their high-dimensional representations. Therein, the principal component analysis decorrelates the distributed representations in cell states of RNNs and allows the comparison to known and fundamental functions. The novel approach is supported by a systematic hyperparameter search strategy that identifies the best neural network architectures and training parameters. The findings of three case studies on fundamental constitutive models (hyperelasticity, elastoplasticity, and viscoelasticity) imply that the proposed strategy can help identify numerical and analytical closed-form solutions to characterize new materials.

Keywords: constitutive modeling, artificial intelligence, explainable AI, recurrent neural networks, principal component analysis

OPEN ACCESS

Edited by:
Norbert Huber,
Helmholtz-Zentrum Hereon, Germany

Reviewed by:
Ralf Denzer,
Lund University, Sweden
Ivan Argatov,
Malmö University, Sweden

***Correspondence:**
Arnd Koeppe
arnd.koeppe@kit.edu

Specialty section:
This article was submitted to Computational Materials Science, a section of the journal Frontiers in Materials

Received: *29 November 2021*
Accepted: *24 December 2021*
Published: *02 February 2022*

Citation:
Koeppe A, Bamer F, Selzer M, Nestler B and Markert B (2022) Explainable Artificial Intelligence for Mechanics: Physics-Explaining Neural Networks for Constitutive Models. Front. Mater. 8:824958. doi: 10.3389/fmats.2021.824958

1 INTRODUCTION

Data-driven models trained with deep learning algorithms have achieved tremendous successes in many research fields (LeCun et al., 2015). As the archetypical deep learning model, (artificial) neural networks and their variants are powerful predictors exceptionally well-suited for spatio-temporal data, such as mechanical tensor fields (Koeppe et al., 2020a). Each successive layer of a deep neural network learns to extract higher-level representations of the input and creates a data-driven model by

supervised learning. Due to the large variety of layers and cells, neural networks are highly modular and successful for many applications (LeCun et al., 2015).

In mechanics and materials sciences, machine learning algorithms accelerate the development of new materials (Bock et al., 2019), provide model order reduction techniques and enable data-driven constitutive models. Likewise, machine and deep learning algorithms have become an active field of research in the related domain of tribology Argatov (2019); Argatov and Chai (2021). As one of the first works in mechanics, Ghaboussi et al. (1991) proposed a unified constitutive model with shallow neural networks that learn from experimental data. Further extensions reduced the required number of experimental samples (Ghaboussi et al., 1998), adjusted the hidden layer dimensionality during training (Ghaboussi and Sidarta, 1998), and approximated the stiffness matrix (Hashash et al., 2004). Theocaris and Panagiotopoulos (1995) used dense neural networks to model kinematic hardening (Theocaris and Panagiotopoulos, 1995) and identify parameters for the failure mode of anisotropic materials (Theocaris et al., 1997). Shin and Pande (2000) and Javadi et al. (2003, 2009); Javadi and Rezania (2009) proposed "intelligent finite elements", neural network constitutive models that were applied to soils under cyclic loading and tunneling processes. Using context neurons in Recurrent Neural Networks (RNNs), Oeser and Freitag (2009), Graf et al. (2012), and Freitag et al. (2013) modeled elastoplastic and viscoelastic materials with fuzzy parameters. Bessa et al. (2017) proposed a data-driven analysis framework for materials with uncertain material parameters. For cantilever beams, Sadeghi and Lotfan (2017) used neural networks for nonlinear system identification and parameter estimation, while Koeppe et al. (2016) used dense neural networks to predict the displacement response. Extensions to continuum models resulted in linear "intelligent meta element" models of a cantilever beam (Koeppe et al., 2018a). To bridge the gap between atomistic and continuum mechanics, Teichert et al. (2019) trained integrable deep neural networks on atomistic scale models and successively approximated free energy functions. Stoffel et al. (2018) used dense neural networks to fit material data from high-velocity shock-wave tube experiments. Heider et al. (2020) investigated the frame invariance for graph-based neural networks predicting anisotropic elastoplastic material behavior. Huang et al. (2020) combined the proper orthogonal decomposition, manual history variables, and dense neural networks for hyperelasticity and plasticity. For materials applications, various data science methods have been successfully applied to simulation data and synthesized microstructures to analyze, characterize, and quantifify (e.g., Zhao et al. (2020); Altschuh et al. (2017)).

Despite the efforts to combine artificial intelligence and mechanics, the main disadvantage of neural networks remains: the learned parameters in black-box neural networks are challenging to interpret and explain (Breiman, 2001). Their high-level representations in deeper layers often elude human interpretation: what the neural network understood and how the individual parameter values can be explained remains incomprehensible. Coupled with insufficient data and limited computation capacities for past efforts, engineers and scientists mistrusted neural networks in favor of simpler models, whose fewer parameters could be easily interpreted and explained. Occam's razor, the well-known problem-solving principle by William of Ockham (ca. 1,287–1,347), became an almost dogmatic imperative: the simpler model with fewer parameters must be chosen if two models are equally accurate. However, with the advent of deep learning, the concept of simple models has become ambiguous. The shared representations common in dense neural networks require potentially infinite numbers of parameters to model arbitrary functions (Hornik et al., 1989). To handle the large dimensionality of mechanical spatio-temporal data, extensive parameter sharing, utilized by recursions (Freitag et al., 2013; Freitag et al., 2017; Koeppe et al., 2017; Koeppe et al., 2018b; Koeppe et al., 2019) and convolutions (Koeppe et al., 2020a; Koeppe et al., 2020b; Wu P. et al., 2020), introduces assumptions based on prior knowledge and user-defined parameters, i.e., hyperparameters, to reduce the number of trainable parameters.

By deriving the prior assumptions on the deep learning algorithm from mechanical knowledge, the recent trend in computational mechanics, enhanced by neural networks, aims towards physics-informed neural networks. Lagaris et al. (1998) and Aarts and van der Veer (2001) imposed boundary and initial conditions on dense neural networks, which were trained to solve unconstrained partial differential equations by differentiating the neural network graphs. In Ramuhalli et al. (2005), the authors designed their neural network by embedding a finite element model, which used the neural network's hidden layer to predict stiffness components and the output layer to predict the external force. Baymani et al. (2010) split the Stokes problem into three Poisson problems, approximated by three neural networks with superimposed boundary condition terms. In Rudd et al. (2014), a constrained backpropagation approach ensures that the boundary conditions are satisfied at each training step. The physics-informed neural network approach, as proposed by Raissi et al. (2019), employed two neural networks that contributed to the objective function. One physics-informed neural network mimicked the balance equation, while the second neural network approximated the potential function and pressure field. Yang and Perdikaris (2019) employed adversarial training to improve the performance of physics-informed neural networks. In Kissas et al. (2020), physics-informed neural networks model arterial blood pressure with magnetic resonance imaging data. Using the FEM to inspire a novel neural network architecture, Koeppe et al. (2020a) developed a deep convolutional recurrent neural network architecture that maps Dirichlet boundary constraints to force response and discretized field quantities in intelligent meta elements. Yao et al. (2020) used physics-informed deep convolutional neural networks to predict mechanical responses. Thus, using physics-informed approaches, neural networks with more parameters became feasible, whose architecture could be designed and explained using prior mechanical knowledge. However, the internal reasonings of the neural networks remained challenging to understand.

Complementary to the physics-informed approach, this work constitutes a new proof-of-concept approach towards a search approach radically different from the aforementioned design approach. Inspired by the explainable Artificial Intelligence (AI) research field (Bach et al., 2015; Arras et al., 2017; Alber et al., 2018; Montavon et al., 2018; Montavon et al., 2019; Samek et al., 2019), this novel approach elucidates the black box of neural networks to provide *physics-explaining* neural networks that complement the existing physics-informed neural networks. As a result, more powerful neural networks may be employed with confidence, since both the architectural design and the learned parameters can be explained.

This work has the objective to efficiently and systematically search neural network architectures for fundamental one-dimensional constitutive models and explain the trained neural networks. For this explanation, we propose a novel explainable AI method that uses the principal component analysis to interpret the cell states in RNNs. For the search, we define a wide hyperparameter search domain, implement an efficient search algorithm, and apply it to hyperelastic, viscoelastic, and elastoplastic constitutive models. After the search, the best neural network architectures are trained until they achieve low errors, and their generalization capabilities are tested. Finally, the developed explainable AI approach compares the temporal behavior of RNN cell states to known solutions to the fundamental physical problem, thereby demonstrating that neural networks without prior physical assumptions have the potential to inform mechanical research.

To the best of the authors' knowledge, this work constitutes a first proof-of-concept study proposing a dedicated explainable AI strategy for neural networks in mechanics, as well as a novel approach within the general field of explainable AI. Unique to RNNs, which are popular in mechanics (Graf et al., 2010; Freitag et al., 2011, Freitag et al., 2017; Cao et al., 2016; Koeppe et al., 2018b, 2019; Wu L. et al., 2020), this work complements popular strategies for classification problems that investigate kernels in convolutional neural networks or investigate the data flow through neural networks, such as layer-wise relevance propagation (Bach et al., 2015; Montavon et al., 2019; Samek et al., 2019). Moreover, our new approach complements unsupervised approaches to find parsimonious and interpretable models for hyperelastic material behavior (Flaschel et al., 2021), which are one of the most recent representatives of the principle of simplicity. Finally, the systematic hyperparameter search strategy offers an alternative strategy to self-designing neural networks, which have successfully modeled anisotropic and elastoplastic constitutive behavior (Fuchs et al., 2021; Heider et al., 2021).

In **Section 2.1**, we briefly review the necessary preliminaries for RNN training as the foundation for the following sections. **Section 2.2** details the systematic hyperparameter search strategy and explainable AI approach for data-driven constitutive models. For fundamental constitutive models, **Section 3** demonstrates the developed approaches in three case studies. Finally, **Section 4** discusses the approach and compares the results to related publications.

2 MATERIALS AND METHODS

2.1 Preliminaries

2.1.1 Training Artificial Neural Networks

Dense feedforward neural networks are the fundamental building block of neural networks. They represent nonlinear parametric mappings NN from inputs $\boldsymbol{x}$ to predictions $\hat{\boldsymbol{y}}$ with L consecutive layers and trainable parameters $\boldsymbol{\theta}$:

$$\text{NN:} \quad \boldsymbol{x}; \boldsymbol{\theta} \longmapsto \hat{\boldsymbol{y}} \quad \text{with} \quad \boldsymbol{\theta} = \left\{ \left(\boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)} \right) \dots \left(\boldsymbol{W}^{(L)}, \boldsymbol{b}^{(L)} \right) \right\}, \tag{1}$$

For dense feedforward neural networks, the parameters $\boldsymbol{\theta}$ include the layer weights $\boldsymbol{W}^{(l)}$ and biases $\boldsymbol{b}^{(l)}$. Each layer l applies a linear transformation to the layer inputs $\boldsymbol{x}^{(l)}$, before a nonlinear activation function $\mathrm{f}^{(l)}$ returns the layer activations $\boldsymbol{a}^{(l)}$

$$\boldsymbol{a}^{(l)} = \mathrm{f}^{(l)}\left(\boldsymbol{z}^{(l)} \right) \quad \text{with} \quad \boldsymbol{z}^{(l)} = \boldsymbol{W}^{(l)} \boldsymbol{x}^{(l)} + \boldsymbol{b}^{(l)} \quad \forall l = 1 \dots L. \tag{2}$$

Assembling and vectorizing these consecutive layers from $\boldsymbol{x} \equiv \boldsymbol{a}^{(0)}$ to $\hat{\boldsymbol{y}} \equiv \boldsymbol{a}^{(L)}$ enables fast computations on Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs).

Using supervised learning, neural network training identifies optimal parameters $\boldsymbol{\theta}^{\mathrm{opt}}$, which minimize the difference between desired outputs and neural network predictions. From a wide variety of input-target samples $(\boldsymbol{x}, \boldsymbol{y})$, the task of predicting physical responses $\hat{\boldsymbol{y}} \approx \boldsymbol{y}$ can be reformulated as a machine learning problem, where the prediction represents the expectation of the data probability distribution $\mathrm{p}^{\mathrm{data}}(\boldsymbol{y} \mid \boldsymbol{x})$:

$$\hat{\boldsymbol{y}}(\boldsymbol{x}) \approx \mathbb{E}\left[\mathrm{p}^{\mathrm{data}}(\boldsymbol{y} \mid \boldsymbol{x}) \right]. \tag{3}$$

Assuming a single-peak distribution, e.g., a normal distribution $\mathrm{p}^{\mathrm{data}}(\boldsymbol{y} \mid \boldsymbol{x}) \approx \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{x})$, the maximum likelihood estimation principle (cf. Goodfellow et al. (2016)) yields a compatible loss function $\mathcal{L}(\boldsymbol{y}, \hat{\boldsymbol{y}})$ for the prediction of physical values, such as the Mean Squared Error (MSE) averaged for batch size M and number of features F, i.e., $\mathcal{L}(\boldsymbol{y}, \hat{\boldsymbol{y}}) \coloneqq (MF)^{-1} \boldsymbol{y} \cdot \hat{\boldsymbol{y}}$. Using this loss function to compute scalar errors ϵ, the contributions of each parameter to the error can be backpropagated to compute gradients and train the neural network with gradient descent. To ensure generalization to unknown samples, the datasets used are split (usually randomly) into training, validation, and test sets. Gradient descent with backpropagation is only performed on the training set, while the validation set safeguards against overfitting of the parameters $\boldsymbol{\theta}$. The test set safeguards against overfitting of the chosen hyperparameter values $\boldsymbol{\gamma}$, which include, e.g., the dimensionalities of each layer dim $(\boldsymbol{a}^{(l)})$ and the number of layers L.

Recurrent neural networks (Rumelhart et al., 1986) introduce time-delayed recursions to feedforward neural networks (Goodfellow et al., 2016). Thus, RNNs use parameter sharing to reuse parameters efficiently for sequential data-driven problems. Such sequential data $\boldsymbol{x}_t$ may be sorted according to the rate-dependent real time or a pseudo time. One straightforward implementation of an RNN cell introduces a single tensor-valued recursion as given by

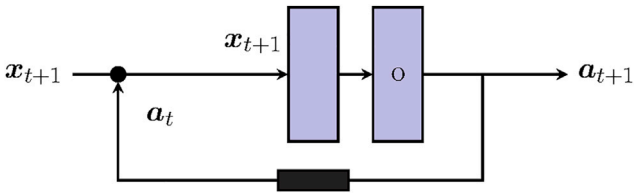


FIGURE 1 | Visualization of a recurrent cell. The recurrent cell consists of two feedforward layers (light violet squares). A time-delayed recursion (black square) loops the activation of the first or last layer back as additional input to the cell.

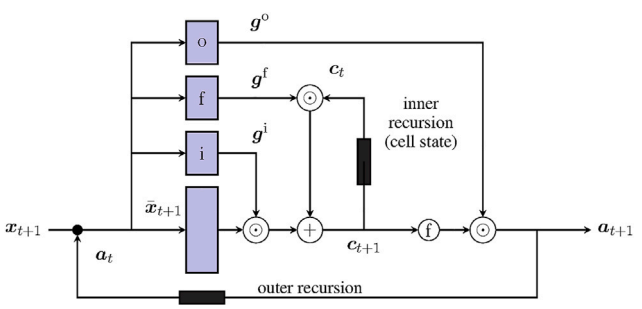


FIGURE 2 | Visualization of an LSTM cell, which consists of multiple feedforward layers (light violet). Three gates, $\boldsymbol{g}^{\mathrm{i}}$, $\boldsymbol{g}^{\mathrm{f}}$, and $\boldsymbol{g}^{\mathrm{o}}$, control the data flow into the cell, within the cell, and out of the cell. Two recursions (black rectangles) ensure stable gradients across many time steps.

$$\bar{\boldsymbol{x}}_{t+1} = \left[\, \boldsymbol{x}_{t+1} \;\; \boldsymbol{a}_t \,\right]^{\mathsf{T}}, \tag{4}$$

$$\boldsymbol{c}_{t+1} = \mathrm{f}\left(\boldsymbol{W}\bar{\boldsymbol{x}}_{t+1} + \boldsymbol{b}\right), \tag{5}$$

$$\boldsymbol{a}_{t+1} = \boldsymbol{W}^{\mathrm{o}}\boldsymbol{c}_{t+1} + \boldsymbol{b}^{\mathrm{o}}. \tag{6}$$

In **Figure 1**, the recurrent cell is depicted as a graph of two conventional dense layers and a time-delayed recursion of previous activation $\boldsymbol{a}_t$. Subsequently embedded into a larger sequence tensor $\boldsymbol{X} = [\boldsymbol{x}_1 \ldots \boldsymbol{x}_T]$, RNNs can be trained with backpropagation through time (Rumelhart et al., 1986).

However, simple RNNs (**Eq. 4** to **Eq. 6**) suffer from unstable gradients and fading memory (Hochreiter and Schmidhuber, 1997; Greff et al., 2015), which motivated the development of RNNs with gates. These gating layers control the data flow, thereby stabilizing the gradients and mitigating the fading memory. The Long Short-Term Memory (LSTM) cell (**Figure 2**) (Hochreiter and Schmidhuber, 1997; Gers et al., 2000) and the Gated Recurrent Unit (GRU) (Cho et al., 2014) are the most common gated RNNs and demonstrated comparable performances (Chung et al., 2014). As in **Eq. 4**, the previous activation $\boldsymbol{a}_t$ is concatenated with the input $\boldsymbol{x}_{t+1}$:

$$\bar{\boldsymbol{x}}_{t+1} = \left[\, \boldsymbol{x}_{t+1} \;\; \boldsymbol{a}_t \,\right]^{\mathsf{T}}, \tag{7}$$

$$\tilde{\boldsymbol{x}}_{t+1} = \bar{\boldsymbol{x}}_{t+1} \quad \text{or} \quad \tilde{\boldsymbol{x}}_{t+1} = \left[\, \bar{\boldsymbol{x}}_{t+1} \;\; \boldsymbol{c}_t \,\right]^{\mathsf{T}}. \tag{8}$$

The gates process the adjusted input $\tilde{\boldsymbol{x}}_{t+1}$ using dense layers, $(\boldsymbol{W}^{\mathrm{i}}, \boldsymbol{b}^{\mathrm{i}})$, $(\boldsymbol{W}^{\mathrm{f}}, \boldsymbol{b}^{\mathrm{f}})$, and $(\boldsymbol{W}^{\mathrm{o}}, \boldsymbol{b}^{\mathrm{o}})$, and apply a sigmoid activation, which yields

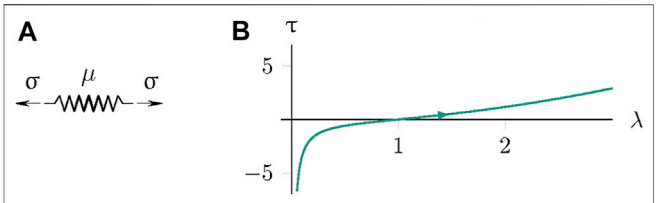


FIGURE 3 | One-dimensional hyperelastic rheological model **(A)** and example load cycle **(B)**.

$$\boldsymbol{g}^{\mathrm{i}} = \mathrm{sig}\left(\boldsymbol{W}^{\mathrm{i}}\tilde{\boldsymbol{x}}_{t+1} + \boldsymbol{b}^{\mathrm{i}}\right), \tag{9}$$

$$\boldsymbol{g}^{\mathrm{f}} = \mathrm{sig}\left(\boldsymbol{W}^{\mathrm{f}}\tilde{\boldsymbol{x}}_{t+1} + \boldsymbol{b}^{\mathrm{f}}\right), \tag{10}$$

$$\boldsymbol{g}^{\mathrm{o}} = \mathrm{sig}\left(\boldsymbol{W}^{\mathrm{o}}\tilde{\boldsymbol{x}}_{t+1} + \boldsymbol{b}^{\mathrm{o}}\right). \tag{11}$$

Therein, $\boldsymbol{g}^{\mathrm{i}}$ is the input gate activation, $\boldsymbol{g}^{\mathrm{f}}$ the forget gate activation, and $\boldsymbol{g}^{\mathrm{o}}$ the output gate activation, whose coefficient values are bounded between zero and one. Similar to Boolean masks, these gates control the data flow in the LSTM cell by

$$\boldsymbol{i}_{t+1} = \boldsymbol{g}^{\mathrm{i}} \odot \mathrm{f}\left(\boldsymbol{W}\bar{\boldsymbol{x}}_{t+1} + \boldsymbol{b}\right), \tag{12}$$

$$\boldsymbol{c}_{t+1} = \boldsymbol{g}^{\mathrm{f}} \odot \boldsymbol{c}_t + \boldsymbol{i}_{t+1}, \tag{13}$$

$$\boldsymbol{a}_{t+1} = \boldsymbol{g}^{\mathrm{o}} \odot \mathrm{f}\left(\boldsymbol{c}_{t+1}\right), \tag{14}$$

where the element-wise product $\odot$ allows the gate tensors $\boldsymbol{g}^{\mathrm{i}}$, $\boldsymbol{g}^{\mathrm{f}}$, and $\boldsymbol{g}^{\mathrm{o}}$ to control the incoming data flow into the cell $\boldsymbol{c}_{t+1}$, to forget information from selected entries, and to output selected information as $\boldsymbol{a}_{t+1}$. Despite being the foundation for the RNN reasoning and long-term memory, the cell states are often regarded as black boxes.

2.1.2 A Selection of Fundamental Constitutive Models

The systematic hyperparameter search and explainable AI approach will be demonstrated on three representative constitutive models, briefly reviewed in the following. These well-known models describe material effects, such as finite, lasting, and rate-dependent deformations. Since the ground truths for these one-dimensional models are known, the resulting neural network architectures from the hyperparameter search can be evaluated. Furthermore, for history-dependent material behaviors, the explainable AI approach can investigate the RNNs that best describe the constitutive models.

First, a hyperelastic constitutive model challenges neural networks to model finite strains and deal with the singular behavior of stretches that approach zero. The one-dimensional Neo-Hooke model (cf. Holzapfel (2000)) is regarded as one of the most straightforward nonlinear elastic material models. Therein, a single parameter μ describes the Cauchy stress τ as a function of the stretch λ:

$$\tau = \mu\left(\lambda^2 - \frac{1}{\lambda}\right). \tag{15}$$

Thus, the Neo-Hooke model offers the highest possible contrast to distributed neural network representations. In **Figure 3**, the corresponding rheological model (**Figure 3A**)

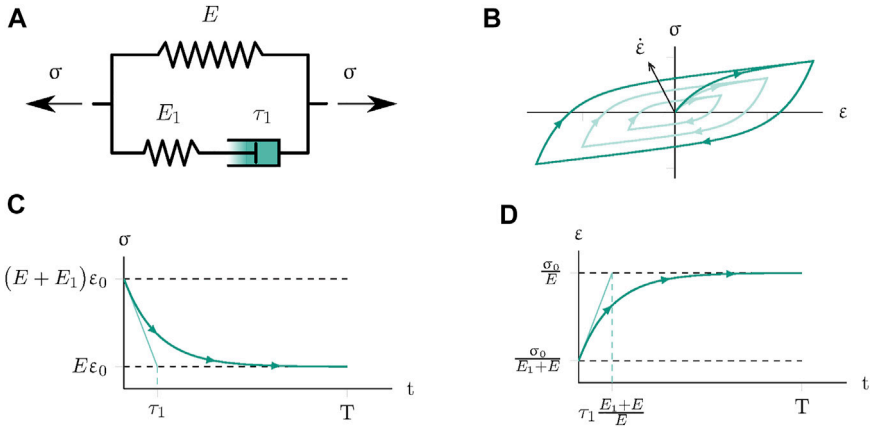


FIGURE 4 | One-dimensional viscoelastic material behavior. **(A)** Poynting-Thomson rheological model. **(B)** Cyclic loading hysteresis for constant strain rates $\dot{\varepsilon}$. **(C)** Stress relaxation response to a unit-sized step $\varepsilon(t) = \varepsilon_0\, H(t)$. **(D)** Creep retardation response to a unit-sized step $\sigma(t) = \sigma_0\, H(t)$.

and a loading and unloading cycle (**Figure 3B**) are depicted. The neural network approximation is challenged by stretch values near the origin, where the stress response exhibits singular behavior.

Second, viscoelasticity represents the archetype of fading-memory material behavior (Truesdell and Noll, 2004), which poses a short-term temporal regression problem to the neural networks. **Figure 4A** introduces the Poynting-Thomson or standard linear viscoelastic solid model as an example for rate-dependent inelastic material behavior. As the elementary example for generalized Maxwell models, which use multiple parallel Maxwell branches to cover more decades in the frequency range, it exhibits the same principal relaxation and creep behavior (Markert, 2005), as depicted in **Figure 4B–D**.

The additive decomposition of the stress σ and the evolution equations for each Maxwell branch i, as reviewed, e.g., in the textbook of Holzapfel (2000), yield

$$3\,\dot{\sigma}_i + \frac{\sigma_i}{\tau_i} = E_i\,\dot{\varepsilon} \quad \forall\, i = 1 \ldots I\,, \quad \text{and} \tag{16}$$

$$\sigma = E\,\varepsilon + \sum_i \sigma_i\,. \tag{17}$$

For each branch i, σ_i represents the stress, τ_i the relaxation time, and E_i the modulus. The strain ε is shared by all Maxwell branches and the elastic spring with modulus E. For $I = 1$, **Eqs 16**, **17** yield

$$\sigma + \tau_1 \dot{\sigma} = E\varepsilon + \tau_1\,(E + E_1)\dot{\varepsilon}. \tag{18}$$

Many known excitations, e.g., unit steps $\sigma(t) = \sigma_0\, H(t)$ or $\varepsilon(t) = \varepsilon_0\, H(t)$ yield closed-form solutions described by the relaxation function $R(t)$ and creep function $C(t)$ (Simo and Hughes, 1998):

$$\sigma(t) = \int_{-\infty}^{t} R\left(t - \tilde{t}\right)\dot{\varepsilon}\left(\tilde{t}\right) d\tilde{t} \quad \text{with} \tag{19}$$

$$R(t) = E + E_1 \exp\left(-\frac{t}{\tau_1}\right), \tag{20}$$

$$\varepsilon(t) = \int_{-\infty}^{t} C\left(t - \tilde{t}\right)\dot{\sigma}\left(\tilde{t}\right) d\tilde{t} \quad \text{with} \tag{21}$$

$$C(t) = \left[\frac{1}{E + E_1} + \frac{E_1}{E(E + E_1)}\left(1 - \exp\left(-\frac{tE}{\tau_1\,(E + SEi)}\right)\right)\right]. \tag{22}$$

For the general case of incremental excitations, numerical integration **Eq. 16** with an implicit backward Euler scheme yields

$$\begin{aligned} \sigma_{t+\Delta t} &= \frac{1}{c_{\sigma 1}}\,\left[c_{e1}\varepsilon_{t+\Delta t} - c_{e0}\varepsilon_t + c_{\sigma 0}\sigma_t\right] \\ \text{with} \quad c_{\sigma 0} &= \frac{\tau_1}{\Delta t}\,, \quad c_{\sigma 1} = 1 + \frac{\tau_1}{\Delta t}, \\ c_{\epsilon 0} &= \frac{\tau_1}{\Delta t}\,(E + E_1)\,, \quad c_{\epsilon 1} = E + \frac{\tau_1}{\Delta t}\,(E + E_1). \end{aligned} \tag{23}$$

Therein, σ_t and $\sigma_{t+\Delta t}$ represent the stresses while ε_t and $\varepsilon_{t+\Delta t}$ respectively represent the strains, respectively at the current and next increment. As a dissipative model, the dissipated energy can be described by the area surrounded by the hystereses, which depend on the strain rates (**Figure 4B**).

Finally, elastoplastic models exhibit path-dependent behavior with lasting deformations, i.e., long-term dependency behavior, and include potential discontinuities in the stress response, which need to be learned by the neural network. Numerical implementations of one-dimensional Prandtl-Reuss plasticity (**Figure 5A**) can be found, e.g., in Simo and Hughes (1998) or de Borst et al. (2012):

$$\sigma = E\,(\varepsilon - \varepsilon^{\mathrm{P}}), \tag{24}$$

$$\begin{aligned} \dot{\varepsilon}^{\mathrm{P}} &= \gamma\,\mathrm{sign}\,(\sigma - q_\sigma) \quad \text{and} \quad \dot{q}_\varepsilon = \gamma \\ &\text{and} \quad \dot{q}_\sigma = \gamma H\,\mathrm{sign}\,(\sigma - q_\sigma), \end{aligned} \tag{25}$$

$$f_y\,(\sigma, q_\varepsilon, q_\sigma) = |\sigma - q_\sigma| - (\sigma_Y + K q_\varepsilon). \tag{26}$$

In **Eqs 24 to 26**, the main history variable is the plastic strain ε^{P}, whose evolution follows the yield step γ if a yield criterion $f_y = 1$ is met. In **Figure 5B**, the characteristic stress-strain curve of one-dimensional Prandtl-Reuss plasticity is shown.

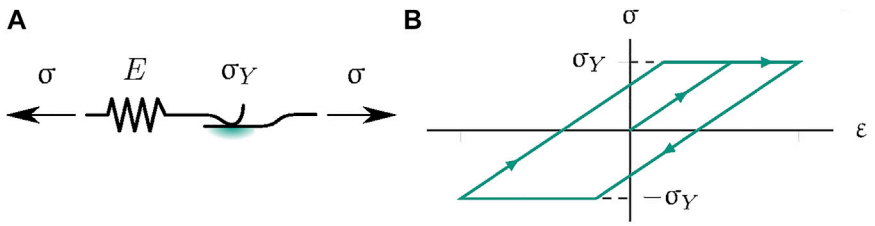


FIGURE 5 | One-dimensional Prandtl-Reuss elastoplastic rheological model **(A)** with perfect plastic material behavior **(B)**.

2.2 Explainable Artificial Intelligence for Mechanics

2.2.1 Data-Driven Constitutive Models for Fundamental Material Behavior

Data-driven constitutive models and intelligent finite elements were one of the first applications of neural networks within the FEM (Ghaboussi et al., 1991; Ghaboussi et al., 1998; Ghaboussi and Sidarta, 1998; Javadi et al., 2009), since they leverage the flexibility of the FEM to the fullest. For a strain-driven problem, data-driven constitutive models can be defined by

$$\text{IC: } \boldsymbol{\varepsilon} \overset{\boldsymbol{Q}}{\longmapsto} \boldsymbol{\sigma}, \tag{27}$$

where $\boldsymbol{\varepsilon}$ represents the strain and $\boldsymbol{\sigma}$ represents the stress. The history variables are gathered in $\boldsymbol{Q}$, which may include both algorithmic history variables, such as the plastic strain ε^{P}, and recurrent neural network cell states $\boldsymbol{C}$. As in the conventional FEM, the unified interface of inputs, outputs, and history variables enables a straightforward substitution of different data-driven constitutive models. For example, data-driven constitutive models, e.g., trained on experimental data (Stoffel et al., 2018), can substitute analytically derived constitutive models with trivial implementation effort. However, this flexibility massively increases the choice of conceivable neural network architectures and their defining hyperparameter configurations of data-driven constitutive models. Often, these hyperparameter configurations are chosen by the user or tuned using brute-force search algorithms.

2.2.2 Systematic Hyperparameter Search

Systematic hyperparameter search strategies constitute elegant and efficient solutions to finding optimal architectures and hyperparameter configurations. Hyperparameter search algorithms automatize the task of identifying advantageous neural network architectures and tuning hyperparameters $\boldsymbol{\gamma}$ to achieve good performance. Since neural networks are nonlinear function approximators with numerous parameters, the error surface is generally non-convex, high dimensional, potentially non-smooth, and noisy (Li et al., 2017). Furthermore, hyperparameters are interdependent, and the effects on the model remain unclear and problem-specific (Li et al., 2017), which often popularized brute force search algorithms, such as grid search (Bergstra and Bengio, 2012).

Grid search is the most fundamental search algorithm that seeks to cover the entire hyperparameter search domain. All possible combinations of user-defined hyperparameter values are tested one by one, with a fixed step size. The number and intervals of tested hyperparameter configurations $\boldsymbol{\Gamma}$ are set arbitrarily by the user, making the approach wasteful, inefficient, and infeasible for large numbers of hyperparameters.

Random search algorithms (Bergstra and Bengio, 2012) yield probabilistic approaches for larger numbers of hyperparameters. With enough trials, random search statistically explores the entire search space by testing a random selection of all possible combinations with varying step sizes. Thus, high-dimensional search spaces are explored faster, which makes random search a widely used search algorithm for hyperparameter-intensive neural networks (Bergstra and Bengio, 2012). Unfortunately, the computational effort of random search remains significant.

Therefore, this work follows approaches that increase the efficiency of the random search algorithm, leveraging the observation that the final performances of neural networks can be approximately assessed after a few epochs n, i.e., iterations that include the entire training dataset. The Successive Halving algorithm (Jamieson and Talwalkar, 2016), for example, trains randomly chosen hyperparameter configurations $\boldsymbol{\Gamma} = \{\boldsymbol{\gamma}^{(1)} \ldots \boldsymbol{\gamma}^{(C)}\}$ for n epochs. After that, the lowest-performing configurations are discarded, while the best-performing configurations are trained further. Thus, the approach focuses the computational resources on promising hyperparameter configurations.

Unfortunately, the number of epochs to train before deciding on discarding low-performing configurations constitutes another hyperparameter, which depends on the mechanical problem. Either Successive Halving can explore more configurations C for fewer epochs n or investigate fewer configurations C for more epochs n per decision. Li et al. (2017) solved this exploration issue with the Hyperband search algorithm (**Algorithm 1**). By gathering groups of configurations in h brackets, Successive Halving can be applied for different numbers of epochs n_h per group. The first bracket s = h maximizes exploration to test many configurations C with Successive Halving, identifying promising positions even for vast search domains and non-convex or non-smooth loss functions. In the last bracket s = 0, fewer configurations are tested for the full number of epochs, similar to a conventional random search. This is advantageous if the search domain is narrow and the objective functions are convex and smooth. Thus, Hyperband combines efficient exploration and investigation of promising architectures and does not introduce additional hyperparameters to be tuned.

In this work, we combine Hyperband with aggressive early stopping, which discards solutions that are unlikely to improve further by monitoring moving averages of the validation errors.

ALGORITHM 1 | Hyperband search (Li et al., 2017).

input: maximum epochs per configuration N, keep quotient η

initialization: number of brackets $h \coloneqq \lfloor \log_\eta(N) \rfloor$, total epochs for Hyperband $B \coloneqq (h+1)N$

for $\mathrm{s} = h \dots 0$ **do**

number of initial configurations in bracket

$$C \coloneqq \left\lceil \frac{B}{N} \frac{\eta^{\mathrm{s}}}{\mathrm{s}+1} \right\rceil$$

maximum epochs per configuration in current bracket

$$n \coloneqq \frac{N}{\eta^{\mathrm{s}}}$$

sample set of hyperparameter configurations

$$\boldsymbol{\Gamma} \coloneqq \left\{ \boldsymbol{\gamma}^{(1)} \dots \boldsymbol{\gamma}^{(C)} \right\} \quad \text{with} \quad \boldsymbol{\gamma} \sim \mathrm{p}^{\mathrm{hyper}}$$

for $\mathrm{h} = 0 \dots \mathrm{s}$ **do**

number of configurations in current Successive Halving step

$$C_h \coloneqq \left\lfloor \frac{C}{\eta^{\mathrm{h}}} \right\rfloor$$

epochs in current Successive Halving step

$$n_h \coloneqq n\eta^{\mathrm{h}}$$

train for n_h epochs and compute performance

$$\boldsymbol{\epsilon} = \left\{ \mathcal{L}(\mathrm{NN}_{\boldsymbol{\gamma}}\left(\boldsymbol{X}^{\mathrm{valid}}; \boldsymbol{\theta}_{\boldsymbol{\gamma}}(n_h)\right), \boldsymbol{Y}^{\mathrm{valid}}) \quad \forall \boldsymbol{\gamma} \in \boldsymbol{\Gamma} \right\}$$

reduce configuration set by η based on performance

$$\boldsymbol{\Gamma} = \mathrm{select\_best\_k\_configurations}\left(\boldsymbol{\Gamma}, \boldsymbol{\epsilon}, \mathrm{k} = \left\lfloor \frac{C}{\eta} \right\rfloor\right)$$

return $\boldsymbol{\gamma}^{\mathrm{best}} = \mathrm{argmin}\, \boldsymbol{\epsilon}(\boldsymbol{\gamma})$ with $\boldsymbol{\gamma} \in \boldsymbol{\Gamma}$

Independent of Hyperband, this additional logic further increases the efficiency of the search strategy for data-driven constitutive models. Thus, this fully automatized strategy identifies optimal neural network architectures for the given training and validation data without user interaction. The resulting hyperparameter configurations represent a 'natural' choice for the given problem, which merits further investigations into why this specific neural network was chosen.

2.2.3 The Novel Explainable Artificial Intelligence Approach

If the problem allows, it is generally possible to identify and train an efficient neural network to achieve accurate results and generalize to unknown data with the systematic approach outlined above. However, in the past, the proven approximation capabilities of neural networks (Hornik et al., 1989) were often shunned because the magical black-box-generated results could not be explained. Given only finite dataset sizes, any machine learning algorithm may learn spurious or accidental correlations. Since neural networks naturally develop distributed high-dimensional representations in deep layers, the 'reasoning' of neural networks is notoriously difficult to verify. Thus, a trained neural network may generalize to validation and test datasets but found it's decision-making on unphysical observations and biases in the dataset. The motivation of explainable AI is to unmask such 'Clever Hans' predictors[1] (Lapuschkin et al., 2019).

Many explainable neural network approaches, such as Layer-wise Relevance Propagation (LRP) (Bach et al., 2015; Arras et al., 2017; Montavon et al., 2018, Montavon et al., 2019), use the neural network graph to trace the activation back to its origin. In particular, such approaches are attractive for classification, because they can explain individual class labels, i.e., from single binary values to multiple inputs. For positive binary values, the activations can be traced back straightforwardly, explaining why the neural network chose the class associated with the binary value. However, multivariate regression problems are faced with a different problem: since the outputs are continuous, the neural network 'reasoning' must be interpretable over the full output ranges, including the origin. In particular, in balance equations, zero-valued residuals are at least as important as non-zero residuals and follow the same physical governing equations. Thus, for mechanical regression tasks, different explainable AI methods are necessary, which focus on the evolution of the high-dimensional representations.

As a new explainable AI approach using mechanical domain knowledge, our proposed approach focuses on the temporal evolution of mechanical quantities. Since time-variant problems in mechanics are often modeled by training RNNs on time-variant and path-dependent mechanical data (Freitag et al., 2011; Cao et al., 2016; Koeppe et al., 2019; Wu L. et al., 2020), we propose to use the Principal Component Analysis (PCA) to investigate recurrent cell states, e.g., in LSTM and GRU cells. Therefore, we interpret the time-variant cell states as statistical variables and use the PCA to identify the major variance directions in the distributed representations. With the original evolution equations and history variables known, major principal components can be compared with the known temporal evolution. If the cell states resemble the mechanical evolution equations of the algorithmic history variables, the neural network correctly understood the fundamental mechanical problem. For future materials, neural networks can thus be trained on new material test data, and the material can be possibly characterized by comparing the cell state principal components to known fundamental evolution equations.

The PCA (Pearson, 1901) constitutes an unsupervised learning algorithm that decorrelated data (**Figure 6**) (Goodfellow et al., 2016). As a linear transformation, the PCA identifies an orthonormal basis by maximizing the variance in the directions of the principal components. In the field of model order reduction and data analysis, the PCA is often used to compute proper orthogonal decompositions, which eliminate undesirable frequencies from mechanical problems and reduce the model dimensionality to achieve computational speed-up (cf. Freitag et al. (2017); Cao et al. (2016); Bamer et al. (2017); Huber (2021); Altschuh et al. (2017)). Used in this explainable AI approach, the PCA is not used to accelerate computations, but to analyze and explain neural network behavior.

To compute the PCA on a dataset with correlated features (**Figure 6A**), the dataset of M samples $\boldsymbol{x}_m$ is collected in a dataset tensor $\boldsymbol{X}$ and centered feature-wise to $\bar{\boldsymbol{X}}$:

$$\boldsymbol{X} = [\boldsymbol{x}_1 \dots \boldsymbol{x}_M]^{\mathsf{T}}, \tag{28}$$

$$\bar{\boldsymbol{X}} = \boldsymbol{X} - \mathbb{E}[\boldsymbol{X}] \tag{29}$$

Applied to the centered dataset tensor $\bar{\boldsymbol{X}}$, the Singular Value Decomposition (SVD),

[1]Apparently, the horse 'Clever Hans' (1895–1916) could count and calculate, but, in fact, interpreted the expressions and gestures of the audience to find the correct answers.

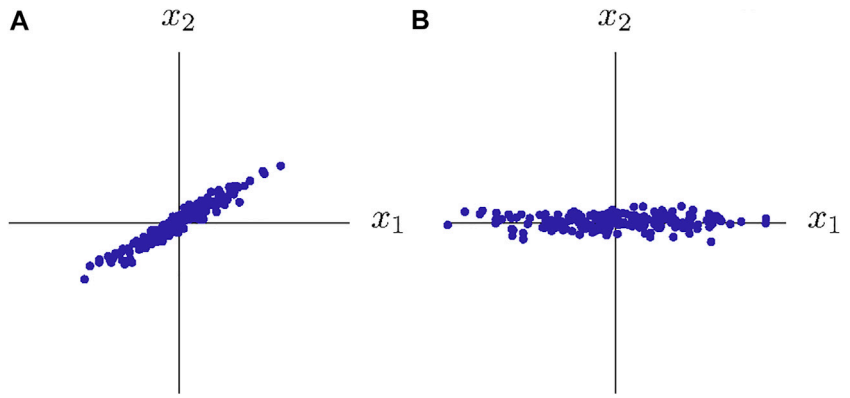


FIGURE 6 | Feature decoupling using the PCA. The PCA identifies the directions with the highest variance in the original data **(A)** and enables the extraction of linearly independent features **(B)**.

$$\text{SVD:} \quad \bar{\boldsymbol{X}} \longmapsto \boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{W}^{\mathsf{T}}, \tag{30}$$

decomposes $\bar{\boldsymbol{X}}$ into the left-singular vectors $\boldsymbol{U}$, the singular-value diagonal matrix $\boldsymbol{\Sigma}$, and the right-singular vectors $\boldsymbol{W}$. Conventionally, most SVD algorithms sort the singular vectors and singular values on the basis of the magnitude of the latter, where the highest singular value in $\boldsymbol{\Sigma}$ corresponds to the highest variance (i.e., frequency) in the dataset. Feature-wise contraction of the centered dataset $\bar{\boldsymbol{X}}$ with $\boldsymbol{W}$ yields the decoupled data tensor $\bar{\boldsymbol{x}}$:

$$\bar{\boldsymbol{x}} = \bar{\boldsymbol{X}}\boldsymbol{W}. \tag{31}$$

The reciprocal square root of the singular values $\sqrt{\boldsymbol{\Sigma}^{-1}}$ scales the decoupled dataset $\bar{\boldsymbol{x}}$ down to unit variance.

To use the PCA on LSTM cell states $\boldsymbol{c}_t$ from **Eq. 13**, the data tensor $\boldsymbol{C}$ of sequence length T is assembled:

$$\boldsymbol{C} \equiv [\boldsymbol{c}_1 \dots \boldsymbol{c}_T]. \tag{32}$$

Equations 29 to **32** yield

$$\bar{\boldsymbol{C}} = \bar{\boldsymbol{C}}\boldsymbol{W} \quad \text{with} \quad \bar{\boldsymbol{C}} = \boldsymbol{C} - \mathbb{E}[\boldsymbol{C}]. \tag{33}$$

The columns $\bar{\boldsymbol{C}}_t$ ($t = 1 \dots T$) of the decoupled state tensor $\bar{\boldsymbol{C}}$ describe the temporal behavior of the cell in the principal axes. The associated singular value Σ_f divided by the sum of all singular values quantifies the relative importance of the corresponding principal components, i.e., how much each principal component explains the cell state variance. Often, most of the variance can be explained using the first three principal components, $\bar{\boldsymbol{c}}^{\mathrm{I}}$, $\bar{\boldsymbol{c}}^{\mathrm{II}}$, and $\bar{\boldsymbol{c}}^{\mathrm{III}}$, which describe the memory response of the majority of the cell units. To investigate the ability to explain the mechanical problem, the neural network's major memory cell responses, $\bar{\boldsymbol{c}}^{\mathrm{I}}$, $\bar{\boldsymbol{c}}^{\mathrm{II}}$, and $\bar{\boldsymbol{c}}^{\mathrm{III}}$, are compared to the algorithmic history variables $\boldsymbol{q}$.

This comparison demonstrates that the neural network understood temporal mechanical problems in line with the physically observed evolution laws.

Note that the ability to generalize, i.e., to achieve a reproducible and equally accurate result on unknown data, as outlined in **Section 2.2.2**, remains independent of the ability to model the result on correct physical assumptions, as described in this subsection. To achieve generalizable results, systematic hyperparameter tuning thus is the necessary prerequisite for the explainable AI approaches.

3 RESULTS

The following three case studies demonstrate the systematic hyperparameter search strategy and the new explainable AI approach. First, the proposed systematic hyperparameter search strategy will be demonstrated in the scope of a case study for an intelligent nonlinear elastic constitutive model. Thereafter, the latter two case studies combine the systematic hyperparameter search strategy with explainable AI, in order to interpret inelastic time-variant constitutive behavior.

For all data-driven constitutive models, the same data-generation strategy provides training, validation, and test data. For a sequence length of $T = 10000$ and $\Omega = 5$ phases of loading and unloading, we sample control values from a random normal distribution $\mathcal{N}(\mu = 0, \sigma = 1)$. Between those control values, a variety of ramping functions, e.g., linear, quadratic, or sinusoidal, interpolate the intermediate values to assemble stress- or strain-controlled loading sequences. The ramping functions are selected to cover a variety of constitutive responses based on the investigated constitutive models. Thereafter, the numerical implementation of each reference constitutive model generates $M^{\mathrm{total}} = 10000$ samples, which are split randomly 70 %-15 %–15% into training, validation, and test set.

Three constitutive models are selected to demonstrate fundamental mechanical material behavior, including finite deformations, long-term temporal behavior, and rate dependency. To enhance interpretability, each one-dimensional constitutive model uses dimensionless and purely academic parameter values. Using preprocessing strategies, such as normalization and augmenting the input with explicit model parameters, models with arbitrary material parameter values can be created (Koeppe et al., 2018b; 2020a). First, an incompressible Neo-Hooke constitutive model ($\mu = \frac{1}{3}$) computes the nonlinear

TABLE 1 | Hyperband search domain for constitutive models.

Variable		Search domain
layer width	$d^{(l)}$	[4, 8, . . . , 128]
neural network depth	L	[2, 3, 4, 5, 6]
Base learning rate	α	$[1 \cdot 10^{-2}, 3 \cdot 10^{-3}, 1 \cdot 10^{-3}, 3 \cdot 10^{-4}, 1 \cdot 10^{-4}]$
batch size	M	[32, 64, 128]
a) activation (dense only)	f	[rect, sig, tanh, elu, splus]
b) cell type (recurrent only)	—	[LSTM, GRU, recurrent – tanh, recurrent – rect]

stress response to stretch-controlled loading. Due to the hyperelastic problem, all loading and unloading phases $\omega = 1...\Omega$ use linear ramps. The investigated time-distributed dense neural network architectures receive the stretch λ as input and learn to predict the Cauchy stress σ. Second, a perfect Prandtl-Reuss elastoplasticity model ($E = 1$, $\sigma_Y = 0.6$) is subjected to strain-controlled loading. The phase interpolation functions are sampled randomly from linear, quadratic, square-root, exponential, sine, and half-sine ramping functions. The investigated RNN architectures use the strain ε to return the stress σ and plastic strain ε^P. Finally, a Poynting-Thomson constitutive model, defined by $E = 1.0$, $E_1 = 0.5$, and $\tau_1 = 0.1667$ is integrated in time (T = 1) with an implicit backward Euler-scheme. The stress-controlled phases $\omega = 1...\Omega$ include linear ramps and constant phases to investigate creeping behavior. The RNN architectures process the stress σ and return the strain ε and the viscous branch stress σ_1. For all neural networks, a final time-distributed dense layer of output size applies a linear transformation to cover the entire range of real values.

For the neural network architecture and the hyperparameters, the Hyperband algorithm (Li et al., 2017) systematically explores and investigates the high-dimensional search domain. During the training of each configuration, the Adam algorithm (Kingma and Ba, 2014) minimizes the MSE on the training set ($N = 51$ epochs) and reports the validation loss to evaluate the configuration in the scope of Hyperband. To avoid artificially penalizing specific weight values, neither L^1 nor L^2 regularization is used, optimizing the unconstrained MSE. In the Hyperband brackets, each step performed by the Successive Halving algorithm eliminates the worst configurations from all configurations by a factor of $\eta = 3.7$. In **Table 1**, the search domains for the recurrent and dense neural network architectures are described. The design variables, i.e., the layer width $d^{(l)}$, the neural network depth L, the base learning rate α and the batch size M vary during all search trials. Moreover, for dense neural networks (hyperelastic material behavior), multiple activation functions are investigated by the algorithm, whereas for recurrent neural networks (viscoelastic and elastoplastic material behavior), the cell type of the reccurent cells become additional design variables. To limit the potentially infinite search space, the layer widths are selected to be identifcal for all layers and the design variables are varied in heurisically defined steps and ranges. After the search, we train the best configuration, as evaluated by the lowest MSE on the validation set, for the full duration of $N = 301$ epochs. The resulting parameter values $\boldsymbol{\theta}^{\text{opt}}$ are used to evaluate the test set to compute the test errors.

For the last two case studies, the novel explainable neural network approach (**Section 2.2.3**), employing the PCA on the cell states, analyzes the best RNN architectures to explain the intelligent inelastic constitutive models.

The Kadi4Mat (Brandt et al., 2021) data management infrastructure stores, links, and describes the data generated in this publication. Published via the direct integration of Zenodo (Koeppe et al., 2021), the dataset includes the generated reference constitutive model dataset, the associated metadata of the constitutive and neural network models, and the serialized trained neural networks for different hyperparameter configurations at different epochs during training. The data storage with the associated metadata and connections enhances the systematic hyperparameter search strategy with the option for long-term data sustainability, e.g., by reusing previous search results and hyperparameter configurations for future models. Furthermore, the links between the dataset and the serialized neural network models provide a starting point for a bottom-up data ontology for machine learning in mechanics and material sciences. Subsequently formalized, the data ontology will provide semantic rules that describe the relations and workflows inherent to the research data, which will enable additional analysis and explainable AI approaches.

3.1 Systematic Investigation of Data-Driven Hyperelastic Constitutive Models

For the hyperelastic problem, the best-performing hyperparameters were a batch size of $M = 64$ and a base learning rate of $\alpha = 3 \cdot 10^{-3}$. A deep feedforward neural network with 5 hidden layers of width 112 and rectifier activation achieved the best performance. After training (**Figure 7**) for 301 epochs, the neural network achieves an MSE ϵ^{MSE} of $1.28 \cdot 10^{-5}$ on the training, $1.01 \cdot 10^{-5}$ on the validation, and $1.01 \cdot 10^{-5}$ on the test set. The same order of error magnitude on all three datasets indicates that the neural network achieved generalization. **Figure 8** visualizes three randomly selected test samples. For the one-dimensional hyperelastic case, the data-driven constitutive model is in perfect agreement with the reference model.

3.2 Explaining Data-Driven Elastoplastic Constitutive Models

For the elastoplastic problem, the systematic search strategy identified a best-performing architecture, characterized by four LSTM cells, with a width of 52 units followed by a linear time-distributed dense

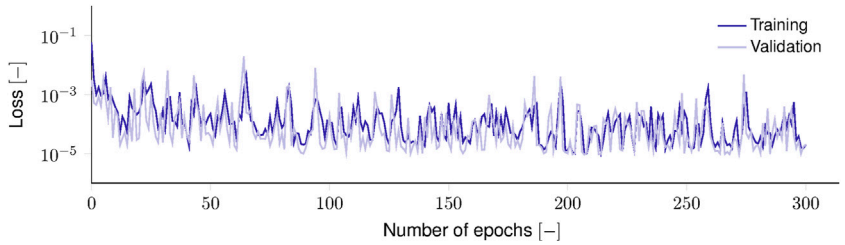


FIGURE 7 | Training and validation loss during hyperelastic constitutive model training. Both losses rapidly reduce within the first 20 epochs before oscillating due to variance shift combined with the singular behavior of the stretch near the origin.

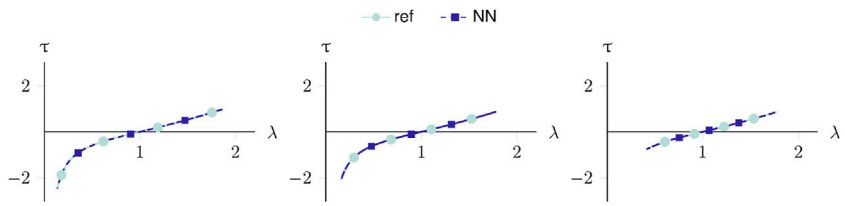


FIGURE 8 | Three randomly selected stress-stretch curves for hyperelastic constitutive behavior. The data-driven constitutive model exactly matches the reference solution.

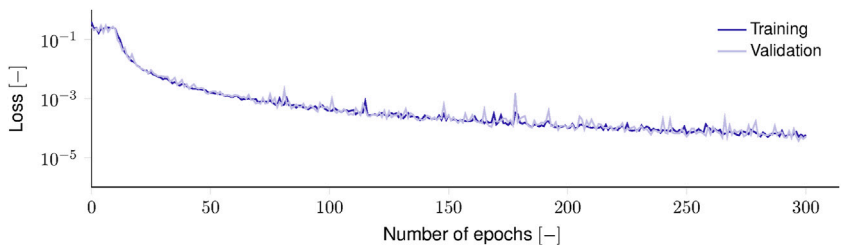


FIGURE 9 | Training and validation loss during elastoplastic constitutive model training.

layer with 2 units. For the training (**Figure 9**), a batch size of $M = 32$ and a base learning rate of $\alpha = 1 \cdot 10^{-3}$ resulted in the best result without overfitting, i.e., an MSE ϵ^{MSE} of $5.72 \cdot 10^{-5}$ on the training, $5.11 \cdot 10^{-5}$ on the validation, and $4.38 \cdot 10^{-5}$ on the test set.

To interpret and explain the RNN behavior, a random sample is extracted from the test set and evaluated using the data-driven constitutive model. The explainable AI approach uses the PCA on the concatenated cell states of all recurrent cells and yields the three principal components with the highest singular values. Expressed in the three major principal components and divided by their singular values, the cell states I, II, and III represent the joint response of the LSTM cell states. **Figure 10** visualizes and compares the evolution of the stresses, strains, and history variables over the entire loading sequence.

The neural network's output predictions and the reference constitutive model match exactly, i.e., the stresses and plastic strain predictions are accurate. By comparing the plastic strain with the internal cell states, the learned function of the LSTM cells becomes apparent, which governs the neural network's decision-making behavior. Without being trained directly, the cell states' principal components learn to approximate the plastic strain evolution (with a negative sign). Since the PCA chooses the principal directions based on the variance, and the output layer can apply arbitrary weighting, the negative sign of the cell state does not affect the result. The second and third principal components do not contribute to the joint cell state response.

3.3 Explaining Data-Driven Viscoelastic Constitutive Models

For the viscoelastic problem, the Hyperband search found the best-performing architecture to be three GRU cells with 120 units, followed by a time-distributed dense layer (2 units and linear activation). For training convergence, a batch size of $M = 32$ and a base learning rate of $\alpha = 1 \cdot 10^{-3}$ achieved the best results. After training for the full 301 epochs (**Figure 11**), the neural network achieved a best MSE ϵ^{MSE} of $2.05 \cdot 10^{-7}$ on the

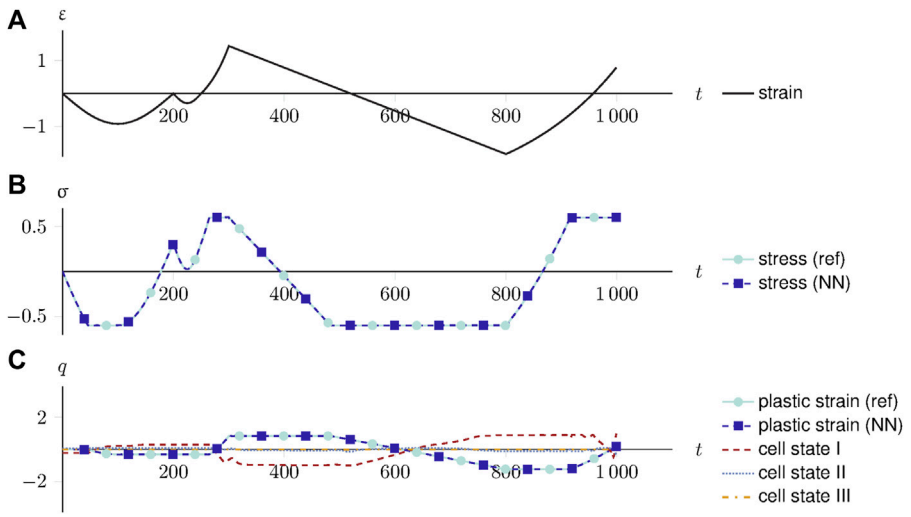


FIGURE 10 | An explainable elastoplastic constitutive model. **(A)** The strain-driven loading over the time increments. **(B)** The stress response of the reference and data-driven constitutive model. **(C)** The plastic strain, compared to the three major principal components of the cell state. The cell states approximate the plastic strain (with a negative sign remedied by the output layer).

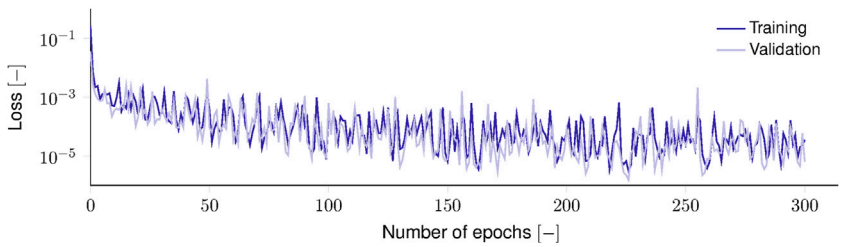


FIGURE 11 | Training and validation loss during viscoelastic constitutive model training.

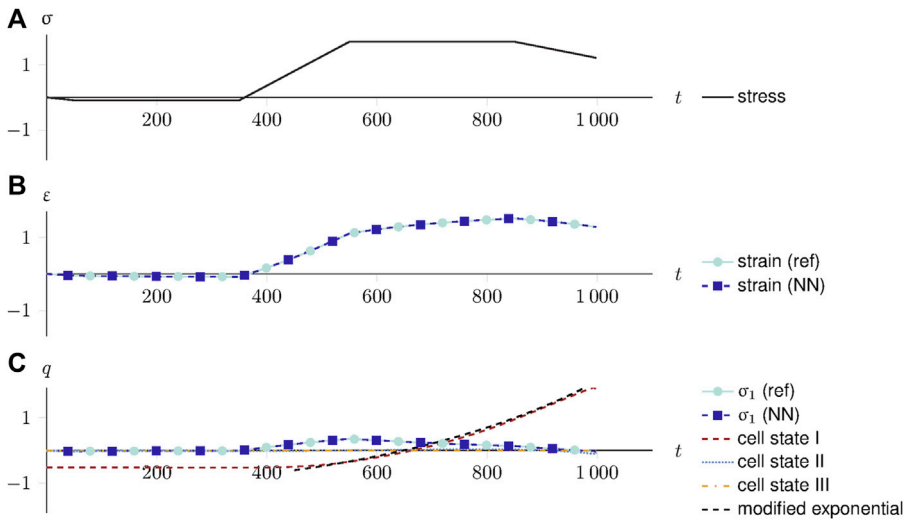


FIGURE 12 | An explainable viscoelastic constitutive model. **(A)** The stress-driven loading over time. **(B)** The strain response of the reference and data-driven constitutive model. **(C)** The branch history stress compared to the three major principal components of the cell state. Instead of mimicking the history variables used to generate training data, the cell states learned a generic solution for viscoelasticity: a modified exponential function that can be shifted and scaled at will by the output layer.

training, $7.36 \cdot 10^{-7}$ on the validation, and $1.15 \cdot 10^{-6}$ on the test set, indicating generalization. Furthermore, since all samples are generated with unit variance and unit stiffness, the model approximates almost at the machine precision of the element-wise definition of the MSE performance metric ($\sim \mathcal{O}(10^7)$ for single-precision floating-point arithmetic).

Figure 12 exemplifies the explainable AI strategy on one randomly generated sample that was not used for training or validation. In the top row, the stress is plotted over the increments, which constitutes the input to the reference constitutive and neural network models. The second row compares the strain response of the reference and neural network models. The final row depicts the history variables q. The branch stress σ_1 is computed by both the reference and data-driven constitutive models. Both strain and branch stress match accurately, as indicated by the low test loss. Finally, the PCA on the GRU cell state over time yields the data-driven constitutive history variables, i.e., the three principal components with the highest singular values, which govern the major evolution of the data-driven constitutive model.

Instead of approximating the history variables σ_1, used to generate the training data with backward Euler time integration, the first principal component of the cell states approximates a modified exponential function, correctly identifying the exponential behavior of viscoelasticity. Shifting, scaling, and changing the sign of a function represent trivial operations for previous and subsequent neural network layers. Thus, the weights of the RNN and the last dense layer can modify the cell output at will to assemble a solution, such as **Eq. 21**. The second and third principal components do not contribute to the joint cell state response.

4 DISCUSSION

4.1 Results and Approach

The objectives of this publication were to systematically and automatically identify and train neural networks for data-driven constitutive models of fundamental material behavior and to explain the resulting recurrent neural networks' behavior.

The test errors ($2.45 \cdot 10^{-6}$ to $4.38 \cdot 10^{-5}$) approach machine precision for the single-precision floating-point arithmetic used to compute them. The neural network performance surpasses the elastoplastic constitutive model for uniaxial tension and compression in the recent works of Huang et al. (2020) ($3.93 \cdot 10^{-2}$) and Alwattar and Mian (2019) ($\sim 1 \cdot 10^{-5}$, as reported on the training set). For each of the case studies, the results were in the same order of magnitude for all datasets, regardless of whether they were shown to the neural networks during training (training dataset) or unknown to the neural network (validation and test dataset). The oscillations in the loss curves for the viscoelastic and hyperelastic cases can be attributed to the shift in randomly selected samples for each batch averaged over each epoch in combination with the stronger continuous nonlinearities of the problems. This suggests, that no overfitting occured and generalization to unknown data was achieved, despite the considerable number of trained parameters in some of the models identified by the search algorithm.

When using manual hyperparameter tuning, mechanical and physical 1257 knowledge, and machine learning expertise, considerably smaller neural networks with fewer parameters with fewer layers and units per layer are conceivable that could have been trained to achieve similar results. Limiting the capacity of machine learning models is one of the first and most important steps in achieving robust performance and generalization to unknown data. This limit is governed by the amount and variance of the data, which is considerably harder to generate in abundance for experimental data (cf. Argatov and Chai (2019)). Here, many best-practice applications and textbooks suggest choosing a neural network capacity slightly larger than strictly necessary and utilizing additional regularization schemes for optimal generalization performance (Goodfellow et al., 2016; Argatov and Chai, 2019). However, the objectives of this study were to demonstrate an automated and inductive strategy that requires minimal prior knowledge to automatically identify and train models and to help understand the resulting models, regardless of how many parameters were used. Therefore, the less-than-minimalistic models identified by the hyperparameter search algorithm posed a more difficult challenge for the novel explainable AI approach.

The results were achieved inductively, with minimal user input, through the application of a Hyperband-inspired systematic search strategy on mechanical data. To further improve the results of the neural networks, several approaches are conceivable, most of which are deductively derived from deep learning and mechanical domain knowledge. Ensemble learning and related regularizers, such as dropout (Srivastava et al., 2014), for example, are known to improve the quality of the results further (Srivastava et al., 2014; Goodfellow et al., 2016). Similarly, physics-informed and physics-guided approaches, as proposed by Raissi et al. (2019), Yang and Perdikaris (2019), or Kissas et al. (2020), use mechanical domain knowledge to improve neural network models of physical systems. In contrast, the data-driven constitutive models trained in this work represent an inductive, fundamental, and accurate approach that offers intuition for possible neural network architectures and hyperparameter configurations for higher-dimensional mechanical problems. In combination with the explainable AI approach, the data-driven constitutive models represent the first step towards *physics-explaining* neural networks.

Even beyond mechanics and materials sciences, applying the PCA to explain neural network cell states constitutes a novel and promising explainable AI method. The PCA investigation proposed in this work can explain the recurrent cell state behavior, despite the considerable size of the investigated architecture, compared to the one-dimensional fundamental problems. Due to the quality of the results and the explainability of the neural network, "Clever-Hans" predictors (Lapuschkin et al., 2019) can be ruled out. For material behavior where multiple physical effects affect the same observed features (e.g., viscoplasticity or ductile fracture), the global PCA used in this proof-of-concept study could be upgraded to provide local decompositions in combination with clustering algorithms.

As an explainable neural network approach in mechanics, the case studies demonstrate how neural networks can help to explain material behavior. For the elastoplastic problem, the recurrent cell state identified the same history variables used in computing the ground truth. For the viscoelastic problem, the data generation used numerical time-integration, but the neural network found a solution similar to the exponential closed-form solution. Therefore, if the ground truth is not known, e.g., when learning from raw experimental data, explainable neural network approaches can possibly identify underlying closed-form solutions. By characterizing new materials or material behavior, e.g., at extreme loading conditions, explainable AI can help guide researchers in mechanics and materials sciences towards new analytic closed-form solutions that elegantly model the materials in the desired ranges.

4.2 Concluding Remarks and Outlook

We proposed a step towards *physics-explaining* neural networks, which inductively complement existing deductive approaches for physics-informed and physics-guided neural networks. To that end, a systematic hyperparameter search strategy was implemented to identify the best neural network architectures and training parameters efficiently. For the analysis of the best neural networks, we proposed a novel explainable AI approach, which uses the PCA to explain the distributed representations in the cell states of RNNs.

The search strategy and explainable AI approach were demonstrated on data-driven constitutive models that learned fundamental material behavior, i.e., one-dimensional hyperelasticity, elastoplasticity, and viscoelasticity. For all case studies, the best neural network architectures achieved test errors in the order of $1 \cdot 10^{-5}$ to $1 \cdot 10^{-6}$. In particular, for hyperelasticity, the test error approached machine precision, despite the singular behavior for stretches approaching zero. For elastoplasticity, the novel explainable AI approach identified that the recurrent cell states learned history variables equivalent to the plastic strain, i.e., the history variables used to generate the original data. Remarkably, for viscoelasticity, the explainable AI approach found that the best performing neural network architecture used an exponential function as the basis for its decisions instead of the algorithmic history variables used to generate the training data.

These findings imply that systematic hyperparameter search, coupled with explainable AI, can help identify and characterize numerical and analytical closed-form solutions for constitutive models independent of the data origin. Thus, new materials can potentially be characterized with data originating from experiments, using the approach proposed in this work.

Future studies will apply and extend the proposed strategies to more complex material models. Of particular interest are viscoelastic materials subjected to strain rates that cover multiple decades, where conventional numerical models require numerous algorithmic history variables. Eventually, new materials, where the analytic closed-form solutions are as-of-yet unknown and numerical solutions are challenging to implement, can be characterized with developments based on the present work. Finally, applications beyond constitutive models are conceivable. For spatio-temporal problems, e.g., as given in Koeppe et al. (2020a), the explainable AI approach outlined in this work needs to be extended to leverage the spatial structure.

In forthcoming work, it is intended to extend the developed artificial intelligence approach within the Kadi4Mat (Brandt et al., 2021) framework to higher dimensional data containing 2D and 3D spatial plus temporal information to predict microstructure-mechanics correlations. The database used to train the neural network algorithms relies on digital twin data from synchronously conducted experiments and simulations of mechanically loaded polycrystalline and multiphase materials. Based on the training, the AI approach is applied to large-scale micromechanics-microstructure simulations so as to provide new insights, e.g., into mechanically induced nucleation events of new phases and grain variants or into microcrack probabilities. The combination of new AI concepts and advanced high-performance materials simulations shall establish an integral component of the research data infrastructure to enable computational methods for an accelerated design of new materials.

DATA AVAILABILITY STATEMENT

The datasets generated and analyzed for this study can be found in the Zenodo repository https://zenodo.org/record/4699219.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

ACKNOWLEDGMENTS

The authors gratefully acknowledge financial support by the Federal Ministry of Education and Research (BMBF) and the Ministry of Science, Research and Art Baden-Württemberg as part of the Excellence Strategy of the German Federal and State Governments, through the projects FestBatt (project number 03XP0174E) and MoMaF - Science Data Center, and the state digitization strategy digital@bw (project number 57). The authors thank Leon Geisen for his editorial support.

REFERENCES

Aarts, L. P., and van der Veer, P. (2001). Neural Network Method for Solving Partial Differential Equations. *Neural Process. Lett.* 14, 261–271. doi:10.1023/A:1012784129883

Alber, M., Lapuschkin, S., Seegerer, P., Hägele, M., Schütt, K. T., Montavon, G., et al. (2018). *iNNvestigate Neural Networks!* ArXiv180804260 Cs Stat.

Altschuh, P., Yabansu, Y. C., Hötzer, J., Selzer, M., Nestler, B., and Kalidindi, S. R. (2017). Data Science Approaches for Microstructure Quantification and Feature Identification in Porous Membranes. *J. Membr. Sci.* 540, 88–97. doi:10.1016/j.memsci.2017.06.020

Alwattar, T., and Mian, A. (2019). Development of an Elastic Material Model for BCC Lattice Cell Structures Using Finite Element Analysis and Neural Networks Approaches. *J. Compos. Sci.* 3, 33. doi:10.3390/jcs3020033

Argatov, I. I., and Chai, Y. S. (2019). An Artificial Neural Network Supported Regression Model for Wear Rate. *Tribology Int.* 138, 211–214. doi:10.1016/j.triboint.2019.05.040

Argatov, I. I., and Chai, Y. S. (2021). Fretting Wear with Variable Coefficient of Friction in Gross Sliding Conditions. *Tribology Int.* 153, 106555. doi:10.1016/j.triboint.2020.106555

Argatov, I. (2019). Artificial Neural Networks (ANNs) as a Novel Modeling Technique in Tribology. *Front. Mech. Eng.* 5, 30. doi:10.3389/fmech.2019.00030

Arras, L., Montavon, G., Müller, K.-R., and Samek, W. (2017). "Explaining Recurrent Neural Network Predictions in Sentiment Analysis," in Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Copenhagen, Denmark (Association for Computational Linguistics), 159–168. doi:10.18653/v1/W17-5221

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. (2015). On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLoS One* 10, e0130140. doi:10.1371/journal.pone.0130140

Bamer, F., Koeppe, A., and Markert, B. (2017). An Efficient Monte Carlo Simulation Strategy Based on Model Order Reduction and Artificial Neural Networks. *Proc. Appl. Math. Mech.* 17, 287–288. doi:10.1002/pamm.201710113

Baymani, M., Kerayechian, A., and Effati, S. (2010). Artificial Neural Networks Approach for Solving Stokes Problem. *Am* 01, 288–292. doi:10.4236/am.2010.14037

Bergstra, J., and Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *J. Mach. Learn. Res.* 13, 281–305. doi:10.5555/2188385.2188395

Bessa, M. A., Bostanabad, R., Liu, Z., Hu, A., Apley, D. W., Brinson, C., et al. (2017). A Framework for Data-Driven Analysis of Materials under Uncertainty: Countering the Curse of Dimensionality. *Comput. Methods Appl. Mech. Eng.* 320, 633–667. doi:10.1016/j.cma.2017.03.037

Bock, F. E., Aydin, R. C., Cyron, C. J., Huber, N., Kalidindi, S. R., and Klusemann, B. (2019). A Review of the Application of Machine Learning and Data Mining Approaches in Continuum Materials Mechanics. *Front. Mater.* 6, 110. doi:10.3389/fmats.2019.00110

Brandt, N., Griem, L., Herrmann, C., Schoof, E., Tosato, G., Zhao, Y., et al. (2021). Kadi4Mat: A Research Data Infrastructure for Materials Science. *Data Sci. J.* 20, 8. doi:10.5334/dsj-2021-008

Breiman, L. (2001). Statistical Modeling: The Two Cultures (With Comments and a Rejoinder by the Author). *Statist. Sci.* 16, 199–231. doi:10.1214/ss/1009213726

Cao, B.-T., Freitag, S., and Meschke, G. (2016). A Hybrid RNN-GPOD Surrogate Model for Real-Time Settlement Predictions in Mechanised Tunnelling. *Adv. Model. Simulation Eng. Sci.* 3, 5. doi:10.1186/s40323-016-0057-9

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., et al. (2014). Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. ArXiv Prepr. ArXiv14061078. doi:10.3115/v1/d14-1179

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. ArXiv14123555 Cs.

de Borst, R., Crisfield, M. A., Remmers, J. J. C., and Verhoosel, C. V. (2012). *Nonlinear Finite Element Analysis of Solids and Structures*. West Sussex, United Kingdom: Wiley.

Flaschel, M., Kumar, S., and De Lorenzis, L. (2021). *Unsupervised Discovery of Interpretable Hyperelastic Constitutive Laws*. ArXiv201013496 Cs.

Freitag, S., Graf, W., Kaliske, M., and Sickert, J. U. (2011). Prediction of Time-dependent Structural Behaviour with Recurrent Neural Networks for Fuzzy Data. *Comput. Struct.* 89, 1971–1981. doi:10.1016/j.compstruc.2011.05.013

Freitag, S., Graf, W., and Kaliske, M. (2013). A Material Description Based on Recurrent Neural Networks for Fuzzy Data and its Application within the Finite Element Method. *Comput. Struct.* 124, 29–37. doi:10.1016/j.compstruc.2012.11.011

Freitag, S., Cao, B. T., Ninić, J., and Meschke, G. (2017). Recurrent Neural Networks and Proper or Thogonal Decomposition With Interval Data for Real-Time Predictions of Mechanised Tunnelling Processes. *Comput. Struct.* 207, 258–273. doi:10.1016/j.compstruc.2017.03.020

Fuchs, A., Heider, Y., Wang, K., Sun, W., and Kaliske, M. (2021). DNN2: A Hyper-Parameter Reinforcement Learning Game for Self-Design of Neural Network Based Elasto-Plastic Constitutive Descriptions. *Comput. Struct.* 249, 106505. doi:10.1016/j.compstruc.2021.106505

Gers, F. A., Schmidhuber, J., and Cummins, F. (2000). Learning to Forget: Continual Prediction with LSTM. *Neural Comput.* 12, 2451–2471. doi:10.1162/089976600300015015

Ghaboussi, J., and Sidarta, D. E. (1998). New Nested Adaptive Neural Networks (NANN) for Constitutive Modeling. *Comput. Geotech.* 22, 29–52. doi:10.1016/S0266-352X(97)00034-7

Ghaboussi, J., Garrett, J., Jr., and Wu, X. (1991). Knowledge-Based Modeling of Material Behavior with Neural Networks. *J. Eng. Mech.* 117, 132–153. doi:10.1061/(asce)0733-9399(1991)117:1(132)

Ghaboussi, J., Pecknold, D. A., Zhang, M., and Haj-Ali, R. M. (1998). Autoprogressive Training of Neural Network Constitutive Models. *Int. J. Numer. Meth. Engng.* 42, 105–126. doi:10.1002/(sici)1097-0207(19980515)42:1<105:aid-nme356>3.0.co;2-v

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

Graf, W., Freitag, S., Kaliske, M., and Sickert, J.-U. (2010). Recurrent Neural Networks for Uncertain Time-Dependent Structural Behavior. *Comput.-Aided Civ. Infrastruct. Eng.* 25, 322–323. doi:10.1111/j.1467-8667.2009.00645.x

Graf, W., Freitag, S., Sickert, J.-U., and Kaliske, M. (2012). Structural Analysis with Fuzzy Data and Neural Network Based Material Description. *Comput.-Aided Civ. Infrastruct. Eng.* 27, 640–654. doi:10.1111/j.1467-8667.2012.00779.x

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J. (2015). *LSTM: A Search Space Odyssey*. ArXiv150304069 Cs.

Hashash, Y. M. A., Jung, S., and Ghaboussi, J. (2004). Numerical Implementation of a Neural Network Based Material Model in Finite Element Analysis. *Int. J. Numer. Methods Eng.* 59, 989–1005. doi:10.1002/nme.905

Heider, Y., Wang, K., and Sun, W. (2020). SO(3)-invariance of Informed-Graph-Based Deep Neural Network for Anisotropic Elastoplastic Materials. *Comput. Methods Appl. Mech. Eng.* 363, 112875. doi:10.1016/j.cma.2020.112875

Heider, Y., Suh, H. S., and Sun, W. (2021). An Offline Multi-Scale Unsaturated Poromechanics Model Enabled by Self-Designed/self-Improved Neural Networks. *Int. J. Numer. Anal. Methods Geomech* 45, 1212–1237. doi:10.1002/nag.3196

Hochreiter, S., and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Comput.* 9, 1735–1780. doi:10.1162/neco.1997.9.8.1735

Holzapfel, G. A. (2000). *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*. Chichester ; New York: Wiley.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer Feedforward Networks Are Universal Approximators. *Neural Netw.* 2, 359–366. doi:10.1016/0893-6080(89)90020-8

Huang, D., Fuhg, J. N., Weißenfels, C., and Wriggers, P. (2020). A Machine Learning Based Plasticity Model Using Proper Orthogonal Decomposition. *Comput. Methods Appl. Mech. Eng.* 365, 113008. doi:10.1016/j.cma.2020.113008

Huber, N. (2021). A Strategy for Dimensionality Reduction and Data Analysis Applied to Microstructure–Property Relationships of Nanoporous Metals. *Materials* 14, 1822. doi:10.3390/ma14081822

Jamieson, K., and Talwalkar, A. (2016). "Non-stochastic Best Arm Identification and Hyperparameter Optimization," in *Artificial Intelligence and Statistics*, 240–248.

Javadi, A., and Rezania, M. (2009). Intelligent Finite Element Method: An Evolutionary Approach to Constitutive Modeling. *Adv. Eng. Inform.* 23, 442–451. doi:10.1016/j.aei.2009.06.008

Javadi, A. A., Tan, T. P., and Zhang, M. (2003). Neural Network for Constitutive Modelling in Finite Element Analysis. *Comput. Assist. Mech. Eng. Sci.* 10, 523–530. doi:10.1016/B978-008044046-0.50086-5

Javadi, A., Tan, T., and Elkassas, A. (2009). "Intelligent Finite Element Method and Application to Simulation of Behavior of Soils under Cyclic Loading," in *Foundations of Computational Intelligence Volume 5. Studies in Computational Intelligence*. Editors A. Abraham, A.-E. Hassanien, and V. Snášel (Springer Berlin Heidelberg). 317–338. doi:10.1007/978-3-642-01536-6_12

Kingma, D., and Ba, J. (2014). Adam: A Method for Stochastic Optimization. ArXiv14126980 Cs.

Kissas, G., Yang, Y., Hwuang, E., Witschey, W. R., Detre, J. A., and Perdikaris, P. (2020). Machine Learning in Cardiovascular Flows Modeling: Predicting Arterial Blood Pressure from Non-invasive 4D Flow MRI Data Using Physics-Informed Neural Networks. *Comput. Methods Appl. Mech. Eng.* 358, 112623. doi:10.1016/j.cma.2019.112623

Koeppe, A., Bamer, F., and Markert, B. (2016). Model Reduction and Submodelling Using Neural Networks. *PAMM* 16, 537–538. doi:10.1002/pamm.201610257

Koeppe, A., Bamer, F., Hernandez Padilla, C. A., and Markert, B. (2017). Neural Network Representation of a Phase-Field Model for Brittle Fracture. *PAMM* 17, 253–254. doi:10.1002/pamm.201710096

Koeppe, A., Bamer, F., and Markert, B. (2018a). An Intelligent Meta-Element for Linear Elastic Continua. *PAMM* 18, e201800283. doi:10.1002/pamm.201800283

Koeppe, A., Hernandez Padilla, C. A., Voshage, M., Schleifenbaum, J. H., and Markert, B. (2018b). Efficient Numerical Modeling of 3D-Printed Lattice-Cell Structures Using Neural Networks. *Manuf. Lett.* 15, 147–150. doi:10.1016/j.mfglet.2018.01.002

Koeppe, A., Bamer, F., and Markert, B. (2019). An Efficient Monte Carlo Strategy for Elasto-Plastic Structures Based on Recurrent Neural Networks. *Acta Mech.* 230, 3279–3293. doi:10.1007/s00707-019-02436-5

Koeppe, A., Bamer, F., and Markert, B. (2020a). An Intelligent Nonlinear Meta Element for Elastoplastic Continua: Deep Learning Using a New Time-Distributed Residual U-Net Architecture. *Comput. Methods Appl. Mech. Eng.* 366, 113088. doi:10.1016/j.cma.2020.113088

Koeppe, A., Hesser, D. F., Mundt, M., Bamer, F., and Markert, B. (2020b). "Mechanik 4.0. Künstliche Intelligenz zur Analyse mechanischer Systeme," in *Handbuch Industrie 4.0: Recht, Technik, Gesellschaft*. Editor W. Frenz (Berlin, Heidelberg: Springer), 553–567. doi:10.1007/978-3-662-58474-3_28

Koeppe, A., Bamer, F., Selzer, M., Nestler, B., and Markert, B. (2021). *Dataset: Explainable Artificial Intelligence for Mechanics: Physics-Informing Neural Networks for Constitutive Models*. [Dataset]. doi:10.5281/zenodo.4699219

Lagaris, I., Likas, A., and Fotiadis, D. (1998). Artificial Neural Networks for Solving Ordinary and Partial Differential Equations. *IEEE Trans. Neural Netw.* 9, 987–1000. doi:10.1109/72.712178

Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., and Müller, K.-R. (2019). Unmasking Clever Hans Predictors and Assessing what Machines Really Learn. *Nat. Commun.* 10, 1096. doi:10.1038/s41467-019-08987-4

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep Learning. *Nature* 521, 436–444. doi:10.1038/nature14539

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., and Talwalkar, A. (2017). Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *J. Mach. Learn. Res.* 18, 6765–6816.

Markert, B. (2005). *Porous Media Viscoelasticity With Application To Polymeric Foams*. No. II-12 in Report/Universität Stuttgart, Institut Für Mechanik (Bauwesen), Lehrstuhl II (Essen: Verl. Glückauf)

Montavon, G., Samek, W., and Müller, K.-R. (2018). Methods for Interpreting and Understanding Deep Neural Networks. *Digital Signal. Process.* 73, 1–15. doi:10.1016/j.dsp.2017.10.011

Montavon, G., Binder, A., Lapuschkin, S., Samek, W., and Müller, K.-R. (2019). "Layer-Wise Relevance Propagation: An Overview," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Lecture Notes in Computer Science*. Editors W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller (Cham: Springer International Publishing), 193–209. doi:10.1007/978-3-030-28954-6_10

Oeser, M., and Freitag, S. (2009). Modeling of Materials with Fading Memory Using Neural Networks. *Int. J. Numer. Meth. Engng.* 78, 843–862. doi:10.1002/nme.2518

Pearson, K. (1901). LIII. On Lines and Planes of Closest Fit to Systems of Points in Space. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* 2, 559–572. doi:10.1080/14786440109462720

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations. *J. Comput. Phys.* 378, 686–707. doi:10.1016/j.jcp.2018.10.045

Ramuhalli, P., Udpa, L., and Udpa, S. (2005). Finite-element Neural Networks for Solving Differential Equations. *IEEE Trans. Neural Netw.* 16, 1381–1392. doi:10.1109/TNN.2005.857945

Rudd, K., Di Muro, G., and Ferrari, S. (2014). A Constrained Backpropagation Approach for the Adaptive Solution of Partial Differential Equations. *IEEE Trans. Neural Netw. Learn. Syst.* 25, 571–584. doi:10.1109/TNNLS.2013.2277601

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning Representations by Back-Propagating Errors. *Nature* 323, 533–536. doi:10.1038/323533a0

Sadeghi, M. H., and Lotfan, S. (2017). Identification of Non-linear Parameter of a Cantilever Beam Model with Boundary Condition Non-linearity in the Presence of Noise: An NSI- and ANN-Based Approach. *Acta Mech.* 228, 4451–4469. doi:10.1007/s00707-017-1947-8

W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller (Editors) (2019). "Explainable AI: Interpreting, Explaining and Visualizing Deep Learning," in *Lecture Notes in Computer Science* (Cham: Springer International Publishing). doi:10.1007/978-3-030-28954-6

Shin, H. S., and Pande, G. N. (2000). On Self-Learning Finite Element Codes Based on Monitored Response of Structures. *Comput. Geotech.* 27, 161–178. doi:10.1016/S0266-352X(00)00016-1

Simo, J. C., and Hughes, T. J. R. (1998). *Computational Inelasticity*. in *Interdisciplinary Applied Mathematics*. New York: Springer-Verlag. doi:10.1007/b98904

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* 15, 1929–1958. doi:10.5555/2627435.2670313

Stoffel, M., Bamer, F., and Markert, B. (2018). Artificial Neural Networks and Intelligent Finite Elements in Non-linear Structural Mechanics. *Thin-Walled Struct.* 131, 102–106. doi:10.1016/j.tws.2018.06.035

Teichert, G. H., Natarajan, A. R., Van der Ven, A., and Garikipati, K. (2019). Machine Learning Materials Physics: Integrable Deep Neural Networks Enable Scale Bridging by Learning Free Energy Functions. *Comput. Methods Appl. Mech. Eng.* 353, 201–216. doi:10.1016/j.cma.2019.05.019

Theocaris, P. S., and Panagiotopoulos, P. D. (1995). Plasticity Including the Bauschinger Effect, Studied by a Neural Network Approach. *Acta Mech.* 113, 63–75. doi:10.1007/BF01212634

Theocaris, P. S., Bisbos, C., and Panagiotopoulos, P. D. (1997). On the Parameter Identification Problem for Failure Criteria in Anisotropic Bodies. *Acta Mech* 123, 37–56. doi:10.1007/BF01178399

Truesdell, C., and Noll, W. (2004). "The Non-Linear Field Theories of Mechanics," in *The Non-Linear Field Theories of Mechanics*. Editors C. Truesdell, W. Noll, and S. S. Antman (Berlin, Heidelberg: Springer), 1–579. doi:10.1007/978-3-662-10388-3_1

Wu, L., Nguyen, V. D., Kilingar, N. G., and Noels, L. (2020a). A Recurrent Neural Network-Accelerated Multi-Scale Model for Elasto-Plastic Heterogeneous Materials Subjected to Random Cyclic and Non-Proportional Loading Paths. *Comput. Methods Appl. Mech. Eng.* 369, 113234. doi:10.1016/j.cma.2020.113234

Wu, P., Sun, J., Chang, X., Zhang, W., Arcucci, R., Guo, Y., et al. (2020b). Data-Driven Reduced Order Model with Temporal Convolutional Neural Network. *Comput. Methods Appl. Mech. Eng.* 360, 112766. doi:10.1016/j.cma.2019.112766

Yang, Y., and Perdikaris, P. (2019). Adversarial Uncertainty Quantification in Physics-Informed Neural Networks. *J. Comput. Phys.* 394, 136–152. doi:10.1016/j.jcp.2019.05.027

Yao, H., Gao, Y., and Liu, Y. (2020). FEA-Net: A Physics-Guided Data-Driven Model for Efficient Mechanical Response Prediction. *Comput. Methods Appl. Mech. Eng.* 363, 112892. doi:10.1016/j.cma.2020.112892

Zhao, Y., Otto, S.-K., Brandt, N., Selzer, M., and Nestler, B. (2020). Application of Random Forests in ToF-SIMS Data. *Proced. Comput. Sci.* 176, 410–419. doi:10.1016/j.procs.2020.08.042

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers in Materials

ORIGINAL RESEARCH
published: 03 February 2022
doi: 10.3389/fmats.2022.821817

Machine Learning Assisted Design of Experiments for Solid State Electrolyte Lithium Aluminum Titanium Phosphate

Yinghan Zhao[1†], *Nikolas Schiffmann*[2†], *Arnd Koeppe*[1]*, *Nico Brandt*[1], *Ethel C. Bucharsky*[2], *Karl G. Schell*[2], *Michael Selzer*[1,3] *and Britta Nestler*[1,3]

[1]Institute for Applied Materials-Computational Materials Science, Karlsruhe Institute of Technology, Karlsruhe, Germany, [2]Institute for Applied Materials-Ceramic Materials and Technologies, Karlsruhe Institute of Technology, Karlsruhe, Germany, [3]Institute for Digital Materials, Karlsruhe University of Applied Sciences, Karlsruhe, Germany

Lithium-ion batteries with solid electrolytes offer safety, higher energy density and higher long-term performance, which are promising alternatives to conventional liquid electrolyte batteries. Lithium aluminum titanium phosphate (LATP) is one potential solid electrolyte candidate due to its high Li-ion conductivity. To evaluate its performance, influences of the experimental factors on the materials design need to be investigated systematically. In this work, a materials design strategy based on machine learning (ML) is employed to design experimental conditions for the synthesis of LATP. In the variation of parameters, we focus on the tolerance against the possible deviations in the concentration of the precursors, as well as the influence of sintering temperature and holding time. Specifically, models built with different design selection strategies are compared based on the training data assembled from previous laboratory experiments. The best one is then chosen to design new experiment parameters, followed by measuring the corresponding properties of the newly synthesized samples. A previously unknown sample with ionic conductivity of 1.09×10^{-3} S cm^{-1} is discovered within several iterations. In order to further understand the mechanisms governing the high ionic conductivity of these samples, the resulting phase compositions and crystal structures are studied with X-ray diffraction, while the microstructures of sintered pellets are investigated by scanning electron microscopy. Our studies demonstrate the advantages of applying machine learning in designing experimental conditions by the synthesis of desired materials, which can effectively help researchers to reduce the number of required experiments.

Keywords: all-solid-state lithium batteries, LATP, machine learning, bayesian optimization, design of experiment

OPEN ACCESS

Edited by:
Surya R. Kalidindi, Georgia Institute of Technology, United States

Reviewed by:
Arghya Bhowmik, Technical University of Denmark, Denmark
Byungchan Han, Yonsei University, South Korea

***Correspondence:**
Arnd Koeppe
arnd.koeppe@kit.edu

†These authors have contributed equally to this work

Specialty section:
This article was submitted to Computational Materials Science, a section of the journal Frontiers in Materials

Received: *24 November 2021*
Accepted: *13 January 2022*
Published: *03 February 2022*

Citation:
Zhao Y, Schiffmann N, Koeppe A, Brandt N, Bucharsky EC, Schell KG, Selzer M and Nestler B (2022) Machine Learning Assisted Design of Experiments for Solid State Electrolyte Lithium Aluminum Titanium Phosphate. Front. Mater. 9:821817. doi: 10.3389/fmats.2022.821817

1 INTRODUCTION

Energy is one of the core issues to be solved in the development of human society. Currently, lithium ion batteries (LIBs) are widely used, as they show great promise as an effective energy storage technology for a wide range of applications from mobile devices to electric vehicles. However, commercial LIBs confront hidden risks which are due to the utilization of fluid electrolytes, which may cause a variety of safety and performance problems, such as the potential ignition of the flammable solvent. To address these problems, lithium-ion batteries with solid electrolytes have

potentials to be safer and longer-lasting alternatives with higher energy density compared to conventional liquid electrolyte batteries by allowing the use of high-voltage cathodes, which can decrease flammability, and suppress dendrite formation (Goodenough and Kim, 2010). However, the principal design challenge of solid electrolytes is their restricted ionic conductivity, which is typically many orders of magnitude lower than that of liquids (10^{-2} S cm^{-1}) (Aravindan et al., 2011). The feasibility of these concepts depends on the applied solid-state electrolyte, for which a wide range of materials is being considered (Manthiram et al., 2017). One of the promising materials is the family of lithium containing NASICON (sodium super ionic conductor) materials, such as lithium aluminum titanium phosphate (LATP, $Li_{1+x}Al_xTi_{2-x}(PO_4)_3$), one of the most often investigated materials (Aono et al., 1990). They have received wide attention as they have emerged as particularly promising solid electrolyte candidates due to their high ionic conductivity, low cost, and stability (Rossbach et al., 2018).

The ionic conductivity of LATP is particularly high for the composition $Li_{1.3}Al_{0.3}Ti_{1.7}(PO_4)_3$, and several studies have reported values up to 10^{-3} S cm^{-1} (Narváez-Semanate and Rodrigues, 2010; Pérez-Estébanez et al., 2014; Bucharsky et al., 2015; Ma et al., 2016). $Li_{1.3}Al_{0.3}Ti_{1.7}(PO_4)_3$ ceramics have been successfully synthesized by different routes, such as calcination of stoichiometric mixtures of oxide precursors (Arbi et al., 2002), by glass crystallization (Narváez-Semanate and Rodrigues, 2010), or by sol-gel (Bucharsky et al., 2015). However, these studies are usually limited to laboratory scale, i.e., in small quantities and under experimental environments. In order to make ceramic electrolytes usable and competitive in the next generation of batteries, it is necessary to identify processing routes for the upscale production. The sol-gel route has already been adopted successfully for the mass production of many materials for industry and therefore provides a good basis for LATP synthesis. For the sake of quality maintenance and reproducibility, the crucial processing parameters have to be identified. As the first step towards the upscaling of LATP production via the sol-gel route, influence of possible deviations in the concentration of the precursors was investigated in our previous study (Schiffmann et al., 2021). This applies especially to phosphoric acid which is difficult to specify due to its hygroscopicity. In the case when precursors are not exactly stoichiometric, this can easily lead to the second-phase formation. In particular, for LATP, such second phases have a great influence on the densification and the ionic conductivity. Hupfer et al. (2017) report how the second phases $AlPO_4$ and $LiTiOPO_4$ can have impacts on the properties of LATP. In this work, synthesis of LATP is studied by varying concentration of the reactants, dwell time, and sintering temperature while microstructures, phase compositions, and ionic conductivities of the samples are further analyzed.

A key challenge in developing better materials is the large potential search space for the optimal chemistries and processing conditions. Traditionally, the development of new materials requires a vast number of experiments guided by intuition, trial and error, and is complemented by simulations and other tools to analyze the mechanism or optimize the design (Wang et al., 2015). As a result, this process is time-consuming, challenging and is often accompanied by detours or serendipity. Recently, the use of machine learning methods to accelerate materials development has received a lot of attention and many advances using this kind of technique have been made in the study of solid-state electrolytes, such as screening fast ion conductor candidates in supervised (Sendek et al., 2017) or unsupervised (Zhang et al., 2019) manner, filtering electrolytes in consideration of suppression of dendrite formation in lithium metal anodes (Ahmad et al., 2018), and developing good candidates combining theoretical calculations as well as experimental data sets (Fujimura et al., 2013).

Among the many approaches, accelerating the research of novel materials through automated experiments (Alberi et al., 2018; Häse et al., 2019; Stein and Gregoire, 2019) instructed by artificial intelligence (AI) (Tran and Ulissi, 2018) has recently attracted a lot of interest. In particular, AI sampling algorithms (Coley et al., 2020) hold great promise for resource-constrained tasks such as materials research, since they can reduce the number of experiments required to achieve a desired property (Vasudevan et al., 2019). Among them, the Bayesian decision-theoretic approach naturally lends itself to adaptive sampling and active learning (Cohn et al., 1996). Hence, a series of active learning methods based on Bayesian optimization can be used to find the optimal material composition or to optimize the experimental parameters. This type of method has been successfully applied in different materials system, such as low thermal hysteresis shape memory alloys (Xue et al., 2016), $BaTiO_3$-based ceramics with better dielectric energy storage density (Yuan et al., 2019), fast ion conductors for rechargeable batteries (Jalem et al., 2018; Harada et al., 2020; Homma et al., 2020; Yang et al., 2020), oxygen evolution reaction catalyst (Rohr et al., 2020), and organic thin films (MacLeod et al., 2020).

In this work, we use a Gaussian process (GP) based Bayesian optimization (Ki Williams, 2006) to optimize the synthesis of a popular electrolyte material for solid-state lithium-ion batteries, LATP ($Li_{1.3}Al_{0.3}Ti_{1.7}(PO_4)_3$). Via the sol-gel route, it is possible to prepare the material at laboratory scale with high purity and with a maximum Li-ion conductivity in the order of 1×10^{-3} S cm^{-1} at room temperature. However, for a potential commercial usage, battery-cell upscaling of the synthesis is required. Based on our previous study (Schiffmann et al., 2021), we further explore the effects of deviations in the concentration of the precursors H_3PO_4, sintering temperature and holding time on the conductivity of the synthesized electrolytes. We use machine learning methods to guide us to reduce the number of required experiments as much as possible to produce LATP with higher ionic conductivity. We train the initial model using the data points from previous experiments (sampled from an equidistant grid and this is noted as grid search) and predict the next optimal experimental configurations. The results show that newly synthesized samples guided by the model can achieve a good performance with the maximum ionic conductivity of 1.09×10^{-3} S cm^{-1}, in the same order of magnitude of the maximum Li-ion conductivity which LATP

can achieve. This method can help to quickly narrow down the search space and assist the experimentalists in designing and analyzing experiments.

2 EXPERIMENTS AND MACHINE LEARNING METHODS

2.1 Experiments

In our experiments, LATP powders are prepared following a modified sol-gel synthesis route described by (Bucharsky et al., 2015). Appropriate amounts of lithium acetate $Li(C_2H_3O_2)\cdot 2H_2O$ (purity $\geq 99\%$, Alfa Aesar GmbH & Co KG, Germany), aluminum nitrate $Al(NO_3)_3\cdot 9H_2O$ (purity $\geq 98.5\%$, Merck KGaA, Germany), titanium-isopropoxide $Ti[OCH(CH_3)_2]_4$ (purity $\geq 98\%$, Merck KGaA, Germany) are used as precursors. Lithium acetate and aluminum nitrate are dissolved in distilled water under constant stirring. Titanium-Isopropoxide is then added dropwise to the solution. By adding the phosphoric acid slowly through a drip funnel, a white gel forms, which is then dried at room temperature for 24 h. The subsequent heat treatment is performed in two steps: first, samples are heat treated at 400°C for 6 h to achieve the precursor formation and to eliminate reaction gases; second, samples are then processed at 900°C for 8 h to complete the reaction to crystalline LATP. One sol-gel batch is prepared with all precursors in stoichiometric quantities (marked as 0.0 wt%). To test whether the sol-gel route is tolerant against possible deviations in the concentration of the precursors, we also explore different sol-gel batches with either an excess up to +7.5 wt%, or a deficiency up to −15.0 wt% of phosphoric acid compared to the stoichiometric composition.

To ensure a high sinterability, the obtained powders are further processed in a planetary ball mill. The pellets are formed by uniaxial pressing and then further densified by cold isostatic pressing at 400 MPa. All pressed samples have a green density of approximately 62% relative density. Samples are sintered at temperatures ranging from 850 to 1,050°C and isothermal sintering time between 30 and 540 min. After sintering, samples are cooled down to room temperature in furnace and their corresponding densities are determined by Archimedes' method. For the ionic conductivity measurements, impedance analysis is performed at room temperature over the frequency range from 0.1 Hz to 1 MHz with an AC amplitude of 50 mV in the frequency response analyzer (AMTEK GmbH, VersaSTAT 4, Pennsylvania, United States). For further details of the experimental part please refer to our previous work (Schiffmann et al., 2021).

To apply machine learning methods in guiding the experimental study, 80 initial data points from the aforementioned phosphoric acid deviation study are used to train the Gaussian process regression based Bayesian optimization (GPR-BO) model. These data points are sampled from an equidistant grid from −22.5 wt% to +7.5 wt% deviation of phosphoric acid compared to the stoichiometric composition. Sintering parameters with temperatures ranging from 800°C up to 1,100°C in steps of 100°C and isothermal durations of 10, 30, 60 and 480 min are applied. To further investigate the effect of these synthesis and sintering conditions on the properties of LATP, the machine learning model is used to predict promising candidate to investigate. Considering the long time needed to synthesize samples with different acid concentrations, we expand the experimental conditions available to the model in two steps: for the first two iterations (1–2), we only allow the model to make a choice among the available samples; for the last two iterations (3–4), we expand the selection range of acid concentrations. Such kind of condition setting is derived from the results of our previous grid search study and offers an efficient compromise that would address the otherwise excessively large search space. In total, we have synthesized 22 new samples in 4 iterations following the model's predictions. Detailed settings for the experiments are listed in **Table 1**.

2.2 Machine Learning Methods

2.2.1 Design Loop

Our method of applying Bayesian optimization (BO) in guiding experiments is schematically illustrated in **Figure 1**. The whole process is mapped as a workflow containing an iterative loop with feedback steps and it is also collectively referred to as "adaptive design." First, the model fits the initial data points. Then the next candidate configuration is predicted and the accompanying experiment and measurements are performed. Finally, the resulting new data point is fed back into the data set for the next iteration. Key ingredients of this process for our problem are as follows: 1) collecting the training data set of the solid state electrolyte LATP, where samples are described by features, here: experimental conditions and their measured properties of interest (e.g., ionic conductivity); 2) training an inference model (Gaussian process regressor) to learn to map the input-output relationship with associated uncertainties. Then, the trained model predicts the outputs (i.e., ionic conductivities) along with their corresponding uncertainties for the whole search space; 3) choosing the combination of experimental parameters, which is expected to produce the material with better characteristics (e.g., higher ionic conductivity) by balancing the trade-off between exploitation and exploration, that is, taking both prediction (of the best known so far) and uncertainty into consideration; 4) performing experiments and measuring the corresponding properties; 5) adding the new sample to the training data set, which allows the subsequent iterative improvement of the inference model. This loop continues until performance (e.g., we have synthesized a sample with a satisfying performance) or a break condition, such as a maximum number of iterations, is met. In this work, the research data infrastructure Kadi4Mat (Brandt et al., 2021) is used to share and manage data for continuously updating the machine learning model. Besides, the whole workflow will also be integrated into this platform and serves as a demonstration for data-driven and machine learning based optimization of solid state electrolyte.

2.2.2 Bayesian Optimization

Bayesian optimization (BO) is a class of machine-learning-based optimization methods focusing on solving the problem arg

TABLE 1 | Range of experimental parameters for selection.

Experimental parameters	Iteration 1–2	Iteration 3–4
	Value range	
Rel. amount of acid wt%	−22.5, −15.0, −7.5, 0.0, 7.5	−22.5, −18.75, −15.0, −11.25, −7.5, −3.75, 0.0, 3.75, 7.5
Temperature °C	800–1,100 (step size 50)	800–1,100 (step size 25)
Time min	10, 20, 30, 40, 50, 60, 90, 120, 240, 360, 480, 540	10, 20, 30, 40, 50, 60, 90, 120, 150, 180, 210, 240, 270, 300, 330, 360, 390, 420, 450, 480, 510, 540

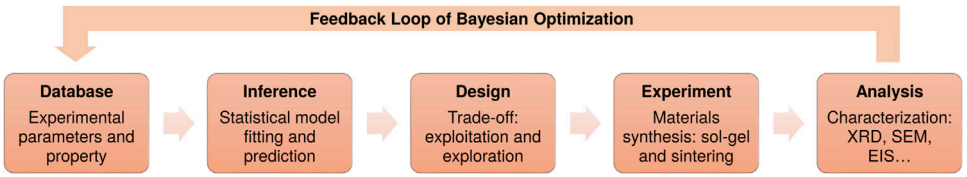


FIGURE 1 | An overview of the Bayesian optimization workflow. It consists of five main components and forms an iterative loop: (i) collecting some initial data points (input-output pairs) from experiments and aggregating them into a database which will be used for training the machine learning model; (ii) fitting an inference model (e.g., a Gaussian process regression model); based on the existing data set; (iii) predicting the property in the search space along with uncertainty, taking both prediction (of the best known so far) and uncertainty (i.e., exploitation vs. exploration) into consideration for selecting the next optimal experimental configuration which has the potential to yield a better property; (iv) performing the experiment to synthesize new sample and (v) validating the new sample's performance using different characterization techniques. The resulting sample is fed back into the initial training data set for the next iteration.

$\max_{x\in\chi} f(x)$ within a domain $\chi \subset \mathbb{R}^d$ as the bounding box. Its ability to optimize expensive black-box derivative-free functions makes BO extremely versatile (Shahriari et al., 2015). Recently, it has become extremely popular for tuning hyper-parameters in machine learning algorithms, especially deep neural networks (Snoek et al., 2012). A typical Bayesian optimization algorithm involves two primary components: a method for statistical inference, typically Gaussian process (GP) (Rasmussen, 2003; Williams and Rasmussen, 2006); and an acquisition function that decides where to sample. For the latter, there are many options such as PI [probability of improvement (Kushner, 1964)], EI [expected improvement (Močkus, 1975; Jones et al., 1998)] or UCB [upper confidence boundary (Auer et al., 2002)]. GP (Gaussian process) is a widely used surrogate for modeling objective functions in Bayesian optimization. The function f is typically assumed to be a GP which is determined by a mean function μ and a covariance kernel K, $f \sim \mathcal{GP}(\mu, K)$. Given the observed data set D, the question would be where the next point to observe the function is. The meta-approach in Bayesian optimization is to design an acquisition function $a(x)$. The acquisition function is usually an inexpensive function, which defines a balance between exploring new areas in the objective space and exploiting areas that are already known to have favorable values (Frazier, 2018). This strategy is important for helping to find the global optimum efficiently instead of being trapped in a local optimum. In short, by adopting this method, the original optimization problem is replaced with another optimization problem based on a much-cheaper function $a(x)$.

PI is one of the earliest acquisition functions designed for Bayesian optimization which suggests maximizing the probability of improvement over the current best observed value $f(x^+)$, where $x^+ = \arg\max_{x\in D_{1:t}} f(x)$ with the observed data set $D_{1:t}$, so that

$$x_{t+1} = \arg\max_{x\in\chi} a_{PI}(x) = \arg\max_{x\in\chi} P\big(f(x) \geq f(x^+)\big) = \arg\max_{x\in\chi} \Phi\left(\frac{\mu_t(x; D_{1:t}) - f(x^+)}{\sigma_t(x; D_{1:t})}\right)$$

where $\Phi(\cdot)$ is the normal cumulative distribution function, μ_t and σ_t are the posterior mean and posterior standard deviation at iteration t.

Alternatively, maximizing the expected improvement (EI) over the current best value can also be chosen, which accounts for the size of the improvement (while PI does not). It can be computed analytically as:

$$x_{t+1} = \arg\max_{x\in\chi} a_{EI}(x) = \arg\max_{x\in\chi} \big[(\mu_t(x; D_{1:t}) - f(x^+))\Phi(Z) + \sigma_t(x; D_{1:t})\phi(Z)\big]$$

$$Z = \frac{\mu_t(x; D_{1:t}) - f(x^+)}{\sigma_t(x; D_{1:t})}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are the cumulative distribution function (CDF) and probability density function (PDF) of the standard normal distribution, respectively.

The acquisition function of UCB takes the form:

$$x_{t+1} = \arg\max_{x\in\chi} a_{UCB}(x) = \arg\max_{x\in\chi} \big[\mu_t(x; D_{1:t}) + \kappa\sigma_t(x; D_{1:t})\big].$$

This function can be intuitively interpreted as a weighted sum of prediction of $f(x)$ and its uncertainty. κ is a tunable hyper-parameter (usually set to be 1.96 as it represents the 95% confidence interval and generally has good performance) which controls how much of the variance in the predicted

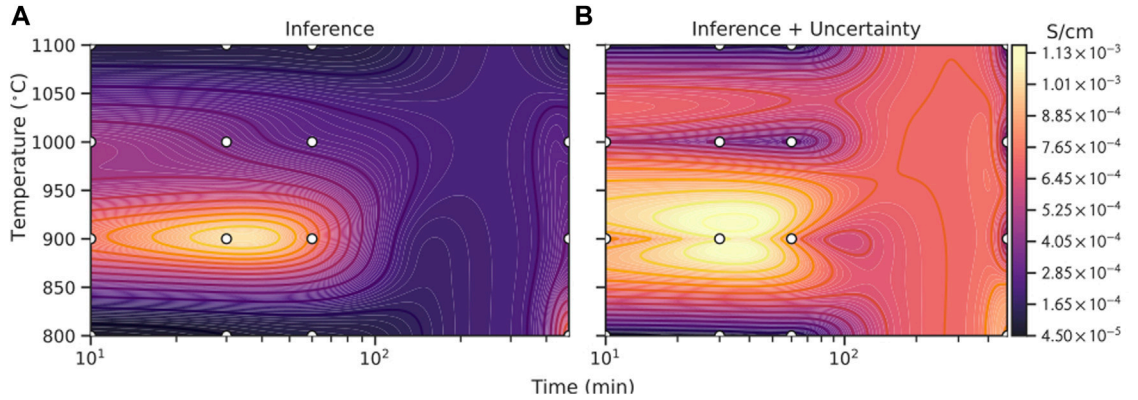


FIGURE 2 | Illustration of the prediction **(A)** based on known data points (white points) and prediction with uncertainty **(B)** from the Gaussian process regression with UCB model in the search space. It shows the ionic conductivity (S cm^{-1}) of LATP with stoichiometric H_3PO_4 (0.0 wt%), where higher values are indicated with brighter color. The x-axis represents the holding time, while the y-axis represents the sintering temperature. It can be seen that the position of the possible optimum (the lightest area) has shifted when considering the uncertainty.

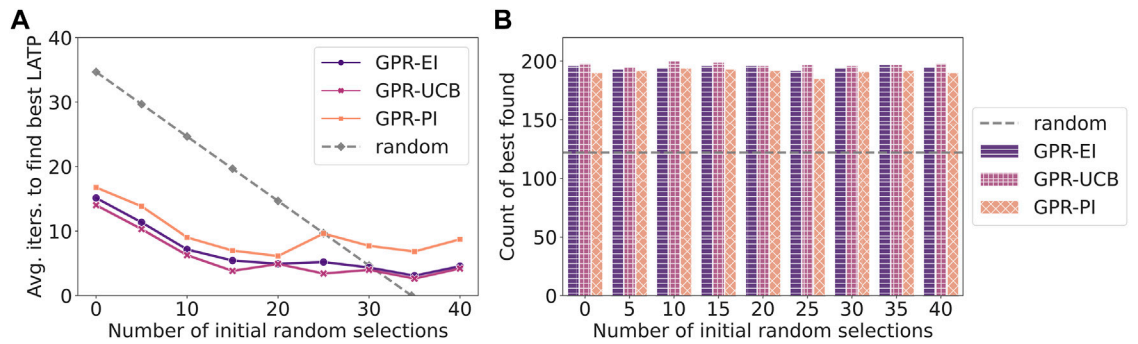


FIGURE 3 | Comparison of GPR based Bayesian optimization with different acquisition functions, EI (dark purple), UCB (light purple), PI (orange) and random selection (grey dashed): **(A)** number of counts to find the global maximum in the repeated 200 virtual experiments, where random selection (grey dashed line) can find 126 times (63%); **(B)** number of additional tries of different strategies needed after *n* initial random selections (x-axis) to find the global maximum of ionic conductivity (1.09×10^{-3} S cm^{-1}) in the training data set (grid search). On average random selection takes about 34.7 tries to arrive at the maximum. It is noticeable that only those cases where the global minimum is found are counted to calculate the average.

values should be taken into account. Higher value favours the exploration over exploitation and vice versa.

To predict optimal experimental parameters in an effective way, the search strategy also needs to follow the aforementioned principle, i.e., combining exploration and exploitation. The model should not only focus on the local region where the known maximum value is located, but also explore the whole search space wisely. Here, we adopt one of the widely used Bayesian optimization methods named GP-UCB [Gaussian Process Upper Confidence Boundary (Srinivas et al., 2009)], which is an intuitive algorithm inspired by the multi-armed bandit problem. We show how the GP-UCB method can be used for materials discovery, as this allows us to choose potential candidates which aims at maximizing the target property of the material. A schematic diagram of the working principle is illustrated in **Figure 2**. This figure shows the fitting of model with the H_3PO_4 acid at stoichiometry (namely, .0 wt%), where the x-axis stands for dwell time and the y-axis represents the sintering temperature. Larger values of the ionic conductivity are marked with brighter color. The left diagram shows what the prediction (inference) based on known data points from the grid search study looks like in the search space and the right diagram illustrates the prediction with uncertainty. It can be clearly seen that when taking the uncertainty into account, the search surface becomes significantly rugged and the location of the possible optimal value has shifted. The model will select regions worth exploring according to both the prediction and the degree of uncertainty.

In addition to UCB, we also compare the above mentioned two other common acquisition functions: PI and EI. The detailed results will be discussed in the model selection section shown in **Figure 3**. In performing the experiments, we greedily choose the next experimental condition for the synthesis of LATP with the best predicted value from the model. In order to make full use of our experiment facilities, it is better that the model can suggest several samples simultaneously in the pre-defined search space in

each iteration of BO. However, one of the limitations of Bayesian optimization is that the acquisition is myopic and permits only a single sample per iteration (Brochu et al., 2010). To alleviate this problem, the so-called Kriging believer approach (Cressie, 1990) is used to suggest 5–6 samples at the same time during each iteration of BO in our study: to suggest more than 1 sample in each iteration, this approach (temporarily) adds each predicted sample to the training data set for updating the model, and then predicts another sample subsequently.

3 RESULTS AND DISCUSSIONS

3.1 Data Set

The initial 80 training data points are listed in **Supplementary Table S1**.

3.2 Model Selection

As there is no clear indication of which optimization strategy to use (according to the "no-free-lunch theorem" (Wolpert and Macready, 1997), there is no universal optimizer for all problems), we compare the optimization efficiencies of BOs with random search, EI, UCB and PI strategies with the 80 initial points from the grid search study. The strategy of achieving the maximum ionic conductivity (1.09×10^{-3} S cm^{-1}) with the least number of average iterations is considered to be optimal. During the experiment of comparing these strategies, random noise is added to the observations and the sample is allowed to be picked more than once (namely, with replacement). In detail, we randomly select a given number of samples from the training data with replacement as initial data points, then train the model using a given acquisition function and count the total number of extra tries (after initial random picks) needed to find the best sample (that is, the one with the largest ionic conductivity) in the grid search study. The model is only allowed to make up to m attempts (m = 80 − number of initial data points) to find the maximum value. This process is called "virtual experiment" and is repeated 200 times with different sets of randomly selected samples. In the overall count, the initial random picks are excluded. Detailed results of comparing different strategies of acquisition functions are shown in **Figure 3**.

Figure 3A illustrates how many times different strategies can find the global maximum (1.09×10^{-3} S cm^{-1}) within the 80 tries in 200 virtual experiments. The grey dashed line represents how many counts the random selection can find the global maximum. Among the 200 virtual experiments, the random selection can find 126 times (63%) and it acts as the base line for evaluating the performance of other models. From the figure it can be seen that GPR model with an EI (dark purple) or UCB (light purple) acquisition function can find the maximum in most virtual experiments ($\geq$95%), where UCB performs slightly better than EI in some cases. In contrast, PI (orange) performs slightly worse than the other two, but still much better than the random selection. We speculate that PI sometimes gets stuck in the local optimum, making it difficult for the model to reach the global maximum. As a result, it fails to find the global maximum in some cases.

Figure 3B illustrates the average number of extra tries (after a given number of initial data points) required for the models with different acquisition functions to find the global maximum. The random selection takes on average 34.7 tries to find the global maximum, which is marked as grey dashed line in the figure. It can be seen that all the three models perform much better than the random selection. Performance of EI and UCB is similar, with UCB being slightly better and it takes the fewest extra tries to achieve the best result. More specifically, the number of extra tries needed to find the global maximum for EI and UCB decreases quickly with more initial data points, which is reasonable as the models' ability to fit and predict is gradually enhanced. After more than 15 initial data points, the gain of introducing more initial data points gradually decreases and the required extra tries finally stabilizes at about 5. This phenomenon is very beneficial for experiments because the model can achieve good performance even with only a small number of initial data points, significantly reducing the number of attempts required. In contrast, the performance of PI is worse than the other two strategies as it requires more steps to obtain the global optimum. Therefore, we choose the model with UCB acquisition function and use it to predict the optimal experimental conditions, as it is more robust and takes fewer steps to reach the global optimum.

3.3 Result of Newly Synthesized Lithium Aluminum Titanium Phosphate Samples

As the search space becomes larger, it is difficult to manually determine the experimental conditions to obtain samples with better performance. Hence, a machine learning model is employed to help to explore the unknown experimental space quickly in order to reduce the number of required trials. The experimental conditions predicted by the model and their measured properties of the resulting new samples, namely, relative density after sintering and ionic conductivity, are listed in **Table 2**. All samples have a green density of approximately 62% so this property is not listed. Starting with 80 data points, we have performed 4 iterations (each predicting 5–6 samples) to optimize the experimental conditions to obtain LATP with higher ionic conductivity. At the end of each iteration, the newly synthesized samples are fed back to the model, which is then retrained. After the update, new experimental conditions are predicted for the next iteration. This process forms a loop, which is repeated until the target number of iterations has been achieved. In total 22 new samples are synthesized.

It can be seen from **Table 2** that the model quickly discovers a new sample (sample No. 3) with second highest ionic conductivity (1.06×10^{-3} S cm^{-1}) in the first iteration. The ionic conductivity of this sample is very close to the highest one of all samples (1.09×10^{-3} S cm^{-1}), which shows a very good performance of our model. The comparison of experimental conditions shows that even though the sintering time of the new sample (540 min) is 60 min longer than that of the known maximum sample (480 min, with a deficiency of –7.25% H_3PO_4 at 900°C), it can still maintain good performance. This indicates

TABLE 2 | Recommended samples using BO and measured properties.

Iteration	No	Experimental parameters			Measured properties	
		rel. H_3PO_4 wt%	Temperature°C	Time min	rel. Density %	Ionic Conductivity S cm^{-1}
1	1	−7.5	1,000	40	87.99	5.67×10^{-4}
	2	−7.5	1,000	360	92.78	7.21×10^{-4}
	3	−7.5	1,000	540	95.81	$\mathbf{1.06 \times 10^{-3}}$
	4	0.0	900	20	96.29	7.23×10^{-4}
	5	−7.5	1,000	240	92.12	6.89×10^{-4}
2	6	−15.0	950	480	78.24	1.27×10^{-4}
	7	−22.5	950	540	80.00	4.93×10^{-5}
	8	7.5	950	540	75.22	6.67×10^{-5}
	9	0.0	850	30	92.56	4.91×10^{-4}
	10	0.0	1,050	30	81.27	6.11×10^{-5}
3	11	.0	900	40	95.80	4.07×10^{-4}
	12	−11.25	1,000	480	97.47	9.06×10^{-4}
	13	−11.25	1,000	510	96.11	7.87×10^{-4}
	14	−11.25	1,000	450	97.52	$\mathbf{1.09 \times 10^{-3}}$
	15	−7.5	1,000	450	95.00	4.02×10^{-4}
	16	−7.5	1,000	510	97.36	4.37×10^{-4}
4	17	−15.0	1,000	450	97.21	7.81×10^{-4}
	18	−11.25	1,000	420	95.25	8.36×10^{-4}
	19	−15.0	1,000	510	96.81	5.36×10^{-4}
	20	−15.0	1,025	450	98.89	5.87×10^{-4}
	21	−11.25	1,000	540	91.31	6.38×10^{-4}
	22	−11.25	1,000	390	91.60	7.10×10^{-4}

that LATP samples synthesized under this condition are stable against long holding time. On the other hand, comparing the densities of samples No. 2 (360 min) and No. 5 (240 min) it can be concluded that shorter holding time is not enough for the samples to get fully sintered as the densities of these samples are less than those with longer holding time. Therefore, the ionic conductivities of these samples are significantly lower than those of the fully sintered ones.

As the Bayesian optimization process is a trade-off process between exploration and exploitation, it is important to explore unknown areas efficiently, which can help to find the global optimum in a scientific way. This can be well reflected in our experiments. It can be seen that in the first iteration, the model explores experimental conditions with holding time at 360 min (sample No. 2) and 240 min (sample No. 5), as the original experimental conditions of grid search have a large gap (uncertainty) of holding time between 60 and 480 min. Similarly, this can also explain why the model in the second iteration explores the temperature intervals that have never been explored before, as these areas are subject to relatively large degrees of uncertainties. Though samples obtained in the second iteration show poor ionic conductivity (on average 1.59×10^{-4} S cm^{-1}), it does not render this iteration a failure. This attempt is reasonable and can help the model to quickly explore this unknown region and excludes the possibility of an optimal value appearing in that region, which effectively reduces the number of tries it needs compared to the exhaustive method. Notably, starting from the third iteration, we expand the range of experimental conditions that can be chosen, more centrally, we expand the extent to which the amount of precursor H_3PO_4 can be adjusted, as its preparation can take a long time. It can be clearly seen that the model also undergoes a competitive process between exploration and exploitation in the third and fourth iterations. It first makes predictions (samples No. 12–14) with experimental conditions with a deficiency of −11.25% H_3PO_4, which has never been explored before (exploration). At this point, the model finds a new sample (sample No. 14) with a value of ionic conductivity which is as large as the previous maximum (1.09×10^{-3} S cm^{-1}). Then the model begins to make the most of this information and makes several attempts (samples No. 18, 21, 22) around this point (exploitation). It can be seen that properties (ionic conductivity) of later samples are inferior to that of sample No. 14, indicating that the model has a good predictive performance. As a result, it can find the optimal value efficiently and quickly, reducing the number of required experiments.

The values of ionic conductivity (black line) during the four iterations and the values of the maximum (grey line) are plotted in **Figure 4**. It can be noted that the overall result shows a step-wise upward trend, illustrating the improvement of new samples during iterations. During the iteration, the model goes through a process of exploration and exploitation, which can be reflected in the fluctuating experimental results of ionic conductivity. The model first selects a sample which yields moderate performance (5.67×10^{-4} S cm^{-1}). Afterwards, the performance of samples increases with iterations and it quickly finds the second highest maximum (sample No. 3). Starting from the sample No. 11, the search space is extended (marked as the vertical grey dashed line in the middle) and the model quickly finds another global maximum (sample No. 14, marked as red star). Another

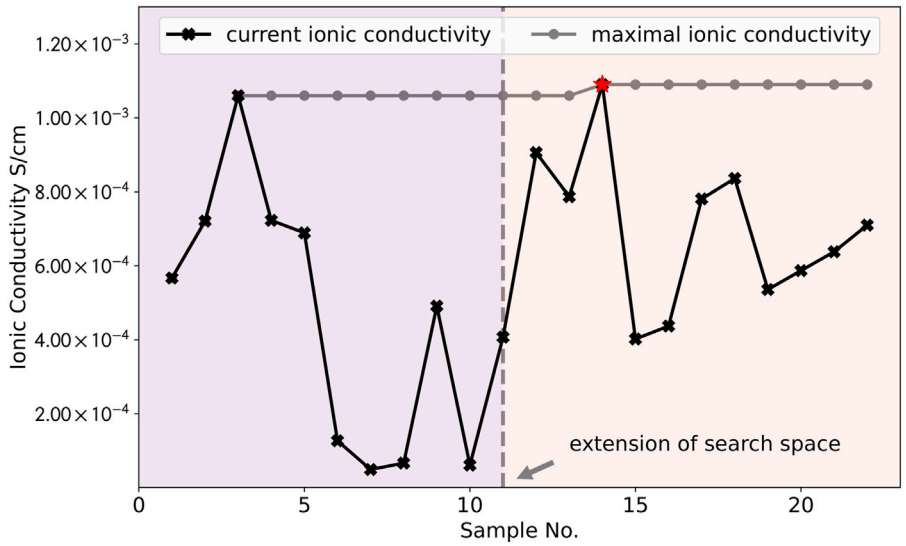


FIGURE 4 | The values of ionic conductivity for each sample (black line) during the four iterations and the values of the maximum (grey line). The results of experiments show a fluctuating trend, reflecting the exploration vs. exploitation in the optimization process. The overall result exhibits a step-wise upward trend and it can be seen that the model has found the largest values of ionic conductivity (marked as red star) within several iterations.

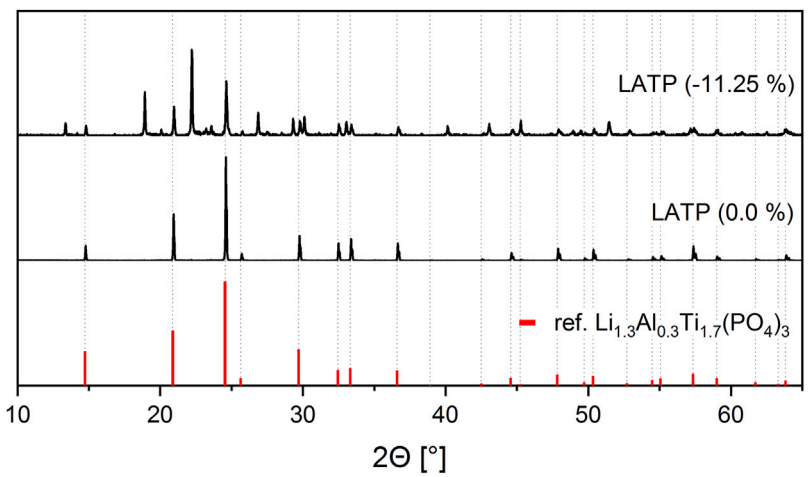


FIGURE 5 | X-ray diffraction patterns for the LATP (−11.25 wt%) sample sintered at 1,000°C for 450 min and one LATP (0.0 wt%) sample sintered at 900°C for 30 min. A standard X-ray diffraction pattern of $Li_{1.3}Al_{0.3}Ti_{1.7}(PO_4)_3$ from the database is shown in red color for reference.

schematic diagram (similar to **Figure 2**) is shown in the supporting material (**Supplementary Figure S1**) to illustrate the above working principle and to better visualize the evolution of changes in the predictions of the model during different iterations. Overall, results of predictions prove that our model has a good ability to help us to find another sample with maximum ionic conductivity under different experimental condition where it has never been explored before. By using the Bayesian optimization model, it can help the experimentalist to quickly narrow the search space and hence can reduce the number of required experiments effectively.

To further explore why these samples (e.g., sample No. 14) have better performances than others, characterization measurement are performed to investigate the mechanisms governing the high ionic conductivity. Details are given in the next section.

3.4 Characterization of Lithium Aluminum Titanium Phosphate Samples

A standard stoichiometric LATP sample reaches the highest ionic conductivity at a sintering temperature of 900°C and a

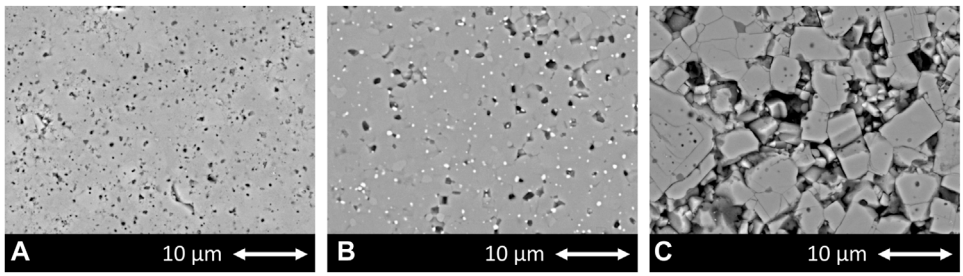


FIGURE 6 | Microstructure of **(A)** LATP (0.0 wt%) sample sintered at 900°C for 30 min; **(B)** LATP (−11.25 wt%) sample sintered at 1,000°C for 450 min and **(C)** LATP (0.0 wt%) sample sintered at 1,000°C for 450 min.

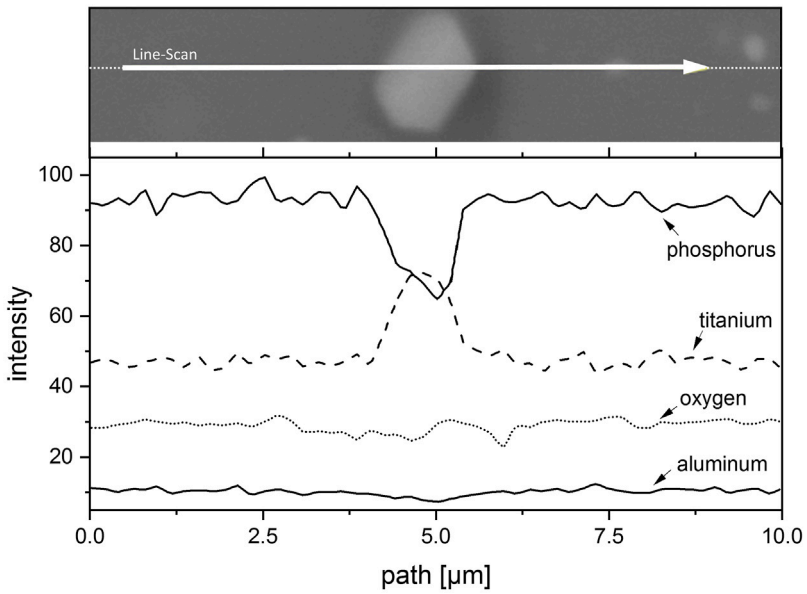


FIGURE 7 | EDX-line-scan of a bright second phase particle in the structure of the sintered LATP (−11.25 wt%) sample. A significant increase can be clearly seen in titanium intensity at the position of the bright region, which indicates that the second phase is highly possible to be TiO_2.

holding time of 30 min. With the aid of machine learning algorithms, the best properties are achieved for the sample No. 14, which is sintered from a LATP batch synthesized with a deficit of −11.25 wt% in phosphoric acid and sintered at 1,000°C for 450 min. For a better understanding why this sample has also reached ionic conductivity performance in the order of 1×10^{-3} S cm^{-1}, even though synthesis and sintering conditions deviated from the standard procedure, the microstructure is analyzed. **Figure 5** shows a comparison of the X-ray diffraction patterns of the standard stoichiometric LATP (0.0 wt%) sample and another LATP (−11.25 wt%) sample. While the stoichiometric sample is still phase pure in the expected NZP-structure after sintering, clear foreign peaks can be seen in the LATP (−11.25 wt%) sample. Among the foreign phases that have formed in addition to the NZP-structure, TiO_2 could be identified as a second phase.

In addition, the different microstructure developments are shown in **Figure 6**. **Figure 6A** shows the LATP (0.0 wt%) sample sintered at 900°C with a holding time of 30 min. There, the ionic conductivity in the order of 1×10^{-3} S cm^{-1} is achieved by a homogeneous and dense microstructure with small and uniform grains as well as very small and finely distributed pores. Despite the significantly different sintering parameters, **Figure 6B** shows a similar homogeneous microstructure with only slightly larger grains and pores for the LATP (−11.25 wt%) sample. The homogeneous grain size and dense structure is not typical for LATP sintered at this high temperature as the comparison in **Figure 6C** with LATP (0.0 wt%) sintered with these parameters shows. There, abnormal grain growth and the related microcracks in these large grains, due to a high thermal expansion anisotropy between a and c lattice parameters, shatter the microstructure and cause a drastic decrease in ionic conductivity (Jackman and Cutler, 2012; Hupfer et al., 2016; Waetzig et al., 2016). Grain

growth seems to be suppressed for the LATP (−11.25 wt%) sample by second-phase particles, which are visible as homogeneously distributed bright dots throughout the structure at the triple points of the grain boundaries. The evaluation of the corresponding XRD-pattern leads to the assumption that the second phase is TiO_2 and this is reinforced by the results of the EDX (Energy dispersive X-Ray) analysis of one of these particles shown in **Figure 7**. The line-scan shows a clear increase in the titanium intensity in the EDX measurement at the position of the bright particle. The interaction between second-phase particles and migrating grain boundaries is known as Zener-type mechanism in ceramic materials and can reduce grain growth. This is likely to be the reason for the moderate grain growth of the LATP (−11.25 wt%) sample at these high sintering temperatures (Rahaman, 2007). It allows for a densification of the microstructure that in turn results in a ionic conductivity in the order of 1×10^{-3} S cm^{-1}. The above analysis serves to explain the possible mechanism why samples like No. 14 have better performances than others, which agrees well with the conclusions from our previous study (Schiffmann et al., 2021) where a deficiency of phosphoric acid in the synthesis can lead to the formation of $LiTiOPO_4$ and TiO_2. It seems that the second phases are the reason for the prevention of abnormal grain growth for sintering temperatures up to 1,000°C. Because the small grains are less susceptible to microcracking, a dense structure with high ionic conductivity is achievable even sintered at these high sintering temperatures. Experimental parameters for other samples, such as the sample No. 3 (1.06×10^{-3} S cm^{-1}) synthesized with parameters (−7.5 wt%, 1,000°C, 540 min), are comparable to one of the samples (−7.5 wt%, 1,000°C, 480 min with ionic conductivity of 1.09×10^{-3} S cm^{-1}) in the paper mentioned above and one can therefore assume that a comparable microstructure is the reason for the high ionic conductivity of this sample as well.

4 CONCLUSION

Our work shows that a data-driven materials design strategy based on Bayesian optimization using Gaussian process regression can be employed in effectively designing experimental conditions for synthesizing LATP, which is one of the potential solid electrolyte candidates for batteries. The whole design strategy is divided into several sections: first, to find the most suitable model for our study, virtual experiments are performed to compare models built with several combinations of design strategies using the training data assembled from previous laboratory studies. In our study, we find that the model with UCB (upper confidence boundary) strategy can achieve the best performance. Second, the best model is then selected to design new experimental parameters in order to synthesize new sample with the largest possible value of ionic conductivity. Third, the corresponding properties of the newly sintered samples are measured and these results are fed back to update the model for the next iteration of the design process. Our results show that within several iterations, newly synthesized samples guided by the model can achieve a good performance with maximum value of 1.09×10^{-3} S cm^{-1}, which is in the same order of magnitude of the maximum Li-ion conductivity that LATP can achieve. In addition, the range of search space can be dynamically adjusted during the experiment, making this method flexible according to researchers' needs. Besides, it can help the researcher to quickly explore the boundary of the range of experimental parameters which may yield samples with good performance, hence it can be assisted in designing experiments in an effective and reasonable way to reduce the number of required experiments. It is worth noting here that the main focus of this work is on single-objective optimization, that is, only the ionic conductivity is paid attention to. Admittedly, this is a simplification of the optimization problem, since other properties (e.g., sintered density) can also affect our interested property. As a result, taking them into consideration and regarding it as a multi-objective optimization problem (previous studies concerning similar question can be found in (Harada et al., 2020; Yang et al., 2020)) may further improve the performance of the model and may hence result in better samples. This question deserves to be explored in details and is left for the future research.

In order to further understand the reasons governing the high ionic conductivity of these samples, the resulting crystal structures and phase compositions are studied with X-ray diffraction and energy dispersive X-Ray analysis, while the microstructures of sintered pellets are investigated by scanning electron microscopy. The formation of secondary phases such as TiO_2, is demonstrated to be substantially influenced by the initial concentration of the precursors, which can influence ionic conductivity, densification behavior, and microstructure evolution.

In summary, our studies demonstrate the advantages of adopting machine learning for an accelerated design of experimental parameters by the synthesis of materials with targeted properties, which can help experimentalists to explore the search space effectively and narrow the parameter range quickly. This is a general method that can be mapped to other research systems and the whole workflow will be kept sustainable within the Kadi4Mat framework, which can reduce the number of required experiments and accelerate the process of developing materials.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding author.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

FUNDING

This work is supported by the German Federal Ministry of Education and Research (BMBF) within the "FESTBATT"

consortium (Grant No. 03XP0174E). Contribution to the integration of the methods into the Kadi4Mat research data infrastructure is provided by the Helmholtz Association of German Research Centres through the program "MTET" (Grant No. 38.02.01).

REFERENCES

Ahmad, Z., Xie, T., Maheshwari, C., Grossman, J. C., and Viswanathan, V. (2018). Machine Learning Enabled Computational Screening of Inorganic Solid Electrolytes for Suppression of Dendrite Formation in Lithium Metal Anodes. *ACS Cent. Sci.* 4, 996–1006. doi:10.1021/acscentsci.8b00229

Alberi, K., Nardelli, M. B., Zakutayev, A., Mitas, L., Curtarolo, S., Jain, A., et al. (2018). The 2019 Materials by Design Roadmap. *J. Phys. D Appl. Phys.* 52, 013001. doi:10.1088/1361-6463/aad926

Aono, H., Sugimoto, E., Sadaoka, Y., Imanaka, N., and Adachi, G.-y. (1990). Ionic Conductivity and Sinterability of Lithium Titanium Phosphate System. *Solid State Ionics* 40-41, 38–42. doi:10.1016/0167-2738(90)90282-v

Aravindan, V., Gnanaraj, J., Madhavi, S., and Liu, H.-K. (2011). Lithium-ion Conducting Electrolyte Salts for Lithium Batteries. *Chem. Eur. J.* 17, 14326–14346. doi:10.1002/chem.201101486

Arbi, K., Mandal, S., Rojo, J. M., and Sanz, J. (2002). Dependence of Ionic Conductivity on Composition of Fast Ionic Conductors Li1+xTi2-xAlx(PO4)3, $0 \leq X \leq 0.7$. A Parallel NMR and Electric Impedance Study. *Chem. Mater.* 14, 1091–1097. doi:10.1021/cm010528i

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learn.* 47, 235–256. doi:10.1023/a:1013689704352

Brandt, N., Griem, L., Herrmann, C., Schoof, E., Tosato, G., Zhao, Y., et al. (2021). Kadi4mat: A Research Data Infrastructure for Materials Science. *Data Sci. J.* 20. doi:10.5334/dsj-2021-008

Brochu, E., Cora, V. M., and De Freitas, N. (2010). *A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning*. arXiv. arXiv preprint arXiv:1012.2599.

Bucharsky, E. C., Schell, K. G., Hintennach, A., and Hoffmann, M. J. (2015). Preparation and Characterization of Sol-Gel Derived High Lithium Ion Conductive NZP-type Ceramics Li1+x AlxTi2−x(PO4)3. *Solid State Ionics* 274, 77–82. doi:10.1016/j.ssi.2015.03.009

Cohn, D. A., Ghahramani, Z., and Jordan, M. I. (1996). Active Learning with Statistical Models. *JAIR* 4, 129–145. doi:10.1613/jair.295

Coley, C. W., Eyke, N. S., and Jensen, K. F. (2020). Autonomous Discovery in the Chemical Sciences Part Ii: Outlook. *Angew. Chem. Int. Ed.* 59, 23414–23436. doi:10.1002/anie.201909989

Cressie, N. (1990). The Origins of Kriging. *Math. Geol.* 22, 239–252. doi:10.1007/bf00889887

Frazier, P. I. (2018). *A Tutorial on Bayesian Optimization*. arXiv. arXiv preprint arXiv:1807.02811.

Fujimura, K., Seko, A., Koyama, Y., Kuwabara, A., Kishida, I., Shitara, K., et al. (2013). Accelerated Materials Design of Lithium Superionic Conductors Based on First-Principles Calculations and Machine Learning Algorithms. *Adv. Energ. Mater.* 3, 980–985. doi:10.1002/aenm.201300060

Goodenough, J. B., and Kim, Y. (2010). Challenges for Rechargeable Li Batteries. *Chem. Mater.* 22, 587–603. doi:10.1021/cm901452z

Häse, F., Roch, L. M., and Aspuru-Guzik, A. (2019). Next-generation Experimentation with Self-Driving Laboratories. *Trends Chem.* 1, 282–291. doi:10.1016/j.trechm.2019.02.007

Harada, M., Takeda, H., Suzuki, S., Nakano, K., Tanibata, N., Nakayama, M., et al. (2020). Bayesian-optimization-guided Experimental Search of Nasicon-type Solid Electrolytes for All-Solid-State Li-Ion Batteries. *J. Mater. Chem. A.* 8, 15103–15109. doi:10.1039/d0ta04441e

Homma, K., Liu, Y., Sumita, M., Tamura, R., Fushimi, N., Iwata, J., et al. (2020). Optimization of a Heterogeneous Ternary Li3PO4-Li3BO3-Li2SO4 Mixture for Li-Ion Conductivity by Machine Learning. *J. Phys. Chem. C* 124, 12865–12870. doi:10.1021/acs.jpcc.9b11654

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fmats.2022.821817/full#supplementary-material

Hupfer, T., Bucharsky, E. C., Schell, K. G., Senyshyn, A., Monchak, M., Hoffmann, M. J., et al. (2016). Evolution of Microstructure and its Relation to Ionic Conductivity in Li1+xAlxTi2−x(PO4)3. *Solid State Ionics* 288, 235–239. doi:10.1016/j.ssi.2016.01.036

Hupfer, T., Bucharsky, E. C., Schell, K. G., and Hoffmann, M. J. (2017). Influence of the Secondary Phase LiTiOPO 4 on the Properties of Li 1+x Al X Ti 2−x (PO 4) 3 (X = 0; 0.3). *Solid State Ionics* 302, 49–53. doi:10.1016/j.ssi.2016.10.008

Jackman, S. D., and Cutler, R. A. (2012). Effect of Microcracking on Ionic Conductivity in Latp. *J. Power Sourc.* 218, 65–72. doi:10.1016/j.jpowsour.2012.06.081

Jalem, R., Kanamori, K., Takeuchi, I., Nakayama, M., Yamasaki, H., and Saito, T. (2018). Bayesian-driven First-Principles Calculations for Accelerating Exploration of Fast Ion Conductors for Rechargeable Battery Application. *Sci. Rep.* 8, 5845–5910. doi:10.1038/s41598-018-23852-y

Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient Global Optimization of Expensive Black-Box Functions. *J. Glob. Optimizat.* 13, 455–492. doi:10.1023/a:1008306431147

Ki Williams, C. (2006). *Gaussian Processes Formachine Learning*. Taylor & Francis Group.

Kushner, H. J. (1964). A New Method of Locating the Maximum point of an Arbitrary Multipeak Curve in the Presence of Noise. *J. Basic Eng.* 86 (1), 97–106. doi:10.1115/1.3653121

Ma, Q., Xu, Q., Tsai, C.-L., Tietz, F., and Guillon, O. (2016). A Novel Sol-Gel Method for Large-Scale Production of Nanopowders: Preparation of Li1.5Al0.5Ti1.5(PO4)3as an Example. *J. Am. Ceram. Soc.* 99, 410–414. doi:10.1111/jace.13997

MacLeod, B. P., Parlane, F. G. L., Morrissey, T. D., Häse, F., Roch, L. M., Dettelbach, K. E., et al. (2020). Self-driving Laboratory for Accelerated Discovery of Thin-Film Materials. *Sci. Adv.* 6, eaaz8867. doi:10.1126/sciadv.aaz8867

Manthiram, A., Yu, X., and Wang, S. (2017). Lithium Battery Chemistries Enabled by Solid-State Electrolytes. *Nat. Rev. Mater.* 2, 1–16. doi:10.1038/natrevmats.2016.103

Močkus, J. (1975). "On Bayesian Methods for Seeking the Extremum," in *Optimization Techniques IFIP Technical Conference* (Springer), 400–404.

Narváez-Semanate, J. L., and Rodrigues, A. C. M. (2010). Microstructure and Ionic Conductivity of Li1+Al Ti2−(PO4)3 NASICON Glass-Ceramics. *Solid State Ionics* 181, 1197–1204. doi:10.1016/j.ssi.2010.05.010

Pérez-Estébanez, M., Isasi-Marín, J., Többens, D. M., Rivera-Calzada, A., and León, C. (2014). A Systematic Study of Nasicon-type Li1+xMxTi2−x(PO4)3 (M: Cr, Al, Fe) by Neutron Diffraction and Impedance Spectroscopy. *Solid State Ionics* 266, 1–8. doi:10.1016/j.ssi.2014.07.018

Rahaman, M. N. (2007). *Sintering of Ceramics*. CRC Press.

Rasmussen, C. E. (2003). "Gaussian Processes in Machine Learning," in *Summer School on Machine Learning* (Springer), 63–71.

Rohr, B., Stein, H. S., Guevarra, D., Wang, Y., Haber, J. A., Aykol, M., et al. (2020). Benchmarking the Acceleration of Materials Discovery by Sequential Learning. *Chem. Sci.* 11, 2696–2706. doi:10.1039/c9sc05999g

Rossbach, A., Tietz, F., and Grieshammer, S. (2018). Structural and Transport Properties of Lithium-Conducting Nasicon Materials. *J. Power Sourc.* 391, 1–9. doi:10.1016/j.jpowsour.2018.04.059

Schiffmann, N., Bucharsky, E. C., Schell, K. G., Fritsch, C. A., Knapp, M., and Hoffmann, M. J. (2021). Upscaling of Latp Synthesis: Stoichiometric Screening of Phase Purity and Microstructure to Ionic Conductivity Maps. *Ionics* 27, 2017–2025. doi:10.1007/s11581-021-03961-x

Sendek, A. D., Yang, Q., Cubuk, E. D., Duerloo, K.-A. N., Cui, Y., and Reed, E. J. (2017). Holistic Computational Structure Screening of More Than 12 000 Candidates for Solid Lithium-Ion Conductor Materials. *Energy Environ. Sci.* 10, 306–320. doi:10.1039/c6ee02697d

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. (2015). Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proc. IEEE* 104, 148–175. doi:10.1109/JPROC.2015.2494218

Snoek, J., Larochelle, H., and Adams, R. P. (2012). *Practical Bayesian Optimization of Machine Learning Algorithms*. arXiv. arXiv preprint arXiv:1206.2944.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. (2009). *Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design*. arXiv. arXiv preprint arXiv:0912.3995.

Stein, H. S., and Gregoire, J. M. (2019). Progress and Prospects for Accelerating Materials Science with Automated and Autonomous Workflows. *Chem. Sci.* 10, 9640–9649. doi:10.1039/c9sc03766g

Tran, K., and Ulissi, Z. W. (2018). Active Learning across Intermetallics to Guide Discovery of Electrocatalysts for CO2 Reduction and H2 Evolution. *Nat. Catal.* 1, 696–703. doi:10.1038/s41929-018-0142-1

Vasudevan, R. K., Choudhary, K., Mehta, A., Smith, R., Kusne, G., Tavazza, F., et al. (2019). Materials Science in the Artificial Intelligence Age: High-Throughput Library Generation, Machine Learning, and a Pathway from Correlations to the Underpinning Physics. *MRS Commun.* 9, 821–838. doi:10.1557/mrc.2019.95

Waetzig, K., Rost, A., Langklotz, U., Matthey, B., and Schilm, J. (2016). An Explanation of the Microcrack Formation in Li 1.3 Al 0.3 Ti 1.7 (PO 4) 3 Ceramics. *J. Eur. Ceram. Soc.* 36, 1995–2001. doi:10.1016/j.jeurceramsoc.2016.02.042

Wang, Y., Richards, W. D., Ong, S. P., Miara, L. J., Kim, J. C., Mo, Y., et al. (2015). Design Principles for Solid-State Lithium Superionic Conductors. *Nat. Mater* 14, 1026–1031. doi:10.1038/nmat4369

Williams, C. K., and Rasmussen, C. E. (2006). *Gaussian Processes for Machine Learning*, Vol. 2. Cambridge, MA: MIT press.

Wolpert, D. H., and Macready, W. G. (1997). No Free Lunch Theorems for Optimization. *IEEE Trans. Evol. Computat.* 1, 67–82. doi:10.1109/4235.585893

Xue, D., Balachandran, P. V., Hogden, J., Theiler, J., Xue, D., and Lookman, T. (2016). Accelerated Search for Materials with Targeted Properties by Adaptive Design. *Nat. Commun.* 7, 11241–11249. doi:10.1038/ncomms11241

Yang, Z., Suzuki, S., Tanibata, N., Takeda, H., Nakayama, M., Karasuyama, M., et al. (2020). Efficient Experimental Search for Discovering a Fast Li-Ion Conductor from a Perovskite-type LixLa(1-x)/3NbO3 (LLNO) Solid-State Electrolyte Using Bayesian Optimization. *J. Phys. Chem. C* 125, 152–160. doi:10.1021/acs.jpcc.0c08887

Yuan, R., Tian, Y., Xue, D., Xue, D., Zhou, Y., Ding, X., et al. (2019). Accelerated Search for BaTiO 3 -Based Ceramics with Large Energy Storage at Low Fields Using Machine Learning and Experimental Design. *Adv. Sci.* 6, 1901395. doi:10.1002/advs.201901395

Zhang, Y., He, X., Chen, Z., Bai, Q., Nolan, A. M., Roberts, C. A., et al. (2019). Unsupervised Discovery of Solid-State Lithium Ion Conductors. *Nat. Commun.* 10, 5260–5267. doi:10.1038/s41467-019-13214-1

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers in Materials

ORIGINAL RESEARCH
published: 14 February 2022
doi: 10.3389/fmats.2021.824441

Efficient Exploration of Microstructure-Property Spaces *via* Active Learning

Lukas Morand[1]*, *Norbert Link*[2], *Tarek Iraki*[2], *Johannes Dornheim*[2,3] and *Dirk Helm*[1]

[1]Fraunhofer Institute for Mechanics of Materials IWM, Freiburg, Germany, [2]Intelligent Systems Research Group ISRG, Karlsruhe University of Applied Sciences, Karlsruhe, Germany, [3]Institute for Applied Materials—Computational Materials Sciences IAM-CMS, Karlsruhe Institute of Technology, Karlsruhe, Germany

In materials design, supervised learning plays an important role for optimization and inverse modeling of microstructure-property relations. To successfully apply supervised learning models, it is essential to train them on suitable data. Here, suitable means that the data covers the microstructure and property space sufficiently and, especially for optimization and inverse modeling, that the property space is explored broadly. For virtual materials design, typically data is generated by numerical simulations, which implies that data pairs can be sampled on demand at arbitrary locations in microstructure space. However, exploring the space of properties remains challenging. To tackle this problem, interactive learning techniques known as active learning can be applied. The present work is the first that investigates the applicability of the active learning strategy query-by-committee for an efficient property space exploration. Furthermore, an extension to active learning strategies is described, which prevents from exploring regions with properties out of scope (i.e., properties that are physically not meaningful or not reachable by manufacturing processes).

Keywords: active learning, adaptive sampling, data generation, inverse modeling, materials design, membership query synthesis, microstructure-property relations, query-by-committee

OPEN ACCESS

Edited by:
Surya R. Kalidindi, Georgia Institute of Technology, United States

Reviewed by:
Anh Tran, Sandia National Laboratories, United States
Tomasz Michno, Kielce University of Technology, Poland

***Correspondence:**
Lukas Morand
lukas.morand@iwm.fraunhofer.de

Specialty section:
This article was submitted to Computational Materials Science, a section of the journal Frontiers in Materials

Received: *29 November 2021*
Accepted: *22 December 2021*
Published: *14 February 2022*

Citation:
Morand L, Link N, Iraki T, Dornheim J and Helm D (2022) Efficient Exploration of Microstructure-Property Spaces via Active Learning. Front. Mater. 8:824441. doi: 10.3389/fmats.2021.824441

1 INTRODUCTION

1.1 Motivation

With regard to natural learning processes, Cohn et al. stated that "in many situations, the learner's most powerful tool is its ability to act, to gather data, and to influence the world it is trying to understand" (Cohn et al., 1996). One attempt to transfer this ability to methods in the field of machine learning is called active learning. Active learning describes an interactive learning process in which machine learning models improve with experience and training (Settles, 2012). It is therefore contrary to the often used approach of gathering data a priori and learn from it afterwards. In fact, active learning couples both, sampling and training, what typically results in an efficient and broad data space exploration.

In terms of materials design, the exploration of data spaces is quite important. For a designer, it is essential to know all possible microstructure configurations and reachable properties of a material in order to increase the performance of a workpiece. How to delineate these configurations and properties is described for example in the microstructure sensitive design (MSD) approach (Adams et al., 2001). Two necessary steps for MSD are 1) to determine the design space yielding a hull of possible microstructures and 2) to calculate the corresponding properties defining a so-called

property closure (Fullwood et al., 2010). For complex high-dimensional microstructure representations, exploring the design space is, however, challenging. It is even more challenging, if gathering data is cumbersome, because it is time-consuming (e.g., running complex numerical simulations), ties up manpower (e.g., labeling data manually) or laborious (e.g., performing experiments).

To the authors knowledge, the present work is the first one that uses the active learning strategy query-by-committee (Burbidge et al., 2007) for generating microstructure-property data in materials science. We show that the active learning strategy can be used to explore microstructure-property spaces efficiently and that the generated data is better suited to train accurate machine learning models than data generated *via* classical sampling approaches. Moreover, we present an extension to active learning approaches that aims to avoid sampling in regions with properties out of scope. This is necessary, as defining bounds for microstructure spaces is not always simple and it can happen that active learning techniques explore regions with properties that are physically not meaningful or not reachable by manufacturing processes.

1.2 Related Work

Active learning techniques can be grouped into three use-case specific classes, namely stream-based selective sampling, pool-based selective sampling and membership query synthesis (Settles, 2009). The first two mentioned are based either on a continuous data-stream or an a priori defined pool of data, from which the active learning algorithm can chose data points to be labeled. In membership query synthesis, in contrast, the algorithm is free of choice at which location new data points are created. Therefore, membership query synthesis is well suited for virtual data generation on the basis of numerical simulations.

Membership query synthesis goes back to Angluin (1988) for classification problems and to Cohn et al. (1996) for regression problems. The technique presented in Cohn et al. (1996) is called variance reduction. It aims to minimize the output variance of a machine learning model in order to minimize the future generalization error. Alternative approaches for regression problems are maximizing the expected change of a machine learning model when seeing new data (Cai et al., 2013) or committee-based approaches like query-by-committee (Burbidge et al., 2007), where the prediction variance of a committee consisting of multiple separately trained machine learning models is minimized. In the present study, we use the latter approach, as it is straightforward to implement and scales to complex models (i.e., neural networks). Recent research in active learning targets the application of deep learning models, see for example Stark et al. (2015) and Wang et al. (2016) for applications using convolutional neural networks and Zhu and Bento (2017), Sinha et al. (2019) and Mayer and Timofte (2020) for generative models.

Instead of actively sampling data spaces, classical space-filling sampling strategies can be used to generate data without considering the learning task, see Fang et al. (2000), Simpson et al. (2001) and Wang and Shan (2006) for an overview. In the following, we list some of the most popular space-filling sampling strategies. Latin hypercube design (McKay et al., 2000) aims to partition the dimensions of the input space into equidistant slices and places data points such that each slice is covered by one data point. Orthogonal arrays (Owen, 1992) are special matrices that define sampling with the aim to sample data spaces uniformly. In particular, orthogonal arrays can be used to generate uniform Latin hypercubes (Tang, 1993). Furthermore, low-discrepancy sequences can be used to cover spaces uniformly with data points, see for example Niederreiter (1988). Among others, popular sequences are the Hammersley sequence (Hammersley and Handscomb, 1964), Halton sequence (Halton, 1964) and Sobol sequence (Sobol, 1967). In addition to these methods, a common sampling strategy is to randomly draw samples from a uniform distribution. However, all of these approaches suffer from the curse of dimensionality, which means that the effort needed to sufficiently sample data spaces grows exponentially with the number of dimensions.

In materials design, the usage of classical sampling strategies is very common. The framework for data-driven analysis of materials that is presented in Bessa et al. (2017) for example, uses data generation on the basis of space-filling sampling methods, especially the Sobol sequence. In Gupta et al. (2015), dual-phase 2D-microstructures are generated by randomly placing particles in a steel matrix. In order to generate spatially resolved dual-phase microstructure volume elements, Liu et al. (2015b) uses evenly distributed data of phase volume fractions. Regarding homogenized microstructural features, in Iraki et al. (2021), Latin hypercube design is used to generate a data set of textures for cold rolled steel sheets. Even special sampling heuristics have been developed for generating sets of microstructure features, like in Johnson and Kurniawan, 2018.

Also, adaptive sampling techniques are used in materials design, however, in the sense of an optimization aiming to identify microstructures with targeted properties. In Liu et al. (2015a) and Paul et al. (2019), specific machine learning-based optimization approaches are presented that efficiently guide sampling to regions in the space of microstructures, where microstructures with desired properties are expected to be located. Further statistic-based approaches exist that use surrogate-based optimization (cf. Forrester and Keane, 2009), see Nikolaev et al. (2016), Balachandran et al. (2016), Lookman et al. (2017) and Lookman et al. (2019). Yet, as these approaches aim to find individual material compositions or microstructures for certain target properties, they are not applicable for sampling microstructure-property spaces broadly.

So far, only few publications exist, which describe the usage of active learning to train a machine learning model while generating microstructure-property data (Jung et al., 2019; Kalidindi, 2019; Castillo et al., 2019). The approaches presented therein are based on variance reduction using Bayesian models, like Gaussian process regression (GPR), cf. Seo et al. (2000). Such Bayesian approaches can have a tremendous advantage when working with experimental measurements. However, the computational complexity of GPR increases cubically with the number of data points. Furthermore, it is worth mentioning the Bayesian approach described in Tran and Wildey (2021) to solve stochastic

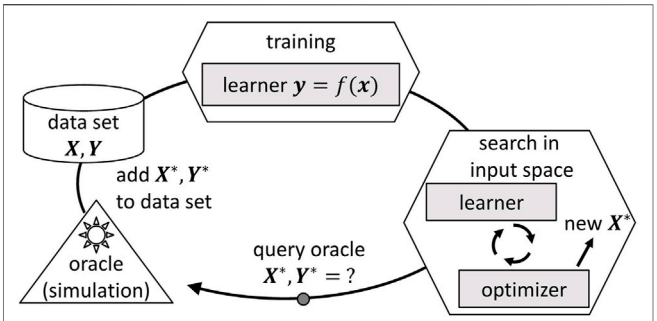


FIGURE 1 | Iterative procedure of membership query synthesis, cf. Settles (2012).

inverse problems for property-structure linkages. It is the aim of the approach to model posterior probabilities for microstructures having desired properties. This is achieved by successively updating an a priori probability distribution with new sampled microstructure-property data points. These are generated by drawing microstructure samples from the actual probability distribution and evaluating their properties. Afterwards, the samples are accepted or rejected depending on a certain criterion. However, this so-called acceptance-rejection sampling can be disadvantageous in terms of sample efficiency, as the decision to accept samples is made after calculating properties.

1.3 Paper Structure

In **Section 2** the concept of membership query synthesis is presented including an extension to avoid sampling in regions out of scope. Additionally, the applied query-by-committee approach is described in detail. In **Section 3**, three numerical examples are shown to demonstrate the advantage of using active learning to sample microstructure-property spaces. The results are discussed in **Section 4**. The work is summarized in **Section 5** and a brief outlook on the application of active learning in virtual materials design is given.

2 METHODS

2.1 Active Learning *via* Membership Query Synthesis

Membership query synthesis follows an iterative procedure that is sketched schematically in **Figure 1**, cf. Settles (2012). The procedure starts with an initial data set of input variables $\boldsymbol{X}_i \in \mathbb{R}^k$ and corresponding target variables $\boldsymbol{Y}_i \in \mathbb{R}^l$. The mapping from input space $\mathcal{X}$ to output space $\mathcal{Y}$ is approximated by a learner:

$$f\colon \mathcal{X} \to \mathcal{Y}, \quad \boldsymbol{y} = f(\boldsymbol{x}), \tag{1}$$

where $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^k$ and $\boldsymbol{y} \in \mathcal{Y} \subset \mathbb{R}^l$. The learner is realized by one or more supervised learning models. To apply active learning, it is essential that the learner's prediction quality can be measured. On the basis of such a measure, an optimization is performed with the objective to find the location $\boldsymbol{X}^\star$ in the input space $\mathcal{X}$ at which the learner's prediction quality is likely worst. At this location, a new data point is generated in order to improve the learner. To get the corresponding target label $\boldsymbol{Y}^\star$, the so-called oracle (in our case a numerical simulation) is queried. The obtained new data tuple $\boldsymbol{X}^\star$, $\boldsymbol{Y}^\star$ is added to the data set and the procedure is repeated.

2.2 Avoid Queries in Regions out of Scope

Following the procedure described in **Section 2.1**, new data points are generated over the whole input space. In many applications, this might be appropriate to improve the learner. However, when the input space bounds cannot be defined adequately, it is probable that the active learning algorithm queries for data in regions that are out of interest for the application case. To avoid sampling in regions out of scope, the original workflow depicted in **Figure 1** can be extended with a region, cf. **Figure 2**. The purpose of the region filter is to limit the optimizer to regions in the input space leading only to output quantities of interest. In order to set up the region filter, the data points $\boldsymbol{X}_i$ get an additional class label $c \in (0, 1)$, which marks if the output values are of interest or out of scope. The bounds in the output space that determine the class label can be defined by the user.

The region filter is realized by a binary classifier that partitions the input space depending on the class label by learning the mapping function

$$g\colon \mathcal{X} \to c, \quad \boldsymbol{c} = g(\boldsymbol{x}). \tag{2}$$

In fact, the described extension is similar to Bayesian optimization approaches that account for unknown constraints using classification methods, see for example Sacher et al. (2018), Heese et al. (2019) and Tran et al. (2019).

Often the amount of data that is out of scope is much lower than the data of interest. If this is the case, one-class classification methods can be used as region filter, such as isolation forests (Liu et al., 2008) or one-class support vector machines (Schölkopf et al., 2001). Both are unsupervised learning methods that delimit the input space, which is covered by data out of scope. Once trained, they are used to estimate the class membership of unseen

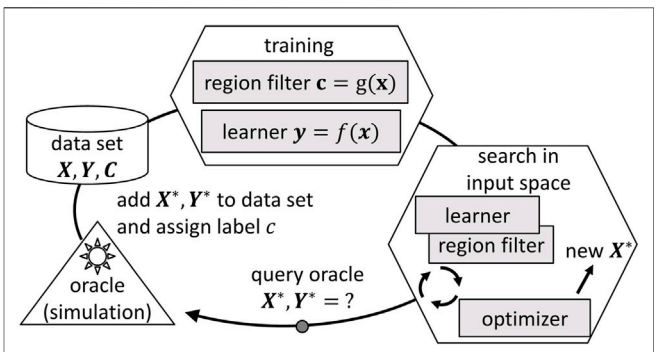


FIGURE 2 | Iterative procedure of membership query synthesis (cf. Settles (2012)) including an extension to avoid sampling in regions out of scope. The class labels of the data points are represented by $C_i \in (0, 1)$.

data points. For more complex classification problems, this can also be achieved by deep learning approaches (cf. Chalapathy and Chawla, 2019), such as autoencoder neural networks (Hinton and Salakhutdinov, 2006; Sakurada and Yairi, 2014).

2.3 Query-By-Committee for Microstructure-Property Space Exploration

Originally, query-by-committee was introduced by Seung et al. (1992) for classification problems. In this study, the query-by-committee approach following Burbidge et al. (2007) for regression problems is applied. In this approach, the learner is realized by a committee of n regression models (here we use feedforward neural networks). Following the workflow depicted in **Figure 1**, the committee members are trained on the actual data set. However, in this work, each neural network is trained only on a subset of the data (RayChaudhuri and Hamey, 1995). In order to query new data points, the microstructure space is searched for the location at which the committee members disagree the most. Disagreement is defined by the variance of the neural network predictions s^2 (Krogh and Vedelsby, 1995):

$$s^2(\boldsymbol{x}) = \sum_{\eta=1}^{n} \left(f_\eta(\boldsymbol{x}) - \bar{f}(\boldsymbol{x}) \right)^2, \tag{3}$$

where $f_\eta(\boldsymbol{x})$ denotes the property prediction of neural network η and $\bar{f}(\boldsymbol{x})$ denotes the mean over all predictions at location $\boldsymbol{x}$. The location to query the next data point is determined by

$$\boldsymbol{X}^* = \arg\max_{\boldsymbol{x}} \left(s^2(\boldsymbol{x}) \right). \tag{4}$$

Certainly, it is challenging here to chose the right number of regression models and to equip them with sufficient complexity (e.g., depth of neural networks). Depending on the mapping to learn, we suggest to assign a lower complexity to the regression models in the beginning, as the amount of initial data is typically low. With an increasing amount of data it is possible to increase the complexity of the regression models, which was, however, not done in this study. Regarding the number of regression models in the committee, the similarity of the query-by-committee approach to Bayesian methods like GPR is worth mentioning here. GPR can be interpreted as a distribution over functions (Williams and Rasmussen, 2006), which is also the case for the query-by-committee approach when the number of committee members goes to infinity. Though, the overall training time of the query-by-committee approach increases linearly with an increasing number of committee members.

In order to extend the query-by-committee approach to avoid sampling in regions out of scope, first, data points that exceed the predefined output bounds need to be filtered out from the actual data set. Then, a classifier is trained on these data points in order to delimit a region in microstructure space. This region is excluded from the optimization by adding a soft constraint to **Eq. 4**:

$$\boldsymbol{X}^* = \arg\max_{\boldsymbol{x}} \left(s^2(\boldsymbol{x}) - W \, \langle \rho(\boldsymbol{x}) \rangle \right), \tag{5}$$

where $\langle \cdot \rangle$ denotes the Macaulay brackets and $\rho(\boldsymbol{x})$ denotes the distance of $\boldsymbol{x}$ to the decision boundary that is defined by the classifier. As $s^2(\boldsymbol{x})$ and $\rho(\boldsymbol{x})$ can be of different magnitudes, the scalar weight factor W is introduced, which needs to be set in order to balance the optimization.

In this work, $\rho(\boldsymbol{x})$ is determined by an isolation forest classifier. Isolation forest is an outlier detection method that consists of an ensemble of decision trees. Each tree partitions the input space randomly until all training data points are isolated. It is assumed that outliers typically lie in partitions with rather short paths in the decision tree structures. On the basis of the path lengths, an anomaly score (in the range of (0, 1)) can be defined for each observation, see Liu et al. (2008) for details. Therein, it is stated that data points with an anomaly score < 0.5 can be regarded as being normal. Consequently, the decision function $\rho(\boldsymbol{x})$ can be defined by shifting the anomaly score to the range $(-0.5, 0.5)$, and, in this work, by multiplying it by -1. The latter needs to be done because the isolation forest is trained only on microstructures that exceed the predefined property bounds. In this respect, two cases can occur in **Eq. 5**. If $\rho(\boldsymbol{x}) > 0$, the optimization is punished such that the optimizer is forced to generate candidate microstructures that do likely not exceed the specified property bounds. Generating such microstructures then leads to $\rho(\boldsymbol{x}) \leq 0$, what does not affect the optimization at all.

To solve **Eqs 4**, **5**, we use the differential evolution algorithm by Storn and Price (1997) as it is implemented in Python package *scipy* (Virtanen et al., 2020). The neural network models and the isolation forest classifier applied in this work are based on the implementation in Python package *scikit-learn* (Pedregosa et al., 2011).

3 RESULTS

3.1 Toy Example: Dirac Delta Function

First, a simplistic extreme case is analyzed. The data generating process considered here is given by an approximation of the Dirac delta function via a Gaussian distribution

$$\delta(x) = \frac{1}{|\alpha|\sqrt{x}} e^{-\left(\frac{x}{\alpha}\right)^2}, \tag{6}$$

with parameter $\alpha = 0.1$. A set of 500 data points is generated by randomly drawing samples of x in the range of $(-50, 50)$. Additionally, 500 data points are generated via query-by-committee. Therefore, a committee of five neural networks is set up, which are all trained on a random subset of 80% of the actual data. The neural networks consist of two layers with five neurons each. To avoid overfitting, early stopping (Prechelt, 1998) and $L2$-regularization (Krogh and Hertz, 1992) is applied with regularization parameter $\lambda = 0.0001$. As activation functions, rectifiers (ReLU) are used. The mean-squared-error loss function between true and predicted $\delta(x)$ is applied and optimized using the limited-memory Broyden-Fletcher-Goldfarb-Shanno (BFGS) optimizer (Liu and Nocedal, 1989). The approach is initialized with 100 randomly drawn samples.

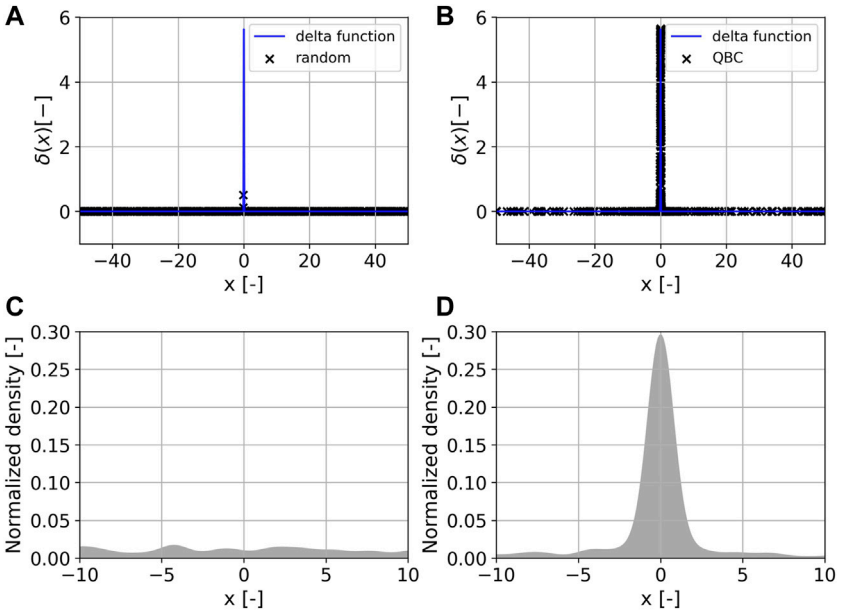


FIGURE 3 | The sampled input-output space of the approximated Dirac delta function $\delta(x)$ is shown above. 500 samples were generated via random sampling **(A)** and query-by-committee, labeled as QBC, **(B)**. Below, the normalized Gaussian kernel density estimation is shown for the data sets generated via random sampling **(C)** and query-by-committee **(D)**.

The resulting 500 data points and their distribution over x are shown in **Figure 3**. The data points generated by random sampling are distributed almost uniformly over the input space. All data points are located in regions of lower $\delta(x)$. The maximum $\delta(x)$-value in the randomly sampled data set equals 0.506 208 09. In contrast, the query-by-committee approach concentrates on sampling the region close to the peak of the approximated delta function. The maximum $\delta(x)$-value in the data set equals 5.64189535, which is very close to the maximum of **Eq. 6**: $\delta(x = 0) = 5.64189584$. The generated data is available online, see Morand et al. (2021).

3.2 Identifying Material Model Parameters

The second example is about inferring material model parameters from given material model responses, which is (like typical materials design problems) an inverse identification problem, cf. Mahnken (2004). To solve it, neural networks can be used to directly learn a mapping from material model responses to model parameters (Yagawa and Okuda, 1996; Huber, 2000). Such an approach is for example applied in Huber and Tsakmakis (1999) to identify constitutive parameters of a finite deformation plasticity model on the basis of spherical indentation tests. As simulating spherical indentation tests is time consuming, the usage of active learning can be beneficial, because it efficiently explores the space of material model parameters and responses. This characteristic can be understood as goal-directed sampling, which is essential for the prediction quality of supervised learning models that are directly trained on inverse relations (Jordan and Rumelhart, 1992).

In this example, we analyze the identification problem described in Morand and Helm (2019), as it requires a special sampling, for which a knowledge-based approach has been developed. The data generating process is defined by the hardening model (cf. Helm, 2006):

$$H(s_\mathrm{p}, \beta, \gamma) = \frac{\gamma}{\beta}\left(1 - e^{-\beta s_\mathrm{p}}\right), \tag{7}$$

where β and γ are material dependent parameters and s_p denotes the accumulated plastic strain. For the purpose of this study, the hardening curves are discretized into 20 equidistantly distributed points in $s_\mathrm{p} \in (0.0, 0.2)$. For sampling, we consider β and $\frac{\gamma}{\beta}$ being inside the ranges (5, 200) and (100, 400), respectively. This yields a parameter identification problem as it is illustrated in **Figure 4**.

In total 2,500 discretized hardening curves $\boldsymbol{H}_i \in \mathbb{R}^{20}$ are generated by varying β and $\frac{\gamma}{\beta}$ using 1) Latin hypercube design, 2) the proposed knowledge-based sampling approach following Morand and Helm (2019) and 3) query-by-committee. The knowledge-based approach from Morand and Helm (2019) is also based on Latin hypercube design, however, the parameter variations in β are manipulated such that the region of lower β-values is sampled more densely (as this region is significant for the shapes of the hardening curves). The configuration of the query-by-committee approach here is the same as in **Section 3.1**, except for the neural network complexity, which is increased to two hidden layers with 10 and 15 neurons. The initial data set consists of 100 randomly sampled data points.

The resulting sets of parameter tuples $(\beta, \frac{\gamma}{\beta})$ chosen by the three sampling strategies are shown in **Figure 5** and are represented in the following by $\boldsymbol{B}_i \in \mathbb{R}^2$. Per definition, Latin hypercube design samples the parameter space almost uniformly. In contrast, the query-by-committee approach samples the parameter space in a similar manner as it is done by the knowledge-based approach. Thereby, the region of lower β-values is sampled even more

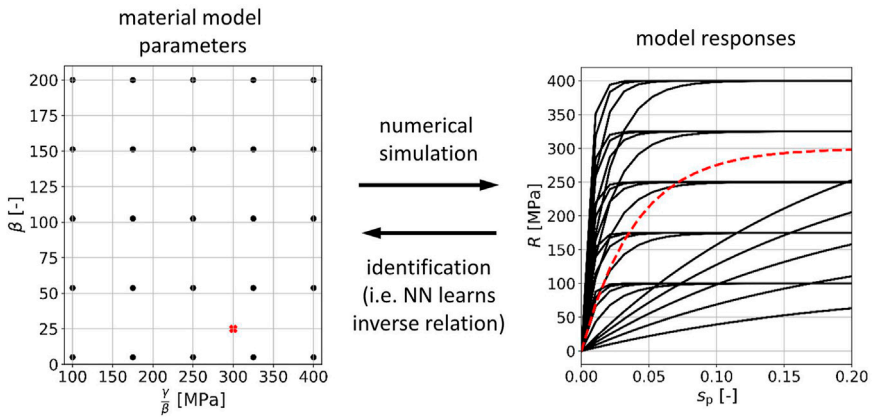


FIGURE 4 | Parameter identification problem. A data base of material model parameters and corresponding responses is set up (black dots and curves), which can be used to train a neural network (NN) on the inverse mapping. After training, the neural network is able to identify parameters for given hardening curves (red dashed line and cross).

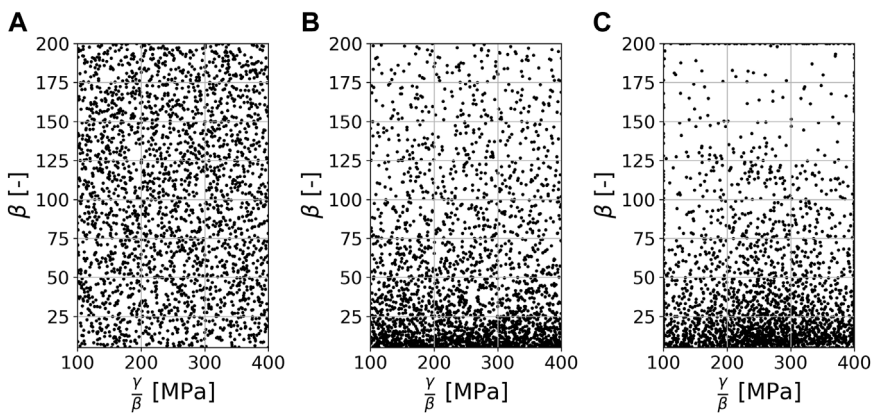


FIGURE 5 | The sampled parameter space of the hardening model described in **Eq. 7**. 2,500 samples were generated using Latin hypercube design **(A)**, the knowledge-based sampling approach following Morand and Helm (2019) **(B)** and query-by-committee **(C)**.

densely and, in contrast to the knowledge-based approach, also the bounds of the parameter space are sampled. Naturally, the three generated data sets have different effects on the prediction quality of supervised learning models.

In order to show these effects, neural networks are trained on the data sets. As there is no ground truth to test the trained models, the generated data is also used for testing. Training and testing is done for both, the forward mapping

$$f\colon \mathcal{B} \to \mathcal{H}, \quad \boldsymbol{h} = f(\boldsymbol{b}), \tag{8}$$

and the inverse mapping as it is outlined in **Figure 4**

$$f^{-1}\colon \mathcal{H} \to \mathcal{B}, \quad \boldsymbol{b} = f^{-1}(\boldsymbol{h}), \tag{9}$$

where $\boldsymbol{b} \in \mathcal{B} \subset \mathbb{R}^2$ and $\boldsymbol{h} \in \mathcal{H} \subset \mathbb{R}^{20}$. Here, $\mathcal{B}$ denotes the space of hardening parameters and $\mathcal{H}$ the space of discretized hardening curves.

The neural networks that learn the forward mapping consist of two hidden layers with 10 and 15 neurons and for learning the inverse mapping they consist of two hidden layers with 15 and 10 neurons. In both cases, the mean-squared-error loss function between true and predicted output quantity is applied and optimized using the limited-memory BFGS optimizer with $L2$ regularization of $\lambda = 0.0001$. Furthermore, early stopping is applied using a random subset of 10% of the training data for validation. Both networks use ReLU activation functions. To measure the performance of the forward models, the absolute error between the predicted curve $\boldsymbol{H}_{\text{pred}}$ and the true curve $\boldsymbol{H}_{\text{true}}$ is given by

$$\Delta H = \frac{1}{20}\sum |\boldsymbol{H}_{\text{pred}} - \boldsymbol{H}_{\text{true}}|. \tag{10}$$

To measure the performance of the inverse models, the curves are reconstructed using the predicted material model parameters (which yields $\boldsymbol{H}_{\text{recon}}$) and compared with the true curves $\boldsymbol{H}_{\text{true}}$:

$$\Delta H = \frac{1}{20}\sum |\boldsymbol{H}_{\text{recon}} - \boldsymbol{H}_{\text{true}}|. \tag{11}$$

TABLE 1 | Averaged mean ΔH for the neural networks that are trained and evaluated using the three data sets. The data sets are generated using LHD (Latin hypercube design), KBS (the knowledge-based sampling approach), and QBC (query-by-committee). The best result for each test set is marked in bold.

Av. mean ΔH (MPa)	Forward model trained with			Inverse model trained with		
	LHD set	KBS set	QBC set	LHD set	KBS set	QBC set
Tested on LHD set	—	3.70	**3.46**	—	**2.60**	3.06
Tested on KBS set	7.92	—	**4.07**	9.10	—	**6.10**
Tested on QBC set	7.88	**4.77**	—	9.77	**5.66**	—

TABLE 2 | Average mean ΔE and Δr for the neural networks trained and evaluated on the three generated data sets: LHD (Latin hypercube design), QBC (query-by-committee) and QBC+ (query-by-committee with extension). The best result for each test set is marked in bold.

Av. mean ΔE (MPa)	Forward model trained with		
	LHD set	QBC set	QBC+ set
Tested on LHD set	—	**125**	133
Tested on QBC set	240	—	**207**
Tested on QBC + set	183	**157**	—
Av. mean Δr (–)	**Forward model trained with**		
	LHD set	**QBC set**	**QBC+ set**
Tested on LHD set	—	**0.035**	0.037
Tested on QBC set	0.081	—	**0.062**
Tested on QBC + set	0.055	**0.045**	—

Training and test runs were performed five times with different random validation splits. The averaged results are listed in **Table 1**. The neural network trained on the data set generated by query-by-committee reaches a similar performance than the neural network trained on the data generated by the knowledge-based approach when tested on the Latin hypercube samples. When tested on each other, the averaged mean is rather low for both approaches. However, one can observe that for modeling the forward relation, the neural network trained with the data generated by query-by-committee is slightly better than the one trained with the data generated by the knowledge-based sampling approach and vice versa for the inverse relation. Comparing both neural networks with the neural network trained on the data generated via Latin hypercube design, the latter is outperformed for every test set. The generated data is available online, see Morand et al. (2021).

3.3 Artificial Rolling Texture Generation

As a third example, we analyze the problem of generating microstructure-property data, which is used to learn a forward mapping as a fundamental basis to solve materials design problems in Iraki et al. (2021). The specific materials design problem tackled therein is the identification of crystallographic textures for given desired material properties of DC04 steel sheets, see **Figure 6** for an illustration. Basically, this is achieved by using a machine learning-based model that approximates the mapping from crystallographic textures to properties combined with an optimization approach. Alternatively, the identification problem can be solved by learning the inverse mapping. In general, solving inverse problems is challenging due to ill-posedness. In this example, the solution of the inverse problem is not guaranteed to exist, and if it exists, it is not guaranteed to be unique (in contrast to the previous example). Here, the uniqueness and existence of a solution is highly depending on the choice of desired properties. If the definition of desired properties is very specific, then it is rather unlikely that a microstructure leading to exact these properties exists. One way to tackle this problem is by defining target property windows (desired properties with tolerances), as is done in Iraki et al. (2021) for example.

In Iraki et al. (2021), texture generation is done based on the rolling texture description model described in Delannay et al. (1999). In this study, the parameter ranges for the texture description model are defined as is described in Iraki et al. (2021). To calculate the properties of interest, a crystal plasticity model is used. The model is of Taylor-type and is set up following Kalidindi et al. (1992). For a detailed description of the Taylor-type crystal plasticity model, see Dornheim et al. (2022) and Iraki et al. (2021). Besides, instead of using a Taylor-type crystal plasticity model, also computationally expensive full-field models can be applied here. For the purpose of our study, we use the Taylor-type crystal plasticity model to determine the Young's moduli E_φ and the Lankford coefficients (r-values) r_φ, both at 0, 45 and 90° to rolling direction for given crystallographic textures. In the following, the generated properties are represented by $\boldsymbol{P}_i \in \mathbb{R}^6$. The material model parameters are chosen to represent DC04 steel (cf. Iraki et al., 2021). However, using the elastic constants for ferrite from Eghtesad and Knezevic, (2020), the Young's modulus is slightly overestimated by our simulations.

In the following, we compare the generation of 5,000 texture-property data pairs using Latin hypercube design and query-by-committee. As r-values of rolled DC04 sheets typically do not exceed values of 5.0, we additionally apply the extension described in **Section 2.2** to suppress generating data in regions leading to $r > 5.0$ (the factor to weight the soft constraint W in **Eq. 5** is set to 100). For the query-by-committee approach, a committee of five neural networks is used with two hidden layers of 24 and 6 neurons. Every committee member is trained on a random subset of 80% of the actual data. The mean-squared-error loss function between true and predicted properties is applied and the limited-memory BFGS optimizer is used. Early stopping and $L2$-regularization with $\lambda = 0.1$ are applied. The activation function used is ReLU. For an initial data set, 100 texture-property data points are sampled randomly.

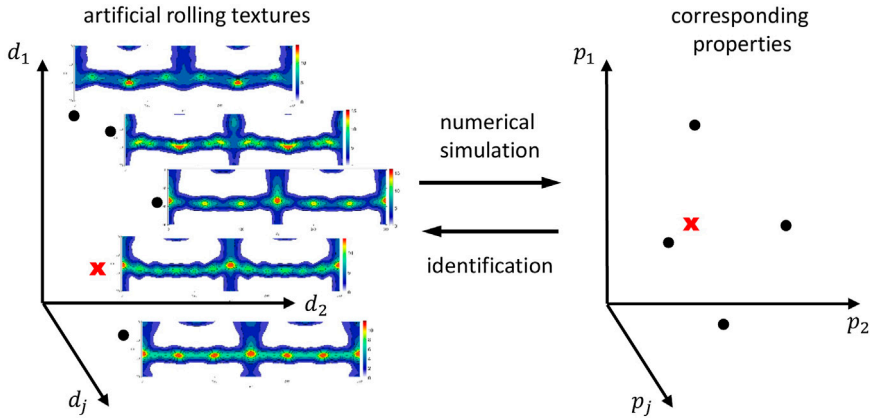


FIGURE 6 | Illustration of the texture identification problem. The space of rolling textures is described by d_j and the space of properties by p_j.

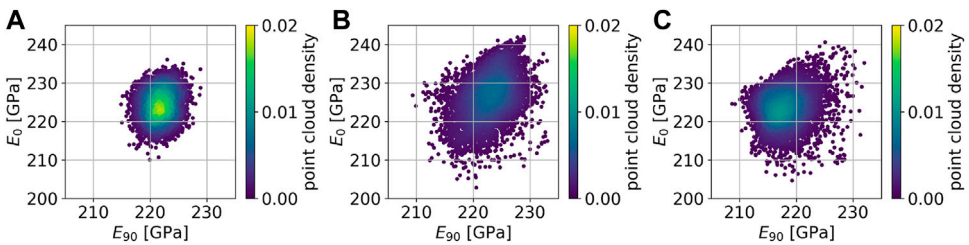


FIGURE 7 | 5,000 sampled texture-property data points projected into property space (E_0, E_{90}). Data points are generated on the basis of Latin hypercube design **(A)**, query-by-committee **(B)** and extended query-by-committee with $r \leq 5.0$ **(C)**.

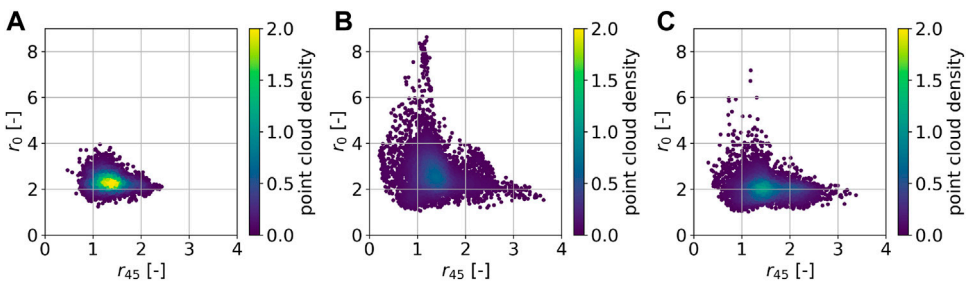


FIGURE 8 | 5,000 sampled texture-property data points projected into property space (r_{45}, r_0). The data points are generated on the basis of Latin hypercube design **(A)**, query-by-committee **(B)** and extended query-by-committee with $r \leq 5.0$ **(C)**.

The obtained data sets are depicted as projections in property space in **Figure 7** (E_0, E_{90}) and **Figure 8** (r_{45}, r_0). The point cloud generated by Latin hypercube design comprises a much smaller region in property space compared to the ones generated by query-by-committee. Furthermore, the point cloud is concentrated at its center. Such a strong concentration cannot be observed in the point clouds generated by query-by-committee. Also the minimum and maximum values in both, E and r, that are found by the active learning strategies are more extreme than by using Latin hypercube design. However, the original query-by-committee sampling approach leads to unrealistic high r-values, cf. **Figure 8B**. In contrast, **Figure 8C** shows that this effect can be minimized by applying the extension to query-by-committee presented in **Section 2.2**. The applied region filter (isolation forest) limits the active learning search space in such a way that textures with high r-values are excluded. Therefore, the amount of textures in the data set that lead to unrealistic high r-values decreases dramatically compared to the data set generated without the query-by-committee extension. The latter includes 141 data points with $r > 5$, while the former includes only 11. The generated data is available online, see Morand et al. (2021).

To evaluate the effect of the applied sampling strategies on supervised learning models, we train and test feedforward neural

TABLE 3 | Average maximum ΔE and Δr for the neural networks trained and evaluated on the three generated data sets: LHD (Latin hypercube design), QBC (query-by-committee) and QBC+ (query-by-committee with extension). The best result for each test set is marked in bold.

Av. max ΔE (MPa)	Forward model trained with		
	LHD set	QBC set	QBC+ set
Tested on LHD set	—	1,347	**1,252**
Tested on QBC set	2,684	—	**1977**
Tested on QBC + set	2,296	**1,227**	—
Av. max Δr (–)	**Forward model trained with**		
	LHD set	**QBC set**	**QBC+ set**
Tested on LHD set	—	**0.345**	0.388
Tested on QBC set	0.921	—	**0.898**
Tested on QBC + set	0.725	**0.483**	—

networks on the generated data. For training and testing, we exclude the data points with $r > 5.0$. In contrast to **Section 3.2**, only the forward mapping is modeled, as the inverse relation cannot be learned directly using feedforward neural networks due to its non-uniqueness. For the forward mapping, first, we approximate the orientation distribution function of the generated textures via symmetric generalized spherical harmonics of degree 12 Bunge (2013). The constants $\boldsymbol{D}_i \in \mathbb{R}^{33}$ of this series expansion are used as texture representation (cf. Kalidindi et al., 2004). The feedforward neural networks are supposed to learn the mapping from texture space $\mathcal{D}$ to property space $\mathcal{P}$

$$f\colon \mathcal{D} \to \mathcal{P}, \quad \boldsymbol{p} = f(\boldsymbol{d}), \tag{12}$$

where $\boldsymbol{p} \in \mathcal{P} \subset \mathbb{R}^6$ and $\boldsymbol{d} \in \mathcal{D} \subset \mathbb{R}^{33}$. Each neural network consists of two hidden layers with 30 and 10 neurons with ReLU activation functions. The mean-squared-error loss function is applied and optimized using the limited memory BFGS optimizer. $L2$ regularization with $\lambda = 0.0001$ is applied as well as early stopping using a subset of 10% of the training data for validation.

The performance measure for the neural networks used in this example is the mean absolute error for the Young's moduli

$$\Delta E = \frac{1}{3}\left(|E_{0,\mathrm{pred}} - E_{0,\mathrm{true}}| + |E_{45,\mathrm{pred}} - E_{45,\mathrm{true}}| + |E_{90,\mathrm{pred}} - E_{90,\mathrm{true}}|\right) \tag{13}$$

and for the r-values

$$\Delta r = \frac{1}{3}\left(|r_{0,\mathrm{pred}} - r_{0,\mathrm{true}}| + |r_{45,\mathrm{pred}} - r_{45,\mathrm{true}}| + |r_{90,\mathrm{pred}} - r_{90,\mathrm{true}}|\right). \tag{14}$$

Table 2 and **3** show the results of the trained neural networks, when tested on the generated data. Training and test runs were performed five times with different random validation splits. The mean and maximum errors were averaged over the data set. Both tables show that the neural networks trained with data generated by query-by-committee outperform the neural networks trained with data generated on the basis of Latin hypercube design. However, the differences in the averaged mean errors are not significantly high. In contrast, regarding the averaged maximum errors, the differences are much higher. When tested on the data set generated by Latin hypercube design, both neural networks that are trained with data generated by query-by-committee achieve similar results.

4 DISCUSSION

In **Section 3.1**, an extreme case is studied to emphasize the advantage of using active learning for the generation of microstructure-property data sets. The peak of the approximated delta function is chosen to be quite steep such that the probability of sampling data points on it by random sampling is rather small. As a result, random sampling covers the peak region of the delta function insufficiently. In contrast, by using query-by-committee, the peak region is explored extensively. As pointed out, even the maximum value in the sampled data set is very close to the maximum value of the approximated delta function. If we imagine the delta function expressing a relation between microstructures and properties, we can easily see the advantage for a designer to gain knowledge about the property peak in order to be able to improve the performance of a workpiece.

In **Section 3.2**, the query-by-committee approach is compared to a classical Latin hypercube design approach and a knowledge-based sampling approach for generating data of hardening model parameters and responses. Originally, the knowledge-based approach was developed in Morand and Helm (2019) to optimally sample the hardening model's parameter space by incorporating knowledge about the model's behavior. The results show that the query-by-committee approach is able to find a similar parameter distribution, but without manually introducing any expert knowledge. The data generated by query-by-committee is equally appropriate for training forward and inverse neural network models, which all outperform the models trained on the data generated by the baseline Latin hypercube design approach. All in all, the results show that by using query-by-committee, sampling can be performed automatically in a goal-directed way without additionally introducing expert knowledge.

Also, the results from **Section 3.3** show that the query-by-committee approach is more suitable to sample microstructure-property spaces than classical space-filling sampling strategies. In this example, a space of artificial rolling textures is sampled aiming to efficiently explore the space of corresponding properties. A comparison of the spread of the generated properties point clouds reveals with which additional possibilities a designer can be equipped, when the design space is sampled via active learning. However, the original query-by-committee approach explores the texture space in regions that lead to unrealistic high properties. By using the extension to membership query synthesis that is presented in this paper, sampling in regions with unrealistic high properties can be suppressed. In fact, still some data points are generated in these regions, which are yet necessary for the binary classifier (region filter) to be trained. Nevertheless, compared to the classical query-by-committee approach, the

amount of property data out of scope is much lower and sampling concentrates on the predefined region of interest. Consequently, when training a supervised learning model on the inverse relation (predicting textures for given properties), more extreme properties can be learned on the basis of data generated by query-by-committee.

Such a positive effect can also be observed on learning the forward relation. The neural networks trained on the data generated by query-by-committee both outperform the model trained with the data generated by Latin hypercube design. However, no significant differences in the averaged mean absolute error (shown in **Table 2**) can be seen. This is due to the fact that most of the data points are located near the center of the point cloud, which is where all the neural networks are quite accurate. In contrast, the differences in the averaged maximum errors are more significant. This is because the data sets sampled by query-by-committee contains more extreme data points than the data set sampled by Latin hypercube design. Furthermore, it can be seen that the neural network trained on the data generated by the extended query-by-committee approach performs worse than the network trained with the data from the original approach. This is a sign that the region filter limits the texture space too rigorously and further adjustment is needed. However, the general concept of the active learning extension is proven, as less samples were generated in regions out of scope compared to the original approach.

5 SUMMARY AND OUTLOOK

The present paper shows that active learning can be used to efficiently explore microstructure-property spaces. By using the active learning approach query-by-committee, the focus of data generation is automatically shifted to sparse regions and nonlinearities. Subsequently, two main advantages of active learning in materials design applications follow: 1) regions in microstructure space that lead to extreme properties are explored extensively and 2) in contrast to classical space-filling sampling strategies, active learning can be used for goal-directed sampling, which is relevant for training direct inverse machine learning models. Future work is, however, necessary to investigate how the size of the committee, the fraction of the data used to train the committee members and the complexity of these affect sampling. Also it is necessary to benchmark the query-by-committee approach to the Bayesian approaches, which are mentioned in the introduction.

In general, a problem for active learning approaches arises, when the input space bounds are not set adequately. Then, regions in microstructure space are explored that lead to properties out of scope. However, sampling in these regions can be suppressed by using the extension presented in this work. Still one drawback of using active learning remains: In contrast to classical sampling strategies, active learning is time-intensive, as in every active learning cycle one or more machine learning models need to be trained and additionally an optimization has to be performed. Yet, the results of the present paper show that by using active learning, less data is needed to sufficiently cover microstructure-property spaces than it is the case for classical sampling strategies.

Therefore, regarding virtual materials design, the application of active learning techniques is suitable when sample-efficiency plays an important role. This is for example the case when data is generated using time-intensive numerical simulations, like for example on the bases of spatially resolved full-field microstructures. Also, active learning can help setting up multi-fidelity data bases by enriching less quality data with precisely sampled high quality simulation data or experimental data. Though, incorporating multi-fidelity data and experimental data has not been studied in this work and is part of future research.

DATA AVAILABILITY STATEMENT

The data sets generated for this study can be found in the Fraunhofer Fordatis repository at https://fordatis.fraunhofer.de/handle/fordatis/219, see Morand et al. (2021).

AUTHOR CONTRIBUTIONS

LM set up the active learning framework, developed the extension to membership query synthesis with a region filter, generated the results and wrote the manuscript. NL had the idea to develop the extension to membership query synthesis. TI, JD, and NL supported in terms of machine learning and helped fundamentally by discussing the approach. DH supported in terms of materials sciences and modeling. All authors contributed to the discussion of the results and to the summary and outlook.

ACKNOWLEDGMENTS

The authors would like to thank the German Research Foundation (DFG) for funding this work, which was carried out within the research project number 415 804 944: “Taylored Material Properties *via* Microstructure Optimization: Machine Learning for Modelling and Inversion of Structure-Property-Relationships and the Application to Sheet Metals”.

REFERENCES

Adams, B. L., Henrie, A., Henrie, B., Lyon, M., Kalidindi, S. R., and Garmestani, H. (2001). Microstructure-Sensitive Design of a Compliant Beam. *J. Mech. Phys. Sol.* 49, 1639–1663. doi:10.1016/s0022-5096(01)00016-3

Angluin, D. (1988). Queries and Concept Learning. *Mach.* 2, 319–342. doi:10.1007/bf00116828

Balachandran, P. V., Xue, D., Theiler, J., Hogden, J., and Lookman, T. (2016). Adaptive Strategies for Materials Design Using Uncertainties. *Sci. Rep.* 6 19660–19669. doi:10.1038/srep19660

Bessa, M. A., Bostanabad, R., Liu, Z., Hu, A., Apley, D. W., Brinson, C., et al. (2017). A Framework for Data-Driven Analysis of Materials under Uncertainty:

Countering the Curse of Dimensionality. *Comput. Methods Appl. Mech. Eng.* 320, 633–667. doi:10.1016/j.cma.2017.03.037

Bunge, H.-J. (2013). *Texture Analysis in Materials Science: Mathematical Methods*. Burlington: Elsevier Science.

Burbidge, R., Rowland, J. J., and King, R. D. (2007). "Active Learning for Regression Based on Query by Committee," in Proceeding of the Intelligent Data Engineering and Automated Learning - IDEAL 2007, 8th International Conference Berlin, Heidelberg: Springer, December 16-19, 2007, 209–218. doi:10.1007/978-3-540-77226-2_22

Cai, W., Zhang, Y., and Zhou, J. (2013). "Maximizing Expected Model Change for Active Learning in Regression," in Proceeding of the IEEE 13th International Conference on Data Mining, Dallas, TX, USA, 7-10 Dec. 2013 (IEEE), 51–60. doi:10.1109/ICDM.2013.104

Castillo, A. R., Joseph, V. R., and Kalidindi, S. R. (2019). Bayesian Sequential Design of Experiments for Extraction of Single-crystal Material Properties from Spherical Indentation Measurements on Polycrystalline Samples. *JOM* 71, 2671–2679. doi:10.1007/s11837-019-03549-x

Chalapathy, R., and Chawla, S. (2019). Deep Learning for Anomaly Detection: A Survey. arXiv: 1901.03407 preprint.

Cohn, D. A., Ghahramani, Z., and Jordan, M. I. (1996). Active Learning with Statistical Models. *JAIR* 4, 129–145. doi:10.1613/jair.295

Delannay, L., Van Houtte, P., and Van Bael, A. (1999). New Parameter Model for Texture Description in Steel Sheets. *Textures and Microstructures* 31, 151–175. doi:10.1155/tsm.31.151

Dornheim, J., Morand, L., Zeitvogel, S., Iraki, T., Link, N., and Helm, D. (2022). Deep Reinforcement Learning Methods for Structure-Guided Processing Path Optimization. *J. Intell. Manufacturing* 33, 333–352. doi:10.1007/s10845-021-01805-z

Eghtesad, A., and Knezevic, M. (2020). High-performance Full-Field Crystal Plasticity with Dislocation-Based Hardening and Slip System Back-Stress Laws: Application to Modeling Deformation of Dual-phase Steels. *J. Mech. Phys. Sol.* 134, 103750. doi:10.1016/j.jmps.2019.103750

Fang, K.-T., Lin, D. K. J., Winker, P., and Zhang, Y. (2000). Uniform Design: Theory and Application. *Technometrics* 42, 237–248. doi:10.1080/00401706.2000.10486045

Forrester, A. I. J., and Keane, A. J. (2009). Recent Advances in Surrogate-Based Optimization. *Prog. Aerospace Sci.* 45, 50–79. doi:10.1016/j.paerosci.2008.11.001

Fullwood, D. T., Niezgoda, S. R., Adams, B. L., and Kalidindi, S. R. (2010). Microstructure Sensitive Design for Performance Optimization. *Prog. Mater. Sci.* 55, 477–562. doi:10.1016/j.pmatsci.2009.08.002

Gupta, A., Cecen, A., Goyal, S., Singh, A. K., and Kalidindi, S. R. (2015). Structure-property Linkages Using a Data Science Approach: Application to a Non-metallic Inclusion/steel Composite System. *Acta. Mater.* 91, 239–254. doi:10.1016/j.actamat.2015.02.045

Halton, J. H. (1964). Algorithm 247: Radical-Inverse Quasi-Random point Sequence. *Commun. ACM* 7, 701–702. doi:10.1145/355588.365104

Hammersley, J., and Handscomb, D. (1964). *Monte Carlo Methods*. New York: John Wiley & Sons.

Heese, R., Walczak, M., Seidel, T., Asprion, N., and Bortz, M. (2019). Optimized Data Exploration Applied to the Simulation of a Chemical Process. *Comput. Chem. Eng.* 124, 326–342. doi:10.1016/j.compchemeng.2019.01.007

Helm, D. (2006). Stress Computation in Finite Thermoviscoplasticity. *Int. J. Plasticity* 22, 1699–1727. doi:10.1016/j.ijplas.2006.02.007

Hinton, G. E., and Salakhutdinov, R. R. (2006). Reducing the Dimensionality of Data with Neural Networks. *Science* 313, 504–507. doi:10.1126/science.1127647

Huber, N. (2000). *Anwendung neuronaler Netze bei nichtlinearen Problemen der Mechanik*: Karlsruhe FZKA 2000. Tech. rep., Forschungszentrum Karlsruhe Technik und Umwelt. Habilitation thesis German.

Huber, N., and Tsakmakis, C. (1999). Determination of Constitutive Properties from Spherical Indentation Data Using Neural Networks. Part I: The Case of Pure Kinematic Hardening in Plasticity Laws. *J. Mech. Phys. Sol.* 47, 1569–1588. doi:10.1016/s0022-5096(98)00109-4

Iraki, T., Morand, L., Dornheim, J., Link, N., and Helm, D. (2021). *A Multi-Task Learning-Based Optimization Approach for Finding Diverse Sets of Material Microstructures with Desired Properties and its Application to Texture Optimization*. arXiv:2111.00916.

Johnson, O. K., and Kurniawan, C. (2018). An Efficient Algorithm for Generating Diverse Microstructure Sets and Delineating Properties Closures. *Acta. Mater.* 147, 313–321. doi:10.1016/j.actamat.2018.01.004

Jordan, M. I., and Rumelhart, D. E. (1992). Forward Models: Supervised Learning with a Distal Teacher. *Cogn. Sci.* 16, 307–354. doi:10.1207/s15516709cog1603_1

Jung, J., Yoon, J. I., Park, H. K., Kim, J. Y., and Kim, H. S. (2019). An Efficient Machine Learning Approach to Establish Structure-Property Linkages. *Comput. Mater. Sci.* 156, 17–25. doi:10.1016/j.commatsci.2018.09.034

Kalidindi, S. R. (2019). A Bayesian Framework for Materials Knowledge Systems. *MRS Commun.* 9, 518–531. doi:10.1557/mrc.2019.56

Kalidindi, S. R., Bronkhorst, C. A., and Anand, L. (1992). Crystallographic Texture Evolution in Bulk Deformation Processing of FCC Metals. *J. Mech. Phys. Sol.* 40, 537–569. doi:10.1016/0022-5096(92)80003-9

Kalidindi, S. R., Houskamp, J. R., Lyons, M., and Adams, B. L. (2004). Microstructure Sensitive Design of an Orthotropic Plate Subjected to Tensile Load. *Int. J. Plasticity* 20, 1561–1575. doi:10.1016/j.ijplas.2003.11.007

Krogh, A., and Hertz, J. A. (1992). "A Simple Weight Decay Can Improve Generalization," in *Advances in Neural Information Processing Systems* Denver, 950–957. doi:10.5555/2998687.2998716

Krogh, A., and Vedelsby, J. (1995). Neural Network Ensembles, Cross Validation, and Active Learning. *Adv. Neural Inf. Process. Syst.* 7, 231–238.

Liu, D. C., and Nocedal, J. (1989). On the Limited Memory BFGS Method for Large Scale Optimization. *Math. programming* 45, 503–528. doi:10.1007/bf01589116

Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2008). "Isolation forest," in *8th IEEE International Conference on Data Mining*. Pisa, Italy: IEEE Xplore, 413–422. doi:10.1109/icdm.2008.17

Liu, R., Kumar, A., Chen, Z., Agrawal, A., Sundararaghavan, V., and Choudhary, A. (2015a). A Predictive Machine Learning Approach for Microstructure Optimization and Materials Design. *Sci. Rep.* 5, 11551–11612. doi:10.1038/srep11551

Liu, R., Yabansu, Y. C., Agrawal, A., Kalidindi, S. R., and Choudhary, A. N. (2015b). Machine Learning Approaches for Elastic Localization Linkages in High-Contrast Composite Materials. *Integr. Mater. Manuf.* 4, 192–208. doi:10.1186/s40192-015-0042-z

Lookman, T., Balachandran, P. V., Xue, D., Hogden, J., and Theiler, J. (2017). Statistical Inference and Adaptive Design for Materials Discovery. *Curr. Opin. Solid State. Mater. Sci.* 21, 121–128. doi:10.1016/j.cossms.2016.10.002

Lookman, T., Balachandran, P. V., Xue, D., and Yuan, R. (2019). Active Learning in Materials Science with Emphasis on Adaptive Sampling Using Uncertainties for Targeted Design. *npj Comput. Mater.* 5, 1–17. doi:10.1038/s41524-019-0153-8

Mahnken, R. (2004). *Identification of Material Parameters for Constitutive Equations*. New York: John Wiley & Sons. chap. 19. 637–655.

Mayer, C., and Timofte, R. (2020). "Adversarial Sampling for Active Learning," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass, CO, USA, 1-5 March 2020 (IEEE), 3071–3079. doi:10.1109/WACV45572.2020.9093556

McKay, M. D., Beckman, R. J., and Conover, W. J. (2000). A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics* 42, 55–61. doi:10.1080/00401706.2000.10485979

Morand, L., and Helm, D. (2019). A Mixture of Experts Approach to Handle Ambiguities in Parameter Identification Problems in Material Modeling. *Comput. Mater. Sci.* 167, 85–91. doi:10.1016/j.commatsci.2019.04.003

Morand, L., Iraki, T., Dornheim, J., Link, N., and Helm, D. (2021). Sets of Exemplary Microstructure-Property Data Generated via Active Learning and Numerical Simulations - Fraunhofer Fordatis Repository. Available at: https://fordatis.fraunhofer.de/handle/fordatis/219.

Niederreiter, H. (1988). Low-discrepancy and Low-Dispersion Sequences. *J. Number Theor.* 30, 51–70. doi:10.1016/0022-314x(88)90025-x

Nikolaev, P., Hooper, D., Webber, F., Rao, R., Decker, K., Krein, M., et al. (2016). Autonomy in Materials Research: a Case Study in Carbon Nanotube Growth. *npj Comput. Mater.* 2, 1–6. doi:10.1038/npjcompumats.2016.31

Owen, A. B. (1992). Orthogonal Arrays for Computer Experiments, Integration and Visualization. *Stat. Sinica* 2, 439–452.

Paul, A., Acar, P., Liao, W.-k., Choudhary, A., Sundararaghavan, V., and Agrawal, A. (2019). Microstructure Optimization with Constrained Design Objectives Using Machine Learning-Based Feedback-Aware Data-Generation. *Comput. Mater. Sci.* 160, 334–351. doi:10.1016/j.commatsci.2019.01.015

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine Learning in Python. *J. Machine Learn. Res.* 12, 2825–2830. doi:10.5555/1953048.2078195

Prechelt, L. (1998). "Early Stopping - but when?" in *Neural Networks: Tricks of the Trade* (Berlin, Heidelberg: Springer), 55–69. doi:10.1007/3-540-49430-8_3

RayChaudhuri, T., and Hamey, L. G. C. (1995). "Minimisation of Data Collection by Active Learning," in Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, 27 Nov.-1 Dec. 1995 (IEEE), 1338–1341. doi:10.1109/ICNN.1995.487351

Sacher, M., Duvigneau, R., Le Maître, O., Durand, M., Berrini, É., Hauville, F., et al. (2018). A Classification Approach to Efficient Global Optimization in Presence of Non-computable Domains. *Struct. Multidisc. Optim.* 58, 1537–1557. doi:10.1007/s00158-018-1981-8

Sakurada, M., and Yairi, T. (2014). "Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction," in *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*. Gold Coast: Association for Computing Machinery, 4–11. doi:10.1145/2689746.2689747

Sambu Seo, S., Wallat, M., Graepel, T., and Obermayer, K. (2000). "Gaussian Process Regression: Active Data Selection and Test Point Rejection," in Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks - IJCNN 2000, Como, Italy, 27-27 July 2000. doi:10.1109/IJCNN.2000.861310

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (2001). Estimating the Support of a High-Dimensional Distribution. *Neural Comput.* 13, 1443–1471. doi:10.1162/089976601750264965

Settles, B. (2012). Active Learning. *Synth. Lectures Artif. Intelligence Machine Learn.* 6, 1–114. doi:10.2200/s00429ed1v01y201207aim018

Settles, B. (2009). Active Learning Literature Survey. 1648. Tech. rep. Madison, Wisconsin: University of Wisconsin–Madison, Department of Computer Science.

Seung, H. S., Opper, M., and Sompolinsky, H. (1992). "Query by Committee," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. Pittsburgh: Association for Computing Machinery, 287–294. doi:10.1145/130385.130417

Simpson, T. W., Lin, D. K., and Chen, W. (2001). Sampling Strategies for Computer Experiments: Design and Analysis. *Int. J. Reliability Appl.* 2, 209–240.

Sinha, S., Ebrahimi, S., and Darrell, T. (2019). "Variational Adversarial Active Learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. Seoul: IEEE Xplore, 5972–5981. doi:10.1109/iccv.2019.00607

Sobol, I. M. (1967). On the Distribution of Points in a Cube and the Approximate Evaluation of Integrals. *USSR Comput. Mathematics Math. Phys.* 7, 86–112. doi:10.1016/0041-5553(67)90144-9

Stark, F., Hazırbas, C., Triebel, R., and Cremers, D. (2015). "Captcha Recognition with Active Deep Learning," in *Workshop New Challenges in Neural Computation* 2015, 94.

Storn, R., and Price, K. (1997). Differential Evolution – a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. Glob. Optimization* 11, 341–359. doi:10.1023/a:1008202821328

Tang, B. (1993). Orthogonal Array-Based Latin Hypercubes. *J. Am. Stat. Assoc.* 88, 1392–1397. doi:10.1080/01621459.1993.10476423

Tran, A., Sun, J., Furlan, J. M., Pagalthivarthi, K. V., Visintainer, R. J., and Wang, Y. (2019). pBO-2GP-3B: A Batch Parallel Known/unknown Constrained Bayesian Optimization with Feasibility Classification and its Applications in Computational Fluid Dynamics. *Comput. Methods Appl. Mech. Eng.* 347, 827–852. doi:10.1016/j.cma.2018.12.033

Tran, A., and Wildey, T. (2021). Solving Stochastic Inverse Problems for Property-Structure Linkages Using Data-Consistent Inversion and Machine Learning. *JOM* 73, 72–89. doi:10.1007/s11837-020-04432-w

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., et al. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* 17, 261–272. doi:10.1038/s41592-019-0686-2

Wang, G. G., and Shan, S. (2006). Review of Metamodeling Techniques in Support of Engineering Design Optimization. *J. Mech. Des.* 129, 370–380. doi:10.1115/1.2429697

Wang, K., Zhang, D., Li, Y., Zhang, R., and Lin, L. (2016). Cost-effective Active Learning for Deep Image Classification. *IEEE Trans. Circuits Syst. Video Technology* 27, 2591–2600. doi:10.1109/TCSVT.2016.2589879

Williams, C. K., and Rasmussen, C. E. (2006). *Gaussian Processes for Machine Learning*. Vol. 2. Cambridge, MA: MIT press.

Yagawa, G., and Okuda, H. (1996). Neural Networks in Computational Mechanics. *Arch. Comput. Methods Eng.* 3, 435–512. doi:10.1007/bf02818935

Zhu, J.-J., and Bento, J. (2017). *Generative Adversarial Active Learning*. arXiv: 1702.07956.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers in Materials

ORIGINAL RESEARCH
published: 22 February 2022
doi: 10.3389/fmats.2022.825043

The AiiDA-Spirit Plugin for Automated Spin-Dynamics Simulations and Multi-Scale Modeling Based on First-Principles Calculations

Philipp Rüßmann[1]*, *Jordi Ribas Sobreviela*[1,2], *Moritz Sallermann*[1,2,3], *Markus Hoffmann*[1,2], *Florian Rhiem*[4] and *Stefan Blügel*[1]

[1]Peter Grünberg Institut (PGI-1) and Institute for Advanced Simulation (IAS-1), Forschungszentrum Jülich, Jülich, Germany, [2]RWTH Aachen University, Aachen, Germany, [3]Science Institute and Faculty of Physical Sciences, University of Iceland, VR-III, Reykjavík, Iceland, [4]Peter Grünberg Institut/Jülich Centre for Neutron Science—Technical Services and Administration (PGI/JCNS-TA), Forschungszentrum Jülich, Jülich, Germany

OPEN ACCESS

Edited by:
Simone Taioli,
European Centre for Theoretical Studies in Nuclear Physics and Related Areas (ECT*), Italy

Reviewed by:
Tommaso Morresi,
UMR7590 Institut de Minéralogie, de Physique des Matériaux et de Cosmochimie (IMPMC), France
Michele Casula,
Université Pierre et Marie Curie, France

***Correspondence:**
Philipp Rüßmann
p.ruessmann@fz-juelich.de

Specialty section:
This article was submitted to Computational Materials Science, a section of the journal Frontiers in Materials

Received: 29 November 2021
Accepted: 21 January 2022
Published: 22 February 2022

Citation:
Rüßmann P, Ribas Sobreviela J, Sallermann M, Hoffmann M, Rhiem F and Blügel S (2022) The AiiDA-Spirit Plugin for Automated Spin-Dynamics Simulations and Multi-Scale Modeling Based on First-Principles Calculations. Front. Mater. 9:825043. doi: 10.3389/fmats.2022.825043

Landau-Lifshitz-Gilbert (LLG) spin-dynamics calculations based on the extended Heisenberg Hamiltonian is an important tool in computational materials science involving magnetic materials. LLG simulations allow to bridge the gap from expensive quantum mechanical calculations with small unit cells to large supercells where the collective behavior of millions of spins can be studied. In this work we present the AiiDA-Spirit plugin that connects the spin-dynamics code Spirit to the AiiDA framework. AiiDA provides a Python interface that facilitates performing high-throughput calculations while automatically augmenting the calculations with metadata describing the data provenance between calculations in a directed acyclic graph. The AiiDA-Spirit interface thus provides an easy way for high-throughput spin-dynamics calculations. The interface to the AiiDA infrastructure furthermore has the advantage that input parameters for the extended Heisenberg model can be extracted from high-throughput first-principles calculations including a proper treatment of the data provenance that ensures reproducibility of the calculation results in accordance to the FAIR principles. We describe the layout of the AiiDA-Spirit plugin and demonstrate its capabilities using selected examples for LLG spin-dynamics and Monte Carlo calculations. Furthermore, the integration with first-principles calculations through AiiDA is demonstrated at the example of γ–Fe, where the complex spin-spiral ground state is investigated.

Keywords: spin-dynamics simulation, high-throughput computation, Landau-Lifshitz-Gilbert equation, Monte-Carlo simulation, spin-spiral state, gamma-Fe, skyrmion, antiskyrmion

1 INTRODUCTION

Magnetic materials play an important role in modern technology. Their most important applications range from electrical motors to the storing and processing of digital information. The performance of such applications crucially relies on the performance of magnets where the knowledge of their magnetic order, the Curie temperature, the magnetic hardness or their chirality plays an important role. Computational materials design of magnetic materials and devices is a complex multi-scale

problem. While quantum mechanical calculations allow to predict the interaction strength among magnetic atoms (Liechtenstein et al., 1987), large scale simulations for nanometer to micrometer length scales are unfeasible due to their computational cost. Mapping these interactions to a classical Heisenberg model allows to bridge the scales from the atomic length scale to the length scale of devices. The classical Heisenberg model is an approximation to the quantum mechanical problem which assumes that the magnetic moments are localized on atoms and can be described as classical vectors which is applicable for a wide range of materials.

Spin-dynamics calculations based on the Landau-Lifshitz-Gilbert (LLG) equation (Landau and Lifshitz, 1935; Gilbert, 2004) are a widely used tool for this multi-scale modeling of magnetic materials (Dupé et al., 2014; Hoffmann et al., 2017; Hoffmann et al., 2021; Weißenhofer et al., 2021), providing access to the collective behavior of millions of spins (Müller et al., 2019). This approach allows to find, for instance, the (non-collinear) magnetic ground state based on an energy minimization of the extended Heisenberg Hamiltonian or to study the dynamics of magnetic solitons such as skyrmions (Mühlbauer et al., 2009; Yu et al., 2010; Heinze et al., 2011; Back et al., 2020) or hopfions (Bogolubsky, 1988; Sutcliffe, 2018; Kent et al., 2021) at finite temperature. In combination with the geodesic nudged elastic band method and the harmonic transition state theory (Bessarab et al., 2012; Bessarab et al., 2015) it furthermore gives insight into the stability of aforementioned objects (Müller et al., 2019).

In this work we introduce the AiiDA-Spirit plugin that connects the Spirit code (Müller et al., 2021) to the AiiDA environment (Huber et al., 2020). AiiDA is an open-source Python framework designed around the FAIR principles of findable, accessible, interoperable and reusable data (Wilkinson et al., 2016) in computational science (Pizzi et al., 2016). Calculations that run through the AiiDA infrastructure are automatically stored as nodes in a database together with all inputs and outputs that are necessary to reproduce the simulation results. This results in an directed acyclic graph that can connect different nodes which can be used to reproduce the data provenance from a final result.

In the context of spin-dynamics simulations, a simulation result could be the magnetic ordering obtained from a minimization of the forces on each spin in an LLG calculation. The outcome of such a simulation will in general depend on input parameters such as the geometry (positions of the spins, size of simulation cell, open or periodic boundary conditions), the exchange coupling constants, or applied external fields as well as temperature noise. But also the starting point for the minimization (e.g., starting from an ordered ferromagnet or from random spin orientations) are important as local minima in the energy landscape can generally be present in which metastable states can be stabilized. To ensure reproducible calculation results, keeping track of the full data provenance of a simulation is necessary.

AiiDA's plugin infrastructure allows to orchestrate and combine different sequences of calculations, possibly using different simulation software and methods, through a common interface. Here, we use this to first generate exchange coupling parameters from DFT calculations using the JuKKR code (The JuKKR developers, 2021) with the help of the AiiDA-KKR plugin (Rüßmann et al., 2021a; Rüßmann et al., 2021b). Then, we proceed with spin-dynamics simulations using the Spirit code (Müller et al., 2019; Müller et al., 2021) via the newly developed AiiDA-Spirit plugin (The AiiDA-Spirit developers, 2021). This allows to include the full history of the input parameter generation for spin-dynamics calculations in the provenance graph of a Spirit simulation. Using AiiDA therefore facilitates multi-scale modeling that combines the predictive power of DFT calculations and the speed and scalability of spin-dynamics simulations in the same framework.

The AiiDA engine (Uhrin et al., 2021) provides a highly scalable infrastructure that is able to deal with thousands of calculations simultaneously. Together with the simple Python interface that AiiDA-Spirit provides, spin-dynamics simulations are possible in an automated way which can be used in a high-throughput fashion. This opens new possibilities for applying the Spirit code in automated setups and as part of complex workflows in conjunction with other simulation methods such as DFT. This new capability allows to integrate Spirit in the toolbox of methods that are used in automated computational materials design for magnetic materials (Himanen et al., 2019).

This paper is structured as follows. First the methods section introduces the theory behind spin-dynamics simulations. Then the AiiDA-Spirit plugin is presented which is then applied to 1) a parameter exploration based on a toy model and a large number of high-throughput AiiDA-Spirit calculations, 2) to a simple Monte Carlo example to find the critical temperature of a model system, and 3) multi-scale modelling combining DFT and LLG calculations at the example of γ–Fe. Finally the paper concludes with a discussion of the results.

2 METHODS

2.1 Spirit Theory

All spin-dynamics simulations shown throughout the paper were performed with the Spirit code (Müller et al., 2019; Müller et al., 2021). The Spirit code provides a framework for atomic-scale spin simulations and combines both a graphical user interface as well as an easy accessible Python API. All simulations performed with Spirit are based on an extended Heisenberg Hamiltonian describing the interaction of spins $\vec{S}_i = \vec{M}_i/\mu_i \quad (\mu_i = |\vec{M}_i|)$ sitting at lattice sites i. It can be written in its most general form as

$$\begin{aligned} H = &- \sum_{\langle ij \rangle} J_{ij}\left(\vec{S}_i \cdot \vec{S}_j\right) - \sum_{\langle ij \rangle} \vec{D}_{ij} \cdot \left(\vec{S}_i \times \vec{S}_j\right) \\ &- \sum_i K_\perp (\vec{S}_i \cdot \hat{\vec{e}}_z)^2 - \sum_i \mu_i \vec{B} \cdot \vec{S}_i \\ &- \frac{\mu_0}{8\pi} \sum_{i \neq j} \frac{3\left(\vec{S}_i \cdot \hat{\vec{r}}_{ij}\right)\left(\vec{S}_j \cdot \hat{\vec{r}}_{ij}\right) - \vec{S}_i \cdot \vec{S}_j}{r_{ij}^3} \\ &- \sum_{\langle ijkl \rangle} K_{ijkl}\left(\vec{S}_i \cdot \vec{S}_j\right)\left(\vec{S}_k \cdot \vec{S}_l\right) \quad . \end{aligned} \tag{1}$$

Here, the first line contains the isotropic and antisymmetric exchange interactions, the later also referred to as Dzyaloshinskii-Moriya

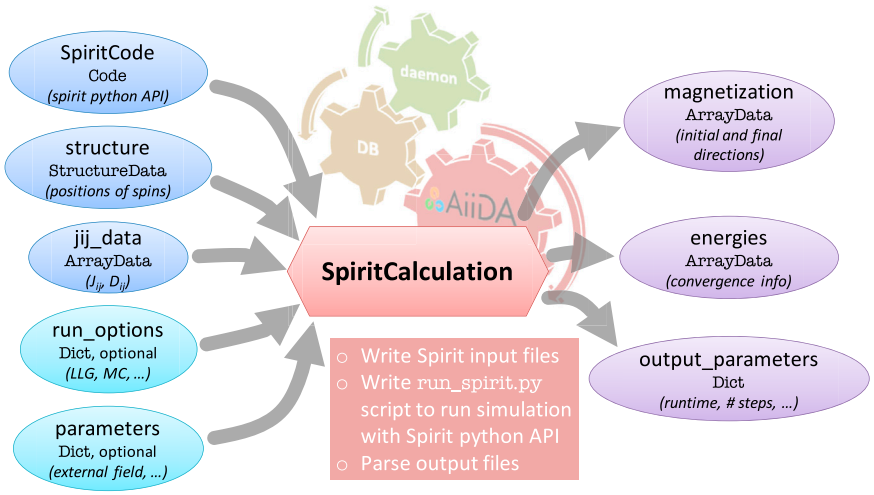


FIGURE 1 | Layout of the SpiritCalculation that is at the heart of the AiiDA-Spirit plugin. On the left hand side the possible input nodes are shown which are translated by the SpiritCalculation into the appropriate input files needed to execute Spirit. The run_options and parameters input nodes are optional that default to a basic LLG calculation starting from random orientation of the spins and without external fields or temperature. The typical output nodes for a LLG calculation are shown on the right hand side.

interaction. The second and third line describe the on-site anisotropy, the Zeeman energy due to an external magnetic field $\vec{B}$, as well as the dipolar contribution. The last term allows to include higher-order exchange interactions (Hoffmann and Blügel, 2020) such as the conventional four-spin or four-spin-four-site interaction (Heinze et al., 2011), the four-spin-three-site interaction (Krönlein et al., 2018), as well as the biquadratic interaction (Szilva et al., 2013). The list of pairs $\langle ij \rangle$ and quadruplets $\langle ijkl \rangle$ as well as their respective parameters J_{ij}, $\vec{D}_{ij}$, and K_{ijkl} can be defined by the user based on the desired use case. Furthermore, the system geometry such as the lattice symmetry and lattice size can be chosen arbitrarily and Spirit allows to introduce defects such as vacancies or atoms of different types. To obtain ground state as well as thermal properties of the investigated system, either the Monte Carlo method based on a Metropolis algorithm or Landau-Lifshitz-Gilbert dynamics can be used.

A more detailed description of the Spirit framework as well as its further functionalities, such as the possibility to calculate lifetimes of magnetic textures based on the combination of geodesic nudged elastic band and harmonic transition state theory calculations, can be found in Ref. (Müller et al., 2019).

2.2 The AiiDA-Spirit Plugin

AiiDA's plugin system allows to combine various simulation codes and methods (to date more than 60 plugins exist already (The AiiDA team, 2021)) on the same footing while augmenting the calculation done through the AiiDA infrastructure with the stored data provenance. Albeit their significance in research on magnetic materials, spin-dynamics calculations have not been at the center of the AiiDA community so far. To the best of our knowledge, besides the AiiDA-Spirit plugin presented here, only a first version of the AiiDA-UppASD plugin (Xu et al., 2021) exists for the UppASD code (Skubic et al., 2008) to combine AiiDA with a spin dynamics simulation engine.

In the context of AiiDA, a calculation plugin needs to be able to generate typical input files that are required to run a calculation through a bash script that will be generated when a calculation is submitted to a computer or as a job on a supercomputer. At the heart of the AiiDA-Spirit plugin lies the SpiritCalculation that connects the Spirit code via the Spirit Python API to AiiDA. The Layout of the SpiritCalculation is shown in **Figure 1**. To run a Spirit calculation a number of input nodes are required:

- a structure node describing the lattice of spins (i.e., their positions in the unit cell),
- an array of the corresponding jij_data that contains the pairwise J_{ij} and $\vec{D}_{ij}$ parameters for the extended Heisenberg Hamiltonian (**Eq. 1**),
- the SpiritCode that is an installation of the Spirit Python API on the computer where the calculation should run,
- and run_options as well as input parameters that control the type of the Spirit run (e.g., LLG or Monte Carlo) or further settings like strength and direction of external fields, respectively.

Additionally, input modes that trigger special features of the Spirit code such as disorder and defects in the structure or pinning of spins to certain directions can be controlled with the corresponding optional input nodes. The SpiritCalculation then implements the functionality to translate this information into the appropriate input files and runs the calculation using the Spirit Python API. The AiiDA daemon automatically takes care of creating a suitable job script, copying necessary input files, and of submitting and monitoring the calculation run. Once the calculation job finishes, important output files are copied back to the retrieved folder in the AiiDA file repository associated to

the AiiDA database. Then, the SpiritParser extracts useful information that should be stored in the database. For the example of a LLG calculation this entails settings such as the number of LLG steps until convergence, the used wall clock time on the computer where the calcualtion ran, an array of the energies (i.e. exchange energy per spin), and the initial and final directions of the spins in the magnetization array.

Apart from the SpiritCalculation and SpiritParser, AiiDA-Spirit comes with some tools that can be used in the typical jupyter notebook environment that is often used in the context of AiiDA. In particular we mention the show_spins tool of AiiDA-Spirit which provides the Spirit visualization capabilities in a simple Python API. This consists of a WebAssembly and WebGL version of the VFRendering library (Vfrendering, 2021) in combination with a JavaScript interface that can be used to visualize the directions of the spins from the web-browser based environment natural to jupyter notebooks.

2.3 DFT-Based Calculation of Exchange Coupling Constants

The density functional theory (DFT) results of this work were produced within the generalized gradient approximation (GGA-PBE) (Perdew et al., 1996) using the full-potential scalar-relativistic Korringa-Kohn-Rostoker Green's function method (KKR) (Ebert et al., 2011) as implemented in the JuKKR code package (The JuKKR developers, 2021). We use an $\ell_{\max} = 3$ cutoff in the angular momentum expansion with an exact description of the atomic cells (Stefanou et al., 1990; Stefanou and Zeller, 1991). After the self-consistent DFT calculations, the method of infinitesimal rotations (Liechtenstein et al., 1987) was used to compute the exchange interaction parameters J_{ij}. The series of DFT calculations in this study are orchestrated using the AiiDA-KKR (Rüßmann et al., 2021a) plugins to the AiiDA infrastructure (Huber et al., 2020). The complete dataset that includes the full provenance of the calculations is made publicly available in the materials cloud repository (Talirz et al., 2020; Rüßmann et al., 2021).

3 RESULTS

3.1 Automated Landau-Lifshitz-Gilbert Calculations for Model Parameter Exploration

To illustrate the usage of AiiDA-Spirit we first consider a toy model consisting of a single layer of spins in a simple-cubic lattice. The complete example is part of the dataset that accompanies this publication (Rüßmann et al., 2021). We assume only nearest neighbor interactions with isotropic exchange $J_1 = 10$ meV and Dzyaloshinskii-Moriya interactions with a strength of $D_1 = 6$ meV. This choice of parameters does not reflect any concrete physical system but is chosen for illustration purposes because it is known to produce skyrmions with small radii. The generation of the corresponding input node for the SpiritCalculation where including the directions of the DMI vectors can be seen in the following code snippet.

TABLE 1 | Parameters for the LLG calculations of the toy model. Arrays are indicated by the square brackets. Except for external_field_magnitude and llg_temperature all parameters are kept fixed in the simulations.

Parameter	Value	Description
n_basis_cells	(50, 50, 1)	Size of the simulation cell
boundary_conditions	(True, True, False)	Periodic boundary conditions
llg_n_iterations	100,000	Number of iterations
llg_damping	0.3	Damping constant
llg_beta	0.1	Non-adiabatic damping
llg_dt	0.001	Time step dt (ps)
llg_force_convergence	10^{-7}	Force convergence parameter
llg_temperature	0…75	Temperature (K)
external_field_magnitude	− 50…50	Magnitude of the external field
external_field_normal	(0.0, 0.0, 1.0)	Direction of the external field
mu_s	(2.0)	Spin moment (μ_B)

```
jijs_expanded = np.array([
    #i  j da db dc  Jij   Dx   Dy   Dz
    [0, 0, 1, 0, 0, 10.0, 6.0, 0.0, 0.0],
    [0, 0, 0, 1, 0, 10.0, 0.0, 6.0, 0.0],
])
jij_data = aiida.orm.ArrayData()
jij_data.set_array('Jij_expanded', jijs_expanded)
```

Here i, j index the lattice site in the unit cell (situated at $\vec{r}_i$, $\vec{r}_j$) and da, db, dc describe offsets into unit cells further away such that $\vec{R}_{ij} = \vec{r}_j - \vec{r}_i + d_a\vec{a} + d_b\vec{b} + d_c\vec{c}$ describes the distance between two spins ($\vec{a}, \vec{b}, \vec{c}$ are the Bravais vectors of the lattice), Jij and Dx, Dy, Dz denote the exchange interaction and the three components of the DMI vector. For this example we consider an isolated layer of spins with periodic boundary conditions in the plane. We choose a supercell for the SpiritCalculation of 50 × 50 × 1 spins. Furthermore we apply an external field of various strength (in the code snippet we show the input parameters for a value of 25 T) in the direction perpendicular to the film in the following code snippet.

```
parameters = aiida.orm.Dict(
    dict={
        'n_basis_cells': [50, 50, 1],
        'boundary_conditions': [True, True, False],
        'external_field_magnitude': 25.0,
        'external_field_normal': [0.0, 0.0, 1.0],
        'mu_s': [2.0], # one value per spin in the unit cell
    })
```

Starting from random orientations of the spins we then perform a time evolution using the LLG method with the Depondt solver (Depondt and Mertens, 2009). The parameters for the LLG calculations are summarized in **Table 1**.

To harness the high-throughput capabilities of the AiiDA-Spirit plugin we perform a series of SpiritCalculations to screen a range of external fields and temperatures. We change the temperature from 0 to 75 K in 2.5 K steps and vary the external field from − 50 T to + 50 T in steps of 2.5 T. The calculations for each parameters set are repeated 5 times starting from different random orientations of the spins for statistical averaging. This amounts to 31 × 41 × 5 = 6,355

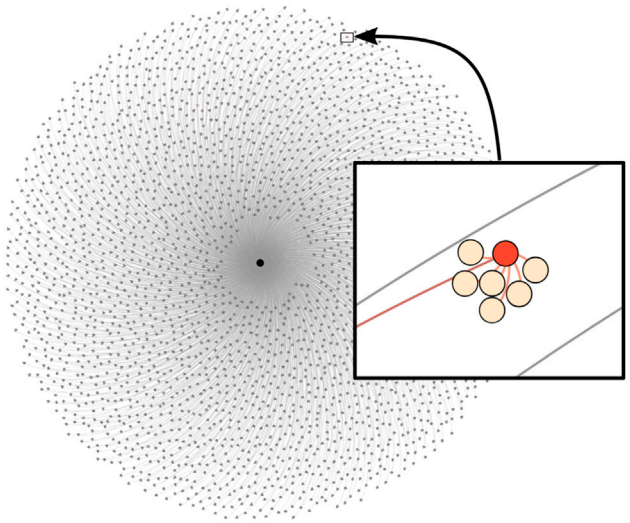


FIGURE 2 | Provenance graph of the SpiritCalculations discussed in **section 3.1**. The graph consists of several thousand calculations that all use the same crystal structure as input (the black circle in the center) with which they are connected. The inset shows a magnified view of one of these calculations (red circle) which is connected to outgoing nodes (colored in light orange).

individual SpiritCalculations that were submitted to an in-house compute cluster. We stress that the AiiDA daemon (Uhrin et al., 2021) conveniently takes care of creating submission scripts and automatically retrieves and parses the outcome of the calculations without the need for any user interaction. A visualization of the dataset and the provenance graph for this application is shown in **Figure 2**.

In order to analyze the outcome of the SpiritCalculations we chose to investigate the topological charge in the simulation cell at the end of the LLG simulation. For a continuous vector field $\vec{m}$ it is defined as

$$\rho_T = \frac{1}{4\pi}\int \vec{m}\cdot\left(\frac{\partial\vec{m}}{\partial x}\times\frac{\partial\vec{m}}{\partial y}\right)\mathrm{d}x\mathrm{d}y\,. \tag{2}$$

We added a custom post-processing step to the SpiritCalculation which uses the get_topological_charge function of the spirit Python API. This function calculates the topological charge from the discretized form of **Eq. 2** as a summation over all contributions of triangles formed by neighboring spins in the simulation cell (Müller et al., 2019).

Figure 3 shows the outcome of these simulations where the topological charge is shown for all 1,271 pairs (T, B_z) together with selected spin configurations of representative calculations marked by the symbols (**Figures 3B–G**). The real-space spin configuration at the end of the LLG calculations were visualized using the show_spins tool of the AiiDA-Spirit plugin. It can be seen that a small external field leads to the appearance of skyrmions which in this case have a topological charge of ± 1, depending whether they form in a ferromagnetic background of spins pointing in $-z$ (**Figure 3C**) or $+z$ (**Figure 3E**) direction. In general, the topological charge counts the difference between

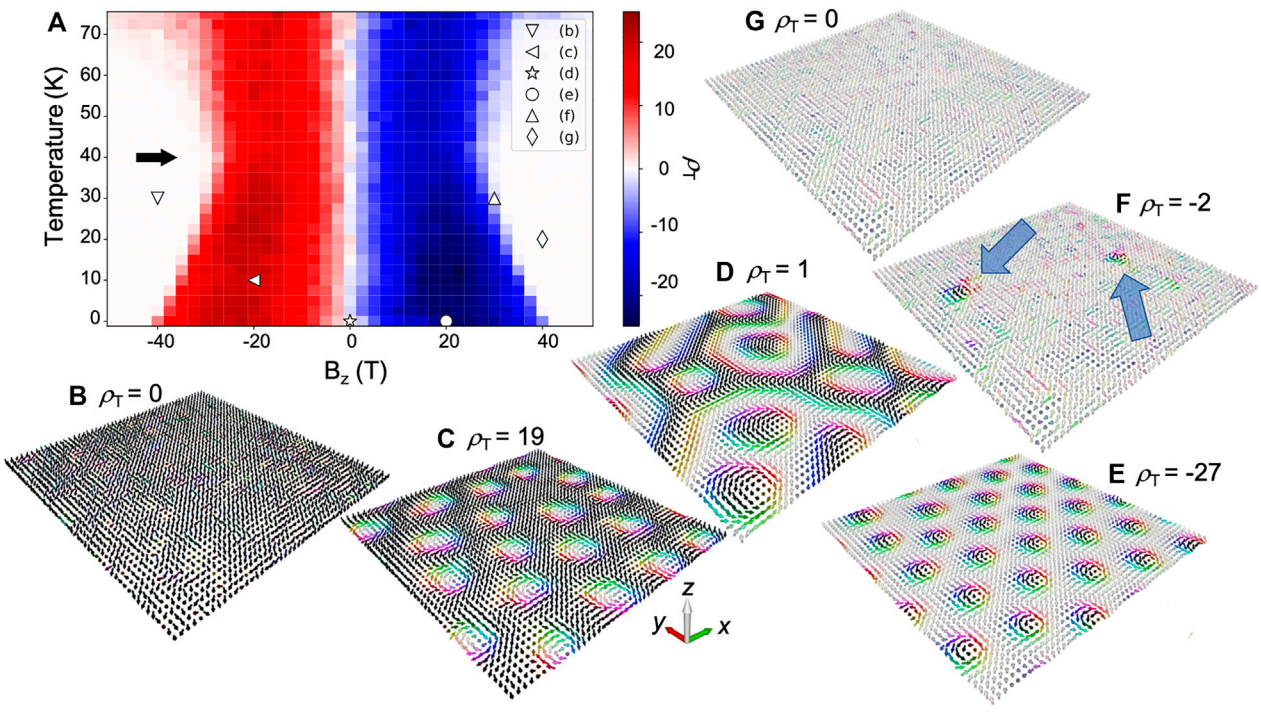


FIGURE 3 | Topological charge of the toy model discussed in the text. **(A)** Dependence of the topological charge ρ_T on the external magnetic field and temperature calculated from the final spin texture after an LLG calculation. The black arrow highlights the inflection point where $|\rho_T(B_z)|$ has a minimum with respect to B_z. For the parameters marked by the symbols **(B–G)** the resulting spin textures are shown in the corresponding panels. The arrows in **(F)** highlight the two skyrmions that result in a topological charge of $\rho_T = -2$.

TABLE 2 | Parameters for the MC calculations of the simple-cubic ferromagnet discussed in the text. Note that the chosen settings result in temperature steps of 0.5 K.

Parameter	Value	Description
n_thermalisation	5,000	Number of thermalization steps before n_samples are taken
n_samples	250,000	Number of samples taken in metropolis algorithm
n_decorrelation	2	Number of decorrelation steps
n_temperatures	40	Number of temperature steps
T_start	25	Start of temperature scanning range
T_end	5	End of temperature scanning range

the amount of skyrmions in up-domains ($\rho_T > 0$) and skyrmions in down-domains ($\rho_T < 0$) as seen for vanishing external field in (**Figure 3D**) where several skyrmions with opposite topological charges lead to a near cancellation of the total topological charge.

At very large fields, the Zeeman exchange coupling term becomes larger than the DMI energy and a homogeneous ferromagnet forms (**Figures 3B,G**). Temperature fluctuations tend to deform the skyrmions (**Figure 3F**) and destabilize them. Thus, at elevated temperatures smaller magnitudes of the external magnetic field leads to a vanishing topological charge. As highlighted by the black arrow in (**Figure 3A**), this is however only true up to a certain critical temperature. For $T > 40$ K the topological charge increases again which can be explained by the energy barrier of skyrmion formation and destruction. At these elevated temperatures the fluctuations of the spin directions are larger than $40\,\mathrm{K} \cdot k_B \approx 3.45$ meV which we conjecture is the energy barrier for skyrmion formation. While the energy barrier can in principle be calculated by performing geodesic nudged elastic band calculations, this is beyond the scope of this paper and therefore will be omitted. The larger temperature fluctuations also prohibit reaching the force convergence criterion set in the LLG calculation which means that the LLG simulation runs until the maximal simulation time of 100 ps is reached. During this simulation time, skyrmions can spontaneously form and disappear which results in a finite topological charge measured at the end of the run. In the future the real time dynamics of skyrmion creation and collapse may be the focus of the investigation. However, this approach may become unfeasible for situations where the skyrmion lifetime is very long compared to the typical time step in LLG calculations. Finally, we highlight that with increasing temperature fluctuations we also find a larger variance in the number of skyrmions when averaging over the five different starting configuration for each pair (T, B_z). This supports our interpretation that skyrmions are spontaneously created and annihilated by temperature fluctuations.

3.2 Curie Temperature Using Monte Carlo

The Monte Carlo (MC) method is a well established tool in physics which, when applied to spin systems, allows to estimate the critical temperature of the magnetic ordering (Curie temperature) (Binder and Heermann, 1997). The Spirit code (Müller et al., 2019) implements a Metropolis algorithm which can be used from AiiDA-Spirit by choosing the mc simulation method (instead of the previously used LLG method). We demonstrate the MC at the example of a simple-cubic ferromagnet with only nearest neighbor interactions $J_1 = 1$ meV. We perform calculations for varying supercell sizes between 10 × 10 × 10 and 40 × 40 × 40 with the MC parameters given in **Table 2**. The results of the calculation are shown in **Figure 4** where, together with the total magnetization M, the isothermal susceptibility

$$\chi = \frac{1}{k_B T}\left(\langle M^2\rangle - \langle M\rangle^2\right) \tag{3}$$

with $M = |\frac{1}{N}\sum_i \vec{S}_i|$ the average magnetization of the sample is shown. We see that $M(T)$ converges with increasing supercell size indicating that boundary effects become less important. The corresponding susceptibilities show a diverging behavior at T_c. Our calculation results agree well with the expected value of $T_c = 1.44 J_1/k_B = 16.71$ K. We stress that these calculations require a series of steps consisting of, for example, thermalization and decorrelation steps at each temperature value in the scanning interval. Within AiiDA-Spirit this complexity is conveniently absorbed in the SpiritCalculation which greatly facilitates the application of MC calculations.

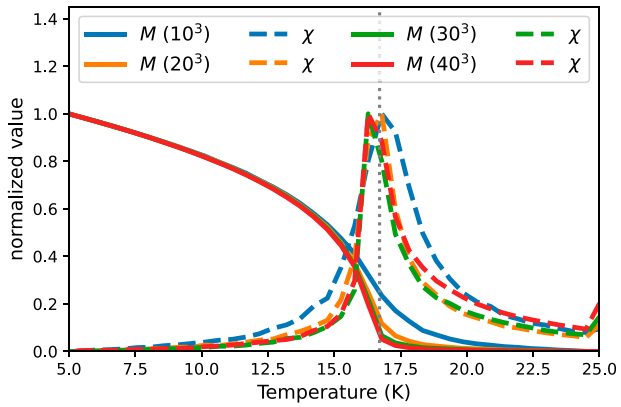


FIGURE 4 | Results of the Monte Carlo calculations for a simple-cubic ferromagnetic with nearest neighbor $J_1 = 1$ meV exchange interactions. Shown are results for 10 × 10 × 10 to 40 × 40 × 40 supercells where the solid lines show the normalized value of the total magnetization M and the dashed lines the corresponding susceptibility χ. The dashed vertical line indicate the expected value of the critical temperature at $T_c = 16.71$ K.

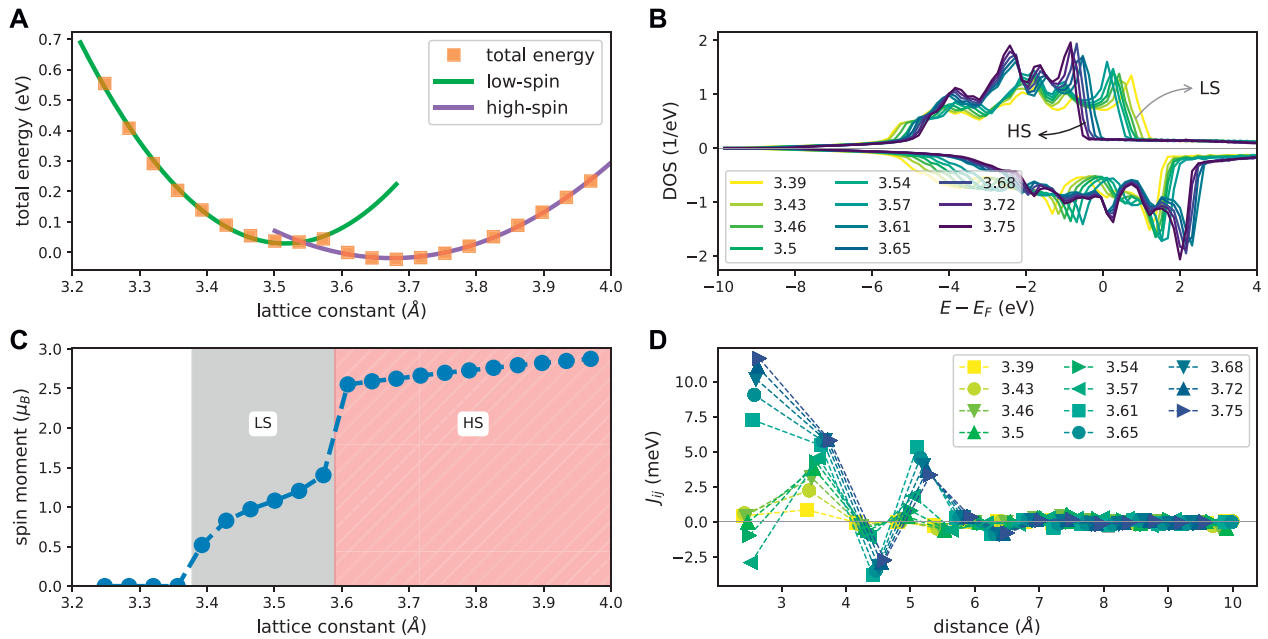


FIGURE 5 | Results of the DFT calculations for γ–Fe. Total energy as a function of the lattice constant **(A)** where the green and violet lines show parabolic fits to low-spin ($a_{lat} < 3.6$Å) and high-spin ($a_{lat} \geq 3.6$Å) states. **(B)** Corresponding density of states (value in the legend indicates the lattice constant) which clearly separates into low-spin (LS) with smaller exchange splitting and high-spin (HS) states with larger exchange splitting. Note that the positive (negative) values of the DOS indicate the majority (minority) spin channels. **(C)** Spin moment as a function of the lattice constant where the transition from non-magnetic (first four data points) over the low-spin state to the high-spin state is evident. **(D)** Resulting exchange coupling constants (J_{ij}) as a function of the pairwise distance between the Fe atoms in γ–Fe (the lattice constant is given in the legend).

3.3 Multi-Scale Modeling: γ–Fe

We now demonstrate how the integration of the Spirit code into the AiiDA framework through the AiiDA-Spirit plugin can facilitate multi-scale modeling for magnetic materials. In this example we first calculate the exchange interaction parameters for γ–Fe using density functional theory which are then passed to the AiiDA-Spirit plugin for LLG simulations.

The γ phase of Fe is a metastable high-temperature phase where the atoms crystallize in the fcc lattice (Knöpfle et al., 2000; Sjöstedt and Nordström, 2002). This has a drastic consequence on the exchange interactions where, in contrast to the ferromagnetic bcc Fe, frustrated exchange interactions can lead to the formation of spin-spirals. Experimentally this structure of Fe can be realized in a Cu matrix (Tsunoda, 1989; Tsunoda et al., 1993). It is known that a variation of the lattice constant of γ–Fe can have drastic consequences for the magnetic ordering (Sjöstedt and Nordström, 2002). Here, we investigate bulk crystals of γ–Fe for varying lattice constants between $a_{lat} = 3.2$ Å and $a_{lat} = 4.0$ Å around the lattice constant of Cu ($a_{lat} = 3.6$ Å).

Figure 5 summarizes the results of the DFT calculations that were done with the AiiDA-KKR plugin (see methods section for numerical details). The total energy as a function of the lattice constant (shown in panel **Figure 5A**) reveals a phase transition from the low-spin state (for $a_{lat} < 3.6$ Å) to the high-spin state ($a_{lat} \geq 3.6$ Å) of γ–Fe as seen in the jump of the spin moment to $\mu_s > 2.5\ \mu_B$. This coincides with a smaller exchange splitting seen in the density of states (**Figure 5B**) and consequently a smaller value of the spin moment (**Figure 5C**). For lattice constants below 3.37 Å we find that the magnetic moment vanishes. Panel (**Figure 5D**) shows the calculated exchange interactions J_{ij} as a function of distance between two Fe atoms. Clearly, the sign of the nearest neighbor interaction shows the most drastic change with the transition from high-spin to low-spin state at smaller lattice constant of γ–Fe. While in the high-spin state the first and second nearest neighbor interaction are both ferromagnetic ($J_{ij} > 0$), for the low-spin state the nearest neighbor interaction changes from being weakly ferromagnetic to antiferromagnetic ($J_{ij} < 0$).

In the following, the consequences of this change for the magnetic ordering are investigated based on a series of LLG calculations using the AiiDA-Spirit plugin. In the DFT calculation we use the primitive cell which contains a single atom in the unit cell. For the spirit calculations we map the calculated exchange interactions onto the conventional unit cell consisting of four atoms. The parameters of the LLG simulations are summarized in **Table 3**. We study the magnetic ordering in a $40 \times 40 \times 40 \times 4 = 256{,}000$ spins supercell without external magnetic fields and at temperature $T = 0$ K. Here we focus on the ground state that forms and therefore neglect effects of temperature fluctuations and external fields which can, for example if T is high enough, overcome the energy barrier between different (metastable) magnetic orderings. We further neglect the influence of anisotropy ($K_{\perp} = 0$) in this work and we also do not include higher order exchange terms ($K_{i,j,k,l} = 0$). We choose open boundary conditions in order to not bias the eventually resulting spin-spiral wavelength by the periodicity of the supercell. **Table 4** summarizes the DFT calculated

TABLE 3 | Parameters for the LLG calculations for γ–Fe. Note that the simulation cell consists of 40 × 40 × 40 × 4 = 256,000 atoms due to the choice of the conventional unit cell with four atoms. The spin moment μ is extracted from the DFT calculation at the respective lattice constant and open boundary conditions are chosen. Parameters not listed here are set to the same value as in **Table 1**.

Parameter	Value	Description
n_basis_cells	(40, 40, 40)	Size of the simulation cell
boundary_conditions	(False, False, False)	Open boundary conditions
llg_temperature	0	Temperature (K)
external_field_magnitude	0	Magnitude of the external field
mu_s	(μ, μ, μ, μ)	Spin moment (μ_B)

TABLE 4 | Input parameters extracted from DFT that are used in the SpiritCalculations for γ–Fe for different lattice constants a_{lat} (given in Å). Listed are the magnetic moment μ (in μ_B per spin) and the exchange coupling parameters J_{ij} for the first seven shells (denoted J_1 to J_7) which are given in meV.

a_{lat}	μ	J_1	J_2	J_3	J_4	J_5	J_6	J_7
3.39	0.52	0.42	0.85	–0.04	–0.03	–0.27	0.00	0.05
3.43	0.83	0.60	2.23	–0.17	0.03	–0.48	0.00	0.04
3.46	0.97	0.42	3.15	–0.31	0.16	–0.57	–0.04	0.00
3.50	1.08	–0.05	3.79	–0.45	0.38	–0.61	–0.07	–0.04
3.54	1.20	–0.98	4.28	–0.64	0.80	–0.62	–0.11	–0.10
3.57	1.40	–2.92	4.56	–1.11	1.83	–0.59	–0.09	–0.16
3.61	2.55	7.27	5.46	–3.82	5.33	0.29	–0.87	0.12
3.65	2.59	9.05	5.75	–3.48	4.57	0.29	–0.87	0.12
3.68	2.62	10.19	5.85	–3.19	4.05	0.30	–0.86	0.12
3.72	2.66	11.03	5.84	–2.96	3.67	0.32	–0.83	0.11
3.75	2.70	11.66	5.79	–2.75	3.33	0.35	–0.80	0.10

values for magnetic moments and exchange coupling constants for varying lattice constants that were used in the respective SpiritCalculations. Note that the exchange coupling constants are only shown up to the seventh shell of neighbors but the calculations included pairs up to the 15th shell that are however much smaller than the values reported in **Table 4**.

We start the discussion of the LLG calculations with the results for γ–Fe in the lattice constant of Cu ($a_{\mathrm{lat}} = 3.61$ Å). **Figure 6A** shows the resulting spin texture at the end of the LLG calculation for the central layer of spins in the yz-plane (shown in the inset **Figure 6B**). We can see that a spin-spiral forms in z-direction with ferromagnetically ordered spins in y-direction. At the open boundaries we see that the missing neighbors on one side influence the magnetic ordering that deviates from the spin-spiral in the center for a distance of about five lattice constants. In order to quantify the spin-spiral we pick the two cardinal directions in this plane (indicated by blue and orange lines in **Figure 5A**) and extract the z-component of the spin S_z. We combine the projections onto the yz-plane from two adjacent layers of spins (indicated by the two grey planes in **Figure 5B**) to not restrict our analysis to a single sub-lattice only. This allows to describe also antiferromagnetic structures in the sub-lattices with ferromagnetic ordering within one sub-lattice, which will be important later. **Figure 6C** shows that, except for boundary effects, S_z stays constants when following the y-direction. Along the z-axis we see a complex beating pattern with the site index that can be decomposed into two π/2-shifted spin-spirals in the two different sub-lattices. The corresponding Fourier transformation

$$\mathcal{F}_j(q) = \frac{1}{\sqrt{2\pi}} \int e^{-iqr_j} S_z(r_j)\, \mathrm{d}r_j \tag{4}$$

with $j = y, z$ computed with the fast Fourier transform algorithm (FFT) is shown in (**Figure 6D**). As expected, the FFT of the predominantly ferromagnetically ordered spins along the y-direction $|\mathcal{F}_y|$ mainly shows a signal at $q = 0$ whereas $|\mathcal{F}_z|$ shows the appearance of four peaks at $q \approx \pm 0.2\, \frac{2\pi}{a_{\mathrm{lat}}}$ and $q \approx \pm 0.8\, \frac{2\pi}{a_{\mathrm{lat}}} = (1 - 0.2)\, 2\pi/a_{\mathrm{lat}}$ which are attributed to the two π/2-shifted oscillations in the two sub-lattices. We point out that the reflection symmetry around $q = 0$ is a consequence of the real-valued input to the FFT and is therefore not discussed further. As seen from the FFT in y-direction and from the corresponding spin texture in (**Figure 6A**) the spins in the direction perpendicular to the propagation direction of the spin spiral (i.e., the z-direction in this example) are ordered ferromagnetically. Therefore the spin-spiral wavevector is $\vec{q} = (0, 0, 0.2)\, 2\pi/a_{\mathrm{lat}}$. After having characterized the spin-spiral ground state of γ–Fe we continue with a discussion of the magnetic ordering depending on the changing exchange coupling parameters with changing lattice constant.

Figure 7 summarizes the LLG calculations for γ–Fe for varying lattice constants using the respective set of exchange parameters shown in **Figure 5D**. The lines in **Figure 7A** show the energy at the end of the LLG calculation starting either from a random spin configuration (E, dashed orange) or from the ferromagnetic (E_{FM}, solid blue) state. We find that for lattice constants $a_{\mathrm{lat}} \geq 3.65$ Å the ferromagnetic state minimizes the energy ($E - E_{\mathrm{FM}} = 0$). We attribute this to the increasing ferromagnetic interaction for nearest neighbor spins in the high-spin state which was discussed with **Figure 5D**. At smaller lattice constants ($a_{\mathrm{lat}} \leq 3.6$ Å), the ferromagnet (FM) is not the ground state anymore. Here we find either a spin-spiral (SS) ground state or an antiferromagnetic (AFM) phase. **Figure 7C** shows three representative images of the SS, AFM and FM states.

In the SS state the magnetization rotates from left to right (i.e. along the y-axis) and shows antiparallel alignment of the rows in z-direction. Along the x-axis (direction perpendicular to the drawn plane) the spins are aligned ferromagnetically, except for boundary effects at the open ends of the simulation cell (seen in the direction of the first layer of spins). In z-direction, adjacent layers are antiferromagnetically ordered. Thus the spin-spiral wavevector for these lattice constants has the

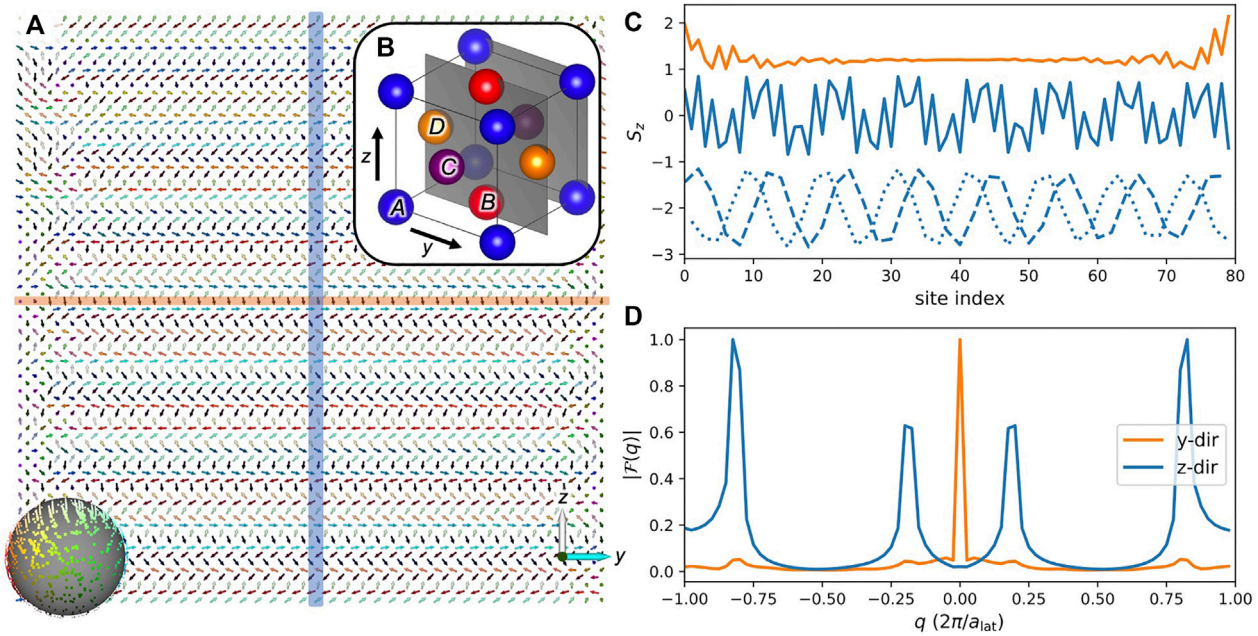


FIGURE 6 | Spin-spiral ground state for γ–Fe at a lattice constant of a_{lat} = 3.61 Å. **(A)** Cut through the central *yz*-plane of the 40 × 40 × 40 supercell. Each unit cell consists of 4 Fe atoms on the sub-lattices *A*, *B*, *C* and *D* shown in the inset **(B)**. The colored points on the grey sphere in the lower left of panel **(A)** show a projection of the direction of the spins onto the unit sphere. **(C)** *z*-component of the spin along the *y*- (orange line) and *z*-directions (solid blue line) indicated in **(A)** with the corresponding colored stripes. Each direction combines the spins from two adjacent *yz*-planes to cover more than a single sub-lattice per direction (i.e. sites in *y*-direction consist of atoms *A*-*B*-*A*-. . .). The dashed and dotted lines in **(C)** are a decomposition of $S_z(z)$ into the sub-lattices *A* and *D*. Note that the orange and dashed and dotted blue lines are shifted by ± 2 with respect to the solid blue line. **(D)** Magnitude of the Fourier transformation of S_z along the *y*- and *z*-directions, i.e. solid orange and blue lines in **(C)**, respectively.

form $\vec{q} = (0, q, 1)\, 2\pi/a_{lat}$ which due to the cubic symmetry of the crystal is equivalent to $\vec{q} = (q, 0, 1)\, 2\pi/a_{lat}$. Note that $\vec{q}_{AF} = (0, 0, 1)\, 2\pi/a_{lat}$ is the antiferromagnet because the distance between two layers in the (0, 0, 1) direction of the fcc lattice is $a_{lat}/2$.

In the AFM phase the direction of the spins separate into four sub-lattices that correspond to the four atoms in the conventional fcc unit cell. Within each sub-lattice the spins are aligned parallel and form a right angle with their neighboring spins from different sub-lattices. This is highlighted with a red box in the middle panel of (**Figure 7C**). In the FM phase (lower panel) all spins point in the same direction. Note that in all these calculations the spins can collectively rotate since we neglected contributions from single-ion anisotropies and do not apply an external field.

The summed magnitude of the Fourier transform along the three cardinal axes

$$|\mathcal{F}(q)| = \sum_{j=x,y,z} |\mathcal{F}_j(q)| \tag{5}$$

is shown in **Figure 7B** for different lattice constants. Note that we have summed here over the symmetry-equivalent directions along the *x*- *y*- and *z*-directions because of the rotational invariance of the complete spin-structure. Starting from the smallest lattice constant of a_{lat} = 3.39 Å we see a peak in $|\mathcal{F}(q)|$ at $\vec{q} = (\pm\, 0.195, 0, 1)\, 2\pi/a_{lat}$ which corresponds to a wavelength of the spin-spiral of 5.13 a_{lat}. Here we focus our discussion on the peak at smaller q values as the same arguments hold for the second peak at $2\pi/a_{lat} - q$ as discussed above. With increasing lattice constant the spin-spiral wavelength increases to 7.82 $a_{lat}/2$ ($\vec{q} = (\pm\, 0.128, 0, 1)\, 2\pi/a_{lat}$) at a lattice constant of a_{lat} = 3.5 Å. For 3.5 Å < a_{lat} < 3.6 Å the AFM state is found, which in the Fourier transform is characterized by the dominating peak at $q = \pm 2\pi/a_{lat}$. Note that we still get a considerable signal at $q = 0$ because we sum over all three cardinal directions and there is ferromagnetic ordering along one direction (see **Figure 7C**). As discussed above, for larger lattice constants the spin-spiral state briefly shows up again at a_{lat} = 3.61 Å with a wavelength of 5.59 a_{lat} which however has ferromagnetically ordered spins in both directions perpendicular to the direction of spin-spiral propagation ($\vec{q} = (\pm\, 0.179, 0, 0)\, 2\pi/a_{lat}$) until for $a_{lat} \geq 3.65$ Å the FM state is found which only shows a significant Fourier amplitude at $q = 0$.

The appearance of the AFM phase for 3.5 Å < a_{lat} < 3.6 Å can be attributed to the sign change of the nearest neighbor interaction from ferro- to antiferromagnetic. To verify this hypothesis we employ a series of LLG calculations through AiiDA-Spirit. We chose to start from the setup of the calculation for a_{lat} = 3.61 Å, which was found to reproduce the spin-spiral phase. We then modify the strength of the nearest neighbor interaction J_1 ranging from − 5 meV to + 15 meV and run LLG calculations starting from the FM state, from the AFM state, from the SS phase and random spin orientations. For the AFM state we construct the row-wise AFM orientation of the spins. All other parameters for the LLG calculation are kept constant. In total this is another set of 84

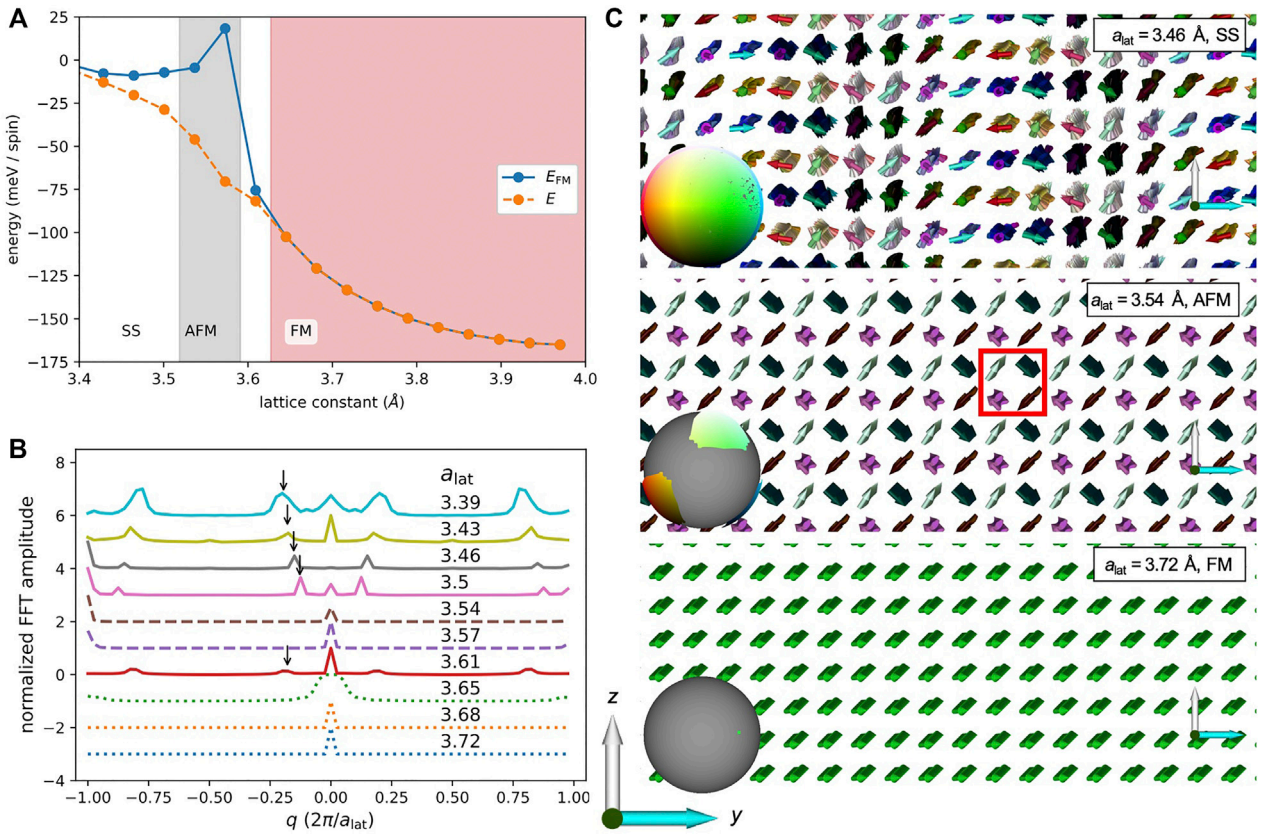


FIGURE 7 | Magnetic ground state in γ–Fe from spin-dynamics simulations via the AiiDA-Spirit plugin. **(A)** Calculated energies per spin of the final state after an LLG calculation. Each data point uses the exchange constants computed from DFT (cf. **Figure 5D**). The solid blue and dashed orange lines indicate the energy computed starting from random spin configuration or the ferromagentic (FM) state. The white, grey and red shaded areas indicate if, respectively, a spin-spiral (SS), antiferromagnetic (AFM) or FM ordering is found to be the ground state. **(B)** Normalized Fourier transform of the z-component of the magnetization in the yz-plane for different lattice constants, shifted for clarity. Solid lines correspond to SS, dashed lines to AFM and dotted lines to FM solutions, respectively. The arrows highlight the principal wavenumber of the spin-spiral and highlight their change with the lattice constant. **(C)** Visualization of representative spin structures for (from top to bottom) SS, AFM and FM states where the red box in the AFM structure highlights a unit cell with the four sub-lattices. The colored points on the grey sphere in the lower left panels show a projection of the direction of the spins onto the unit sphere.

SpiritCalculations where we find that starting from a random spin configuration coincides with starting from the SS. The random starting point is therefore omitted in the following discussion. **Figures 8A,B** show the dependence of the energy per spin at the end of the LLG calculations. We point out that the LLG calculations for the FM and AFM states are converged in the very first iteration which indicates that the FM and AFM phases are local minima in the energy landscape. We find that the SS state is lowest in energy with a maximal energy gain of ~ 7 meV/spin in the transition region where $E_{FM} - E_{AFM}$ changes sign. From the final spin structure of the spin-spiral solution we proceed with an analysis of the Fourier components as introduced in equations 4 and 5. This is shown in **Figure 8C**. As highlighted by the grey line, we see an increase in the spin-spiral wavevector with increasing J_1 up to the point where E_{FM} and E_{AFM} cross around $J_1 = 7.3$ meV. As in the previous analysis for changing lattice constant we find that the spin-spiral state is characterized by two wavevectors at q and $2\pi/a_{lat} - q$ (grey dashed line). Furthermore, the Fourier transform for all states show significant signals at $q = 0$ (indicating parallel spins) and $q = \pm 2\pi/a_{lat}$ (indicating antiparallel spins).

Overall we can conclude that the resulting spin-texture in the 256,000 spin unit cell with open boundary conditions is a result of the complex competition of distance-dependent exchange couplings that favor ferromagnetic or antiferromagnetic alignments of spins or can compete and stabilize spin-spiral ground states.

4 DISCUSSION

In this article we have presented the AiiDA-Spirit plugin that connects the spin-dynamics code Spirit to the AiiDA framework. AiiDA enables high-throughput calculations while automatically keeping track of the data provenance (Huber et al., 2020). We have demonstrated the capabilities of the AiiDA-Spirit plugin with three examples; 1) high-throughput spin-dynamics calculations based on the Landau-Lifshitz-Gilbert (LLG)

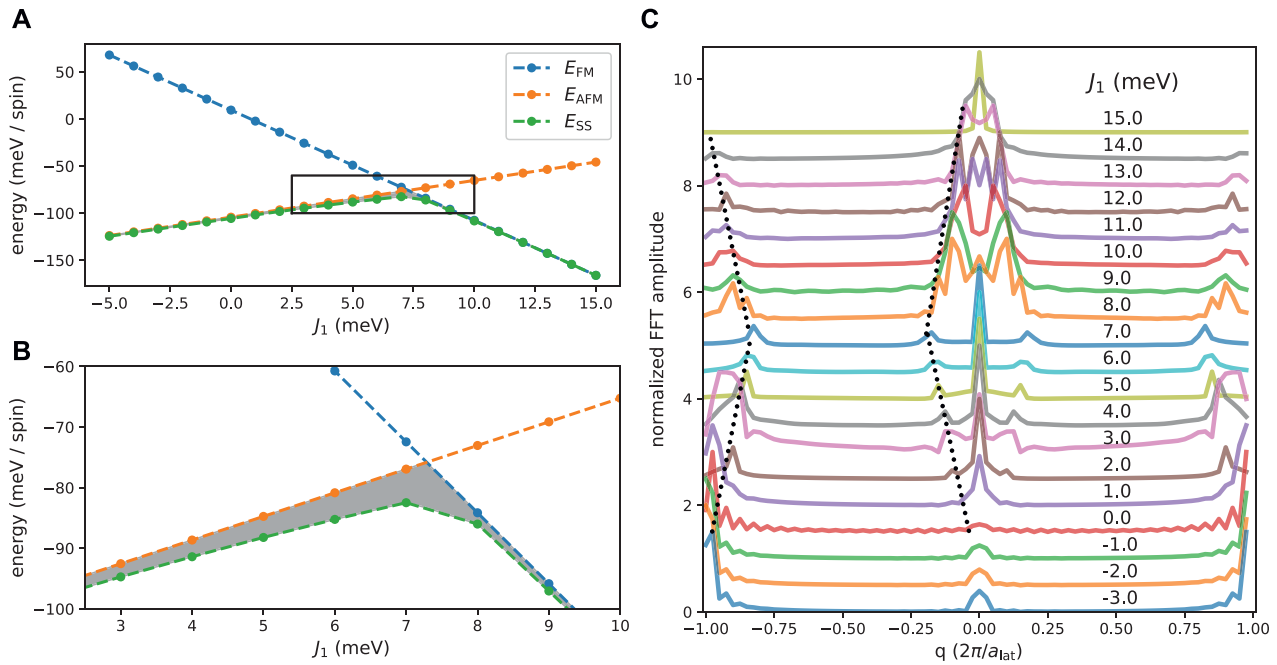


FIGURE 8 | (A) Spin-spiral (SS) energies as a function of the nearest neighbor interaction J_1 in comparison to ferromagnetic (FM) and antiferromagnetic (AFM) states. Exchange coupling constants were taken from the a_{lat} = 3.61 Å calculation and the nearest neighbor couplings J_1 were modified in the range from – 5 to 15 meV. **(B)** Zoom into the transition region marked by the black box in **(A)**. The grey area highlights the energy gain in the SS state compared to FM or AFM states. **(C)** Normalized Fourier transform of the *z*-component of the magnetization in the *yz*-plane. The spectra are shifted for clarity. Dashed lines indicate where $E_{AFM} - E_{SS} < 2$ meV and dotted lines are used for $E_{FM} - E_{SS} < 2$ meV. The solid and dashed grey lines are guides to the eye highlighting the change in the spin-spiral wavevector with J_1.

equation for a toy model that shows skyrmions, 2) Monte Carlo calculations for finding the critical temperature of a simple-cubic model ferromagnet, and 3) multi-scale modelling combining density functional calculations with spin-dynamics simulations for γ–Fe.

In our high-throughput LLG calculations we performed more than 6,000 simulations of a model system consisting of a 2D lattice of spins in the simple-cubic lattice. The model parameters were chosen such that topologically nontrivial skyrmions appear in the magnetic textures. We varied the temperature and the external magnetic field as external parameters and investigate the change in the topological charge, which is a measure of the number of skyrmions that appear in the system. We find that, starting from $T = 0$, the transition to the homogeneous ferromagnetic phase happens at lower magnetic fields. At a certain critical temperature however the number of skyrmions starts increasing again. We interpret this as the surpassing of the energy barrier for skyrmion formation which can be overcome by temperature fluctuations of the spins. This goes hand in hand with a larger variance in the topological charge that we measure from averaging multiple runs for each pair of (T, B_z). These calculations demonstrate the possibility to employ the AiiDA-Spirit plugin for high-throughput spin-dynamics simulations which make parameter exploration easier accessible.

In our Monte Carlo calculations we showed how the complex series of calculations necessary for finding the ordering temperature of a simple-cubic ferromagnet (several calculations across the transition region from ferromagnetically ordered to paramagnetic state have to be performed) can be found from a single SpiritCalculation of the AiiDA-Spirit plugin. Our simulation result is in good agreement with the theoretically expected result. The ease-of-use for these calculations facilitate the incorporation of AiiDA-Spirit calculations in complex workflows in materials informatics for magnetic materials. Here, finding the critical temperature of a magnetic material is a very common problem.

Finally, we discussed the use case of LLG calculations for the study of the magnetic ordering of γ–Fe, which is the high-temperature fcc phase of Fe. From experiments, where Fe clusters were embedded in a Cu matrix, it is known that a spin-spiral ground state with wavevector $\vec{q} = (0.1, 0, 1)\, 2\pi/a_{lat}$ is found for γ–Fe around the lattice constant of Cu (Tsunoda, 1989; Tsunoda et al., 1993). Note that $q = (0, 0, 2\pi/a_{lat})$ correspond to the antiferromagnet since the distance between two layers in the (0, 0, 1) direction of the fcc lattice is $a_{lat}/2$. Theoretically, this wavevector was reproduced from first-principles calculations with good agreement where $\vec{q} = (0.15, 0, 1)\, 2\pi/a_{lat}$ (Knöpfle et al., 2000) and $\vec{q} = (0.16, 0, 1)\, 2\pi/a_{lat}$ (Sjöstedt and Nordström, 2002) were found. However, a significantly different lattice constant compared to the lattice constant of Cu is required for γ–Fe in the calculation compared to the experiments, which makes this agreement unsatisfactory (Sjöstedt and Nordström, 2002). In our work, instead of looking for the spin-spiral energies from first-principles calculations, we chose to explore the predictive power of a combination of DFT and LLG calculations for

γ–Fe with changing lattice constants. We found a change from the low-spin to high-spin ground state in our DFT results that were performed with the JuKKR code (The JuKKR developers, 2021) through the AiiDA-KKR plugin (Rüßmann et al., 2021a; Rüßmann et al., 2021b). This agrees well with earlier DFT calculations where a similar change in the spin moment of the Fe atoms from $\mu \sim 1\ \mu_B$ to $> 2.5\,\mu_B$ is seen (Knöpfle et al., 2000; Sjöstedt and Nordström, 2002).

In contrast to the spin-spiral energy calculations of Refs. (Knöpfle et al., 2000; Sjöstedt and Nordström, 2002) we calculate the exchange parameters for the extended Heisenberg model from the method of infinitesimal rotations (Liechtenstein et al., 1987) around the collinear, ferromagnetically ordered state. These parameters are then used in the SpiritCalculations where the collective magnetic ordering is investigated in a 256,000 spin supercell. We find a strong influence of exchange interactions on the lattice constant of γ–Fe which results in a competition of ferromagnetic, antiferromagnetic and spin-spiral orderings. In our analysis of the spin-spiral wavevectors we chose to study the Fourier transform of the z-component of the spin around the three cardinal axes which are symmetry-equivalent in our approach. At the lattice constant of Cu ($a_{lat} = 3.6$ Å) we find a spin-spiral with $\vec{q} = (0.2, 0, 0)\, 2\pi/a_{lat}$ in contrast to the spin-spiral $\vec{q}_{exp} = (0.1, 0, 1)\, 2\pi/a_{lat}$ found experimentally which has an antiferromagnetic component (Tsunoda, 1989). We attribute this discrepancy to neglecting the change in the spin moment with the spin-spiral wavevector in our simulations based on the Heisenberg Hamiltonian. This change is known to be significant and can be as large as $0.8\ \mu_B$ (Sjöstedt and Nordström, 2002). For lattice constants $a_{lat} \leq 3.5$ we do find the correct spin-spiral with $\vec{q} = (q, 0, 1)\, 2\pi/a_{lat}$ that reappears after the spin-spiral with ferromagnetic ordering perpendicular to $\vec{q}$ transforms into the ordered antiferromagnet. With smaller lattice constant the spin-spiral wavevector increases from $\vec{q} = (0.13, 0, 1)\, 2\pi/a_{lat}$ at $a_{lat} = 3.5$ Å up to $\vec{q} = (0.2, 0, 1)\, 2\pi/a_{lat}$ at $a_{lat} = 3.39$ Å which is in reasonable agreement with earlier calculation results (Knöpfle et al., 2000; Sjöstedt and Nordström, 2002). Incorporating a change of the spin moment with the spin-spiral wavevector might further improve our agreement to the earlier *ab initio* results of Knöpfle *et al.* (Knöpfle et al., 2000) and Sjöstedt and Nordström (Sjöstedt and Nordström, 2002) and also the experimental spin-spiral wavevector (Tsunoda, 1989). The need to include spin moment change is also observed in the high-pressure ε-phase of iron (Lebert et al., 2019). Furthermore, including higher order exchange interactions ($K_{i,j,k,l} \neq 0$) could also be important. Especially for magnetically frustrated systems, those additional terms can be essential in describing the magnetic ground state as seen, for example, in iron chalcogenides where the biquadratic term is required for a correct description based on the Heisenberg model (Glasbrenner et al., 2015).

The change of the ordering to the antiferromagnetic state and then the reappearance of the spin-spiral state at even smaller lattice constant compared to the lattice constant of Cu on the other hand agrees well with the previously stated observation of competing magnetic orders, which are very close in energy and could coexist (Sjöstedt and Nordström, 2002). We further demonstrated the sensitivity of the magnetic ordering with a numerical experiment where we chose to modify the strength of the nearest neighbor exchange interaction J_1. The resulting strong change in the spin-spiral wavevector and the magnetic ordering highlights the rich energy landscape that is underlying the complex magnetic ordering in γ–Fe.

In conclusion, we have shown how augmenting spin-dynamics calculations with the Spirit code through the AiiDA-Spirit plugin enables high-throughput spin-dynamics simulations via the AiiDA infrastructure. This was applied to model systems and, in combination with DFT calculations through the AiiDA-KKR plugin, to the multi-scale problem of the magnetic ordering in γ–Fe. Our results demonstrate that typical spin-dynamics simulations benefit from the possibility to run a large number of calculations in a high-throughput fashion. Automation of SpiritCalculations through AiiDA can be a great asset when complex model parameter spaces (i.e. external fields, temperatures, different geometries, ...) are screened in order to find structure-property relations of magnetic materials. The feature of AiiDA to keep track of the data provenance is here indispensable to get reproducible results and to eventually engineer recipes for the creation and control of unconventional topological solitons in magnetic structures such as skyrmions or hopfions in the future.

DATA AVAILABILITY STATEMENT

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found below: https://archive.materialscloud.org/record/2021.203 Materials Cloud Archive 2021.203 (2021). doi: 10.24435/materialscloud:9s-tx.

AUTHOR CONTRIBUTIONS

PR and JRS programmed the first version of AiiDA-Spirit and MS and FR contributed to the further development of the plugin where FR was responsible for the spin_view functionality of AiiDA-Spirit. PR performed the DFT calculations and PR and MS performed the AiiDA-Spirit calculations. All authors discussed the results and contributed in writing the manuscript.

FUNDING

We acknowledge support by the Joint Lab Virtual Materials Design (JL-VMD) and thank for computing time granted by the JARA Vergabegremium (project number jara0191) and provided on the JARA Partition part of the supercomputer CLAIX at RWTH Aachen University. This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy—Cluster of Excellence Matter and Light for Quantum Computing (ML4Q) EXC 2004/1—390534769.

REFERENCES

Back, C., Cros, V., Ebert, H., Everschor-Sitte, K., Fert, A., Garst, M., et al. (2020). The 2020 Skyrmionics Roadmap. *J. Phys. D: Appl. Phys.* 53, 363001. doi:10.1088/1361-6463/ab8418

Bessarab, P. F., Uzdin, V. M., and Jónsson, H. (2012). Harmonic Transition-State Theory of thermal Spin Transitions. *Phys. Rev. B* 85, 184409. doi:10.1103/physrevb.85.184409

Bessarab, P. F., Uzdin, V. M., and Jónsson, H. (2015). Method for Finding Mechanism and Activation Energy of Magnetic Transitions, Applied to Skyrmion and Antivortex Annihilation. *Comput. Phys. Commun.* 196, 335–347. doi:10.1016/j.cpc.2015.07.001

Binder, K., and Heermann, D. W. (1997). *Monte Carlo Simulation in Statistical Physics*. Berlin: Springer-Verlag.

Bogolubsky, I. L. (1988). Three-dimensional Topological Solitons in the Lattice Model of a Magnet with Competing Interactions. *Phys. Lett. A* 126, 511–514. doi:10.1016/0375-9601(88)90049-7

Depondt, P., and Mertens, F. G. (2009). Spin Dynamics Simulations of Two-Dimensional Clusters with Heisenberg and Dipole-Dipole Interactions. *J. Phys. Condens. Matter* 21, 336005. doi:10.1088/0953-8984/21/33/336005

Dupé, B., Hoffmann, M., Paillard, C., and Heinze, S. (2014). Tailoring Magnetic Skyrmions in Ultra-thin Transition Metal Films. *Nat. Commun.* 5, 4030–4036. doi:10.1038/ncomms5030

Ebert, H., Ködderitzsch, D., and Minár, J. (2011). Calculating Condensed Matter Properties Using the KKR-Green's Function Method-Recent Developments and Applications. *Rep. Prog. Phys.* 74, 096501. doi:10.1088/0034-4885/74/9/096501

Gilbert, T. L. (2004). Classics in Magnetics A Phenomenological Theory of Damping in Ferromagnetic Materials. *IEEE Trans. Magn.* 40, 3443–3449. doi:10.1109/tmag.2004.836740

Glasbrenner, J. K., Mazin, I. I., Jeschke, H. O., Hirschfeld, P. J., Fernandes, R. M., and Valentí, R. (2015). Effect of Magnetic Frustration on Nematicity and Superconductivity in Iron Chalcogenides. *Nat. Phys* 11, 953–958. doi:10.1038/nphys3434

Heinze, S., Von Bergmann, K., Menzel, M., Brede, J., Kubetzka, A., Wiesendanger, R., et al. (2011). Spontaneous Atomic-Scale Magnetic Skyrmion Lattice in Two Dimensions. *Nat. Phys* 7, 713–718. doi:10.1038/nphys2045

Himanen, L., Geurts, A., Foster, A. S., and Rinke, P. (2019). Data-Driven Materials Science: Status, Challenges, and Perspectives. *Adv. Sci.* 6, 1900808. doi:10.1002/advs.201900808

Hoffmann, M., and Blügel, S. (2020). Systematic Derivation of Realistic Spin Models for Beyond-Heisenberg Solids. *Phys. Rev. B* 101, 024418. doi:10.1103/physrevb.101.024418

Hoffmann, M., Zimmermann, B., Müller, G. P., Schürhoff, D., Kiselev, N. S., Melcher, C., et al. (2017). Antiskyrmions Stabilized at Interfaces by Anisotropic Dzyaloshinskii-Moriya Interactions. *Nat. Commun.* 8, 308–309. doi:10.1038/s41467-017-00313-0

Hoffmann, M., Müller, G. P., Melcher, C., and Blügel, S. (2021). Skyrmion-antiskyrmion Racetrack Memory in Rank-One Dmi Materials. *Front. Phys.* 9, 668. doi:10.3389/fphy.2021.769873

Huber, S. P., Zoupanos, S., Uhrin, M., Talirz, L., Kahle, L., Häuselmann, R., et al. (2020). AiiDA 1.0, a Scalable Computational Infrastructure for Automated Reproducible Workflows and Data Provenance. *Sci. Data* 7, 300. doi:10.1038/s41597-020-00638-4

Kent, N., Reynolds, N., Raftrey, D., Campbell, I. T. G., Virasawmy, S., Dhuey, S., et al. (2021). Creation and Observation of Hopfions in Magnetic Multilayer Systems. *Nat. Commun.* 12, 1562–1567. doi:10.1038/s41467-021-21846-5

Knöpfle, K., Sandratskii, L. M., and Kübler, J. (2000). Spin Spiral Ground State of γ-Iron. *Phys. Rev. B* 62, 5564–5569. doi:10.1103/PhysRevB.62.5564

Krönlein, A., Schmitt, M., Hoffmann, M., Kemmer, J., Seubert, N., Vogt, M., et al. (2018). Magnetic Ground State Stabilized by Three-Site Interactions: Fe/Rh(111). *Phys. Rev. Lett.* 120, 207202. doi:10.1103/physrevlett.120.207202

Landau, L., and Lifshitz, E. (1935). On the Theory of the Dispersion of Magnetic Permeability in Ferromagnetic Bodies. *Phys. Z. Sowjet.* 851, 153.

Lebert, B. W., Gorni, T., Casula, M., Klotz, S., Baudelet, F., Ablett, J. M., et al. (2019). Epsilon Iron as a Spin-Smectic State. *Proc. Natl. Acad. Sci. USA* 116, 20280–20285. doi:10.1073/pnas.1904575116

Liechtenstein, A. I., Katsnelson, M. I., Antropov, V. P., and Gubanov, V. A. (1987). Local Spin Density Functional Approach to the Theory of Exchange Interactions in Ferromagnetic Metals and Alloys. *J. Magnetism Magn. Mater.* 67, 65–74. doi:10.1016/0304-8853(87)90721-9

Müller, G. P., Sallermann, M., Mavros, S., Rhiem, F., Schürhoff, D., Meyer, I., et al. (2021). Spirit: Spin Simulation Software. Available at: https://github.com/spirit-code/spirit (Accessed November 25, 2021).

Müller, G. P., Hoffmann, M., Dißelkamp, C., Schürhoff, D., Mavros, S., Sallermann, M., et al. (2019). Spirit : Multifunctional Framework for Atomistic Spin Simulations. *Phys. Rev. B* 99, 224414. doi:10.1103/PhysRevB.99.224414

Mühlbauer, S., Binz, B., Jonietz, F., Pfleiderer, C., Rosch, A., Neubauer, A., et al. (2009). Skyrmion Lattice in a Chiral Magnet. *Science* 323, 915–919. doi:10.1126/science.1166767

Perdew, J. P., Burke, K., and Ernzerhof, M. (1996). Generalized Gradient Approximation Made Simple. *Phys. Rev. Lett.* 77, 3865–3868. doi:10.1103/PhysRevLett.77.3865

Pizzi, G., Cepellotti, A., Sabatini, R., Marzari, N., and Kozinsky, B. (2016). AiiDA: Automated Interactive Infrastructure and Database for Computational Science. *Comput. Mater. Sci.* 111, 218–230. doi:10.1016/j.commatsci.2015.09.013

Rüßmann, P., Bertoldo, F., Bröder, J., Wasmer, J., Mozumder, R., Chico, J., et al. (2021). *JuDFTteam/aiida-kkr: AiiDA Plugin for the JuKKR Codes*. doi:10.5281/zenodo.3628251

Rüßmann, P., Bertoldo, F., and Blügel, S. (2021). The AiiDA-KKR Plugin and its Application to High-Throughput Impurity Embedding into a Topological Insulator. *Npj Comput. Mater.* 7, 13. doi:10.1038/s41524-020-00482-5

Rüßmann, P., Ribas Sobreviela, J., Sallermann, M., Hoffmann, M., Rhiem, F., and Blügel, S. (2021b). The AiiDA-Spirit Plugin for Automated Spin-Dynamics Simulations and Multi-Scale Modelling Based on First-Principles Calculations. *Mater. Cloud Archive*, 203. doi:10.24435/materialscloud:9s-tx

Sjöstedt, E., and Nordström, L. (2002). Noncollinear Full-Potential Studies of γ−Fe. *Phys. Rev. B* 66, 014447. doi:10.1103/PhysRevB.66.014447

Skubic, B., Hellsvik, J., Nordström, L., and Eriksson, O. (2008). A Method for Atomistic Spin Dynamics Simulations: Implementation and Examples. *J. Phys. Condens. Matter* 20, 315203. doi:10.1088/0953-8984/20/31/315203

Stefanou, N., Akai, H., and Zeller, R. (1990). An Efficient Numerical Method to Calculate Shape Truncation Functions for Wigner-Seitz Atomic Polyhedra. *Comput. Phys. Commun.* 60, 231–238. doi:10.1016/0010-4655(90)90009-P

Stefanou, N., and Zeller, R. (1991). Calculation of Shape-Truncation Functions for Voronoi Polyhedra. *J. Phys. Condens. Matter* 3, 7599–7606. doi:10.1088/0953-8984/3/39/006

Sutcliffe, P. (2018). Hopfions in Chiral Magnets. *J. Phys. A: Math. Theor.* 51, 375401. doi:10.1088/1751-8121/aad521

Szilva, A., Costa, M., Bergman, A., Szunyogh, L., Nordström, L., and Eriksson, O. (2013). Interatomic Exchange Interactions for Finite-Temperature Magnetism and Nonequilibrium Spin Dynamics. *Phys. Rev. Lett.* 111, 127204. doi:10.1103/physrevlett.111.127204

Talirz, L., Kumbhar, S., Passaro, E., Yakutovich, A. V., Granata, V., Gargiulo, F., et al. (2020). Materials Cloud, a Platform for Open Computational Science. *Sci. Data* 7, 299. doi:10.1038/s41597-020-00637-5

The AiiDA-Spirit developers (2021). The Aiida-Spirit Plugin. Available at: https://github.com/JuDFTteam/aiida-spirit (Accessed November 25, 2021)

The AiiDA team (2021). *AiiDA Plugin Registry*. Available at: https://aiidateam.github.io/aiida-registry/ (Accessed November 25, 2021).

The JuKKR developers (2021). The Jülich KKR Codes. Available at: https://jukkr.fz-juelich.de (Accessed November 25, 2021).

Tsunoda, Y., Nishioka, Y., and Nicklow, R. M. (1993). Spin Fluctuations in Small γ-Fe Precipitates. *J. Magnetism Magn. Mater.* 128, 133–137. doi:10.1016/0304-8853(93)90867-2

Tsunoda, Y. (1989). Spin-density Wave in Cubic γ-Fe and $\gamma Fe_{100-x}Co_x$ precipitates in Cu. *J. Phys. Condens. Matter* 1, 10427–10438. doi:10.1088/0953-8984/1/51/015

Uhrin, M., Huber, S. P., Yu, J., Marzari, N., and Pizzi, G. (2021). Workflows in AiiDA: Engineering a High-Throughput, Event-Based Engine for Robust and Modular Computational Workflows. *Comput. Mater. Sci.* 187, 110086. doi:10.1016/j.commatsci.2020.110086

Vfrendering (2021). A Vector Field Rendering Library. Available at: https://github.com/FlorianRhiem/VFRendering (Accessed November 25, 2021).

Weißenhofer, M., Rózsa, L., and Nowak, U. (2021). Skyrmion Dynamics at Finite Temperatures: Beyond Thiele's Equation. *Phys. Rev. Lett.* 127, 047203. doi:10.1103/PhysRevLett.127.047203

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., et al. (2016). The FAIR Guiding Principles for Scientific Data Management and Stewardship. *Sci. Data* 3, 160018. doi:10.1038/sdata.2016.18

Xu, Q., Bergman, A., Delin, A., and Chico, J. (2021). The UppASD-AiiDA Plugin. Available at: https://github.com/UppASD/aiida-uppasd (Accessed November 25, 2021).

Yu, X. Z., Onose, Y., Kanazawa, N., Park, J. H., Han, J. H., Matsui, Y., et al. (2010). Real-space Observation of a Two-Dimensional Skyrmion crystal. *Nature* 465, 901–904. doi:10.1038/nature09124

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers in Materials

ORIGINAL RESEARCH
published: 28 February 2022
doi: 10.3389/fmats.2022.837006

Training Deep Neural Networks to Reconstruct Nanoporous Structures From FIB Tomography Images Using Synthetic Training Data

Trushal Sardhara [1], Roland C. Aydin [2], Yong Li [3], Nicolas Piché [4], Raynald Gauvin [5], Christian J. Cyron [1,2] and Martin Ritter [6]*

[1]Institute for Continuum and Material Mechanics, Hamburg University of Technology, Hamburg, Germany, [2]Institute of Material Systems Modeling, Helmholtz-Zentrum Hereon, Geesthacht, Germany, [3]Institute of Materials Physics and Technology, Hamburg University of Technology, Hamburg, Germany, [4]Object Research Systems, Montreal, QC, Canada, [5]Department of Mining and Materials Engineering, McGill University, Montreal, QC, Canada, [6]Electron Microscopy Unit, Hamburg University of Technology, Hamburg, Germany

Focused ion beam (FIB) tomography is a destructive technique used to collect three-dimensional (3D) structural information at a resolution of a few nanometers. For FIB tomography, a material sample is degraded by layer-wise milling. After each layer, the current surface is imaged by a scanning electron microscope (SEM), providing a consecutive series of cross-sections of the three-dimensional material sample. Especially for nanoporous materials, the reconstruction of the 3D microstructure of the material, from the information collected during FIB tomography, is impaired by the so-called *shine-through effect*. This effect prevents a unique mapping between voxel intensity values and material phase (e.g., solid or void). It often substantially reduces the accuracy of conventional methods for image segmentation. Here we demonstrate how machine learning can be used to tackle this problem. A bottleneck in doing so is the availability of sufficient training data. To overcome this problem, we present a novel approach to generate synthetic training data in the form of FIB-SEM images generated by Monte Carlo simulations. Based on this approach, we compare the performance of different machine learning architectures for segmenting FIB tomography data of nanoporous materials. We demonstrate that two-dimensional (2D) convolutional neural network (CNN) architectures processing a group of adjacent slices as input data as well as 3D CNN perform best and can enhance the segmentation performance significantly.

Keywords: electron microscopy, synthetic training data, 3D reconstruction, semantic segmentation, SEM simulation, 3D CNN, 2D CNN with adjacent slices, machine learning

OPEN ACCESS

Edited by:
Surya R. Kalidindi, Georgia Institute of Technology, United States

Reviewed by:
Stefan G. Stanciu, Politehnica University of Bucharest, Romania
David Montes De Oca Zapiain, Sandia National Laboratories, United States

****Correspondence:***
Trushal Sardhara
trushal.sardhara@tuhh.de

Specialty section:
This article was submitted to Computational Materials Science, a section of the journal Frontiers in Materials

Received: *16 December 2021*
Accepted: *27 January 2022*
Published: *28 February 2022*

Citation:
Sardhara T, Aydin RC, Li Y, Piché N, Gauvin R, Cyron CJ and Ritter M (2022) Training Deep Neural Networks to Reconstruct Nanoporous Structures From FIB Tomography Images Using Synthetic Training Data. Front. Mater. 9:837006. doi: 10.3389/fmats.2022.837006

1 INTRODUCTION

Nanoporous materials bear great potential in microtechnology, chemical engineering, biomedical engineering, energy technology and electronics and communication technology. So-called FIB tomography combines the sequential removal of material layers by FIB with SEM imaging. It is a powerful technique for 3D imaging of nanoporous materials with a resolution of approx. 1 nm in the SEM plane and 10 nm in the out-of-plane direction (Knott et al., 2008).

However, accurate 3D reconstruction of nanoporous structures remains a challenge because of the so-called *shine-through effect* in FIB tomography data (Prill et al., 2013). Due to this effect, the

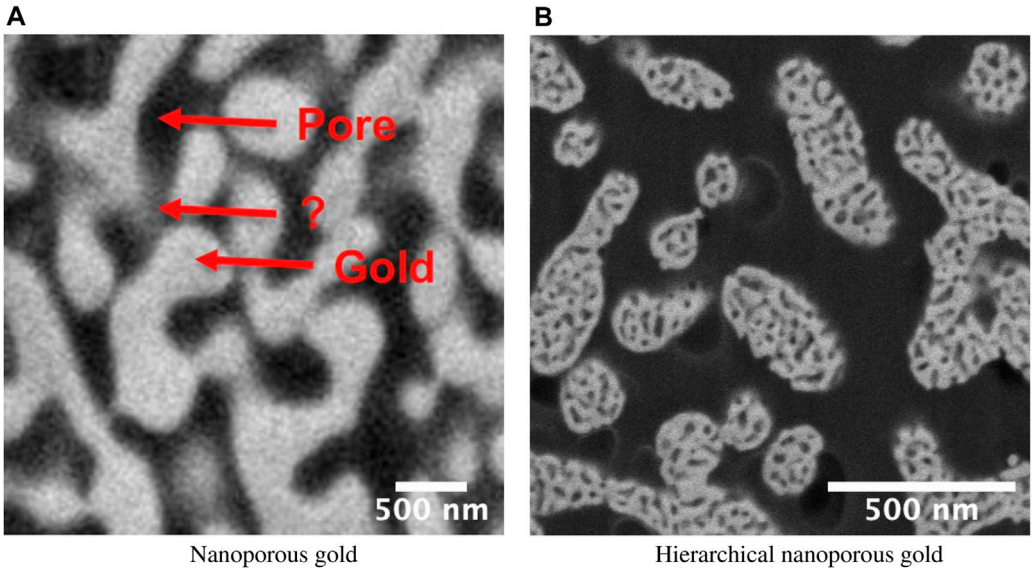


FIGURE 1 | Back-scattered electron scanning microscope images of epoxy-infiltrated nanoporous gold (npg) and hierarchical nanoporous gold (hnpg). **(A)** nanoporous gold structures below the cross-section plane shine through the epoxy-filled pores so that for some pixels it is unclear whether they belong to the solid (gold) or the pore (epoxy) phase (arrow with the question mark). **(B)** The influence of the shine-through effect is increased in hierarchical nanoporous gold due to the small pore sizes within the ligaments.

intensity of pixels in the SEM images generally depends not only on the material at the respective position in the plane currently imaged but also on structures in deeper layers. This effect occurs because these structures may shine through the nanopores up to the surface currently imaged by SEM, in case of back-scattered electron (BSE) imaging even in infiltrated nanoporous materials. Hence, there is no unique mapping between the intensity of a voxel in FIB tomography data and the material composition at exactly the position of this voxel. This ambiguity makes segmentation of FIB tomography data of nanoporous materials highly non-trivial (**Figure 1A**).

Classical methods like thresholding work best for standard materials without nanopores (Salzer et al., 2015). However, they fail for nanoporous materials with strong *shine-through effects* because of the ambiguity in the local voxel intensity. Machine learning algorithms like random forests or the k-means algorithm can help classify material and pores (Rogge and Ritter, 2018; Fager et al., 2020). However, deep learning-based (DL) methods, especially convolutional neural networks (CNN), bear the potential to outperform such methods when processing images. Over the last years, CNNs have more and more outperformed such classical methods across all disciplines (Krizhevsky et al., 2012; Girshick et al., 2014). For example, CNNs were used for the semantic segmentation of electron microscopy images of neuronal membranes (Ciresan et al., 2012). For the segmentation of FIB tomography images of porous membranes, the deep learning architecture ResUNet was applied, using initial training data generated by a random forest algorithm (Tracey et al., 2019). It is thus consequential to apply convolutional neural networks like U-Net, which was originally developed for biomedical images (Ronneberger et al., 2015), with some modifications also to FIB tomography data (Fend et al., 2021).

Due to *shine-through effects* in FIB tomography datasets, structures are visible through several subsequent SEM slices. Taking this information into account is an important step towards accurate segmentation of FIB tomography data of nanoporous materials. The machine learning architecture called CNN 2.5D has recently been reported to be particularly powerful (Vu et al., 2020) to incorporate such partial spatial information in a specific direction. CNN 2.5D feeds several adjacent slices into channels of a 2D CNN architecture. A similar approach, but with 3D kernels, is pursued by two other recently proposed machine learning architectures, 3D U-Net (Çiçek et al., 2016), and VNet (Milletari et al., 2016). The latter seeks to prevent information loss when the network grows deeper (Milletari et al., 2016).

Deep learning requires, however, a large amount of training data. In the context of the semantic segmentation of FIB tomography data, a sufficient amount of images is required whose pixels are labelled as belonging to specific categories (e.g., the solid or the pore phase in a nanoporous material). Obtaining sufficient training data from experiments can be expensive and time-consuming. To overcome this problem, synthetic training data is frequently used (Nikolenko, 2019).

In electron microscopy, two steps are required to generate synthetic training images for deep learning segmentation methods. The first step is the generation of a realistic geometric structure. The second step is the computer simulation of the FIB tomography of this structure, i.e., synthetic back-scattered electron (BSE) imaging data.

For the first step (Fend et al., 2021), previously used geometric primitives like spheres and cylinders. However, these do not adequately resemble the microstructure of the nanoporous materials of our interest. Therefore, these are suitable only to a limited extent for generating synthetic training data for the case we are interested in.

For the second step, Monte Carlo simulations of electron microscopy imaging can be performed. These simulate the trajectories of numerous electrons, thereby providing realistic information on the appearance of SEM images of specific structures. In order to perform such simulations on very simple geometries, first programs were developed more than two decades ago and have seen continuous improvements (Lowney J, 1994; Lowney, 1995; Karabekov et al., 2003; Zhang et al., 2012; Hovington et al., 1997; Gauvin et al., 2006).

In this paper, we compare different deep learning architectures for accurately segmenting FIB tomography data of nanoporous structures despite *shine-through effects*. We present a novel approach to generate synthetic FIB-SEM images using Monte Carlo (MC) simulations to overcome the lack of training data for deep learning methods. To obtain as realistic synthetic training data as possible, these simulations are not performed on geometries consisting of simple geometric primitives. Instead, we compare three different ways to generate largely realistic microstructures and use the most promising of them, the so-called levelled-wave algorithm (Li et al., 2020) as a basis for our study. Using the in silico training data generated this way, we demonstrate that 2D CNN with a group of adjacent slices as input data and 3D CNN can surpass the segmentation performance of classical methods by more than 100%. In the absence of ground-truth data, we measure the segmentation performance with a novel approach, which exploits specific geometrical properties of nanoporous gold and hierarchical nanoporous gold, such as isotropy.

2 MATERIALS AND METHODS

2.1 Generation of Synthetic Training Data

Synthetic FIB tomography data can be generated in two steps. The first step is the generation of virtual microstructures, and the second step is the generation of virtual SEM images of these microstructures using MC simulations.

2.1.1 Generation of Virtual Microstructures

To generate artificial microstructures that closely resemble the ones of nanoporous materials, we compared three different methods: the levelled wave method (LWM), self-similarity method (SSM) and random pore generation method (RPGM).

2.1.1.1 Levelled Wave Method

Nanoporous materials are often produced by dealloying. Theoretical analysis reveals (Li et al., 2020) that this leads to a microstructure whose geometry can be described by a superimposition of several wave vectors with an identical wavelength but different random orientations (Li et al., 2020). Subsequently, the Gaussian random field generated this way is subjected to a thresholding algorithm to divide it into a solid and a pore phase, resulting in microstructures as illustrated in **Figure 2A**.

2.1.1.2 Self-Similarity Method

A structure is called self-similar if it resembles exactly or partly itself. In this method, a hierarchical microstructure is generated using the thresholded images of a real nanoporous gold structure, hence the name "self-similarity method (SSM)". In the first step, FIB-SEM images of nanoporous gold are segmented to get binary images identifying the solid phase (intensity 255) and pore phase (intensity 0) using the k-means algorithm. Then, these binary images are resized using bilinear interpolation (Press et al., 1992) according to required voxel dimensions. In the next step, a mask, smaller in size than the binary images from the previous step, is prepared by resizing binary slices and rotating them at a random angle. Final output images are then calculated by performing arithmetic AND operations on binary images with masks using convolution (**Supplementary Figure S13**). These AND operations with masks generated from the original binary structure make the final structure self-similar. One slice from SSM is shown in **Figure 2B**.

2.1.1.3 Random Pore Generation Method

RPGM is a relatively simple method. In the first step, a volume that is fully solid (intensity 255) is chosen. Subsequently, void spheres are introduced at random locations and with a radius drawn from a Gaussian random distribution using a masking operation (**Supplementary Figure S14**). The advantage of RPGM is that it is a straightforward method. Its disadvantage is that it produces microstructures that exhibit considerable differences compared to actual nanoporous materials, limiting their value for training accurate machine learning-based segmentation algorithms in our case. The final sample geometry generated by this method is shown in **Figure 2C**.

2.1.2 Monte Carlo Simulation of Electron Microscopy

The virtual microstructures generated in the above-described ways are used in Monte Carlo simulations to generate synthetic FIB tomography data. The simulation of the BSE images is performed using the software MCXray. This software is an extension of the Monte Carlo simulation tools Casino (Hovington et al., 1997) and Win X-Ray (Gauvin et al., 2006). It was developed by (Gauvin and Michaud, 2009) and then incorporated in the Dragonfly Software [ORS, Montreal, Canada] (Object Research Systems, 2018). MCXray allows simulations of complex microstructures even consisting of different materials. Simulated BSE images of virtual microstructures generated by the above described three methods are presented in **Figure 3**. In all the Monte Carlo simulations for this paper, we assume beam energy of 2 kV, a beam current of 5×10^{-11} A, FIB sample stage tilt angle of 0 *degrees*, and detector to sample distance of 104 *mm*. We performed these simulations for nanoporous gold and hierarchical nanoporous gold, and considered pores in vacuum (**Figure 3**) as well as infiltrated with epoxy resin (see for example **Figures 4–B**). These simulations have a high *shine-through effect*

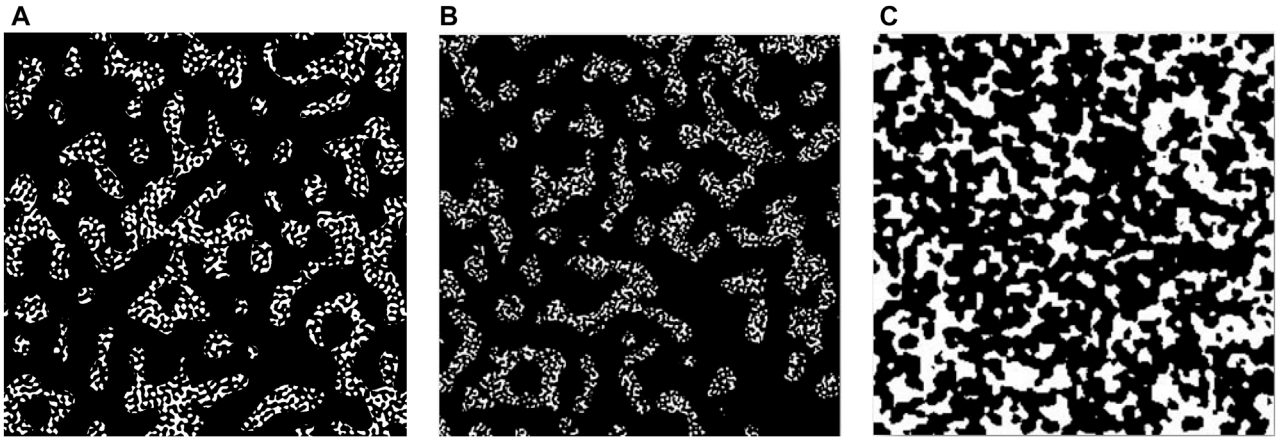


FIGURE 2 | Virtual microstructures generated by **(A)** levelled wave method **(B)** self-similarity method **(C)** random pore generation method.

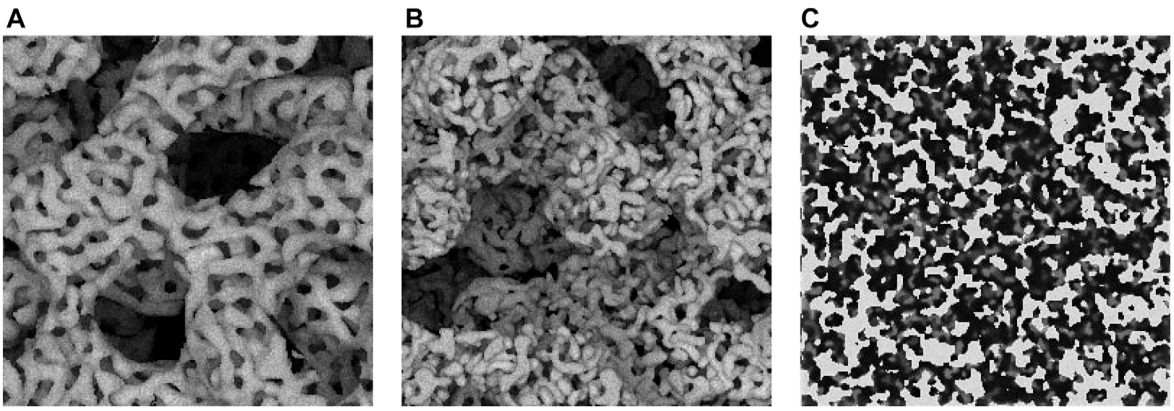


FIGURE 3 | Simulated BSE images using Monte Carlo simulation method and data generated using **(A)** LWM as initial virtual structure **(B)** SSM as initial virtual structure **(C)** RPGM as initial virtual structure.

and mimic images from a less surface sensitive through-the-lens (TLD) back-scattered electron detector.

2.2 Preprocessing of Training Data

In our study, we applied noise and blur filters to make the simulated images more similar to actual FIB tomography data. Subsequently, we applied the online data augmentation technique to increase the size of the training dataset.

2.2.1 Adding Noise

Electron microscopy images typically exhibit two types of noise (Cizmar et al., 2008). Primary electrons can generate Poisson noise, and the rest of the electrons from five noise sources can generate Gaussian noise (Timischl et al., 2012). Not all five sources are equally important, and noise added by detection systems is often assumed to be negligible (Sim et al., 2004; Goldstein, 2003). MCXray (Gauvin and Michaud, 2009) simulations of BSE images naturally include Gaussian noise (Hovington et al., 1997). We added the remaining Poisson noise and some additional Gaussian noise to understand the effect of noise in synthetic SEM images. To this end, we used the Scikit image library[1] in this project. First, Poisson noise was added to the image, and then Gaussian noise, to get a realistic noisy simulated BSE image. For the Gaussian noise, we heuristically chose a zero mean value and a variance of 0.001. After adding the noise, the intensities of the image were renormalized to a range from -1 to 1, converting the noisy image thereby to a standard unsigned 8-bit image. As training data for our study, we used these resulting noisy images.

2.2.2 Blurring of Edges

In the images generated by MCXray simulations[2], the edges were observed to be unrealistically sharp. Simple solutions like applying Gaussian filtering to the whole image may not work

[1]https://scikit-image.org

[2]Dragonfly software version 2021.1.0.118 13. Blurring of edges may not require in future versions of the software

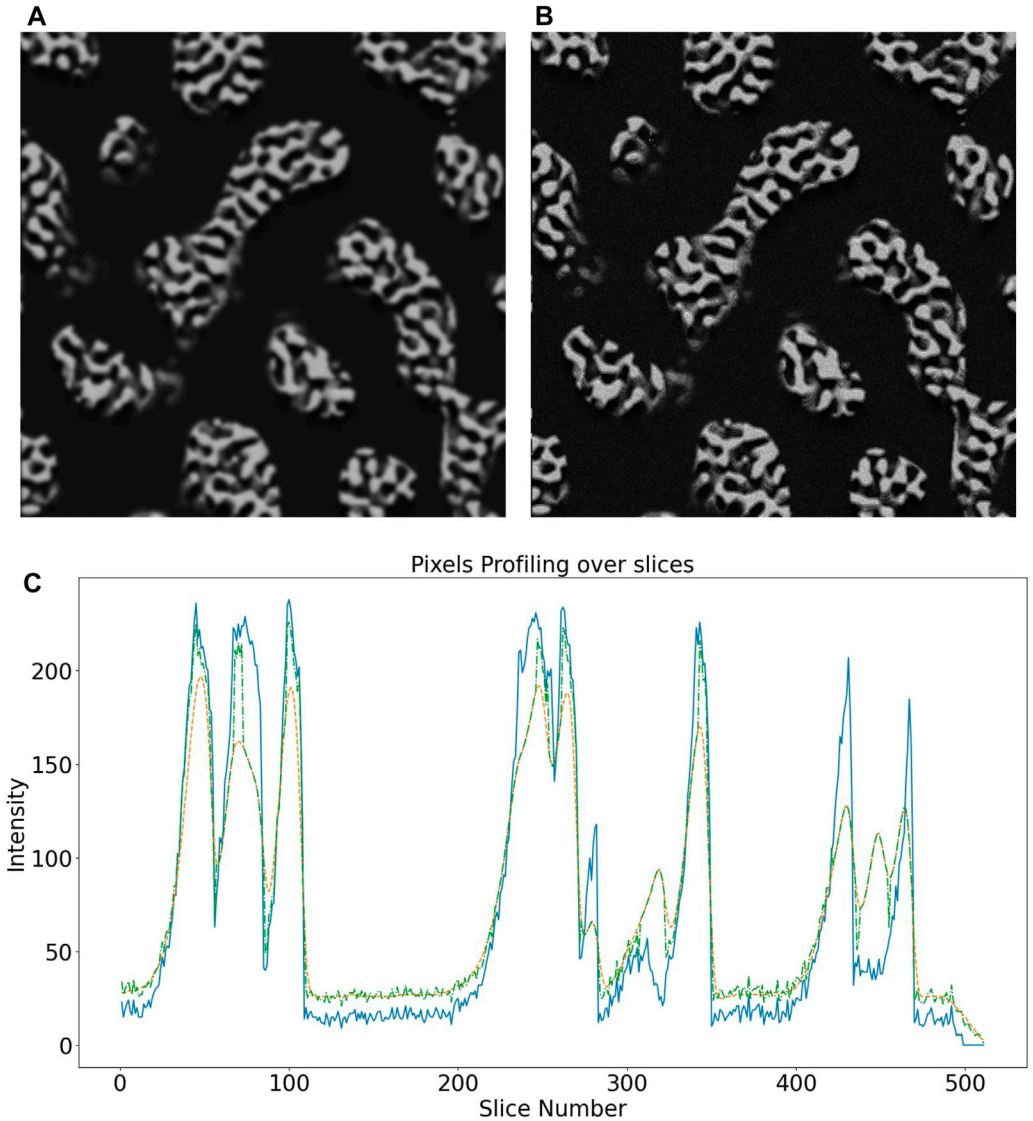


FIGURE 4 | Synthetic SEM image after applying **(A)** standard Gaussian blur filter and **(B)** the 4-step blurring procedure introduced herein **(C)** intensities of pixel located at a random location in the xy-plane across slices before blurring (blue), after applying Gaussian blur filter (orange) and 4-step blurring procedure (green).

as it may remove necessary Gaussian randomness of intensities from the solid ligaments in the simulated image. Therefore, it is very important to blur only the edges of the ligaments. This can be achieved by the following steps of masking and blurring.

1. Find edges of the solid ligaments
2. Generate mask with maximum weights at the edges from step (1)
3. Blur whole original image
4. Blend source image and blurred image from step (3) by mask from step (2)

The difference between this procedure and standard Gaussian blurring is illustrated in **Figure 4**. The image generated by the above 4-step procedure looks more realistic.

2.2.3 Data Augmentation

Data augmentation is a very powerful technique in deep learning when there is not enough training data available (Wang et al., 2017). Herein, we used online data upsampling; during the training process itself, the training data was augmented by applying random flips, rotations, brightness changes, and stretch transforms.

2.3 Machine Learning Architectures for Segmentation

In our synthetic FIB tomography data - as in real data - *shine-through effects* occur. Hence, it can be expected that accurate segmentation is not possible by processing image data layer by layer but rather in the group of the layers. Herein, we tested two machine learning architectures for segmentation that address this need.

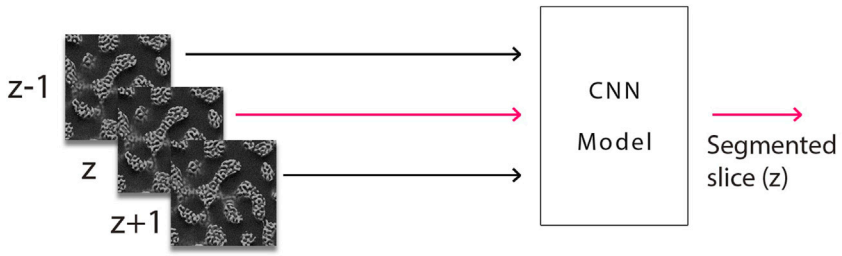


FIGURE 5 | Basic block diagram of 2D CNN receiving as input for the segmentation of slice *z* also the slices *z* – 1 (above) and *z* + 1 (below).

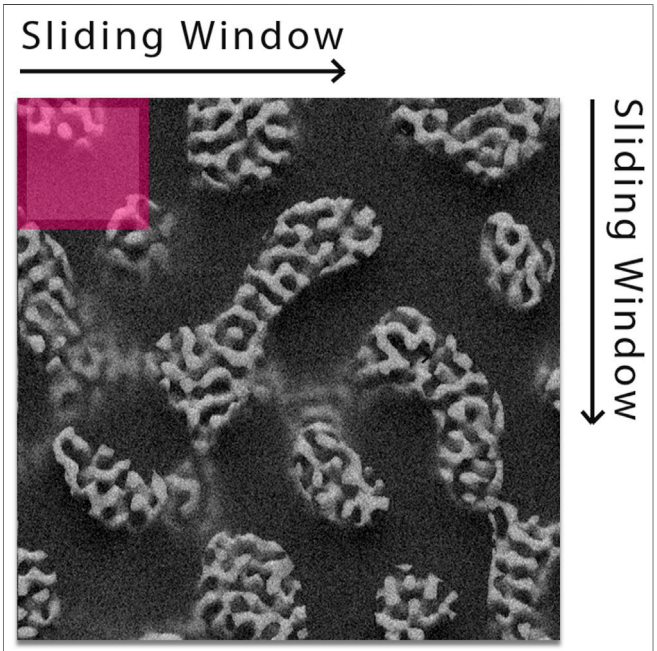


FIGURE 6 | Sliding window method.

2.3.1 2D CNN With Adjacent Slices

Shine-through effects establish a correlation between adjacent slices. To account for this in semantic segmentation, one has to segment different slices not separately but rather process information from several adjacent slides in each segmentation operation. As a way to avoid computationally costly 3D convolutions, one can process information from several adjacent layers simultaneously by 2D CNN (**Figure 5**). To this end, we used the 2D CNN architecture U-Net (Ronneberger et al., 2015). A general limitation of this approach is that the number of slices that can be included at a time is naturally limited by the number of slices available in the 3D image stack that forms the FIB tomography data. Indeed this limits the depth to which segmentation can be performed.

2.3.2 3D CNN

While 2D CNNs are computationally cheaper than 3D CNN, they may always be prone to miss out on recognizing some spatial features. To overcome this problem, we also tested 3D CNN. These use full 3D convolutions. We compared the 3D CNN architectures 3D U-Net (Çiçek et al., 2016), VNet (Milletari et al., 2016) and ResUNet3D with minor modifications. In the U-Net architecture (Ronneberger et al., 2015) we used padding in the convolutional blocks to retain the original image size. Moreover, we also added residual connections in one of our 3D U-Net models. We considered the number of encoding blocks as a hyperparameter and tuned it to improve performance.

TABLE 1 | Parameters used for training 2D CNN with adjacent slices.

Parameter	Value
Patch size	64
Stride	0.5
Batch size	64
Epochs	100 with early stopping with patience = 10
Loss	Dice loss
Optimizer	Adam
Learning rate	0.00001 with reducing it by factor of 0.10 with patience of 10

TABLE 2 | Parameters used for training 3D CNN.

Parameter	Value
Patch size	64
Stride	0.5
Batch size	1
Epochs	100 with early stopping with patience = 10
Loss	Dice Loss
Optimizer	Adam
Learning rate	0.0001 with reducing it by factor of 0.10 with patience of 10

2.4 Training of Neural Networks

All machine learning architectures were implemented using PyTorch; data loaders were written in *Python*, and models were trained on Tesla K80 GPUs.

2.4.1 2D CNN With Adjacent Slices

Input to the deep neural network was provided using the sliding window technique (**Figure 6**), where small patches (64 × 64 pixels) were generated from the original training dataset and used as the final training set for the network. We used 3, 5, 7 and 9 slices as the number of adjacent slices, making the input

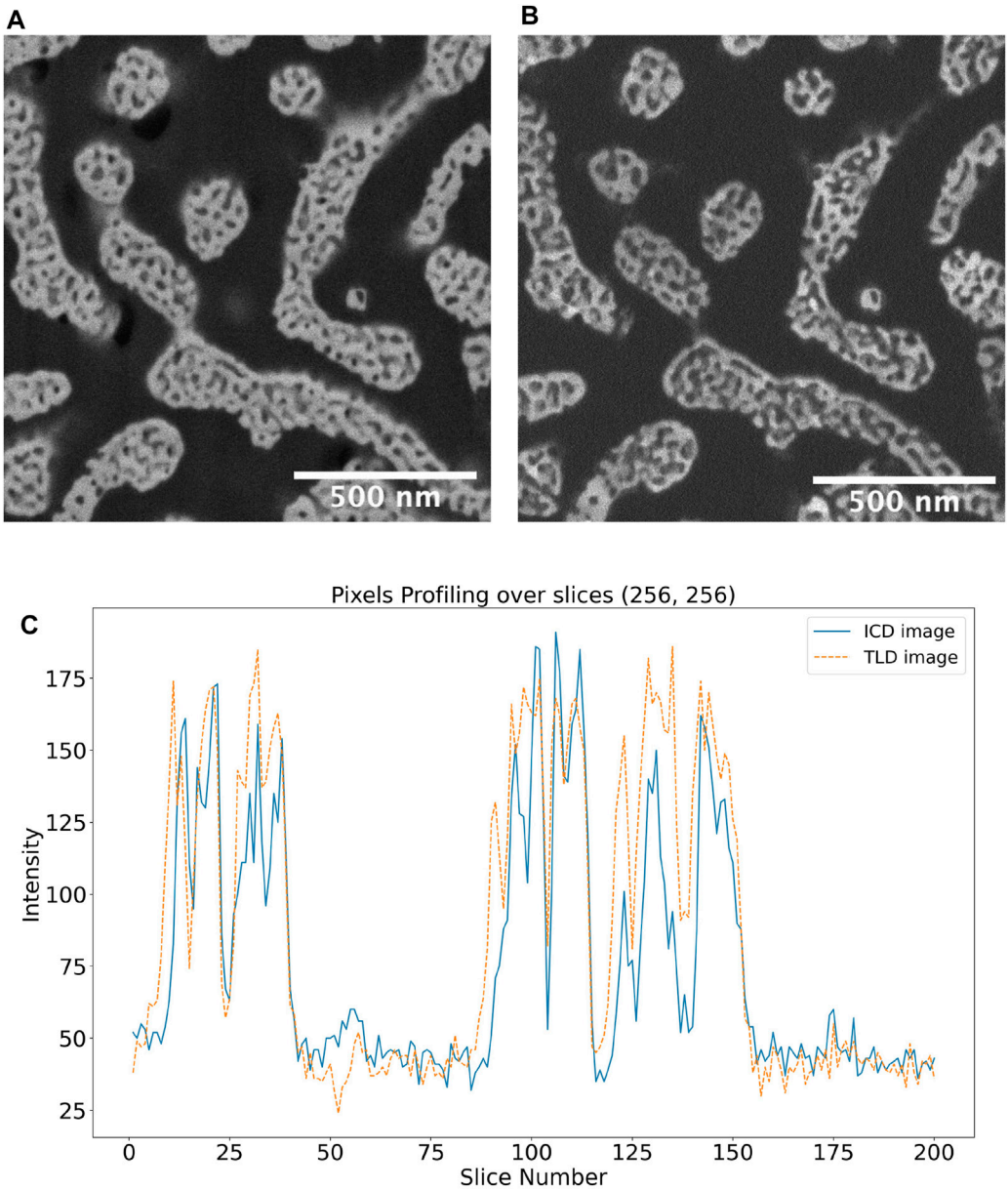


FIGURE 7 | SEM images of hierarchical nanoporous gold simultaneously recorded with **(A)** TLD and **(B)** ICD detector **(C)** intensities of pixel located at a random location in the xy-plane in SEM TLD (orange) and ICD (orange) images.

window size 64 × 64 × 3, 5, 7 and 9, to understand the effect of the number of adjacent slices on the segmentation performance. Additional parameters used for the training process are specified in **Table 1**.

2.4.2 3D CNN

We trained and compared a total of three 3D CNN models, namely U-Net3D, ResUNet and VNet. We used the same sliding window technique for all of them. However, unlike for the 2D CNN, the window was moving in all three spatial directions with a given stride. The training parameters are summarized in **Table 2**. Inspired by (Milletari et al., 2016), we used a squared Dice loss layer with the necessary smoothness value to avoid zero division in all three 3D CNN architectures. We used the mean Dice loss as the final loss value to account for possible data imbalance (Milletari et al., 2016).

2.5 Evaluation Criteria

Due to the unavailability of labelled datasets, the CNN models were trained based on synthetic training and validation data. Moreover, to evaluate the performance of various segmentation methods for real BSE images, we used the concept of anisotropy (**Supplementary Section S1**). The *shine-through effect* makes the images anisotropic in the *z*-direction (though the underlying material microstructure is in a statistical sense isotropic, that is,

TABLE 3 | Performance of different segmentation methods applied to dataset npg40.

	Target data	DL method (SSM)	DL method (LWM)	DL method (RPGM)	Otsu's algorithm	k-means clustering (k = 3)
Solid volume fraction (ϕ)	0.34	0.34	0.34	0.36	0.55	0.49
Relative error of solid volume fraction (%)	0.00	−1.99	0.50	4.33	33.69	17.94
Dice coefficient	1.00	0.98	0.99	0.89	0.75	0.78

TABLE 4 | Impact of preprocessing on segmentation performance for real hnpg_epoxy dataset.

DL model	Preprocessing	ϕ (TLD)	ϕ (ICD)	MP	e_{L2} TPCF	e_{L2} LPF	$e_{D_{L2}}$
2D CNN with adjacent slices	No	0.11	0.08	0.08	0.12	0.07	0.02
2D CNN with adjacent slices	Yes	0.16	0.10	0.10	0.07	0.008	0.004
3D CNN	No	0.19	0.11	0.12	0.15	0.18	0.12
3D CNN	Yes	0.18	0.13	0.10	0.13	0.12	0.08

lacking any preferred direction). Therefore, if the segmented images are isotropic, then it can be concluded that the segmentation method has (likely) been able to filter the *shine-through effect* and performs well. To measure anisotropy of the segmented images, we used two-point correlation functions (TPCF), **Supplementary Figures S7, S8**, and lineal path functions (LPF) for segmented ICD images (**Supplementary Figures S10, S11**) using our method and Otsu's thresholding method. This set of real imaging data, referred to henceforth as hnpg_epoxy dataset, has a voxel size of $2.6 \times 2.6 \times 10\ nm^3$, which was interpolated to $5.2 \times 5.2 \times 5.2\ \text{nm}^3$ using bicubic interpolation. To quantify anisotropy of the segmented images, we calculated the TPCF and LPF in the x-, y- and *z*-direction. Subsequently, we computed the relative L2 differences for the TPCF and LPF in *z*-direction and compared them to the relative L2 differences for the TPCF and LPF in x- and *y*-direction, respectively. Generally, the relative L2 difference between two functions f and g can be computed in a discrete setting with n given data points as

$$e_{L2}(f,g) = \frac{2 \times \sqrt{\sum_{i=1}^{n} (f_i - g_i)^2}}{\left(\sqrt{\sum_{i=1}^{n} (f_i)^2} + \sqrt{\sum_{i=1}^{n} (g_i)^2}\right)} \tag{1}$$

where the f_i, g_i with $i = 1, \ldots, n$ are the given data points. Finally, we averaged the L2 differences of the *z*-direction compared to the x- and *y*-direction. This average is referred to henceforth (e.g., in **Table 4**) as TPCF or LPF e_{L2} difference. The larger both are, the higher the anisotropy of the segmented images, which can be considered a hint that the associated segmentation method has not been able to filter *shine-through effects*.

As an additional measure of anisotropy, we calculated the average diameters of ligaments in xy-, yz- and xz-planes using lineal path functions (D_{xy}, D_{yz}, D_{xz}). Then, we computed the averaged relative difference $e_{D_{L2}}$ of the diameters in the xz- and yz-planes compared to the one in the xy-plane as

$$e_{D_{L2}} = \frac{1}{2}\left(\sqrt{\frac{\left(D_{xz} - D_{xy}\right)^2}{D_{xy}^2}} + \sqrt{\frac{\left(D_{yz} - D_{xy}\right)^2}{D_{xy}^2}}\right) \tag{2}$$

TABLE 5 | Specifications of real FIB-SEM gold datasets used in this study. Note: All dataset discussed in the table have epoxy material as pore filling.

Dataset name	Gold structure	Pixel size [nm^3]	Detector type
hnpg epoxy ICD	hnpg	2.6_2.6_10	ICD
hnpg epoxy TLD	hnpg	2.6_2.6_10	TLD
npg ICD	npg	3.4_3.4_7	ICD
npg TLD	npg	3.4_3.4_7	TLD

Generally, the average diameter of ligaments in the *z*-direction is expected to be larger than in the x- and *y*-direction in the presence of non-filtered *shine-through effects*. Therefore, pronounced *shine-through effects* can be expected to result in large $e_{D_{L2}}$.

In the absence of ground truth, we also checked the congruency between images from two different sensors. We segmented FIB-SEM images from a low-loss in-column back-scattered electron detector (ICD) and compared the results with segmented images from the TLD. The ICD detector is situated at an upper position in the column, ensuring that only the most elastically scattered electrons (i.e., back-scattered electrons) are collected. The signal is highly sensitive to Z-contrast with almost no topographical information. Also, low energy loss increases the probability of near-surface interaction and therefore near-surface information (Ritter, 2019) (**Figures 7–B**). Therefore, one can expect the ICD images to have relatively small *shine-through effects* anyway so that they can provide at least some hint at the (not exactly known) ground truth. We calculated the fraction of misplaced pixels by

$$MP = \frac{(TP - IP)}{TP} \times 100 \tag{3}$$

where TP is the total number of pixels, IP the number of pixels identically segmented for TLD and ICD images, and MP is the fraction of pixels where the segmentation of TLD and associated ICD images disagreed.

TABLE 6 | Performance of different segmentation methods applied to synthetic LWM dataset. MP is here computed not using ICD images as reference but exact synthetic microstructure generated by the LWM.

Model name	ϕ (TLD)	ϕ (LWM)	MP	e_{L2} TPCF	e_{L2} LPF	$e_{D_{L2}}$
k-means clustering	0.33	0.12	0.26	0.83	0.75	0.95
Otsu's algorithm	0.54	0.12	0.43	0.96	0.65	0.71
2D CNN	0.12	0.12	0.04	0.18	0.03	0.02
2D CNN with adjacent slices	0.13	0.12	0.03	0.18	0.02	0.02
3D CNN	0.12	0.12	0.03	0.17	0.01	0.003

TABLE 7 | Performance of different segmentation methods applied to real hnpg_epoxy_TLD dataset.

Model name	ϕ (TLD)	ϕ (ICD)	MP	e_{L2} TPCF	e_{L2} LPF	$e_{D_{L2}}$
k-means clustering	0.39	0.13	0.39	0.20	0.41	0.37
Otsu's algorithm	0.38	0.29	0.12	0.21	0.29	0.24
2D CNN	0.20	0.10	0.16	0.14	0.05	0.01
2D CNN with adjacent slices	0.16	0.10	0.10	0.07	0.008	0.004
3D CNN	0.18	0.13	0.10	0.13	0.12	0.08

TABLE 8 | Performance of different segmentation methods applied to real npg_TLD dataset.

Model name	ϕ (TLD)	ϕ (ICD)	MP	e_{L2} TPCF	e_{L2} LPF	$e_{D_{L2}}$
k-means clustering	0.33	0.24	0.57	0.46	0.28	0.24
Otsu's algorithm	0.54	0.40	0.19	0.40	0.18	0.12
2D CNN	0.24	0.21	0.18	0.38	0.20	0.10
2D CNN with adjacent slices	0.40	0.36	0.17	0.16	0.07	0.05
3D CNN	0.44	0.31	0.20	0.12	0.06	0.04

3 RESULTS AND DISCUSSION

3.1 Selection of Best Synthetic Training Data

The first question we sought to address was which of the three above described methods of generating synthetic training data (LWM, SSM, RPGM) is most suitable for training deep neural networks for semantic segmentation. We performed a study using the CNN architecture U-Net to answer this question. We trained the U-Net on the above three types of synthetic data, splitting them into training data (60%), validation data (20%), and test data (20%). Subsequently, we compared the performance of the trained U-Nets to two classical segmentation methods, Otsu's thresholding algorithm (Otsu, 1979) and k-means clustering (Lloyd, 1982) with k = 3.

To evaluate the performance of these segmentation methods, we generated a synthetic dataset called npg40, which was prepared using MC simulations on synthetic microstructures generated by k-means segmentation of real nanoporous gold SEM images. The performance of the models was then evaluated using metrics as the relative error of solid volume fraction and the Dice coefficients. As shown in **Table 3**, the deep learning model trained on the LWM data exhibits the lowest solid volume fraction error and highest Dice coefficient. Therefore, we trained all our deep learning models, shown below, using synthetic LWM data only after this preliminary test.

3.2 Role of Preprocessing

It is instructive to compare the performance of segmentation methods trained on synthetic training data with and without preprocessing ("blur the edges" and subsequent addition of noise). To this end, we trained 3D CNN and 2D CNN processing adjacent slices (on synthetic LWM data) and subsequently compared their segmentation performance on the (real) hnpg_epoxy dataset. **Table 4** reveals that preprocessing increases segmentation performance for both 2D CNN processing adjacent slices and 3D CNN, underlining that preprocessing indeed helps to generate more realistic synthetic training data.

3.3 Semantic Segmentation

We validated our trained models and their performance both by the segmentation of one synthetic LWM dataset and four sets of real FIB-SEM images (see **Table 5**). Results are presented in **Tables 6**, **7**, and **8**. In these tables, it is apparent that the L2 differences for the TPCF and LPF and also the ligament diameter anisotropy is much lower for the novel segmentation methods based on deep learning introduced in this paper compared to classical methods like Otsu's method or k-means clustering (with k = 3). This suggests that our novel deep learning methods filter *shine-through effects* much better than classical methods and thus produce a geometry much closer to the real one. The excessive solid volume fraction of the segmented ICD data compared to the associated TLD data suggests that, in particular, Otsu's thresholding method in many

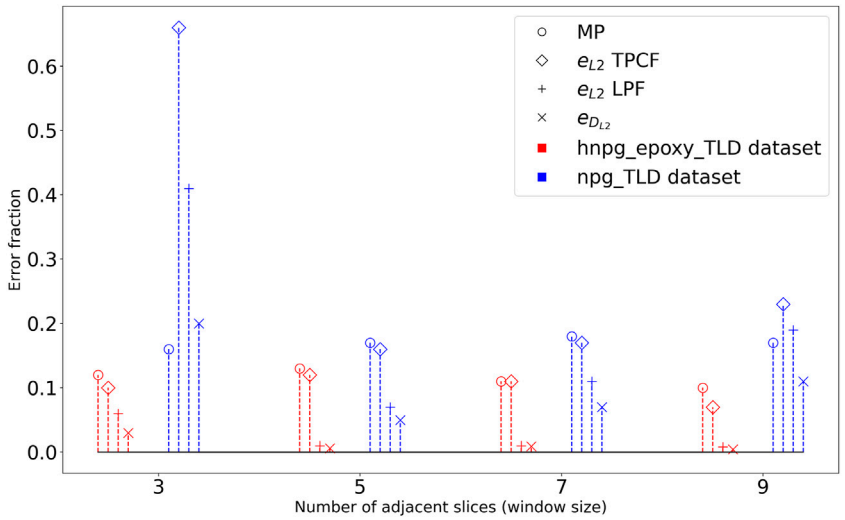


FIGURE 8 | Effect of window size when using 2D CNNs with adjacent slices for segmentation for the real datasets from **Table 5**.

cases fails to classify pixels visible due to the *shine-through effect*. Our deep learning-based segmentation methods exhibit much lower MP values compared to Otsu's method and k-means clustering. That is, they succeed in more cases to classify pixels in TLD and ICD imaging data identically. Following the concept of sensor fusion, also this is indicative of a better performance of our deep learning-based segmentation methods. It is interesting to discuss why there is a particularly large number of misplaced pixels for both real data sets when using k-means clustering. To explain this, it is important to note that we selected k = 3 for k-means clustering to account for the presence of a total of three different clusters, namely a solid phase, a pore phase and artifacts (i.e. pixels due to the shine-through effect). This assumption is certainly reasonable for TLD images because of their sensitivity to shine-through effects. By contrast, ICD images are much less prone to this problem, so k = 3 may no longer be a good assumption, resulting in poor performance of the associated segmentation.

Among the CNN-based segmentation methods, the 2D CNN with adjacent slices and the 3D CNN performed better than the standard 2D CNN. They probably can generalize better to the different datasets. For example, the very low solid volume fraction in **Table 8** for the 2D CNN segmentation suggests that this method fails to classify solid pixels for the npg_TLD dataset.

We also studied the effects of window size (number of adjacent slices) on segmentation performance. It is evident from **Figure 8** that 2D CNN processing adjacent slices provided the best performance for windows of sizes 9 and 5 for the hnpg_epoxy_TLD and the npg_TLD datasets, respectively.

4 CONCLUSION

Deep learning can play an important role in the segmentation of FIB tomography data. A potential caveat is the limited availability of training data. We demonstrated that the lack of training data can be overcome by generating virtual microstructures and simulating them using the MCXray method, providing ample synthetic FIB tomography training data. We compared three different methods to generate realistic synthetic nanoporous geometries. Our results reveal that the training data generated by the levelled wave method (LWM) is most effective for deep learning of image segmentation. A major problem in this context is the typically missing availability of ground truth. That is, for real materials, most often, there is no information available that would characterize their microstructure more accurately and reliably than segmented FIB tomography data, which makes it naturally difficult to evaluate the performance of a specific segmentation method. Herein, we introduced a novel approach to measure the extent to which *shine-through effects* - which can be expected to be the dominant source of errors in the semantic segmentation of images of nanoporous materials - are filtered out by different segmentation methods. This method did not require any full ground truth but rather exploited that nanoporous materials typically exhibit an isotropic microstructure. This way, the degree of anisotropy of the segmented images could be used as a proxy of the segmentation error. We tested different deep learning architectures for the segmentation of FIB tomography data and identified 2D CNNs with adjacent slices as image channels and 3D CNNs as the best architectures of the ones tested herein. Generally, 3D CNNs were found to be computationally more expensive to train than 2D CNN with adjacent slices.

DATA AVAILABILITY STATEMENT

The datasets generated and analyzed for this study can be found in TUHH Open Research repository (https://tore.tuhh.de/handle/11420/11060).

AUTHOR CONTRIBUTIONS

TS, RA, CC, and MR contributed to conception and design of the study. YL generated the LWM database. NP and RG helped

with the formation of Monte Carlo simulations. TS wrote the first draft of the manuscript. All authors contributed to manuscript revision, read, and approved the submitted version.

FUNDING

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – SFB 986 – Project number 192 346 071.

ACKNOWLEDGMENTS

We thank Stefan Bartels for his contribution by providing us the necessary computation power to generate the simulated images in this study.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fmats.2022.837006/full#supplementary-material

REFERENCES

Ciresan, D., Giusti, A., Gambardella, L., and Schmidhuber, J. (2012). Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images. *Adv. Neural Inf. Process. Syst.* 25, 2843–2851.

Cizmar, P., Vladár, A. E., Ming, B., and Postek, M. T. (2008). For the National Institute of Standards and TechnologySimulated Sem Images for Resolution Measurement. *Scanning* 30, 381–391. doi:10.1002/sca.20120

Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., and Ronneberger, O. (2016). "3d U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation," in International conference on medical image computing and computer-assisted intervention (Berlin, Germany: Springer), 424–432. doi:10.1007/978-3-319-46723-8_49

Fager, C., Röding, M., Olsson, A., Lorén, N., von Corswant, C., Särkkä, A., et al. (2020). Optimization of FIB-SEM Tomography and Reconstruction for Soft, Porous, and Poorly Conducting Materials. *Microsc. Microanal* 26, 837–845. doi:10.1017/s1431927620001592

Fend, C., Moghiseh, A., Redenbach, C., and Schladitz, K. (2021). Reconstruction of Highly Porous Structures from FIB-SEM Using a Deep Neural Network Trained on Synthetic Images. *J. Microsc.* 281, 16–27. doi:10.1111/jmi.12944

Gauvin, R., Lifshin, E., Demers, H., Horny, P., and Campbell, H. (2006). Win x-ray: A New Monte Carlo Program that Computes X-ray Spectra Obtained with a Scanning Electron Microscope. *Microsc. Microanal.* 12, 49–64. doi:10.1017/s1431927606060089

Gauvin, R., and Michaud, P. (2009). Mc x-ray, a New Monte Carlo Program for Quantitative X-ray Microanalysis of Real Materials. *Microsc. Microanal* 15, 488–489. doi:10.1017/s1431927609092423

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 580–587. doi:10.1109/cvpr.2014.81

Goldstein, J. (2003). *Scanning Electron Microscopy and X-Ray Microanalysis*. Berlin, Germany: Springer US.

Hovington, P., Drouin, D., and Gauvin, R. (1997). Casino: A New Monte Carlo Code in C Language for Electron Beam Interaction—Part I: Description of the Program. *Scanning* 19, 1–14. doi:10.1002/sca.4950190104

Karabekov, A., Zoran, O., Rosenberg, Z., and Eytan, G. (2003). Using Monte Carlo Simulation for Accurate Critical Dimension Metrology of Super Small Isolated Poly-Lines. *Scanning: J. Scanning Microscopies* 25, 291–296.

Knott, G., Marchman, H., Wall, D., and Lich, B. (2008). Serial Section Scanning Electron Microscopy of Adult Brain Tissue Using Focused Ion Beam Milling. *J. Neurosci.* 28, 2959–2964. doi:10.1523/jneurosci.3189-07.2008

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet Classification with Deep Convolutional Neural Networks. *Commun. ACM* 60, 84–90. doi:10.1145/3065386

Li, Y., Dinh Ngô, B.-N., Markmann, J., and Weissmüller, J. (2020). Datasets for the Microstructure of Nanoscale Metal Network Structures and for its Evolution during Coarsening. *Data in brief* 29, 105030. doi:10.1016/j.dib.2019.105030

Lloyd, S. (1982). Least Squares Quantization in Pcm. *IEEE Trans. Inform. Theor.* 28, 129–137. doi:10.1109/tit.1982.1056489

Lowney J, M. E. (1994). *User's Manual for the Program Monsel-1: Monte Carlo Simulation of Sem Signals for Linewidth Metrology*. Gaithersburg, Maryland, U.S.: NIST special publication.

Lowney, J. (1995). *Monsel-ii Monte Carlo Simulation of Sem Signals for Linewidth Metrology*. Gaithersburg, Maryland, U.S.: NIST special publication.

Milletari, F., Navab, N., and Ahmadi, S.-A. (2016). "V-net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation," in 2016 fourth international conference on 3D vision (3DV) (IEEE), 565–571. doi:10.1109/3dv.2016.79

Nikolenko, S. (2019). "Synthetic Data in Deep Learning," in School-conference "Approximation and Data Analysis 2019", 21.

Object Research Systems (2018). *Object Research Systems (ORS) Inc*. Montreal, Canada: Object Research Systems (ORS) Inc. Dragonfly 3.6 [computer software].

Otsu, N. (1979). A Threshold Selection Method from gray-level Histograms. *IEEE Trans. Syst. Man. Cybern.* 9, 62–66. doi:10.1109/tsmc.1979.4310076

Press, W. H., Teukolsky, S. A., Flannery, B. P., and Vetterling, W. T. (1992). *Numerical Recipes in Fortran 77: Volume 1, Volume 1 of Fortran Numerical Recipes: The Art of Scientific Computing*. Cambridge: Cambridge University Press.

Prill, T., Schladitz, K., Jeulin, D., Faessel, M., and Wieser, C. (2013). Morphological Segmentation of Fib-Sem Data of Highly Porous media. *J. Microsc.* 250, 77–87. doi:10.1111/jmi.12021

Ritter, M. (2019). Comparison of 3d-Fib Tomography Volume Reconstruction with Various In-Column Detectors on an Epoxy-Infiltrated Nonoporous Metal Test Sample," in EMAS 2019—16th European Workshop on Modern Developments and Applications in Microbeam Analysis.

Rogge, F., and Ritter, M. (2018). Cluster Analysis for Fib Tomography of Nanoporous Materials," in EMAS 2019—19th International Microscopy Congress (IMC19).

Ronneberger, O., Fischer, P., and Brox, T. (2015). "U-net: Convolutional Networks for Biomedical Image Segmentation," in International Conference on Medical image computing and computer-assisted intervention (Berlin, Germany: Springer), 234–241. doi:10.1007/978-3-319-24574-4_28

Salzer, M., Prill, T., Spettl, A., Jeulin, D., Schladitz, K., and Schmidt, V. (2015). Quantitative Comparison of Segmentation Algorithms for Fib-Sem Images of Porous media. *J. Microsc.* 257, 23–30. doi:10.1111/jmi.12182

Sim, K. S., Thong, J. T. L., and Phang, J. C. H. (2004). Effect of Shot Noise and Secondary Emission Noise in Scanning Electron Microscope Images. *Scanning* 26, 36–40. doi:10.1002/sca.4950260106

Timischl, F., Date, M., and Nemoto, S. (2012). A Statistical Model of Signal-Noise in Scanning Electron Microscopy. *Scanning* 34, 137–144. doi:10.1002/sca.20282

Tracey, J., Lin, S. S., Jankovic, J., Zhu, A., and Zhang, S. (2019). Iterative Machine Learning Method for Pore-Back Artifact Mitigation in High Porosity Membrane Fib-Sem Image Segmentation. *Microsc. Microanal* 25, 186–187. doi:10.1017/s1431927619001661

Vu, M. H., Grimbergen, G., Nyholm, T., and Löfstedt, T. (2020). Evaluation of Multislice Inputs to Convolutional Neural Networks for Medical Image Segmentation. *Med. Phys.* 47, 6216–6231. doi:10.1002/mp.14391

Wang, J., and Perez, L. (2017). The Effectiveness of Data Augmentation in Image Classification Using Deep Learning. *Convolutional Neural Networks Vis. Recognit* 11, 1–8.

Zhang, P., Wang, H. Y., Li, Y. G., Mao, S. F., and Ding, Z. J. (2012). Monte Carlo Simulation of Secondary Electron Images for Real Sample Structures in Scanning Electron Microscopy. *Scanning* 34, 145–150. doi:10.1002/sca.20288

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers in Materials

ORIGINAL RESEARCH
published: 11 March 2022
doi: 10.3389/fmats.2022.851085

Development of a Robust CNN Model for Capturing Microstructure-Property Linkages and Building Property Closures Supporting Material Design

Andrew Mann[1] *and Surya R. Kalidindi*[1,2]*

[1]*Georgia Institute of Technology, School of Materials Science and Engineering, Atlanta, GA, United States,* [2]*Georgia Institute of Technology, George W. Woodruff School of Mechanical Engineering, Atlanta, GA, United States*

Recent works have demonstrated the viability of convolutional neural networks (CNN) for capturing the highly non-linear microstructure-property linkages in high contrast composite material systems. In this work, we develop a new CNN architecture that utilizes a drastically reduced number of trainable parameters for building these linkages, compared to the benchmarks in current literature. This is accomplished by creating CNN architectures that completely avoid the use of fully connected layers, while using the 2-point spatial correlations of the microstructure as the input to the CNN. In addition to increased robustness (because of the much smaller number of trainable parameters), the CNN models developed in this work facilitate the construction of property closures at very low computational cost. This is because it allows for easy exploration of the space of valid 2-point spatial correlations, which is known to be a convex hull. Consequently, one can generate new sets of valid 2-point spatial correlations from previously available valid sets of 2-point spatial correlations, simply as convex combinations. This work demonstrates the significant benefits of utilizing 2-point spatial correlations as the input to the CNN, in place of the voxelated discrete microstructures used in current benchmarks.

Keywords: convolutional neural networks, property closures, 2-point spatial correlations, convex hull, microstructure design

OPEN ACCESS

Edited by:
Roberto Brighenti,
University of Parma, Italy

Reviewed by:
Niaz Abdolrahim,
University of Rochester, United States
Hamid Akbarzadeh,
McGill University, Canada

***Correspondence:**
Surya R. Kalidindi
surya.kalidindi@me.gatech.edu

Specialty section:
This article was submitted to Computational Materials Science, a section of the journal Frontiers in Materials

Received: *09 January 2022*
Accepted: *23 February 2022*
Published: *11 March 2022*

Citation:
Mann A and Kalidindi SR (2022) Development of a Robust CNN Model for Capturing Microstructure-Property Linkages and Building Property Closures Supporting Material Design. Front. Mater. 9:851085. doi: 10.3389/fmats.2022.851085

1 INTRODUCTION

The microstructure[1] of a material has a causal relationship with its effective anisotropic properties. Therefore, it should be theoretically possible to design the microstructure for optimal performance, which is typically specified in terms of a set of desired effective material properties. In practice, the microstructure-property relationships are most commonly explored using computationally expensive physics-based simulation tools (Ghosh et al., 1995; Kalidindi and Schoenfeld, 2000; Roters et al., 2010; Wargo et al., 2012; Brands et al., 2016). However, such computational tools allow exploration mainly in the forward direction, i.e., going from given microstructures to the estimation of their effective properties. Microstructure design can be achieved through iterative evaluations of the forward model to minimize a suitably defined objective function on the targeted properties.

[1]This term is generally used to refer to the salient details of the material internal structure which often includes statistical information on the size, shape, and placement of the different material local states (e.g., thermodynamic phases).

However, the high computational expense of the physics-based forward models poses a major hurdle for such design efforts.

Many efforts in prior literature have aimed to reduce the computational cost of the forward models described above, since inverse solutions often require the execution of a very large number of forward trials. Such efforts have explored both analytical approaches as well as emergent data-driven approaches. The most impactful efforts utilizing analytical approaches have been based on Kroner's perturbation expansion of the solution for the effective elastic stiffness of a composite material (Kröner, 1971). In this formalism, the effective material property is expressed as a series whose terms systematically utilize increasingly higher-order spatial correlations (i.e., n-point spatial correlations) of the different material local states in the microstructure. In most practical applications, the series expansion is truncated to include up to 2-point spatial correlations (Torquato, 2002; Adams et al., 2013; Kalidindi, 2015). An implicit benefit of these analytical approaches is that they permit formal application of optimization methods (many of which require computation of the gradients of the specified objective function with respect to microstructural variables) in solving microstructure design problems. Examples of these efforts can be found in the development and application of the Microstructure-Sensitive Design (MSD) framework (Fullwood et al., 2007; Fullwood et al., 2008a; Fullwood et al., 2010; Adams et al., 2013). Prior MSD efforts have demonstrated the viability of designing simple microstructures (e.g., composites with two isotropic phases, single phase polycrystalline materials) to meet designer specified target properties (Adams et al., 2001; Fast et al., 2008; Knezevic et al., 2008; Shaffer et al., 2010; Adams et al., 2013). The application of the MSD framework has been largely confined to simple microstructures and simple physics due to the difficulties encountered in computing the convolution integrals involved in the series expansions. Some of the main hurdles encountered arise from the need to find Green's function solutions for the specific governing field equations and the reliable computation of the principal value (Fullwood et al., 2008a; Fullwood et al., 2010; Adams et al., 2013). Furthermore, the perturbation series expansion has also been shown to be limited in practice to moderate contrast (refers to the degree to which the local properties can change from one location to another in the microstructure) material systems (Kalidindi et al., 2006; Fullwood et al., 2010). This limitation is due to the challenges encountered in achieving convergence in the series expansion with the systematic inclusion of higher-order spatial correlations (Torquato, 2002; Fullwood et al., 2008a; Fullwood et al., 2010).

Data-driven approaches have aimed to overcome the shortcomings of the analytical approaches described above by producing low-computational cost surrogates trained on the high-computational cost physics-based numerical models [e.g., representative volume elements modeled by finite element models (FEM)]. If these surrogates exhibit adequate accuracy, their low computational cost clearly justifies their use in microstructure design efforts. A prime example of these efforts can be seen in the Materials Knowledge System (MKS) (Kalidindi et al., 2010; Landi et al., 2010; Kalidindi, 2015; Brough et al., 2017) framework, which employs a novel feature engineering approach for material microstructures by combining the formalism of the n-point spatial correlations mentioned above with machine learning tools such as the principal component analysis (PCA). The MKS framework learns the salient (low-dimensional) microstructure features in a completely unsupervised manner. These low-dimensional features are then used to build data-driven surrogate models for the reliable prediction of a broad range of material properties of interest. In typical MKS applications, these surrogate models are trained on datasets generated by physics-based numerical tools. The viability of the MKS approach has been demonstrated on a broad class of material structures and applications (Cecen et al., 2014; Brough et al., 2017; Latypov et al., 2019). In recent extensions of the MKS framework (Cecen et al., 2018; Yang et al., 2018; Eidel, 2021), convolutional neural network (CNN) based surrogates have been explored, which bypass the feature engineering steps and build structure-property relationships directly from the input voxelated microstructure volumes. These CNN-based surrogates have demonstrated excellent accuracy, even for high contrast composites (Yang et al., 2018; Eidel, 2021). Although the CNN-based models offer a highly accurate and low-computational cost tool to predict the effective property of a given microstructure, they encounter certain limitations that arise from the difficulty of incorporating known physical concepts into the CNN-based surrogate models. For example, when one imposes periodic boundary conditions on a representative volume element (RVE) of a microstructure, the predicted effective property exhibits translational invariance[2]. The most commonly used CNN architectures do not exhibit this characteristic implicitly. Most importantly, CNN-based models are prone to model overfit due to their large number of tunable parameters. Despite these limitations, recent work has demonstrated the positive impact of data-driven methods on topology optimization (Kollmann et al., 2020; Yilin et al., 2021) and inverse design of microstructures (Jung et al., 2020; Tan et al., 2020).

This work aims to combine the advantages of both the analytical and data-driven approaches described above. First, this work employs the microstructure hull concept introduced in the MSD framework, which represents the complete space of physically realizable structures in a compact and convex space. Second, this work builds CNN-based surrogates using the 2-point spatial correlation maps as inputs, as opposed to using the voxelated microstructures directly. The approaches described in this work offer many advantages: 1) The use of 2-point spatial correlations as input to the CNN models automatically imparts translational invariance. 2) The change of the input to the CNN, from the voxelated microstructures (RVEs) to the 2-point spatial correlation maps, is expected to produce a more accurate and robust surrogate model (compared to current benchmarks)

[2]This implies that if one extends the original RVE in all directions utilizing periodicity and takes a new RVE of the same size but with a different starting point, its effective property would be exactly the same as the original RVE.

with a significantly smaller number of trained parameters (and the associated training cost). 3) The proposed strategy allows one to explore the complete space of possible property combinations in a highly practical manner (by limiting the exploration to the 2-point spatial correlations hull); this construct has been termed as property closure in prior work (Proust and Kalidindi, 2006; Fullwood et al., 2007; Wu et al., 2007; Knezevic et al., 2008; Fullwood et al., 2010; Adams et al., 2013). The property closure produced in this work represents a significant advance from the closures produced in prior literature in terms of both computational cost and accuracy.

2 BACKGROUND

2.1 Microstructure Quantification

Discretized representations have been used extensively in the mathematical representation of the material microstructures (Adams et al., 2013). In these, the microstructure is most conveniently represented as an array, m_s^h, whose values reflect the volume fraction of the material local state, h, in the spatial bin, s, in the RVE. In this formalism, the local states are used to index the salient local material attributes (e.g., thermodynamic phase identifiers) and the spatial bins are produced through a uniform tessellation of the RVE (equivalent to pixels in 2-D and voxels in 3-D). Implicit in this representation is the assumption that there exist a finite number of distinct material local states, $h = 1, 2, \ldots, H$, which are allowed to occupy each spatial bin in the RVE, $s = 1, 2, \ldots, S$, to define the microstructure of interest. In this study, we will employ a 3-D vector index $\boldsymbol{s} = (s_1, s_2, s_3)$ for indexing the spatial bins in a 3-D RVE.

Of primary interest to this paper are the 2-point spatial correlations, which are captured in a discretized representation as an array denoted by $f_{\boldsymbol{r}}^{hh'}$. The elements of this array reflect the probability of finding local states h and h' in the RVE separated by a discretized vector indexed by an integer array $\boldsymbol{r}$ (very similar to $\boldsymbol{s}$). Mathematically, 2-point spatial correlations are defined as (Kalidindi, 2015)

$$f_{\boldsymbol{r}}^{hh'} = \frac{1}{S}\sum_{\boldsymbol{s}} m_{\boldsymbol{s}}^{h} m_{\boldsymbol{s}+\boldsymbol{r}}^{h'} \tag{1}$$

Where S denotes the total number of spatial bins in the RVE. In practice, $f_{\boldsymbol{r}}^{hh'}$ are most efficiently computed by taking advantage of the fast Fourier transform (FFT) algorithm (Niezgoda et al., 2008; Cecen et al., 2016).

2.2 Microstructure Hulls and Property Closures

We will restrict our attention in this work to periodic eigen microstructures. Eigen microstructures are defined as a special class of microstructures where each spatial bin is fully occupied by only one material local state. In other words, m_s^h are allowed to take only the values of either zero or one. Most experimentally observed microstructures are commonly depicted as eigen microstructures, with the spatial bin size limited by the resolution limits of the characterization machine (e.g., microscope). Moreover, most structural composites exhibit thermodynamic phase regions separated by sharp boundaries. Therefore, in practice, the discretization error arising from the use of eigen microstructure representations is largely restricted to the voxels next to the phase boundaries; this error can be controlled through the selection of a sufficiently small spatial bin size. The assumption of periodicity makes the microstructure representations consistent with the typically imposed boundary conditions in the finite element modeling of the RVEs for the estimation of their effective (bulk) mechanical properties (e.g., elastic stiffness, yield strength) (Cecen et al., 2014; Brough et al., 2017; Latypov et al., 2019).

Prior work in the development of the MSD framework (Niezgoda et al., 2008) has demonstrated that the complete space of 2-point spatial correlations (i.e., the set of all theoretically possible 2-point spatial correlations) can be depicted as a convex (and compact) hull. Generally referred as a microstructure hull, this construct delineates the complete space of inputs (i.e., design space) that needs to be considered in microstructure design. It should be recognized that the space of the 2-point spatial correlations is significantly smaller than the space of all microstructures, since microstructures related to each other by translations and/or inversions have the exact same set of 2-point spatial correlations (implied from **Eq. 1**). This is indeed one of the main advantages of using spatial correlations to represent the microstructure in design efforts; the microstructures that have been filtered out exhibit the exact same effective mechanical properties as the ones retained in the 2-point spatial correlations hull. Therefore, they effectively remove many of the redundancies in the design space. Although higher-order spatial correlations (i.e., 3-point spatial correlations and higher) are known to influence the effective properties, they are expected to have (currently unknown) non-linear relationships with the 2-point spatial correlations, at least for the class of eigen microstructures considered in this work. This can be inferred from the fact that it is possible to reconstruct exactly the eigen microstructures from their 2-point spatial correlations (Fullwood et al., 2008b). One of the important practical consequences of the concepts presented above is that one can construct a new set of valid 2-point spatial correlations as a convex combination of the 2-point spatial correlations of known microstructures. This realization offers an attractive avenue for exploring efficiently the space of microstructures without having to instantiate them directly. In other words, we can explore the space of $f_{\boldsymbol{r}}^{hh'}$ much more easily than the space of $m_{\boldsymbol{s}}^{h}$. This is because it is not possible to span the space of eigen microstructures simply as convex combinations of previously known eigen microstructures.

Another advance from the MSD framework related to the present work is the concept of a property closure (Proust and Kalidindi, 2006; Wu et al., 2007; Fullwood et al., 2007; Knezevic et al., 2008; Fullwood et al., 2010; Adams et al., 2013). The property closure delineates the complete set of effective (bulk) property combinations in a selected material system, which are theoretically realizable through the modulation of its microstructure. Property closures are extremely valuable in

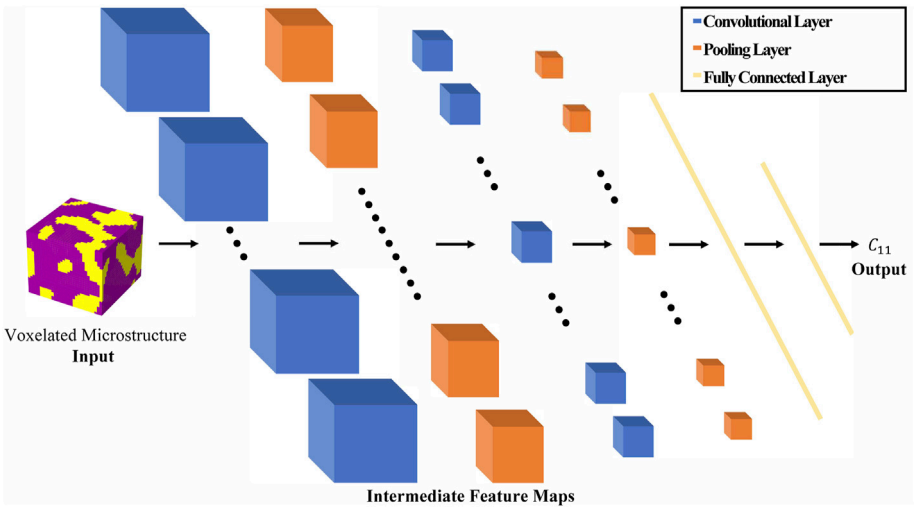


FIGURE 1 | Schematic of a typical CNN architecture consisting of convolutional layers, pooling layers, and fully connected layers.

engineering design because they represent the complete set of property combinations that can be leveraged for the optimization of the part performance. This is particularly important for heterogeneous design where the microstructures are intentionally varied throughout the part to optimize the overall part performance. Prior work in the MSD framework focused on computationally efficient algorithms for mapping the microstructure hulls into property closures. As already mentioned, the case studies reported to date in the MSD framework have relied on Green's function-based analytical models for microstructure-property relationships, which were themselves restricted to relatively simple material physics (i.e., constitutive models) and low to moderate contrast composites (Kalidindi et al., 2006; Proust and Kalidindi, 2006; Adams et al., 2013).

2.3 Convolutional Neural Networks

Neural networks (Schmidhuber, 2015) have shown to be powerful tools for learning highly complex non-linear mappings between selected inputs and the targets (i.e., outputs) in a wide range of application domains. Indeed, under certain conditions, neural networks can be shown to be universal function approximators (Cybenko, 1989; Pinkus, 1999). Convolutional neural networks (CNNs) are a special class of neural networks that perform exceptionally well for problems involving spatial fields as inputs. CNNs have been successfully deployed in a variety of image analyses and machine vision applications (Lecun et al., 1998; He et al., 2016; Krizhevsky et al., 2017). Since microstructures are spatial fields, CNNs are ideally suited to explore microstructure-property relationships (Yang et al., 2018; Rao and Liu, 2020; Eidel, 2021). The central advantage of CNNs is that they circumvent the need for explicit feature engineering of the complex input spatial fields. In other words, the feature engineering occurs implicitly in the CNN during the model training process.

The primary components of a typical CNN are the convolutional layers, the pooling layers, and the fully connected layers, all of which are used to transform systematically the input into the desired target. **Figure 1** depicts schematically a typical CNN architecture, where each block represents the transformed input (i.e., feature map) and the mathematical operations between the blocks are performed using one of the types of layers described above. The number of transformation layers and their characteristics (e.g., number of channels in each layer, type of non-linear activation employed, kernel size) are considered as hyperparameters of the CNN architecture, and are generally optimized for a specific application through multiple trials. As the size of the network is increased, the model accuracy and the computational cost of the training generally increases. However, increases in network size are often accompanied by increases in the number of learned (i.e., model-fit) parameters. Consequently, larger networks are prone to be model over-fits, especially when using a limited training dataset. Model over-fit is generally assessed through some form of cross-validation (Bishop, 2006; Hastie et al., 2009). Therefore, one aims to build a *robust* CNN model that provides high model accuracy, while avoiding over-fit.

The reader is referred to various excellent texts (e.g., LeCun et al., 2015; Emmert-Streib et al., 2020; Zhang et al., 2021) for an introduction to the basics of machine learning. Here, we briefly present the primary components of the CNNs discussed in this work. A convolutional layer applies a linear transformation (performed as a convolution of a learned kernel on the input) followed by a non-linear activation applied pointwise on the feature map. The PReLU activation function defined below has been used in this work:

$$\mathrm{PReLU}(x) = \begin{cases} x, & if\ x \geq 0 \\ ax, & otherwise \end{cases} \tag{2}$$

A stride (Goodfellow et al., 2016; Zhang et al., 2021) can be implemented in the convolution operation to effectively coarsen the input, which helps reduce the size of the transformed feature map (this is needed as most CNNs start with high-dimensional spatial maps as input and produce low dimensional targets; see **Figure 1**. Pooling (Goodfellow et al., 2016; Zhang et al., 2021) is another dimensionality reduction technique that is used extensively in CNN architectures. Specifically, this work employs global average pooling where a feature map produced in an output channel is simply replaced by its average. Most CNN architectures implement fully connected layers in the last few transformation layers. Unlike the convolution layers, fully connected layers treat each input feature independently in the linear transformation. Consequently, fully connected layers dramatically increase the expressivity of the CNN models along with a concomitant increase in the number of model-fit parameters. Consequently, the fully connected layers also make the CNN models prone to over-fit.

CNNs have been successfully employed to model the microstructure-property relationships in heterogeneous (composite) material systems (Yang et al., 2018; Rao and Liu, 2020; Eidel, 2021). However, the CNN architectures designed for predicting the effective property of a microstructure have not differed significantly from the CNNs designed for machine vision problems (Lecun et al., 1998). For example, Yang et al. (2018) employed a CNN to predict the C_{1111} component of the effective elastic stiffness tensor of a composite using a voxelated representation of its RVE as the input. Their CNN model contained approximately 3.2 million trainable parameters, the majority of which reside in the fully connected layers. More recently, Eidel (2021) improved the CNN model from Yang et al. (2018) by significantly reducing the number of neurons in the fully connected layers, which correspondingly reduces the number of trainable parameters. However, the CNN architecture still has approximately 2.9 million trainable parameters, of which approximately 1.8 million parameters are in the fully connected layers. Because the models are trained using relatively small datasets, typically containing only about 10^4 (Yang et al., 2018; Eidel, 2021) training datapoints, the chances of the CNN model being an over-fit are significant. As already discussed, the CNN architectures used in the prior microstructure-property modeling efforts exhibit the following additional limitations: 1) the use of discrete microstructures as input makes it difficult to employ them in exploring inverse microstructure-design solutions (because of the difficulty in delineating the unimaginably large input domain), and 2) they do not automatically reflect the desired translation invariance.

3 NEW PROTOCOL FOR BUILDING PROPERTY CLOSURES

This work proposes a new protocol for constructing property closures that leverages the prior advances made in both the MSD and MKS frameworks. More specifically, the proposed protocol combines the concepts of 2-point spatial correlations and their hulls developed in the MSD framework (Adams et al., 2013) with a new CNN architecture that avoids the use of fully connected layers. The proposed protocol involves two main steps: 1) building a robust surrogate model that captures the 2-point spatial correlations-property linkage of interest using the new CNN architecture, which employs a much lower number of model fit-parameters compared to current benchmarks (Yang et al., 2018; Eidel, 2021), and 2) constructing the property closure by systematically exploring the 2-point spatial correlations hull with the new CNN model. This new protocol is developed and demonstrated in this paper for constructing the property closure for selected components of the effective elastic stiffness tensor in a high-contrast composite material system.

3.1 Convolutional Neural Network Model for Microstructure-Property Linkages

The first step of the proposed protocol for building property closures is to establish a robust surrogate model that takes 2-point spatial correlations as the input and predicts the effective properties of interest. The many benefits that could come from the use of 2-point spatial correlations as the input instead of the discrete microstructure have already been discussed earlier. It is emphasized here that the features identified by the 2-point spatial correlations are expected to serve as universal features for all effective anisotropic material properties of interest (Garmestani et al., 1998; Cecen et al., 2014; Gupta et al., 2015; Kalidindi, 2015; Paulson et al., 2017; Yabansu et al., 2020; Generale and Kalidindi, 2021). Therefore, it should be possible to create microstructure-property surrogates capable of concurrently predicting multiple effective anisotropic material properties. It is further emphasized that the relationships of interest have to be necessarily formulated in the direction of microstructure → property (and not in the inverse direction) as these are expected to be many-to-one relationships. In other words, microstructures exhibiting different 2-point spatial correlations can produce the exact same effective property, because the effective property reflects a suitably averaged bulk response of the material[3].

A few example RVEs and their 2-point spatial correlation maps are presented in **Figure 2**. It should be noted that 2-point spatial correlation maps are continuous spatial fields with a natural origin corresponding to the zero vector (i.e., $\boldsymbol{r} = 0$ in **Eq. 1**). Note that the 2-point spatial correlation maps exhibit a sharp peak at $\boldsymbol{r} = 0$ and generally decrease with increasing $|\boldsymbol{r}|$. More specifically, the peak value is equal to the phase volume fraction, while the asymptotic value at large $|\boldsymbol{r}|$ is equal to the square of the phase volume fraction. Additionally, the 2-point spatial correlations capture a number of other important details of the microstructure, including size and shape distributions (Fullwood et al., 2010; Kalidindi, 2015). Indeed, it is seen from **Figure 2** that the shape of the central peak in the 2-point spatial correlations mimics the average shape of the features in the

[3]The reader is pointed to prior work on iso-property surfaces in microstructure hulls (Knezevic and Kalidindi, 2007; Fullwood et al., 2010) to visualize the many-to-one microstructure-property linkages.

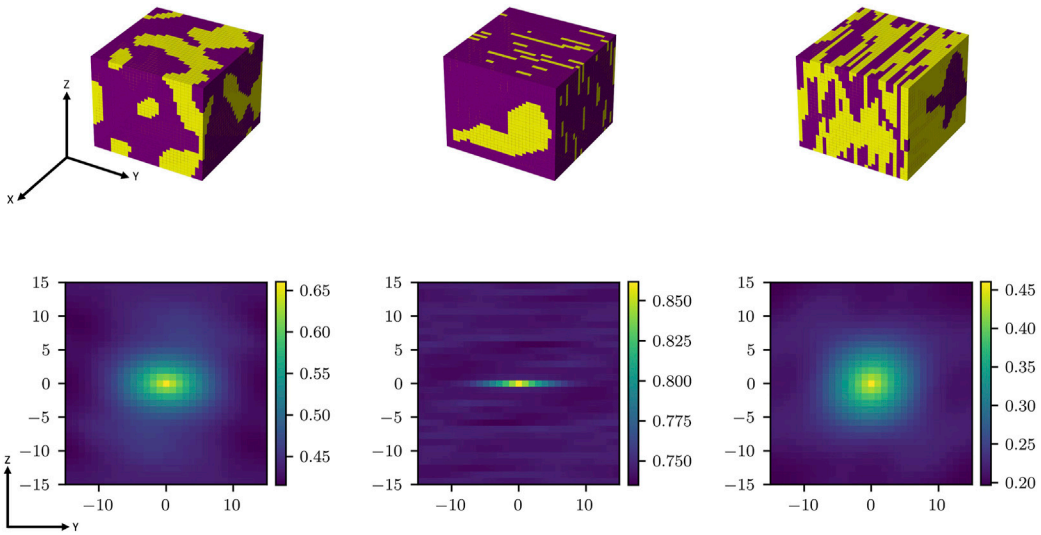


FIGURE 2 | Example 3-D RVEs with the corresponding Y-Z mid-sections of their 2-point spatial correlations. It is seen that the 2-point autocorrelations capture various salient measures of the microstructures (for example, the center peak in these plots reflects the phase volume fraction and the shape of the central peak region reflects the average shape of the phase regions).

microstructure. The presence of the natural origin at $\boldsymbol{r} = 0$ imparts the desired translational invariance of the effective properties described earlier. The 2-point spatial correlations maps used as inputs to the CNNs developed in this work include vectors with x, y, and z components in the integer range $\{-15, \ldots, 0, \ldots, 15\}$. Although the vectors defining spatial correlations have physical units of length, they are used here with arbitrary units (these default to voxels of unit size in this work), as is typically done in FEM of composite RVEs. This is because the material constitutive law (i.e., Hooke's law for elasticity) employed in these models does not contain any characteristic length scales. Consequently, the values of the effective stiffness of the RVE are completely independent of the physical length of each voxel in the RVE. However, one must pay attention to the voxelization itself in creating the RVEs. The use of coarser voxels will lead to inaccurate representation of the smallest features (i.e., phase regions) in the RVE, while the use of finer voxels will increase the computational cost. Prior work (Latypov et al., 2019; Marshall and Kalidindi, 2021) has utilized successfully RVEs of resolution $27\ X\ 27\ X\ 27$ in modelling the homogenized plastic response of composite microstructures. The resolution of the RVEs in this work was increased to $31\ X\ 31\ X\ 31$ to allow for a slightly improved representation of the RVEs used to capture the salient microstructure-property linkages for the present application. Further increase in the RVE resolutions was not possible for the present study because of the high computational cost involved. As a result of the considerations described above, the 2-point spatial correlation map used as input to the CNN was standardized as a 3-D array of size $31\ X\ 31\ X\ 31$ (with $\boldsymbol{r} = 0$ corresponding to the element (16, 16, 16) of this array). Note that the elements of this array take continuous values only in the range $[0, 1]$.

Prior work (Garmestani et al., 1998; Cecen et al., 2014; Gupta et al., 2015; Kalidindi, 2015; Paulson et al., 2017) has also shown that the 2-point spatial correlations can serve as universal features for correlating the microstructure to its many different effective (bulk) properties. The theoretical justification for this claim is most clearly seen in the statistical continuum theories formulated by Kröner (1971). Consequently, the use of 2-point spatial correlations as input can offer attractive avenues for creating high-fidelity multi-output CNN models, where each output corresponds to a different effective property of interest. In other words, one can aim to build CNN architectures that learn the common salient microstructure features that are capable of making sufficiently accurate predictions for the different effective properties of interest. Such multi-output CNNs would implicitly account for cross-correlations between the different effective properties of the RVE, making the predictions more reliable and robust. In this work, we will specifically explore multi-output CNNs for the predictions of C_{1111} and C_{1212} components of the effective elastic stiffness tensor for high contrast composites.

The use of 2-point spatial correlations, instead of the voxelated microstructures, as input to the CNN essentially constitutes feature engineering. It should be recognized that most applications of CNNs do not apply any feature engineering steps. In fact, CNNs are generally touted as model building approaches that do not require feature-engineering. However, for our application, the established physics (i.e., statistical continuum mechanics theories (Kröner, 1971; Torquato, 2002)) has already proven that the 2-point spatial correlations can serve as versatile microstructural features with a number of desired characteristics described earlier. CNN models are ill-equipped to learn the 2-point spatial correlations from the discrete microstructures by themselves, because the

TABLE 1 | Examples of different CNN architectures explored in this work along with the relevant benchmarks from literature. The notation a@b/c indicates a axaxa kernel, applied with a stride of c, and b channels. The default value of stride, when not mentioned, is one.

Model	Yang et al. (2018)	Eidel (2021)	A	B	C	D
Input	**51 × 51 × 51 Microstructure**	**51 × 51 × 51 Microstructure**	**31 × 31 × 31 Spatial Correlations**	**31 × 31 × 31 Microstructure**	**31 × 31 × 31 Spatial Correlations**	**31 × 31 × 31 Spatial Correlations**
L1	3@16	3@16	5@8/2	5@8/2	5@64/2	5@8/2
L2	MaxPool	MaxPool	3@16	3@16	3@32	3@16
L3	3@32	3@32	3@32	3@32	3@16	3@32
L4	MaxPool	MaxPool	3@64	3@64	3@8	3@64
L5	3@64	3@64	1@2	1@2	1@2	3@4
L6	MaxPool	MaxPool	AvgPool	AvgPool	AvgPool	6@2
L7	3@128	3@128	—	—	—	—
L8	MaxPool	MaxPool	—	—	—	—
L9	3@256	3@256	—	—	—	—
L10	MaxPool	MaxPool	—	—	—	—
L11	FC-2048	FC-256	—	—	—	—
L12	FC-1024	FC-128	—	—	—	—
Number of Outputs	1	27	2	2	2	2

auto-correlations and cross-correlations are not easily approximated by the various transformation layers used in the CNNs. Thus, it is likely to be much more beneficial to first compute the 2-point spatial correlations as a feature engineering step, and subsequently use them as inputs into a CNN model for predicting the effective properties. One of the main benefits anticipated would be a dramatic reduction in the number of model fit parameters. A critical evaluation of this hypothesis is one of the main goals of this work.

Towards the goals described above, we primarily focused on designing CNN architectures for our study that do not have any fully connected layers. After a few trials, we arrived at Model A (see **Table 1**) that showed better accuracy than the benchmarks with far fewer number of trainable parameters (discussed in more detail in the next section). We believe that this dramatic reduction of model complexity is attributable to the fact that we are using the 2-point spatial correlations as input to the CNN, in place of the voxelated microstructure. In order to validate this hypothesis, we created Model B (see **Table 1**), with the only difference from Model A coming from the use of the voxelated microstructure as the input to the CNN. It was also generally observed that the CNN models produced in this work needed far few layers compared to the benchmarks. This is because of the already feature-engineered inputs (i.e., 2-point spatial correlations). Furthermore, the architectures explored in this work achieved the needed dimensionality reduction in the feature maps by using stride in the first convolutional layer. A final dimensionality reduction is accomplished using global average pooling in the final layer, effectively transforming the feature maps into scalar outputs (i.e., targets). Note that the CNN architectures explored in our work are drastically simplified compared to the current benchmarks (see **Table 1**).

The architectures of Models A and B follow the current benchmarks in that they start with a relatively small number of channels in the first layer and systematically increase the number of channels in the subsequent layers. Consequently, these conventional architectures aim to initially identify a smaller number of macroscale features and then further transform them in the subsequent layers into the salient features that strongly correlate with the output. We also explored the possible benefits of inverting this architecture, i.e., starting with a large number of channels in the first layer and systematically reducing the number of channels in subsequent layers. The general idea of these inverted architectures is that they allow the initial capture of a large number of potential features, and subsequently transform them to a smaller number of salient features. Another advantage is that these inverted architectures are more naturally aligned with the dimensionality reduction needed in our application - from the higher-dimensional input to the low-dimensional output. Model C (see **Table 1**) shows an example of this architecture. It was also observed that the global average pooling in the last layer was essential for producing high-fidelity CNN models for our application. In an effort to critically validate this concept, we created several CNN architectures that avoided the use of the average pooling layer and accomplished the necessary dimensionality reduction exclusively through the use of convolutional layers. Model D (see **Table 1**) shows one such architecture, which is very similar to Model A. Note also that the change from Model A architecture to Model D architecture actually increases the number of trainable parameters and allows for richer non-linear transformations (i.e., increases model expressivity). It will be shown in the next section that this increase in model expressivity dies not necessarily result in an improvement in model fidelity. The performance of Models A through D will be discussed extensively in the next section.

The dataset employed in this work to train the CNN networks consisted of 20,480 two-phase microstructures and their corresponding FEM-estimated C_{1111} and C_{1212} components of

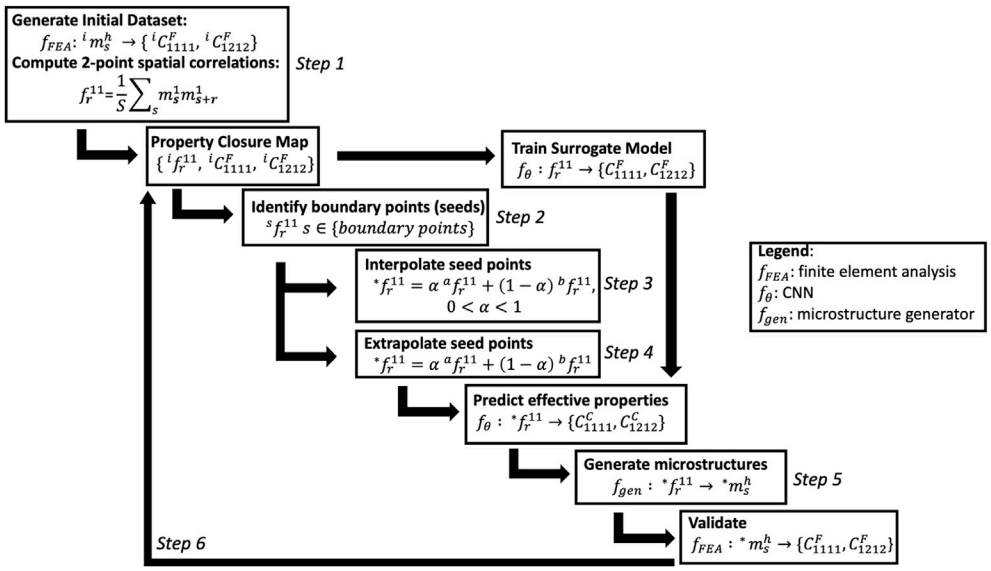


FIGURE 3 | A flow chart of the proposed protocol.

the effective elastic stiffness tensor. The details of the generation of the microstructures and the finite element models used in the estimation of the effective elastic properties have been described in prior work (Kelly and Kalidindi, 2021). The microstructures exhibited volume fractions ranging from 0 to 100%. For the two-phase microstructures studied in this work, only the f_r^{11} are independent (Niezgoda et al., 2008). Consequently, only these are used as input to the CNNs. The finite element models used for the estimation of the effective properties used periodic boundary conditions (Landi et al., 2010). Specifically, the C_{1111} component of the elastic stiffness tensor was estimated by imposing an average uniaxial strain of $\bar{\varepsilon}_{11} = 0.001$ on the RVE, computing the average stress $\bar{\sigma}_{11}$ from the FE simulation, and dividing it by the imposed strain (0.001). The C_{1212} component of the effective elastic stiffness tensor was similarly evaluated by imposing an average shear strain. The elastic properties of the two isotropic phases used in this study were assigned using Poisson ratios (ν_1, ν_2) and Young's Moduli (E_1, E_2). The following specific values were used: $\nu_1 = \nu_2 = 0.3$, $E_1 = 120$ and $E_2 = 6,000$ (as already noted, in the FEM simulations, these are associated with arbitrary but consistent units). These selections correspond to a high contrast ratio of 50 (defined by the ratio $\frac{E_2}{E_1}$) for the composite material system studied here. Because of the linearity of the elasticity problem, the CNN model generated here can be applied to any composite material system with the same contrast value and the same values of the Poisson ratios, simply by applying a suitable scaling factor. It should be noted that although the level of contrast explored in this work is relatively high, it is still far less than the infinite contrast experienced in porous solids (for example, commonly encountered in additively manufactured components). We do believe that the microstructure design strategy proposed here is extendable to such extremely high contrast composites.

3.2 Property Closure Construction

In principle, the property closure should be constructed by mapping the complete 2-point spatial correlations space to the property space of interest. As already mentioned earlier, the protocol developed and implemented in this study aims to take full advantage of the fact that the complete space of the 2-point spatial correlations delineates a convex hull. The mapping between the 2-point spatial correlations and the effective properties of interest will be approximated by a suitable multi-output CNN model. Although the 2-point spatial correlations space is continuous and convex, it is still too large to explore using brute-force approaches. A clever strategy is therefore needed to successfully explore the complete space of 2-point spatial correlations and map it into the property space of interest. The following protocol (see **Figure 3**) is designed and implemented in this work:

Step 1: Create a large initial set of voxelated eigen microstructures and compute their 2-point spatial correlations. Additionally, estimate their corresponding effective properties using suitable finite element models (i.e., applying periodic boundary conditions). Build an initial estimate of the property closure using this initial dataset.

Step 2: Using a suitable algorithm (such as a convex hull algorithm), identify the boundary points of the current estimate of the property closure. This work employed the Quickhull algorithm (Barber et al., 1996), which efficiently identifies the boundary points of a convex hull defined by a set of points. These boundary points reflect extreme combinations of the properties of interest (within the current estimate of the property closure). The boundary points are updated after each iteration of the proposed protocol until the area enclosed by the

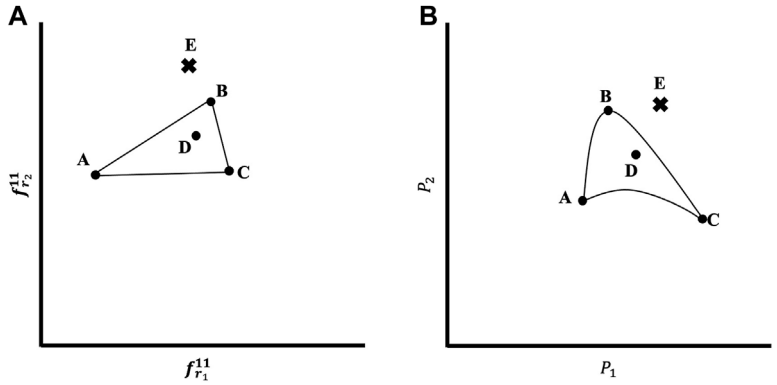


FIGURE 4 | (A) Schematic illustration of the generation of new microstructures (defined in terms of 2-point spatial correlations) as interpolations or extrapolations. **(B)** Schematic depiction of the non-linear mapping of microstructures to the property space using a CNN model developed in this work.

boundary points does not change significantly. The microstructures corresponding to these boundary points are identified as seeds for the generation of new microstructures of interest in the next step.

Step 3: Let ${}^{a}f_r^{11}$ and ${}^{b}f_r^{11}$ represent the 2-point spatial correlations of two selected seeds identified in Step 2. Generate new microstructures by taking convex combinations of the 2-point spatial correlations of any selected pair of seeds from Step 2. In other words, create new microstructures as ${}^{*}f_r^{11} = \alpha {}^{a}f_r^{11} + (1-\alpha) {}^{b}f_r^{11}$ with $0 < \alpha < 1$. Create as many new microstructures as needed by selecting different seeds and varying the weights of the convex combination. Ideally, one may want to focus their expansion efforts on regions of the property closure that are currently lightly populated. In other words, take convex combinations of the 2-point spatial correlations that correspond to effective property points that reside in the sparse regions of the property closure. One can also use more than two seeds at a time. As long as the weights are positive and sum to one, the generated microstructure represents convex combination of the seeds. As an example, **Figure 4A** depicts generation of new microstructures using three seeds, labelled as A, B, and C. All of the convex combinations are represented by points inside the triangle ABC. Microstructure D represents an example of such an interpolation. Using the CNN surrogate model, estimate the effective properties of interest for all of the new microstructures generated in this step. Add these new estimated properties to the set of points that currently approximates the property closure. Note the mapping between the microstructure space and the property space is expected to be highly non-linear, as illustrated in **Figure 4B**. Since a CNN provides a smooth mapping between the input and the output (Hastie et al., 2009), the triangular region ABC in the microstructure space would map to a (non-linearly) distorted but continuous triangular region with curvilinear sides in the property space.

Step 4: In this step, we will focus on extrapolations (Step 3 only used interpolations) by essentially following the same process as in Step 3, while relaxing the requirement that all weights are positive. Extrapolations are usually produced by using at least one negative weight, while requiring the weights add to one. However, we will only allow acceptable new microstructures by requiring that all values of ${}^{*}f_r^{11}$ lie between zero and one (this condition is automatically satisfied in the interpolations in Step 3). Microstructure E in **Figure 4A** represents an example of the generation of a new microstructure through an extrapolation. Once again, it might be prudent to focus the generation of new microstructures in this step to the sparsely populated regions in the current estimate of the property closure.

Step 5: Validate the new microstructures added in Step 3 and 4 as needed. In particular, we note that our confidence is much higher in the 2-point spatial correlations generated as interpolations in Step 3, compared to those generated in Step 4 as extrapolations. In this work, we only validated selected new points on the expanded boundaries of the property closure. For the validation, one would have to generate a discrete microstructure corresponding to the known 2-point spatial correlations using one of the established approaches in literature (Fullwood et al., 2008b; Robertson and Kalidindi, 2021), and estimate its effective property using a suitable FEM simulation.

Step 6: Iterate Steps 2–5 as needed, while continuously adding the validated new points collected in each iteration into the current estimate of the property closure. This might necessitate re-training of the CNN model after each iteration.

The central hypotheses behind the protocol described above is that the interpolations and extrapolations of the 2-point spatial correlations can effectively explore the complete microstructure space. Moreover, since the interpolations and extrapolations are conducted using promising seeds, the protocol naturally allows

TABLE 2 | Summary of the performance of selected CNN models produced in this work and their comparison with benchmarks form literature.

Model	No. of Trainable Parameters	Dataset (train/valid/test)	NMAE (%)	
			C_{1111}	C_{1212}
Yang et al. (2018)	3,272 K	3,819/1,881/2,850	3.10	—
Eidel (2021)	2,981 K	7,000/2,000/1,000	1.11	1.15
A	73.9 K	12,288/3,072/5,210	0.72	0.66
B	73.9 K	12,288/3,072/5,210	1.73	1.66
C	80.8 K	12,288/3,072/5,210	0.90	0.80
D	82.5 K	12,288/3,072/5,210	0.96	0.86

targeted exploration of promising regions of the property closure. As will be shown in the next section, the CNN model facilitates a sufficiently accurate non-linear mapping of the 2-point spatial correlations into the property space.

4 RESULTS AND DISCUSSION

The CNN architectures described in **Section 3.1** were implemented in PyTorch (Paszke et al., 2019), and the property closure construction protocol described in **Section 3.2** was implemented in a Python code. The set of 20, 480 data points described above was split into independent train (60%), validation (15%), and test (25%) groups. Training was conducted on a single NVIDIA V100 GPU with 16GB of memory, utilizing the Adam optimizer (Kingma and Ba, 2017) with a cosine annealing learning rate (Loshchilov and Hutter, 2017) and a training batch size of 32. The Mean Absolute Error (MAE) loss function was utilized in this work for training the CNN model. The Mean Squared Error (MSE) loss function was also explored. However, it was found that the models trained utilizing MAE produced improved learning characteristics compared to the models trained with MSE for the present application. The CNN architectures were trained for multiple epochs until the MAE loss function converged to a minimum value. Based on multiple trials, the number of epochs was fixed at 480 epochs. Using a fixed number of epochs allows for a critical comparison of the performances of the different CNN architectures explored in this work.

4.1 Development of the Convolutional Neural Network Model for Microstructure-Stiffness Linkages

As previously described, the development of a robust CNN model requires multiple trials in which the hyperparameters of the CNN architecture are systematically varied to evaluate their influence on the model fidelity. The main types of architectures explored were summarized in **Table 1**. The accuracy of the different CNN models produced were evaluated using the Normalized MAE (NMAE) percentage defined as

$$NMAE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{S_i - \hat{S}_i}{S_{average}}\right| \times 100 \tag{3}$$

Where S_i represents the ground-truth value (established here using FEM) for the output, $\hat{S}_i$ represents the CNN prediction, and $S_{average}$ denotes the ensemble average of the ground-truth values from a set of N observations.

Table 2 summarizes the NMAE percentages for some of the best models produced in this work, along with the corresponding values from the benchmarks reported in literature (Eidel., 2021, Yang et al., 2018). The table also summarizes the number of trainable parameters in each model as well as the sizes of the training/validation/test sizes employed in building and validating each model. As the table shows, the number of trainable parameters for each of the CNN models developed in this work is significantly smaller than those used in the current benchmarks. This is primarily because we have built our CNN models without using any fully connected layers. It is also worth noting that Eidel (2021) drastically reduced the number of neurons in the fully connected layers, when compared to Yang et al. (2018). However, even relatively smaller fully connected layers produce a very large number of trainable parameters. This is mainly because the feature maps produced at the end of the convolutional layers in the benchmark models are high-dimensional, and their reduction to a small number of features that are correlated to the outputs using the fully connected layers introduces a large number of trainable parameters. Although the Eidel (2021) model demonstrated higher accuracy than the Yang et al. (2018) model, it should be recognized that the former used significantly more training data and a much smaller test data set. As a result of these important differences between them, it is not possible to conclude conclusively that the Eidel (2021) model performance is demonstrably better than that of the Yang et al. (2018) model. However, it is clear from **Table 2** that the performances of Models A, C, and D (all of which used 2-point spatial correlations as the input) are significantly better than the benchmarks, both in terms of the prediction accuracy as well the number of trainable parameters.

Comparing Models A and B, it becomes clear that changing the input to the CNN from the discrete microstructure to its 2-point spatial correlations produced a marked improvement in the model accuracy. This confirms the central hypotheses we laid out earlier. Since Model B exhibited comparable or better performance than the benchmarks (keeping in mind the larger and more diverse test sets used in this study) with a significantly smaller number of trainable parameters, it is argued that the CNN architectures without fully connected layers produce more robust models for our applications.

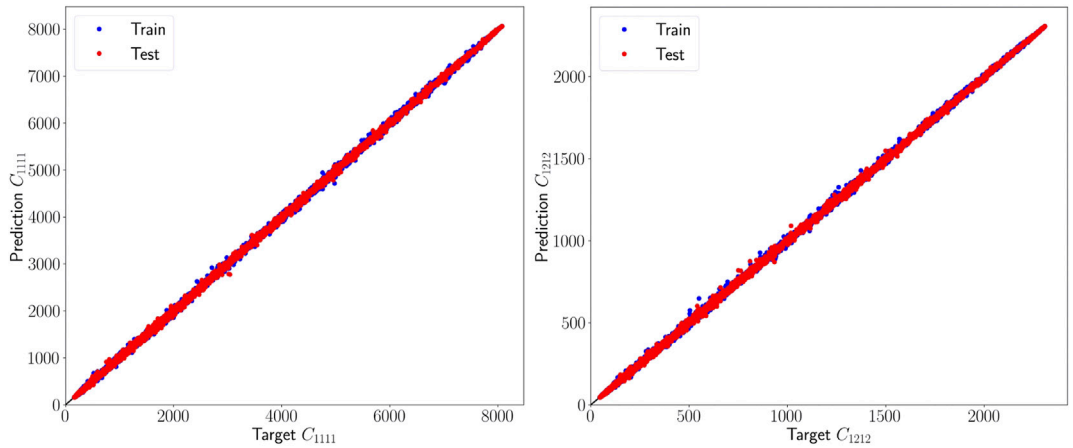


FIGURE 5 | Parity plot showing the accuracy of Model A. The test points (red) are superimposed on the train points (blue). It is seen that both the train and test sets exhibit high levels of accuracy consistent with each other.

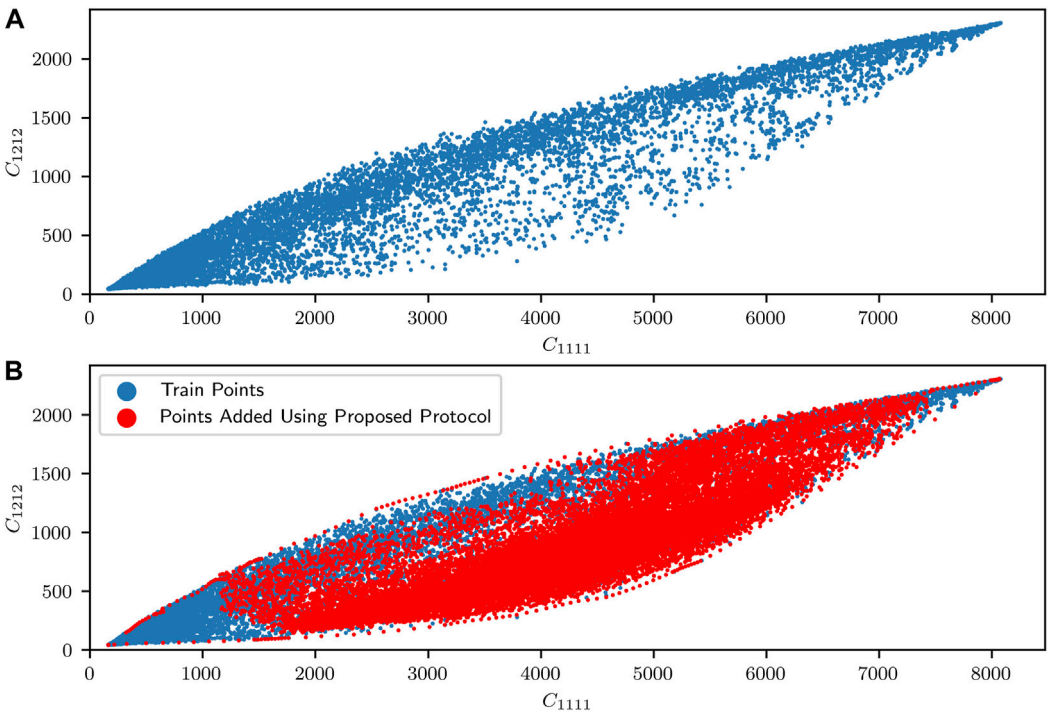


FIGURE 6 | (A) Original estimate of the property closure produced using the dataset generated to initially train the CNN surrogate model. **(B)** Updated property closure using the protocol presented in **Section 3.2**. The red points represent the new points generated in the process of building the property closure.

The improved performance of Model A over Model D underscores the need and benefits of using global average pooling as the final transformation layer for the microstructure-property CNNs. This is counter-intuitive, especially since Model D actually exhibits a higher model expressivity (i.e., it is capable of capturing more non-linear mappings). We believe the main reason for the improved performance of Model A over Model D is that the global average pooling in the last layer essentially serves as a model tree, where predictions from multiple models are averaged to produce the final prediction. Model tree strategies have been shown to improve the robustness of the surrogate models in other applications (Ho, 1995; Breiman, 2001).

Overall, it is seen that Model A outperforms all the CNN models developed to date. Most impressively, it used only 73.9 K model parameters. This is 1-2 orders of magnitude lower than the number of trainable parameters used in the current benchmarks for the same problem. As such, this model represents a significant advance in the proper use of CNNs in capturing the highly non-linear and elusive microstructure-property linkages of interest to materials design efforts. The high accuracy and robustness of the multi-output Model A can also be confirmed in the parity plots shown in **Figure 5**. The fact that the architecture of Model A is able to make simultaneous accurate predictions for multiple effective properties opens new research avenues for future materials design efforts.

4.2 Elastic Property Closure Using Convolutional Neural Network Model

The protocol established in **Section 3.2** for building property closures was implemented here using Model A. **Figure 6A** shows the initial estimate of the property closure using all 20,480 data points generated for training and validating the CNN model in **Section 4.1**. It is clear that the property closure space is not well sampled by the initial dataset. This is to be expected because the protocols used to generate the microstructures only aimed to cover as many diverse microstructures as possible. They were not in any way informed by the effective properties associated with the microstructures. Also, generating the microstructures that produce a uniform sampling of the property space are especially difficult in our application, because the input is essentially a discretized 3-D eigen microstructure.

As already described, the central advantage of the protocol described in **Section 3.2** is that it allows us to generate new datapoints in selected regions of the property space. This is evident in **Figure 6B**, where 26,825 new datapoints (shown in red) were generated using interpolations and extrapolations in the convex hull of the 2-point spatial correlations. Note that these new datapoints were targeted to lie in specific regions of the property closure. This ability to generate new microstructures corresponding to selected regions of the property space at low computational cost is unprecedented, and is only possible because the CNN model was established using 2-point spatial correlations as the input. The central consequence of the property closure shown in **Figure 6B** is that it is now possible to trivially produce a large number of microstructures that correspond to any designer-specified combination of properties within the property closure.

It is emphasized here that the property closure presented in **Figure 6B** is the first of its kind. All previously reported property closures either used grossly simplified descriptions of the microstructure (e.g., one-point statistics) or substantially degraded models (e.g., truncated expansions, primitive bounds). As such, the property closure presented in **Figure 6B** represents the most accurate depiction to date of the property closure for the selected problem. Although we restricted our attention in this work to a two-phase composite with a high-contrast in the elastic properties of its constituent phases, the framework presented here is extensible to much more complicated class of composite (i.e., heterogeneous) microstructures and their different properties of interest (e.g., yield strength, conductivity, permeability).

5 CONCLUSION

In this work, a new CNN architecture is proposed that takes as input the 2-point spatial correlations of a voxelated eigen microstructure and predicts its effective properties of interest. Although CNNs are generally viewed as a model building technique that bypasses explicit feature engineering, it was observed that transforming the voxelated microstructure into its 2-point spatial correlations before inputting them into the CNN model dramatically improved the model accuracy and robustness. Specifically, it was shown that it is possible to build CNN models exhibiting ~0.7% test NMAE for simultaneous predictions of two different elastic stiffness components for a high-contrast (=50) 3-D composite microstructure. This unprecedented model accuracy and robustness was made possible by avoiding the use of fully connected layers, using a global average pooling in the final layer, and using 2-point spatial correlations as input to the CNN. It was also demonstrated that the CNN model produced in this work is capable of producing the most accurate elastic property closure available today for the selected high-contrast composite material system.

DATA AVAILABILITY STATEMENT

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found below: https://www.dropbox.com/sh/i9yls9d8aba2sy6/AABG-MAWABZ9gS947Phx7kgya?dl=0

AUTHOR CONTRIBUTIONS

AM and SK conceptualized the work. AM developed the code and generated figures and tables. AM and SK wrote and edited the manuscript.

FUNDING

The authors acknowledge funding from NSF 2027105. This work utilized the Hive computing cluster supported by NSF 1828187 and managed by PACE at Georgia Institute of Technology, United States.

ACKNOWLEDGMENTS

The authors thank Conlain Kelly and Andreas Robertson for their helpful comments in guiding this work.

REFERENCES

Adams, B. L., Kalidindi, S., and Fullwood, D. T. (2013). *Microstructure-sensitive Design for Performance Optimization*. Waltham, MA: Butterworth-Heinemann

Adams, B. L., Henrie, A., Henrie, B., Lyon, M., Kalidindi, S. R., and Garmestani, H. (2001). Microstructure-sensitive Design of a Compliant Beam. *J. Mech. Phys. Sol.* 49, 1639–1663. doi:10.1016/S0022-5096(01)00016-3

Barber, C. B., Dobkin, D. P., and Huhdanpaa, H. (1996). The Quickhull Algorithm for Convex Hulls. *ACM Trans. Math. Softw.* 22, 469–483. doi:10.1145/235815.235821

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning, Information Science and Statistics*. New York: Springer.

Brands, D., Balzani, D., Scheunemann, L., Schröder, J., Richter, H., and Raabe, D. (2016). Computational Modeling of Dual-phase Steels Based on Representative Three-Dimensional Microstructures Obtained from EBSD Data. *Arch. Appl. Mech.* 86, 575–598. doi:10.1007/s00419-015-1044-1

Breiman, L. (2001). Random Forests. *Mach. Learn.* 45, 5–32. doi:10.1023/A:1010933404324

Brough, D. B., Wheeler, D., and Kalidindi, S. R. (2017a). Materials Knowledge Systems in Python-A Data Science Framework for Accelerated Development of Hierarchical Materials. *Integr. Mater. Manuf Innov.* 6, 36–53. doi:10.1007/s40192-017-0089-0

Brough, D. B., Wheeler, D., Warren, J. A., and Kalidindi, S. R. (2017b). Microstructure-based Knowledge Systems for Capturing Process-Structure Evolution Linkages. *Curr. Opin. Solid State. Mater. Sci.* 21, 129–140. doi:10.1016/j.cossms.2016.05.002

Cecen, A., Dai, H., Yabansu, Y. C., Kalidindi, S. R., and Song, L. (2018). Material Structure-Property Linkages Using Three-Dimensional Convolutional Neural Networks. *Acta Materialia* 146, 76–84. doi:10.1016/j.actamat.2017.11.053

Cecen, A., Fast, T., and Kalidindi, S. R. (2016). Versatile Algorithms for the Computation of 2-point Spatial Correlations in Quantifying Material Structure. *Integr. Mater. Manuf Innov.* 5, 1–15. doi:10.1186/s40192-015-0044-x

Çeçen, A., Fast, T., Kumbur, E. C., and Kalidindi, S. R. (2014). A Data-Driven Approach to Establishing Microstructure-Property Relationships in Porous Transport Layers of Polymer Electrolyte Fuel Cells. *J. Power Sourc.* 245, 144–153. doi:10.1016/j.jpowsour.2013.06.100

Cybenko, G. (1989). Approximation by Superpositions of a Sigmoidal Function. *Math. Control. Signal. Syst.* 2, 303–314. doi:10.1007/BF02551274

Eidel, B. (2021). *Deep Convolutional Neural Networks Predict Elasticity Tensors and Their Bounds in Homogenization*. Preprint. arXiv:2109.03020.

Emmert-Streib, F., Yang, Z., Feng, H., Tripathi, S., and Dehmer, M. (2020). An Introductory Review of Deep Learning for Prediction Models with Big Data. *Front. Artif. Intell.* 3, 4. doi:10.3389/frai.2020.00004

Fast, T., Knezevic, M., and Kalidindi, S. R. (2008). Application of Microstructure Sensitive Design to Structural Components Produced from Hexagonal Polycrystalline Metals. *Comput. Mater. Sci.* 43, 374–383. doi:10.1016/j.commatsci.2007.12.002

Fullwood, D. T., Adams, B. L., and Kalidindi, S. R. (2008a). A strong Contrast Homogenization Formulation for Multi-phase Anisotropic Materials. *J. Mech. Phys. Sol.* 56, 2287–2297. doi:10.1016/j.jmps.2008.01.003

Fullwood, D. T., Adams, B. L., and Kalidindi, S. R. (2007). Generalized Pareto Front Methods Applied to Second-Order Material Property Closures. *Comput. Mater. Sci.* 38, 788–799. doi:10.1016/j.commatsci.2006.05.016

Fullwood, D. T., Niezgoda, S. R., Adams, B. L., and Kalidindi, S. R. (2010). Microstructure Sensitive Design for Performance Optimization. *Prog. Mater. Sci.* 55, 477–562. doi:10.1016/j.pmatsci.2009.08.002

Fullwood, D. T., Niezgoda, S. R., and Kalidindi, S. R. (2008b). Microstructure Reconstructions from 2-point Statistics Using Phase-Recovery Algorithms. *Acta Materialia* 56, 942–948. doi:10.1016/j.actamat.2007.10.044

Garmestani, H., Lin, S., and Adams, B. L. (1998). Statistical Continuum Theory for Inelastic Behavior of a Two-phase Medium. *Int. J. Plasticity* 14, 719–731. doi:10.1016/S0749-6419(98)00019-9

Generale, A. P., and Kalidindi, S. R. (2021). Reduced-order Models for Microstructure-Sensitive Effective thermal Conductivity of Woven Ceramic Matrix Composites with Residual Porosity. *Compos. Structures* 274, 114399. doi:10.1016/j.compstruct.2021.114399

Ghosh, S., Lee, K., and Moorthy, S. (1995). Multiple Scale Analysis of Heterogeneous Elastic Structures Using Homogenization Theory and Voronoi Cell Finite Element Method. *Int. J. Sol. Structures* 32, 27–62. doi:10.1016/0020-7683(94)00097-G

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning, Adaptive Computation and Machine Learning*. Cambridge, Massachusetts: The MIT Press.

Gupta, A., Cecen, A., Goyal, S., Singh, A. K., and Kalidindi, S. R. (2015). Structure-property Linkages Using a Data Science Approach: Application to a Non-metallic Inclusion/steel Composite System. *Acta Materialia* 91, 239–254. doi:10.1016/j.actamat.2015.02.045

Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. in" *Springer Series in Statistics*. 2nd ed. (New York, NY: Springer)

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep Residual Learning for Image Recognition. in" 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Presented at the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). USA: IEEE, Las VegasNV, 770–778. doi:10.1109/CVPR.2016.90

Jung, J., Yoon, J. I., Park, H. K., Jo, H., and Kim, H. S. (2020). Microstructure Design Using Machine Learning Generated Low Dimensional and Continuous Design Space. *Materialia* 11, 100690. doi:10.1016/j.mtla.2020.100690

Kalidindi, S. R., Binci, M., Fullwood, D., and Adams, B. L. (2006). Elastic Properties Closures Using Second-Order Homogenization Theories: Case Studies in Composites of Two Isotropic Constituents. *Acta Materialia* 54, 3117–3126. doi:10.1016/j.actamat.2006.03.005

Kalidindi, S. R. (2015). *Hierarchical Materials Informatics: Novel Analytics for Materials Data*. Amsterdam: Elsevier.

Kalidindi, S. R., Niezgoda, S. R., Landi, G., Vachhani, S., and Fast, T. (2010). A Novel Framework for Building Materials Knowledge Systems. *Comput. Mater. Contin.* 17, 103. doi:10.3970/cmc.2010.017.103

Kalidindi, S. R., and Schoenfeld, S. E. (2000). On the Prediction of Yield Surfaces by the crystal Plasticity Models for Fcc Polycrystals. *Mater. Sci. Eng. A* 293, 120–129. doi:10.1016/S0921-5093(00)01048-0

Kelly, C., and Kalidindi, S. R. (2021). Recurrent Localization Networks Applied to the Lippmann-Schwinger Equation. *Comput. Mater. Sci.* 192, 110356. doi:10.1016/j.commatsci.2021.110356

Kingma, D. P., and Ba, J. (2017). *Adam: A Method for Stochastic Optimization*. Preprint. arXiv:1412.6980 Cs.

Knezevic, M., and Kalidindi, S. R. (2007). Fast Computation of First-Order Elastic-Plastic Closures for Polycrystalline Cubic-Orthorhombic Microstructures. *Comput. Mater. Sci.* 39, 643–648. doi:10.1016/j.commatsci.2006.08.025

Knezevic, M., Kalidindi, S. R., and Mishra, R. K. (2008). Delineation of First-Order Closures for Plastic Properties Requiring Explicit Consideration of Strain Hardening and Crystallographic Texture Evolution. *Int. J. Plasticity* 24, 327–342. doi:10.1016/j.ijplas.2007.05.002

Kollmann, H. T., Abueidda, D. W., Koric, S., Guleryuz, E., and Sobh, N. A. (2020). Deep Learning for Topology Optimization of 2D Metamaterials. *Mater. Des.* 196, 109098. doi:10.1016/j.matdes.2020.109098

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* 60, 84–90. doi:10.1145/3065386

Kröner, E. (1971). *Statistical Continuum Mechanics. Statistical Continuum Mechanics, CISM International Centre for Mechanical Sciences*. Vienna: Springer Vienna. doi:10.1007/978-3-7091-2862-6

Landi, G., Niezgoda, S. R., and Kalidindi, S. R. (2010). Multi-scale Modeling of Elastic Response of Three-Dimensional Voxel-Based Microstructure Datasets Using Novel DFT-Based Knowledge Systems. *Acta Materialia* 58, 2716–2725. doi:10.1016/j.actamat.2010.01.007

Latypov, M. I., Toth, L. S., and Kalidindi, S. R. (2019). Materials Knowledge System for Nonlinear Composites. *Comp. Methods Appl. Mech. Eng.* 346, 180–196. doi:10.1016/j.cma.2018.11.034

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep Learning. *Nature* 521, 436–444. doi:10.1038/nature14539

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based Learning Applied to Document Recognition. *Proc. IEEE* 86, 2278–2324. doi:10.1109/5.726791

Loshchilov, I., and Hutter, F. (2017). *SGDR: Stochastic Gradient Descent with Warm Restarts*. Preprint. arXiv:1608.03983 Cs Math.

Marshall, A., and Kalidindi, S. R. (2021). Autonomous Development of a Machine-Learning Model for the Plastic Response of Two-phase Composites from Micromechanical Finite Element Models. *JOM* 73, 2085–2095. doi:10.1007/s11837-021-04696-w

Niezgoda, S. R., Fullwood, D. T., and Kalidindi, S. R. (2008). Delineation of the Space of 2-point Correlations in a Composite Material System. *Acta Materialia* 56, 5285–5292. doi:10.1016/j.actamat.2008.07.005

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. Preprint. arXiv: 1912.01703. Editors H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc).

Paulson, N. H., Priddy, M. W., McDowell, D. L., and Kalidindi, S. R. (2017). Reduced-order Structure-Property Linkages for Polycrystalline Microstructures Based on 2-point Statistics. *Acta Materialia* 129, 428–438. doi:10.1016/j.actamat.2017.03.009

Pinkus, A. (1999). Approximation Theory of the MLP Model in Neural Networks. *Acta Numerica* 8, 143–195. doi:10.1017/S0962492900002919

Proust, G., and Kalidindi, S. (2006). Procedures for Construction of Anisotropic Elastic-Plastic Property Closures for Face-Centered Cubic Polycrystals Using First-Order Bounding Relations. *J. Mech. Phys. Sol.* 54, 1744–1762. doi:10.1016/j.jmps.2006.01.010

Rao, C., and Liu, Y. (2020). Three-dimensional Convolutional Neural Network (3D-CNN) for Heterogeneous Material Homogenization. *Comput. Mater. Sci.* 184, 109850. doi:10.1016/j.commatsci.2020.109850

Robertson, A. E., and Kalidindi, S. R. (2021). Efficient Generation of Anisotropic N-Field Microstructures from 2-Point Statistics Using Multi-Output Gaussian Random Fields. *SSRN J* In Press. doi:10.2139/ssrn.3949516

Roters, F., Eisenlohr, P., Hantcherli, L., Tjahjanto, D. D., Bieler, T. R., and Raabe, D. (2010). Overview of Constitutive Laws, Kinematics, Homogenization and Multiscale Methods in crystal Plasticity Finite-Element Modeling: Theory, Experiments, Applications. *Acta Materialia* 58, 1152–1211. doi:10.1016/j.actamat.2009.10.058

Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. *Neural Networks* 61, 85–117. doi:10.1016/j.neunet.2014.09.003

Shaffer, J. B., Knezevic, M., and Kalidindi, S. R. (2010). Building Texture Evolution Networks for Deformation Processing of Polycrystalline Fcc Metals Using Spectral Approaches: Applications to Process Design for Targeted Performance. *Int. J. Plasticity* 26, 1183–1194. doi:10.1016/j.ijplas.2010.03.010

Tan, R. K., Zhang, N. L., and Ye, W. (2020). A Deep Learning-Based Method for the Design of Microstructural Materials. *Struct. Multidisc Optim* 61, 1417–1438. doi:10.1007/s00158-019-02424-2

Tin Kam Ho, T. K. (1995). "Random Decision Forests," in Proceedings of 3rd International Conference on Document Analysis and Recognition. Presented at the 3rd International Conference on Document Analysis and Recognition (Montreal, Que: IEEE Comput. Soc. PressCanada), 278–282. doi:10.1109/ICDAR.1995.598994

Torquato, S. (2002). *Random Heterogeneous Materials: Microstructure and Macroscopic Properties, Interdisciplinary Applied Mathematics*. New York: Springer.

Wargo, E. A., Hanna, A. C., Çeçen, A., Kalidindi, S. R., and Kumbur, E. C. (2012). Selection of Representative Volume Elements for Pore-Scale Analysis of Transport in Fuel Cell Materials. *J. Power Sourc.* 197, 168–179. doi:10.1016/j.jpowsour.2011.09.035

Wu, X., Proust, G., Knezevic, M., and Kalidindi, S. R. (2007). Elastic-plastic Property Closures for Hexagonal Close-Packed Polycrystalline Metals Using First-Order Bounding Theories. *Acta Materialia* 55, 2729–2737. doi:10.1016/j.actamat.2006.12.010

Yabansu, Y. C., Altschuh, P., Hötzer, J., Selzer, M., Nestler, B., and Kalidindi, S. R. (2020). A Digital Workflow for Learning the Reduced-Order Structure-Property Linkages for Permeability of Porous Membranes. *Acta Materialia* 195, 668–680. doi:10.1016/j.actamat.2020.06.003

Yang, Z., Yabansu, Y. C., Al-Bahrani, R., Liao, W.-k., Choudhary, A. N., Kalidindi, S. R., et al. (2018). Deep Learning Approaches for Mining Structure-Property Linkages in High Contrast Composites from Simulation Datasets. *Comput. Mater. Sci.* 151, 278–287. doi:10.1016/j.commatsci.2018.05.014

Yilin, G., Fuh Ying Hsi, J., and Wen Feng, L. (2021). Multiscale Topology Optimisation with Nonparametric Microstructures Using Three-Dimensional Convolutional Neural Network (3D-CNN) Models. *Virtual Phys. Prototyping* 16, 306–317. doi:10.1080/17452759.2021.1913783

Zhang, A., Lipton, Z. C., Li, M., and Smola, A. J. (2021). *Dive into Deep Learning*. Preprint. arXiv:2106.11342 Cs.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers in Materials

ORIGINAL RESEARCH
published: 15 March 2022
doi: 10.3389/fmats.2022.840644

Automated Virtual Design of Organic Semiconductors Based on Metal-Organic Frameworks

Mersad Mostaghimi[1], *Celso R. C. Rêgo*[1], *Ritesh Haldar*[2], *Christof Wöll*[3], *Wolfgang Wenzel*[1] *and Mariana Kozlowska*[1]*

[1]Institute of Nanotechnology, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, [2]Tata Institute of Fundamental Research Hyderabad, Hyderabad, India, [3]Institute of Functional Interfaces, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

The arrangement of organic semiconductor molecules in a material can be modulated using different supramolecular approaches, including the metal–organic framework (MOF) approach. These arrangements result in different frameworks topologies and structures. Fabrication of materials comprising optimized assemblies and functional molecules enables efficient tailoring of material properties, including electronic responses. Since semiconducting properties are sensitive to subtle changes in the nanostructure of the material, the exploitation of MOFs has promising potential in the development of new materials with designed structure and function. Based on decade-long method development, virtual design strategies have become ever more important, and such design methods profit from the availability of automated tools. Such tools enable screening of huge libraries of organic molecules in *in silico* models of the structure of three-dimensional nanoscale assemblies as the prerequisite to predict their functionality. In this report, we present and demonstrate the application of an automated workflow tool developed for MOFs of the primitive cubic (PCU) topology. We use pentacene-based ditopic linkers of a varied chemical composition and pillar linkers of different molecular sizes to automatically generate PCU MOFs, sample their structural dynamics at finite temperature, and predict electronic coupling matrix elements in vibrationally averaged assemblies. We demonstrate the change of the intermolecular ordering in the resulting MOFs and its impact on the semiconducting properties. This development lays the basis of an extendable framework to automatically model a wide variety of MOFs and characterize their function with respect to properties, such as conduction properties, absorption, and interaction with light. The developed workflow protocol and tools are available at https://github.com/KIT-Workflows/PCU-MOF.

Keywords: organic semiconductors, pentacene, metal–organic framework, virtual design, workflow, structure builder, molecular dynamics, stacking

OPEN ACCESS

Edited by:
Simone Taioli, European Centre for Theoretical Studies in Nuclear Physics and Related Areas (ECT), Italy*

Reviewed by:
Giovanni Garberoglio, Bruno Kessler Foundation (FBK), Italy
Tim Kowalczyk, Western Washington University, United States

***Correspondence:**
Mariana Kozlowska
mariana.kozlowska@kit.edu

Specialty section:
This article was submitted to Computational Materials Science, a section of the journal Frontiers in Materials

Received: *21 December 2021*
Accepted: *31 January 2022*
Published: *15 March 2022*

Citation:
Mostaghimi M, Rêgo CRC, Haldar R, Wöll C, Wenzel W and Kozlowska M (2022) Automated Virtual Design of Organic Semiconductors Based on Metal-Organic Frameworks. Front. Mater. 9:840644. doi: 10.3389/fmats.2022.840644

1 INTRODUCTION

Electronic properties of organic semiconductors (OSCs) and their optoelectronic response highly depend on the structural arrangement of the molecules and their vibrational freedom in the material (Pohl and Pohl, 2020; D'Avino et al., 2016). They determine the electronic coupling between the molecules, which impacts the mechanism and the properties of the electronic conduction (Kera et al., 2009; Mailman et al., 2017; Haldar et al., 2021). The packing geometry of OSC is driven by the subtle intermolecular interactions, which are difficult to control; therefore, crystal engineering remains a significant challenge. The arrangement of molecules depends on the crystallization conditions, which, in most cases, cannot be easily controlled. This limits the fabrication of new OSC with desired orientation of the molecular units and also the optimization of the semiconducting properties, which result from the molecular packing in a material. A precise control over the architecture of the material, which incorporates the OSC components at well-defined positions and orientations, can be realized by a reticular chemistry approach (Freund et al., 2021), applied for the synthesis of metal–organic frameworks (MOFs). This approach has been successfully utilized to obtain materials with a controlled three-dimensional assembly of organic molecules (Alvaro et al., 2007; Liu et al., 2011; Xu et al., 2016; Liu and Wöll, 2017), opening new perspectives in the fabrication of OSC materials with programmable functionality (Garg et al., 2019; Haldar et al., 2019; Xie et al., 2020).

Metal–organic frameworks are materials that consist of organic molecules, i.e., linkers, and inorganic polynuclear clusters (nodes), known as secondary building units (SBUs) (Rowsell and Yaghi, 2004) (**Figure 1**). Several bipartite nets, known in reticular chemistry (O'Keeffe et al., 2008), represent binary frameworks used for the construction of MOF structures of various topologies (Kalmutzki et al., 2018). By combining different metal ions and organic linkers in diverse topologies, a considerable variety of MOFs can be realized, and around 100,000 different MOF structures have been synthesized so far (Chung et al., 2014; Lyu et al., 2020). They represent a unique variety of structures, and their properties can be tailored through rational design (Freund et al., 2021) and by choosing the appropriate combinations. As one may guess, an unprecedented number of possible combinations occurs, which led also to the creation of huge databases of hypothetical MOF structures and screening of new MOF candidates (Moosavi et al., 2020). MOFs are widely used for gas sorption (Lin et al., 2020; Qiao et al., 2020) and storage (Li et al., 2018; Connolly et al., 2020), sensors (Kreno et al., 2012), or solar cells (Goswami et al., 2016). They are good candidates for establishing new OSC with tailored electronic properties by ordering organic linkers in a network manner with the specific framework topology (Öhrström, 2015; Neumann et al., 2016). However, owing to their porous structure and relatively large distances between neighboring OSC linker molecules, MOFs are typical wide-bandgap semiconductors, but their electronic conduction can be increased by realizing structures that enable efficient charge transfer through-bonds (Narayan et al., 2012) or space (Neumann et al., 2016), as well as by π-stacking (Liu and Wöll, 2017; Haldar et al., 2021; Zojer and Winkler, 2021). The through-bond pathway depends on continuous SBUs comprising metals and ligand moieties; their high charge mobility and small band gap are a result of the proper orbital overlap and well-matched energy levels. Through-space electronic conduction can be enhanced by incorporating guest molecules like TCNQ or fullerenes in MOF pores (Neumann et al., 2016; Liu et al., 2019). It can be also photoswitched by light (Heinke et al., 2014; Kanj et al., 2018; Garg et al., 2019; Haldar et al., 2020). Upscaling of MOF electronic conduction *via* the incorporation of highly efficient OSC molecules with π-stacking within MOF scaffold, e.g., pentacene, has been also realized (Haldar et al., 2021). Hence, electronic conduction in MOFs is based on several different mechanisms (Xie et al., 2020); therefore, it follows different design strategies. For example, in the last case, due to the localized frustrated rotations of the pentacene cores in the assembled structure, charge carrier mobility is significantly decreased. More rigid linkers (with limited vibrational flexibility) or other types of assemblies with the designed orientation of OSC units in the MOF scaffold are sought after (Haldar et al., 2019).

Considering the enormous chemical space of MOF components, a multitude of new candidates with interesting properties can be realized. Unfortunately, attempting to access them all experimentally *via* MOF fabrication is expensive, time-consuming, and based on trial and error (Stock and Biswas, 2012). Predictive methods for MOF construction, design, and selection of the most prominent candidates with a desired functionality will significantly accelerate the discovery of new formulations and reduce costs (Mancuso et al., 2020). There are several free and commercial tools or molecular editors for the initial construction of a representative MOF model, e.g., Avogadro (Nefedov et al., 2021) and Materials Studio (Dassault Systèmes, 2020). Often, they require time-consuming manual work and prior knowledge of the topological details and metal–organic interactions in MOF. Advanced algorithms using the reverse topological or top-down approach are implemented, using graph theory, in the tool of Boyd and Woo (2016), and using known nets (topologies) of MOFs that follow a recursive, geometry-based assembly of SBUs and linkers (O'Keeffe et al., 2008), in codes like *stk* (Turcani et al., 2021), *Zeo++* (Martin and Haranczyk, 2014), *ToBaCCo* (Colón et al., 2017), and *AuToGraFS* (Addicoat et al., 2014a). All of these tools can be used to prepare a basic MOF model for visualization, whereas subsequent structure optimization and the calculation of electronic and/or optoelectronic properties should be done separately. Some of the tools require programming knowledge, and they have limited automation functionality towards virtual design of new MOFs comprising various components, triggering new interesting phenomena.

In this report, we present an automated, transferable, and user-friendly workflow for the generation, optimization, and structure–property prediction of new MOF materials. We use the SimStack workflow environment (https://www.simstack.de/) to combine several modules, called *WaNos* (Workflow active

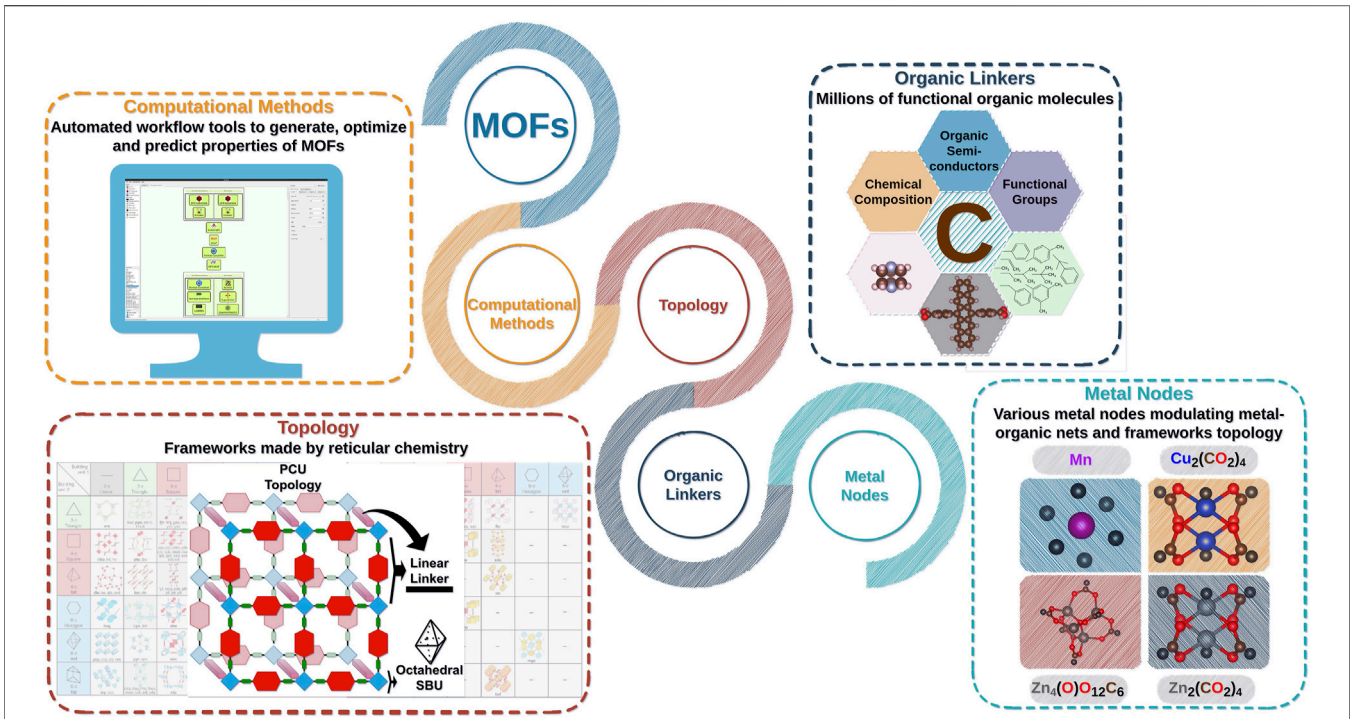


FIGURE 1 | Reticular chemistry applied for the fabrication of MOFs with diverse metal–organic networks, components (organic molecules and metal nodes), and properties. Virtual design of new MOF materials by automated workflows investigated in this report.

Nodes), responsible for the specific tasks or calculation(s) involved in the MOF building protocol and up to the analysis of specific MOF properties, e.g., in the case presented here, electronic coupling elements towards semiconducting MOFs. Each WaNo uses as an input the output from the previous WaNo, enabling an easy transfer of data between compute units and construction of different workflows, as well as a combination of different WaNos aiming at desired purposes. The workflow, introduced here for a MOF design and calculation, is as follows:

- Adaptable: User-friendly graphical user interface (GUI) and automated dataflow between multiple module functions allow easy adaptation of the workflow to specific user requirements;
- Reproducible: Once set up, it will generate the same results independently of the user environment. Automatic flow of data and processes allows an error-free data generation and storage of the work recipes for other investigations;
- Efficient: It significantly reduces the time to prepare and carry out similar calculations that have the same methodological flow, but different starting input data;
- Expandable: By replacing method modules (WaNos), the workflow can be extended to related applications and to diverse MOF topologies. Moreover, it can be used as a part of other workflow(s);
- Transferable: It can be used by users outside the group which has developed it without facing implementation issues.

Considering the variety of MOF materials and possible attained properties, multiple options for such workflow developments are possible. In the present study, we focus on the primitive cubic (PCU) topology of MOFs (O'Keeffe and Yaghi, 2012) (**Figure 1**) that is a widely used topology found in fabricated MOF materials (Li et al., 2007; Luo et al., 2017; Wei et al., 2019; Xing et al., 2019; Yao et al., 2019). The PCU topology features the network that describes the MOF-2 family, which has paddle-wheel nodes, such as $Zn_2(CO_2)_4$ and $Cu_2(CO_2)_4$, connected by linkers (Li et al., 1998; Xing et al., 2019; Goldman et al., 2020; Qiao et al., 2020; Yazdanparast et al., 2020) and the MOF-5 family with octahedral nodes, such as $Zn_4(O)O_{12}C_6$ (Li et al., 1999; Schoedel et al., 2016). It can be formed also from SBUs based on Mg, Cd, Pb, or rare earth metals (Bhattacharya et al., 2014; Luo et al., 2017; Lin et al., 2018; Yao et al., 2019). In addition, MOF films grown using a layer-by-layer technique on the pre-functionalized surfaces, e.g., surface-anchored MOFs (SURMOFs) (Liu et al., 2012; Zhuang et al., 2016), using $Cu_2(CO_2)_4$ and $Zn_2(CO_2)_4$ SBUs are often assembled in PCU-type networks. Owing to the fact that SURMOFs allow us to control the assembly type and the linker order in a material (Heinke et al., 2014; Heinke and Wöll, 2019), crystal structure engineering of such MOFs comprising OSC and chromophore molecules will allow us to investigate new systems systematically and accelerate the selection of the most promising MOF candidates. Virtual screening of various organic linker molecules and their assembly in the MOF allows computation of structure–property relationships that can be accessed by the

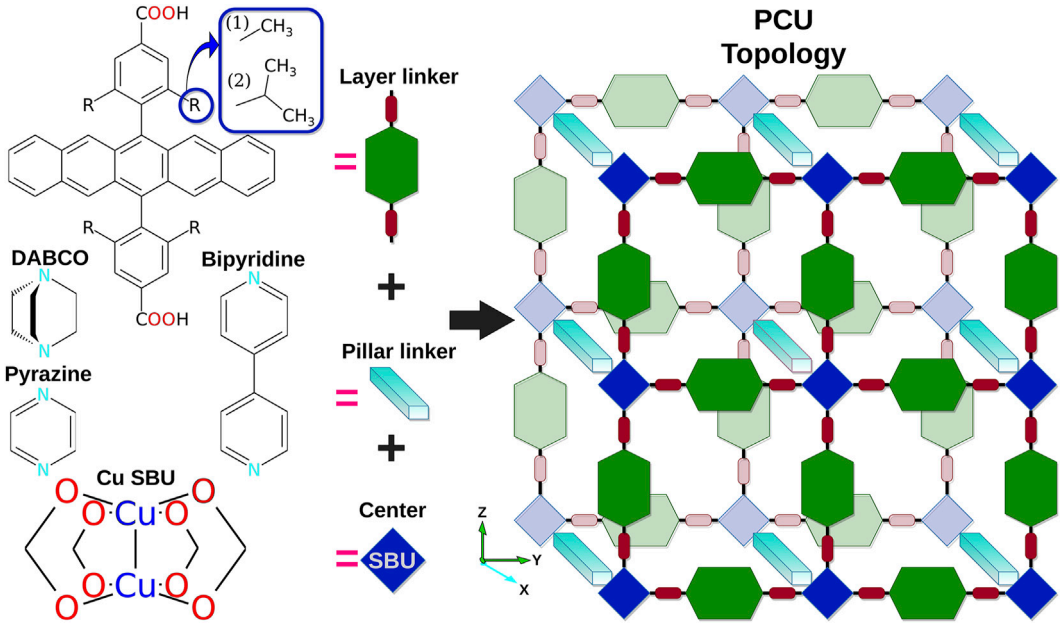


FIGURE 2 | Representation of MOF materials of PCU topology investigated in the present report. Layer and pillar linkers are marked in green and cyan, respectively. Copper secondary building units (SBUs) are depicted in blue.

automated workflows reported here, including the dynamical behavior of the structure at finite temperature.

We demonstrate the architecture of such a workflow and its exemplary usage for the prediction of electronic coupling elements between pentacene cores in a PCU SURMOF structures built from ditopic pentacene layer linkers and DABCO (DABCO = 1,4-diazabicyclo(2.2.2)octane), pyrazine (pyz = 1,4-diazine), and bipyridine (bipy = 4,4′-bipyridine) pillar linkers (**Figure 2**). Pentacene (Pn) is an organic molecule, which is characterized by high charge mobility in its different crystal forms (Jurchescu et al., 2007; Virkar et al., 2009). It was previously integrated in the SURMOF-2 type of MOF (Haldar et al., 2021), where the stacking of pentacene cores of the linkers, localized at a distance of approximately 5.8 Å, was suggested to be not optimal for efficient charge carrier transport caused by ineffective overlap between the neighboring pentacene cores (Zojer and Winkler, 2021) and weak π–π interactions, which are dynamically changing over time, as observed in MD simulations at 300 K (Haldar et al., 2021). These simulations explained increased vibrational flexibility of molecules in the material and the significant decrease of the effective electronic coupling.

With PCU SURMOF type of assembly, investigated here, we aim to use automated workflows to build and calculate properties of new MOFs and illustrate the change of their semiconducting properties as a function of the controlled separation between the pentacene cores (by different pillar linkers) that change with the stacking type and the dynamics of OSC cores in a material. With this work, we initiate a new approach for further virtual design of versatile MOF assemblies that will enable the upscaling of the functionality of single molecules, such as pentacene, to yield functional solid materials or thin films of macroscopic dimensions in future investigations.

2 METHODS AND COMPUTATIONAL DETAILS

MOFs are modular periodic systems, consisting of metal (oxo)-nodes (SBUs) and organic linker molecules. Depending on the nature of both components, different three-dimensional nets are formed. Virtual design of new MOF constructs requires the combination of several algorithms (methodologies and codes) that are linked together in one workflow, enabling automated data transfer and on-the-fly property prediction. Here, we create an automated methodology to build, optimize, and analyze MOFs of the PCU topology and implement it within SimStack. MOF systems and components used for the protocol demonstration are depicted in **Figure 2**. Three ditopic linkers, differing by the R-substitution in the linking phenyl group and based on pentacene cores, were used as layer linkers, i.e., connected to the Cu-node by Cu···O coordination bonds (on *y*- and *z*-axes), while DABCO, pyrazine, and bipyridine were used as pillar linkers with Cu···N coordination (*x*-axis).

In SimStack, each module or WaNo implements a graphical user interface for the backend codes, which performs required calculations. Each WaNo is maximally generalized, therefore can be used to calculate various molecules and types of materials, and can be used for tasks in conceptually different workflows. The PCU-MOF workflow consists of several WaNos for software codes, such as TURBOMOLE (Balasubramani et al., 2020),

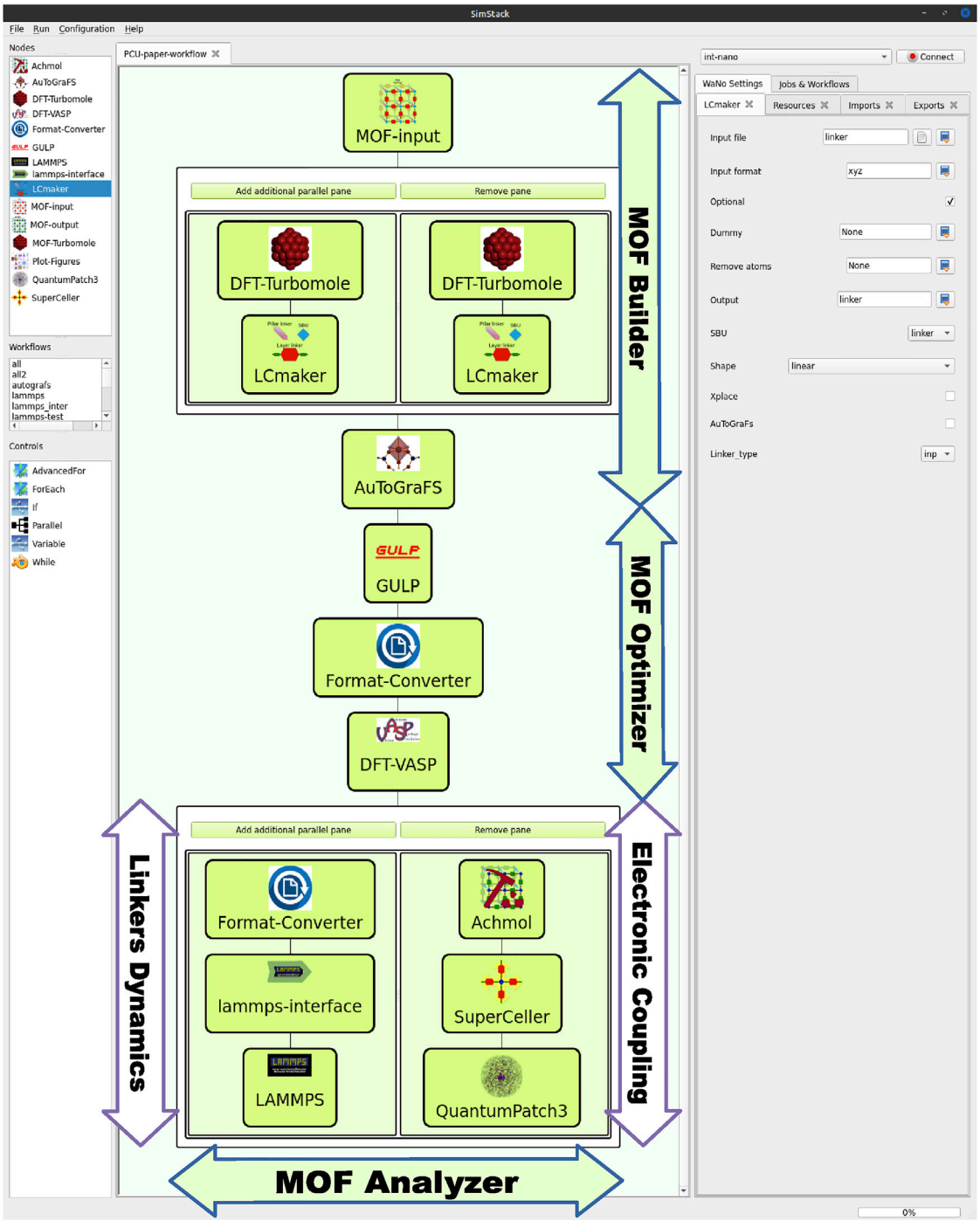


FIGURE 3 | PCU-MOF workflow within the SimStack client to build, optimize, and analyze MOF materials for automated structure–property predictions and design.

AuToGraFS (Automatic Topological Generator for Framework Structures) (Addicoat et al., 2014a), GULP (General Utility Lattice Program) (Gale and Rohl, 2003), VASP (Vienna Ab initio Simulation Package) (Kresse and Hafner, 1993), LAMMPS (Large-scale Atomic Molecular Massively Parallel Simulator) (Boyd et al., 2017; Plimpton et al., 2020), and interfaces between codes. We also developed several new codes and algorithms, i.e., LCmaker, Achmol (Assembler of Chemical Molecules), and SuperCeller.

To create an automated workflow for MOF design, we have used several available WaNos, *DFT-Turbomole* (Schlöder and Rêgo, 2022), *DFT-VASP* (Ricardo, 2021), *QuantumPatch3* (Friederich et al., 2014), and *Format-Converter*, while all other WaNos (*MOF-input*, *LCmaker*, *AuToGraFS*, *GULP*,

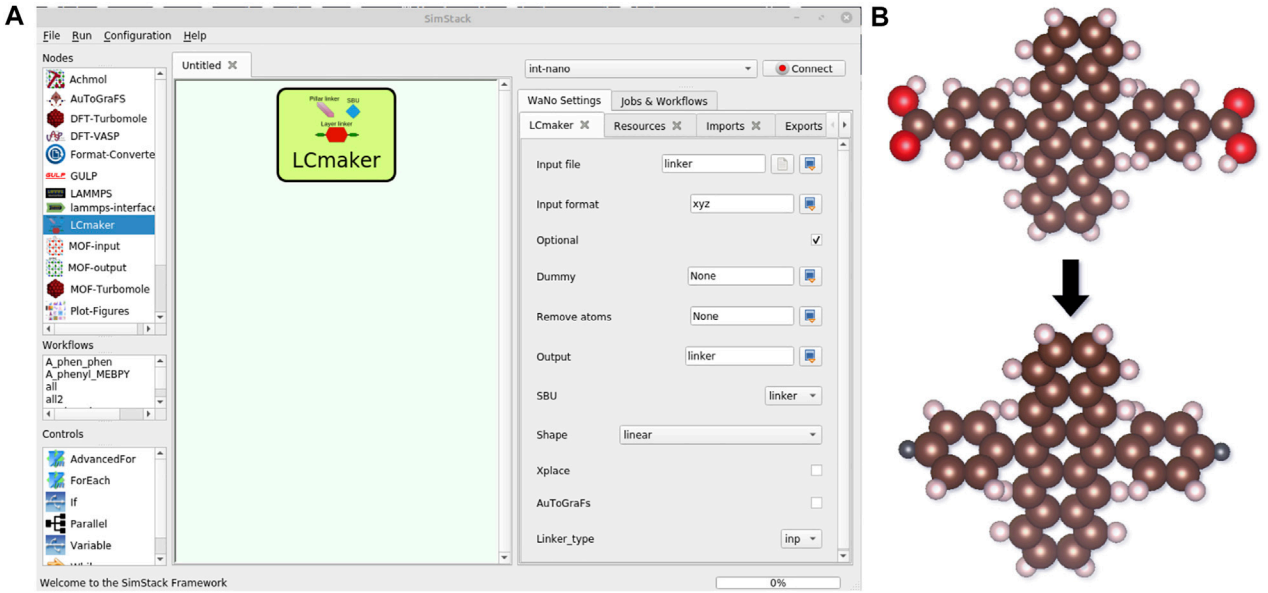


FIGURE 4 | *LCmaker* WaNo window in SimStack **(A)**. Options in the right window are set for the PCU topology. Description of all options available is given in **Supplementary Table S1**. Main functionality of *LCmaker* **(B)**.

lammps-interface, *LAMMPS*, *Achmol*, and *SuperCeller*) were programmed for the purpose of MOF design and are reported here for the first time. The workflow consists of three central parts, i.e., 1) **MOF Builder**, 2) **MOF Optimizer**, and 3) **MOF Analyzer** (**Figure 3**). The **MOF Builder** consists of the general WaNo *MOF-input*, where the user specifies linker molecules, metal node type, and desired topology; *DFT-Turbomole*, where optimization of input linkers is done using the TURBOMOLE software; and *LCmaker* and *AuToGraFS* used to build the periodic MOF model. To generalize the standard of atomic units in the entire workflow, both in converting formats and in developing the back-end software in different parts of the workflow, we employed the Atomic Simulation Environment (ASE) (Larsen et al., 2017). The **MOF Optimizer** includes WaNo *GULP* for the pre-optimization of MOFs with force-field potentials, *DFT-VASP* for MOF optimization using the quantum mechanical approach (density functional theory, DFT), and *Format-Converter* designed to convert different file formats between software. The **MOF Analyzer** is a multifunctional part of a workflow, which is assigned to calculate different properties of MOF based on both 1) optimized structures at 0 K (right branch, including *Achmol* and *SuperCeller* WaNos) and 2) sampled MOF structures during molecular dynamics (MD) simulations at finite temperature using *LAMMPS*. In the present study, we focus on the calculation of the electronic coupling matrix elements, J_{if}, between the initial state of a donor and the final state of the acceptor, used in the Marcus theory of charge transfer (CT), i.e., in a hopping mechanism (Marcus, 1993) to estimate the semiconducting properties of PCU MOFs. According to this theory (**Eq. 1**), the charge transfer rate, k_{CT}, is defined as

$$k_{CT} = \frac{2\pi}{\hbar}|J_{if}|^2\sqrt{\frac{1}{4\pi k_B T\lambda}}\exp\left(\frac{-(\Delta G^0+\lambda)^2}{4\lambda k_B T}\right). \quad (1)$$

Here, λ is the reorganization energy, connected to the change in the equilibrium geometry of both donor and acceptor upon change of the charged state, k_B is the Boltzmann constant, T is the temperature, and ΔG^0 is the total Gibbs free energy change (i.e., energy difference between frontier molecular orbitals involved in CT). Electronic coupling, J_{ij}, between adjacent OSC cores in a MOF was calculated using the overlap between molecular orbitals obtained from the DFT, as implemented in the WaNo *QuantumPatch3*. A detailed description of algorithms and computational details is given below.

2.1 Metal-Organic Framework Builder

In this part, the initial model of a MOF is constructed from the user-defined linker molecules and implemented SBU nodes. Here, we describe the PCU topology of MOFs, and we selected as an example Cu metal nodes in the paddlewheel shape (Cu_pw6.inp). The workflow uses the AuToGraFS software; therefore, all SBUs and topologies available in AuToGraFS can be accessed by our tool too, enabling many extensions. Input data, i.e., linker molecules and SBUs, are specified in *MOF-input* WaNo. Since we have used two linker types, layer and pillar, that have different chemical compositions, two parallel optimization procedures of linker molecules have to be performed in *DFT-Turbomole*. The choice of different functionals, basis sets, and other DFT parameters is implemented in this WaNo. All linkers studied here were optimized utilizing the B3LYP functional (Lee et al., 1988; Becke, 1993) with the def2-SV(P) basis set (Weigend and Ahlrichs, 2005) and Grimme D3 dispersion correction

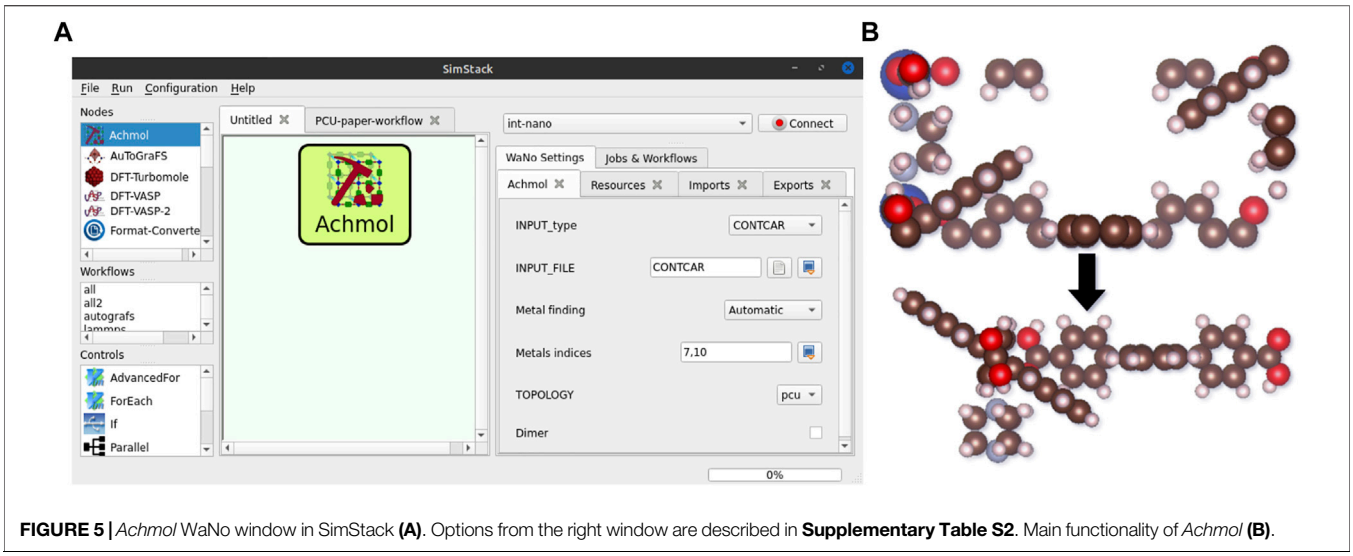


FIGURE 5 | *Achmol* WaNo window in SimStack **(A)**. Options from the right window are described in **Supplementary Table S2**. Main functionality of *Achmol* **(B)**.

(Grimme et al., 2010), using default optimization criteria. Optimized structures are then automatically passed to *LCmaker*.

LCmaker makes a customized "inp" or "xyz" format files for different versions of AuToGraFS and automatically generates the joint points, called dummies, in the places where the organic linker should be linked to the metal node. *LCmaker* supports two algorithms for dummy selection: (1) for carboxyl group connections, i.e., by removing COOH groups from layer linkers and exchanging them by the dummy points (**Figure 4B**), and (2) for the nitrogen-terminated pillar linkers by the dummy on the N atom. Moreover, there is also a possibility to define dummies *via* the user-defined atom indexes. Options available in the *LCmaker* are depicted in **Figure 4** and listed in **Supplementary Table S1**.

As mentioned before, there are several algorithms to apply metal center–organic linker connections for the specific MOF topology. We built MOF structures using AuToGraFS (Addicoat et al., 2014a) for three main reasons: 1) the simplicity of MOF construction based on manageable input and output interfaces, 2) its ability to support almost all MOF topologies, 3) its ability to be linked to a fully featured force field UFF4MOF (Universal Force Field for MOF) necessary for further optimization of structures of arbitrary frameworks using, e.g., GULP software (Addicoat et al., 2014b; Coupry et al., 2016). Moreover, UFF4MOF is implemented in LAMMPS package, enabling lots of further investigations of specific MOF properties, including MD simulations. AuToGraFS has also implemented additional atom types found in the Computation-Ready Experimental (CoRE) database (Chung et al., 2014); it shows comparability with experimental results and enables our workflows to be extended to various MOF systems. AuToGraFS was implemented in our workflow in the respective WaNo node called *AuToGraFS* (**Figure 3**). This WaNo enables use of the code without any prior programming knowledge and manual preparation of necessary files. Together with *LCmaker*, *AuToGraFS* is automatically generating MOF periodic representation that can be processed in the further steps of the workflow.

TABLE 1 | Unit cell parameters of PCU-type MOF structures studied.

Structure	Cell parameters (X × Y × Z in Å)
$Cu_2(Pn)_2(DABCO)$	9.24 × 19.50 × 19.50
$Cu_2(Pn)_2(pyz)$	9.23 × 19.50 × 19.50
$Cu_2(Pn)_2(bipy)$	13.64 × 19.42 × 19.42
$Cu_2(Me\text{-}Pn)_2(DABCO)$	9.47 × 19.59 × 19.58
$Cu_2(iPr\text{-}Pn)_2(DABCO)$	9.73 × 19.54 × 19.60

2.2 Metal–Organic Framework Optimizer

Two consecutive optimization schemes are used to optimize the initially generated MOF model: pre-optimization with the UFF4MOF force field using GULP (Gale and Rohl, 2003; Addicoat et al., 2014b) and later optimization based on DFT using VASP. Such a DFT-optimized MOF structure represents one of the local minima structures necessary for setting up the next steps of the workflow. For the case study presented here, DFT calculations in the *DFT-VASP* WaNo (Ricardo, 2021) were performed using the PBE functional (Perdew et al., 19961997) with the Tkatchenko-Scheffler method with iterative Hirshfeld partitioning (Bučko et al., 2013) using VASP version 5.4.4. The plane wave energy cutoff was set to 500 eV, and the *k*-point grid was 3 × 2 × 2. The input–output file formats between codes were handled *via Format-Converter* WaNo (**Figure 3**).

2.3 Metal-Organic Framework Analyzer

From this step, functionality of the workflow can be user specific, and different WaNos can be added to the workflow to calculate desired properties of MOF materials. For the semiconducting properties, which can be triggered *via* different electronic conduction mechanisms, DFT-optimized MOF structures from *DFT-VASP* may be used:

- To calculate band structure, from which couplings and effective masses of electrons and holes can be derived,

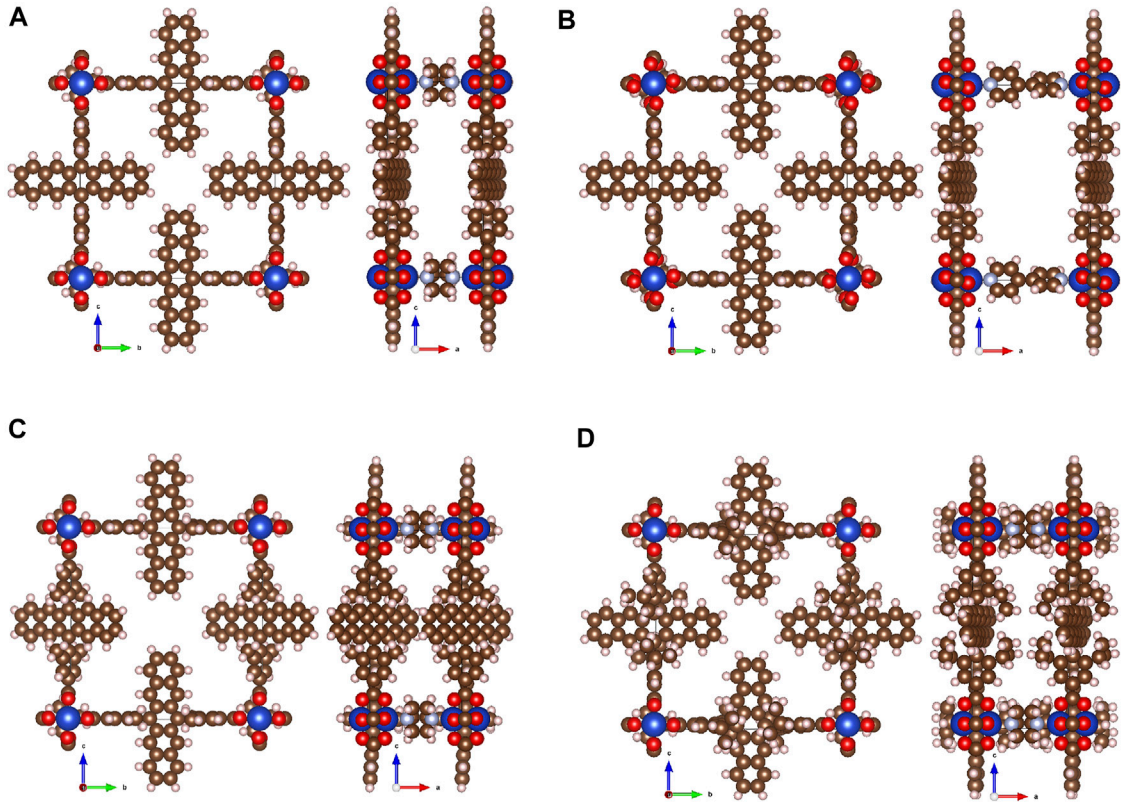


FIGURE 6 | Representative structures of MOF of PCU topology obtained automatically after MOF Builder and MOF Optimizer parts of the workflow starting from coordinate files of linkers and specification of SBU type: $Cu_2(Pn)_2(DABCO)$ **(A)**, $Cu_2(Pn)_2(bipy)$ **(B)**, $Cu_2(Me\text{-}Pn)_2(DABCO)$ **(C)**, $Cu_2(iPr\text{-}Pn)_2(DABCO)$ **(D)**.

- To calculate electronic couplings between linkers based on molecular orbitals self-consistently converged at the explicit MOF environment, which are used in hopping equations, e.g., **Eq. 1**,
- To sample temperature-dependent MOF structures with further calculation of couplings from snapshots in MD.

Since our previous studies have proven that the vibrational flexibility of linkers in a MOF impacts charge carrier mobilities measured experimentally (Haldar et al., 2021), in the present study, we focus on the electronic couplings between OSC cores in PCU-type MOF assemblies based on the MD sampled structures at 298 K (left branch in **MOF Analyzer** in **Figure 3**). However, in the PCU-MOF workflow, we demonstrate two possible pathways that may be used to estimate electronic couplings in the hopping scenario.

2.3.1 Electronic Coupling in DFT-Optimized Metal-Organic Frameworks

If the electronic conduction of a MOF is based on the charge carrier hopping between OSC linkers in a MOF scaffold with no or negligible impact of the metal atoms linking them (Liu et al., 2019; Haldar et al., 2021), metal atoms can be omitted. To extract optimized MOF in XYZ format from the periodic file formats and provide essential files for the *QuantumPatch3* WaNo (cml files with linker molecules in the MOF supercell fragment), where couplings are calculated for each linker separately (Friederich et al., 2014), we developed a python-based code that uses the Open Babel Toolbox within ASE (O'Boyle et al., 2011). It is known that atom positions in periodic calculations are mapped to the unit cell (**Figure 5**); therefore, direct conversion of files after VASP to Cartesian format, keeping atoms grouped to each linker, is not possible. This is done in our workflow by the Achmol (Assembler of Chemical Molecules) code implemented in *Achmol* WaNo. It takes the unit cell of the optimized MOF structure and creates a supercell out of it, e.g., $2 \times 2 \times 2$. Then, it removes metal atoms and searches for groups (clusters) of complete linkers (the correct amount of atoms that belong to a specific molecule) based on a condition of volume minimization of the center of mass of the linkers. It finds complete linkers based on the number of atoms of separated linkers after removing metals using a connectivity matrix. The graph of molecules is the main pattern to find complete linkers after optimization (**Figure 5B**). The main output of *Achmol* is the Cartesian file with three linkers (on *x*-, *y*-, and *z*-axes, periodically repeated to represent the MOF) with atoms grouped to a specific linker molecule, e.g., atoms 1–49 belong to linker 1, while atoms 50 to 100 to linker 2. All extracted linkers are then automatically hydrogenated to keep neutral charge consistence and can be used to create user-defined clusters, specified in WaNo *SuperCeller*, necessary for further calculations in *QuantumPatch3*. *Achmol* can post-process output files of periodic calculations in CONTCAR, xml and cif formats, e.g., vasprun.xml or sample.cif. Options available in

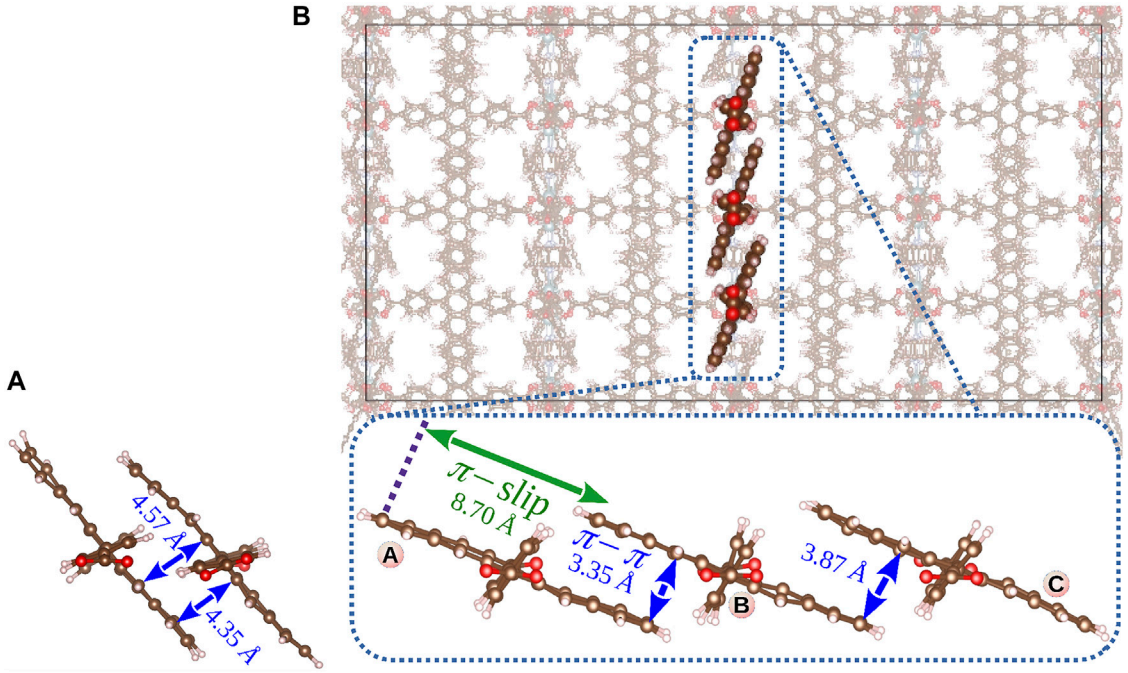


FIGURE 7 | Molecular packing of the pentacene cores in a MOF as a function of MOF topology and the COM distance between ditopic linkers: dimer structure from QM/MM MD simulation of Zn paddlewheel SURMOF-2 reported by Haldar et al. (2021) (COM of 5.8 Å) **(A)**. $Cu_2(Pn)_2(DABCO)$ PCU-type MOF simulated in the present study and the trimer of layer linkers formed during MD simulation at 298 K (COM of 9.2 Å) **(B)**.

the *Achmol* WaNo are depicted in **Figure 5** and listed in **Supplementary Table S2**. Functionality of *SuperCeller* is listed in the **Supplementary Material** (**Supplementary Figure S2**; **Supplementary Table S3**).

2.3.2 Electronic Coupling From Molecular Dynamics Sampling

MOF structures obtained after **MOF Builder** and **MOF Optimizer** can be used as starting structures in MD simulations at finite temperature in either VASP (*ab initio* MD) or LAMMPS (force-field-based MD). Considering the high computational cost and limited simulation times of *ab initio* MD, we performed MOF structure sampling using UFF4MOF as implemented in LAMMPS. For that, two WaNos were created: *lammps-interface* and *LAMMPS* (**Figure 3**). *lammps-interface* contains a list of options that should be specified by a user to set up automatically input files to run MD. It passes all files to *LAMMPS*, which performs the actual MD run, including equilibration and production. For the MOF systems studied here, we performed MD simulations of 4 × 4 × 4 supercells for 10 ns (with a timestep of 1 fs) at 298 K. Production runs were preceded with 30 ps equilibration. All simulations were done in a canonical ensemble (NVT).

To calculate electronic couplings between neighboring ditopic linkers structured in a MOF during 10 ns MD, we extracted dimers from 1,000 snapshots (every 10 ps) and performed separate calculations using the Quantum Patch approach (Friederich et al., 2014). Couplings were calculated as an overlap between the highest occupied molecular orbitals, HOMO (hole transport), and the lowest unoccupied molecular orbitals, LUMO (electron transport), in dimers using the B3LYP functional with the def2-SVP basis set.

3 RESULTS AND DISCUSSION

The workflow protocol described in **Section 2** was applied to automatically generate and optimize five PCU MOF systems with the subsequent structure sampling by MD simulations to collect vibrationally averaged assemblies of pristine and modified (methyl and isopropyl) pentacene ditopic linkers. The cell parameters of optimized structures are listed in **Table 1**.

Representative structures are depicted in **Figure 6** and **Supplementary Figure S3**. Due to the similar size of DABCO and pyrazine pillar linkers, the geometrical parameters of both systems are also similar, resulting in center-of-mass (COM) distances between neighboring linkers of approximately 9.2 Å, which is much larger than the distance observed for the previously reported SURMOF-2 structures (Haldar et al., 2021) (around 5.8 Å, **Figure 7A**). PCU MOF with a bipyrazine pillar has an even larger COM (around 13.6 Å, **Figure 6B**).

MD calculations of MOFs with pristine pentacene cores and a DABCO (or pyrazine) pillar revealed system stabilization in a nanosecond scale with the formation of self-assemblies depicted in **Figure 7B**. Dynamical movements of linkers in these MOFs are decoupled, and after starting an MD production run, some dimers are stabilized faster (within 2 ns) and others slower (within 7 ns), e.g., dimers "A-B" and "B-C" in **Figures 8B,C**.

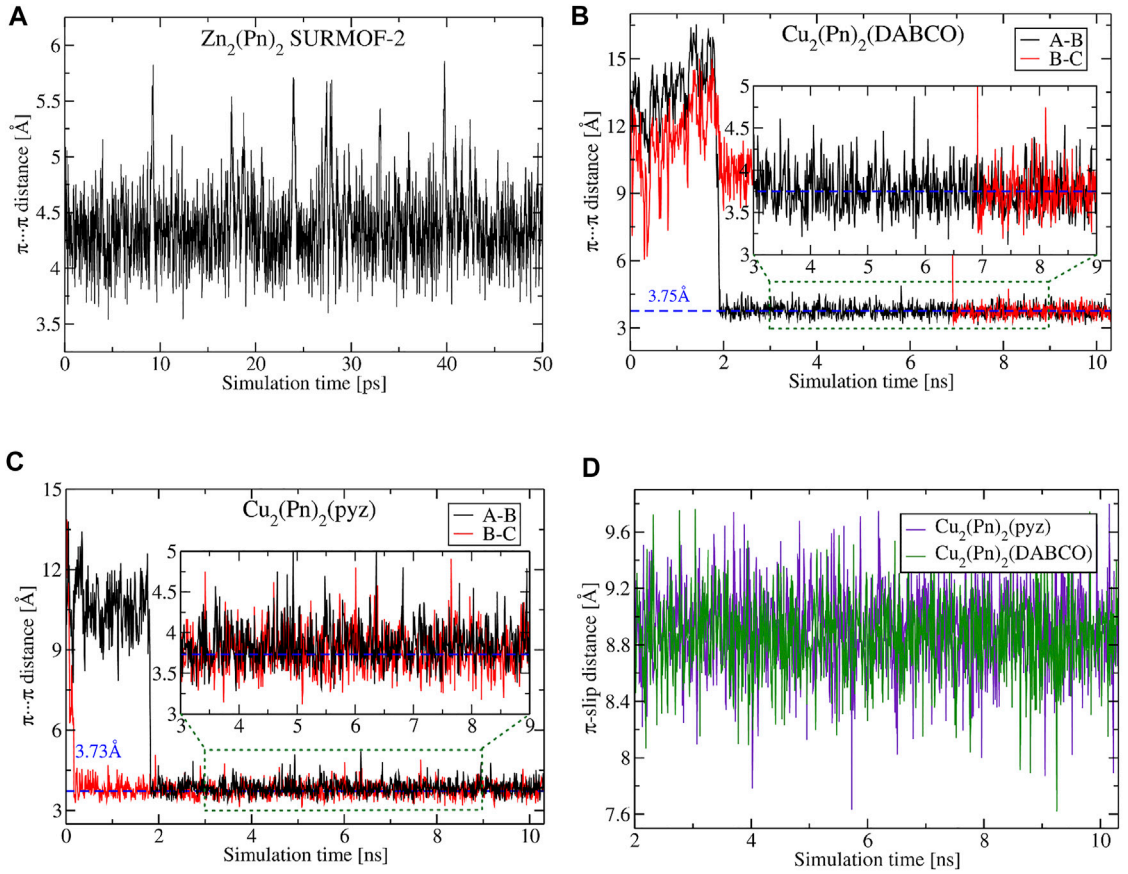


FIGURE 8 | Dynamical change of π–π interaction and π-slip distances between pentacene cores in MOF materials of different topologies and COM distances obtained from MD simulations at room temperature. The π–π distance in a dimer from SURMOF-2 reported by Haldar et al. (2021) **(A)**, in a dimer from $Cu_2(Pn)_2(DABCO)$ **(B)** and $Cu_2(Pn)_2(pyz)$ **(C)**. The comparison of π-slip distances between layer linkers in $Cu_2(Pn)_2(DABCO)$ and $Cu_2(Pn)_2(pyz)$ **(D)**. Plotted structural parameters and "A"-, "B"-, "C"-marked linkers are depicted in **Figure 7**.

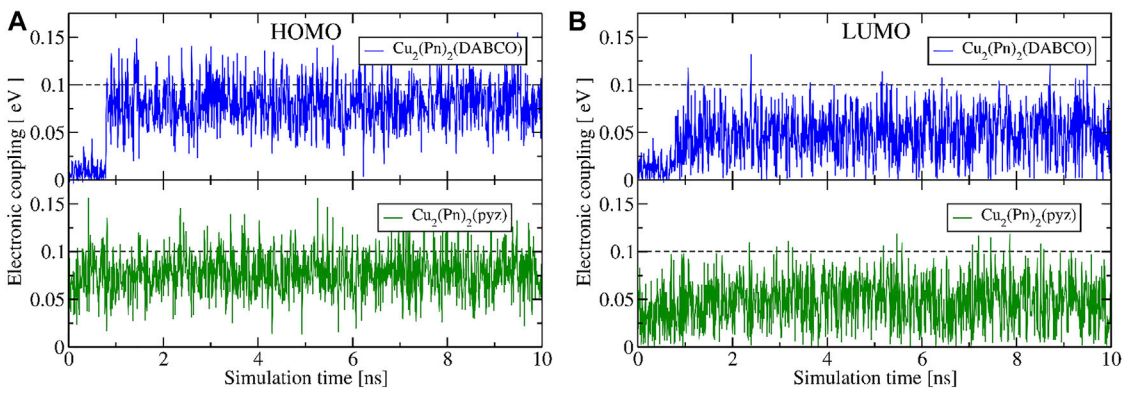


FIGURE 9 | Absolute electronic coupling elements (B3LYP/def2-SVP) between HOMO **(A)** and LUMO **(B)** orbitals of ditopic pentacene linkers extracted from the force-field MD simulations of $Cu_2(Pn)_2(DABCO)$ (in blue) and $Cu_2(Pn)_2(pyz)$ (in green) PCU-MOFs at 298 K.

However, during a 10 ns MD run, all 128 molecules in simulated supercells reached stable assemblies (similar to the ones shown in **Figure 7B**) without further structural changes in material. Both π–π interaction distances and π-slip are significantly changed in comparison to the dimers formed in SURMOF-2 (**Figure 7A**); they are also much more

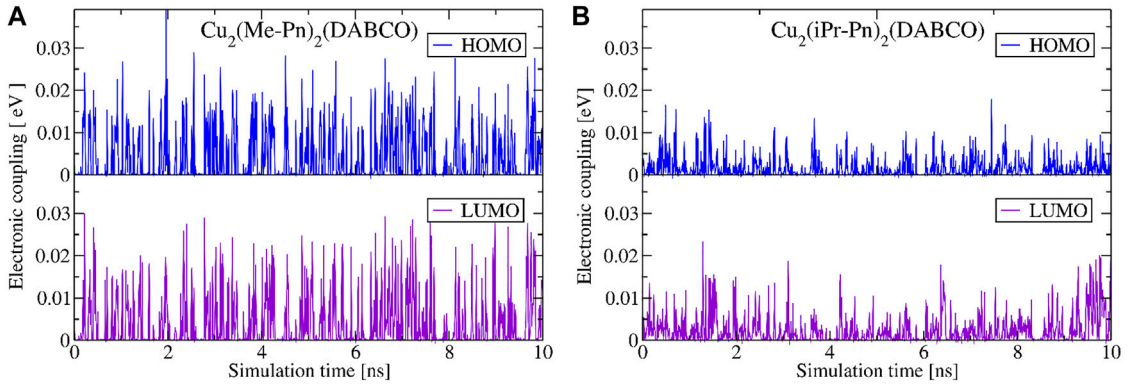


FIGURE 10 | Absolute electronic coupling elements (B3LYP/def2-SVP) between HOMO (in blue) and LUMO (in purple) orbitals of functionalized ditopic pentacene linkers (Me-Pn and iPr-Pn, **Figure 2**) extracted from the force-field MD simulations of Cu_2(Me-Pn)$_2$(DABCO) **(A)** and Cu_2(iPr-Pn)$_2$(pyz) **(B)** PCU-MOFs at 298 K.

stable at room temperature than observed previously (**Figure 8**).

The average values of π–π distances for both $Cu_2(Pn)_2$(DABCO) and $Cu_2(Pn)_2$(pyz) are around 3.75 Å (3.74 and 3.77–3.86 Å, respectively), and the dispersity of their change is relatively small with the variance of 0.058–0.066 Å^2 (for comparison, see plotted π–π distances for SURMOF-2 in **Figure 8A**). A larger COM distance between metal nodes in the PCU MOF, modulated by pillar linkers, results in different packing of pentacene cores in a material that enables more efficient overlap between orbitals, as was also suggested by Zojer and Winkler (2021) by screening different distances between ditopic pentacene linkers. New structuring of OSC cores impacts electronic couplings between HOMO and LUMO orbitals and, therefore, significantly stabilizes average effective coupling (**Figure 9**). From 1,000 snapshots collected in MD, the average value of absolute coupling between HOMO and LUMO orbitals of linkers in $Cu_2(Pn)_2$(DABCO) is 74.5 and 45.2 meV, respectively. The same is also observed for $Cu_2(Pn)_2$(pyz): J_{HOMO} of 77.3 meV and J_{LUMO} of 47.2 meV. The average values of couplings in the last 2 ns of MD are 82.2 and 78.3 meV between HOMO orbitals and 52.0 and 46.7 meV between LUMO orbitals of MOFs with DABCO and pyrazine pillars, respectively. An actual state transition of the electronic coupling is depicted in **Supplementary Figures S8, S9**.

Increasing the COM distance further, e.g., by introducing bipyridine (**Figure 6B**), resulted in the reduction of interactions between the pentacene cores (**Supplementary Figure S4**). Taking into account the energy barrier for the CCCC dihedral angle rotation between the OSC linked to two phenyl rings (**Supplementary Figure S5**) and possible barrier-free rotations by 60–120°, large separation of ditopic linkers eliminates any possibility for an overlap between the adjacent molecules (**Supplementary Figure S6**). Therefore, the $Cu_2(Pn)_2$(bipy) MOF is not a good candidate for charge transport. For this reason, we were not continuing analysis of electronic coupling matrix elements for this system.

Another approach to modify packing of molecules in a MOF can be realized *via* introduction of the steric control units (SCU) (Haldar et al., 2019). We investigated two SCUs also in the present study *via* chemical modification of ditopic pentacene linkers by methyl and isopropyl groups. Hence, we replaced the hydrogen atoms in phenyl rings with methyl groups (Me = CH_3) or isopropyl groups (iPr = $CH(CH_3)_2$) (**Figures 2** and **6C,D**). Both systems were automatically structured in a MOF with the DABCO pillars, i.e., structures Cu_2(Me-Pn)$_2$(DABCO) and Cu_2(iPr-Pn)$_2$(DABCO). Even if the unit cells of these PCU MOFs are very similar, packing of ditopic linkers in the material and the dynamical behavior of the pentacene cores towards finding stable aggregates differ significantly. Selected snapshots of the neighboring OSC cores are depicted in **Supplementary Figure S7**. Addition of SCU introduces steric repulsions between neighboring ditopic linkers, disabling the formation of stable assemblies of pentacene OSC cores, as was observed for $Cu_2(Pn)_2$(DABCO) and $Cu_2(Pn)_2$(pyz). The COM distance is similar in all four cases, but packing modulated by the SCU allows only for a small coupling between the orbitals (**Figure 10**). This packing is not stable at room temperature, suggesting worse semiconducting properties of such MOFs. Moreover, the hindrance of efficient packing of pentacene cores depends on the size of the SCU: the addition of isopropyl group in Cu_2(iPr-Pn)$_2$(DABCO) reduces charge carrier transport in the material (**Figure 10B**), resulting in average electronic coupling of 1.7 meV (HOMO orbitals) and 3.9 meV (LUMO orbitals).

From several examples studied in this report, it is seen that subtle modifications in the chemical composition of MOF components or type of MOF topology result in significant differences in the assemblies of OSC pentacene cores and their dynamically averaged stacking. Introduction of steric control units induces additional steric repulsion, suppressing preferable orientation of pentacene cores (which is observed in the case of pristine ditopic linkers), lowering charge carrier transport. Spatial control of the separation distance between OSC cores *via* the length of the pillar linker allows us to optimize stacking and increase the average effective couplings for the hole transport up to 75 meV, retaining structural stability at finite temperature. Virtual screening of other linker molecules, their packing in a MOF, and resulting electronic properties is possible using automated workflows reported in this study.

4 CONCLUSION

We developed an automated workflow tool for virtual design of new MOF materials of the PCU topology with the subsequent prediction of electronic properties. This reproducible, transferable, and

adaptable computational protocol (Mostaghimi et al., 2020) allows us to build MOF structures, optimize them using different levels of theory, and analyze electronic coupling matrix elements in dynamically averaged assemblies at finite temperature. We demonstrate the functionality of the developed software by preliminary *in silico* crystal engineering of molecular assemblies of pentacene OSC. For this, we constructed MOF structures based on pristine and chemically functionalized pentacene ditopic linkers (as layer linkers) and DABCO, pyrazine, and bipyridine (as pillar linkers). Copper paddlewheel nodes were used to keep the three-dimensional order of organic molecules in a material.

All structures were automatically optimized and further investigated using the MD approach. We have shown the change of the stacking type between ditopic linkers with pentacene OSC cores as a function of the intermolecular separation in MOFs controlled by different sizes of MOF pillars. We have proven that stacking of pristine ditopic linkers is more efficient, where the metal node distances are around 9 Å, enabling better overlap of HOMO and LUMO orbitals of OSC cores, increasing both hole and electron conduction, respectively. Moreover, the analysis of the π–π interactions and π-slip changes during 10 ns MD production runs confirmed higher MOF nanostructure stability at room temperature, addressing the concept reported by Haldar et al. (2021) and supporting hypotheses postulated by Zojer and Winkler (2021). The average effective electronic coupling between OSC in $Cu_2(Pn)_2$(DABCO) and $Cu_2(Pn)_2$(pyz) was calculated to be 78–82 and 46–52 meV between HOMO and LUMO orbitals, respectively.

Using the developed workflow, we have demonstrated the reduction of the electronic conduction propensity of PCU-type MOF materials fabricated from ditopic pentacene linkers with introduced steric control units (methyl and isopropyl groups). This observation resulted from steric repulsions between neighboring organic linkers in the MOF scaffold, suppressing the formation of the optimal interaction assemblies with high molecular orbital overlap integrals. The exceptional sensitivity of the semiconducting properties of MOFs upon the specific details of the molecular assembly inside a material, modulated by subtle chemical modifications of building components, illustrates the importance of screening protocols for both organic molecules and their structuring in MOFs. The availability of the automated workflow tools reported in this study significantly simplifies the tasks involved in these investigations.

DATA AVAILABILITY STATEMENT

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found in the article/**Supplementary Material**.

AUTHOR CONTRIBUTIONS

MM: software and WaNos development (*LCmaker*, *AuToGraFS*, *GULP*, *lammps-interface*, *LAMMPS*, *Achmol*, *SuperCeller*), investigation, visualization, drafting the paper; CR: SimStack support, WaNos development (*DFT-VASP*, *Format-Converter*), paper review; RH: paper review and editing; CW: paper review and editing; WW: conceptualization, resources, funding acquisition, paper review and editing; MK: conceptualization, supervision, investigation, funding acquisition, paper writing, review and editing.

FUNDING

This research has been funded by Deutsche Forschungsgemeinschaft (DFG) SPP priority program COORNETs "Coordination Networks: Building Blocks for Functional Systems" (SPP1928, grant number WE 1863/37-1). Authors acknowledge funding by the DFG under the Germany Excellence Strategy *via* the Excellence Cluster 3D Matter Made to Order (Grant No. EXC-2082/1–390761711) and the GRK 2450 "Scale bridging methods of computational nanoscience", as also the Virtual Materials Design initiative (Virtmat) funded by KIT. WW and CR thank the German Federal Ministry of Education and Research (BMBF) for financial support of the project Innovation-Platform MaterialDigital (www.materialdigital.de) through project funding FKZ no: 13XP5094A. MK acknowledges funding by the Ministry of Science, Research and Art of Baden-Württemberg (Germany) under Brigitte-Schlieben-Lange-Programm.

ACKNOWLEDGMENTS

Authors are grateful to Yohanes Pramudya and Modan Liu for fruitful discussions. We acknowledge support by the KIT-Publication Fund of the Karlsruhe Institute of Technology.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fmats.2022.840644/full#supplementary-material

REFERENCES

Addicoat, M. A., Coupry, D. E., and Heine, T. (2014). AuToGraFS: Automatic Topological Generator for Framework Structures. *The J. Phys. Chem. A* 118, 9607–9614. doi:10.1021/jp507643v

Addicoat, M. A., Vankova, N., Akter, I. F., and Heine, T. (2014). Extension of the Universal Force Field to Metal–Organic Frameworks. *J. Chem. Theor. Comput.* 10, 880–891. doi:10.1021/ct400952t

Alvaro, M., Carbonell, E., Ferrer, B., Llabrés i Xamena, F. X., and Garcia, H. (2007). Semiconductor Behavior of a Metal-Organic Framework (MOF). *Chemistry (Weinheim an Der Bergstrasse, Germany)* 13, 5106–5112. doi:10.1002/chem.200601003

Balasubramani, S. G., Chen, G. P., Coriani, S., Diedenhofen, M., Frank, M. S., Franzke, Y. J., et al. (2020). TURBOMOLE: Modular Program Suite for Ab Initio Quantum-Chemical and Condensed-Matter Simulations. *J. Chem. Phys.* 152, 184107. doi:10.1063/5.0004635

Becke, A. D. (1993). Density-functional Thermochemistry. III. The Role of Exact Exchange. *J. Chem. Phys.* 98, 5648–5652. doi:10.1063/1.464913

Bhattacharya, S., Gnanavel, M., Bhattacharyya, A. J., and Natarajan, S. (2014). Organization of Mn-Clusters in Pcu and Bcu Networks: Synthesis, Structure, and Properties. *Cryst. Growth Des.* 14, 310–325. doi:10.1021/cg401587r

Boyd, P. G., Moosavi, S. M., Witman, M., and Smit, B. (2017). Force-Field Prediction of Materials Properties in Metal-Organic Frameworks. *J. Phys. Chem. Lett.* 8, 357–363. doi:10.1021/acs.jpclett.6b02532

Boyd, P. G., and Woo, T. K. (2016). A Generalized Method for Constructing Hypothetical Nanoporous Materials of Any Net Topology from Graph Theory. *CrystEngComm* 18, 3777–3792. doi:10.1039/C6CE00407E

Bučko, T., Lebègue, S., Hafner, J., and Ángyán, J. G. (2013). Improved Density Dependent Correction for the Description of London Dispersion Forces. *J. Chem. Theor. Comput.* 9, 4293–4299. doi:10.1021/ct400694h

Chung, Y. G., Camp, J., Haranczyk, M., Sikora, B. J., Bury, W., Krungleviciute, V., et al. (2014). Computation-Ready, Experimental Metal–Organic Frameworks: A Tool to Enable High-Throughput Screening of Nanoporous Crystals. *Chem. Mater.* 26, 6185–6192. doi:10.1021/cm502594j

Colón, Y. J., Gómez-Gualdrón, D. A., and Snurr, R. Q. (2017). Topologically Guided, Automated Construction of Metal–Organic Frameworks and Their Evaluation for Energy-Related Applications. *Cryst. Growth Des.* 17, 5801–5810. doi:10.1021/acs.cgd.7b00848

Connolly, B. M., Madden, D. G., Wheatley, A. E. H., and Fairen-Jimenez, D. (2020). Shaping the Future of Fuel: Monolithic Metal–Organic Frameworks for High-Density Gas Storage. *J. Am. Chem. Soc.* 142, 8541–8549. doi:10.1021/jacs.0c00270

Coupry, D. E., Addicoat, M. A., and Heine, T. (2016). Extension of the Universal Force Field for Metal-Organic Frameworks. *J. Chem. Theor. Comput.* 12, 5215–5225. doi:10.1021/acs.jctc.6b00664

Dassault Systèmes (2020). BIOVIA Materials Studio - BIOVIA - Dassault Systèmes®. Available at: https://www.3ds.com/products-services/biovia/products/molecular-modeling-simulation/biovia-materials-studio/.

D'Avino, G., Muccioli, L., Castet, F., Poelking, C., Andrienko, D., Soos, Z. G., et al. (2016). Electrostatic Phenomena in Organic Semiconductors: Fundamentals and Implications for Photovoltaics. *J. Phys. Condensed Matter* 28, 433002. doi:10.1088/0953-8984/28/43/433002

Freund, R., Canossa, S., Cohen, S. M., Yan, W., Deng, H., Guillerm, V., et al. (2021). 25 Years of Reticular Chemistry. *Angew. Chem. Int. Edition* 60, 23946–23974. doi:10.1002/anie.202101644

Friederich, P., Symalla, F., Meded, V., Neumann, T., and Wenzel, W. (2014). Ab Initio treatment of Disorder Effects in Amorphous Organic Materials: Toward Parameter Free Materials Simulation. *J. Chem. Theor. Comput.* 10, 3720–3725. doi:10.1021/ct500418f

Gale, J. D., and Rohl, A. L. (2003). The General Utility Lattice Program (GULP). *Mol. Simulation* 29, 291–341. doi:10.1080/0892702031000104887

Garg, S., Schwartz, H., Kozlowska, M., Kanj, A. B., Müller, K., Wenzel, W., et al. (2019). Conductance Photoswitching of Metal–Organic Frameworks with Embedded Spiropyran. *Angew. Chem. Int. Edition* 58, 1193–1197. doi:10.1002/anie.201811458

Goldman, A., Gil-Hernández, B., Millan, S., Gökpinar, S., Heering, C., Boldog, I., et al. (2020). Flexible Bifunctional Monoethylphosphonate/carboxylates of Zn(II) and Co(II) Reinforced with DABCO Co-ligand: Paradigmatic Structural Organization with Pcu Topology. *CrystEngComm* 22, 2933–2944. doi:10.1039/D0CE00275E

Goswami, S., Ma, L., Martinson, A. B. F., Wasielewski, M. R., Farha, O. K., and Hupp, J. T. (2016). Toward Metal–Organic Framework-Based Solar Cells: Enhancing Directional Exciton Transport by Collapsing Three-Dimensional Film Structures. *ACS Appl. Mater. Inter.* 8, 30863–30870. doi:10.1021/acsami.6b08552

Grimme, S., Antony, J., Ehrlich, S., and Krieg, H. (2010). A Consistent and Accurate Ab Initio Parametrization of Density Functional Dispersion Correction (DFT-D) for the 94 Elements H-Pu. *J. Chem. Phys.* 132, 154104. doi:10.1063/1.3382344

Haldar, R., Heinke, L., and Wöll, C. (2020). Advanced Photoresponsive Materials Using the Metal–Organic Framework Approach. *Adv. Mater.* 32, 1905227. doi:10.1002/adma.201905227

Haldar, R., Kozlowska, M., Ganschow, M., Ghosh, S., Jakoby, M., Chen, H., et al. (2021). Interplay of Structural Dynamics and Electronic Effects in an Engineered Assembly of Pentacene in a Metal–Organic Framework. *Chem. Sci.* 12, 4477–4483. doi:10.1039/D0SC07073D

Haldar, R., Mazel, A., Krstic, M., Zhang, Q., Jakoby, M., Howard, I. A., et al. (2019). A De Novo Strategy for Predictive crystal Engineering to Tune Excitonic Coupling. *Nat. Commun.* 10, 2048. doi:10.1038/s41467-019-10011-8

Heinke, L., Cakici, M., Dommaschk, M., Grosjean, S., Herges, R., Bräse, S., et al. (2014). Photoswitching in Two-Component Surface-Mounted Metal–Organic Frameworks: Optically Triggered Release from a Molecular Container. *ACS Nano* 8, 1463–1467. doi:10.1021/nn405469g

Heinke, L., and Wöll, C. (2019). Surface-Mounted Metal–Organic Frameworks: Crystalline and Porous Molecular Assemblies for Fundamental Insights and Advanced Applications. *Adv. Mater.* 31, 1806324. doi:10.1002/adma.201806324

Jurchescu, O. D., Popinciuc, M., van Wees, B. J., and Palstra, T. T. M. (2007). Interface-controlled, High-Mobility Organic Transistors. *Adv. Mater.* 19, 688–692. doi:10.1002/adma.200600929

Kalmutzki, M. J., Hanikel, N., and Yaghi, O. M. (2018). Secondary Building Units as the Turning point in the Development of the Reticular Chemistry of MOFs. *Sci. Adv.* 4, eaat9180. doi:10.1126/sciadv.aat9180

Kanj, A. B., Müller, K., and Heinke, L. (2018). Stimuli-Responsive Metal-Organic Frameworks with Photoswitchable Azobenzene Side Groups. *Macromolecular Rapid Commun.* 39, 1700239. doi:10.1002/marc.201700239

Kera, S., Yamane, H., and Ueno, N. (2009). First-principles Measurements of Charge Mobility in Organic Semiconductors: Valence Hole–Vibration Coupling in Organic Ultrathin Films. *Prog. Surf. Sci.* 84, 135–154. doi:10.1016/j.progsurf.2009.03.002

Kreno, L. E., Leong, K., Farha, O. K., Allendorf, M., Van Duyne, R. P., and Hupp, J. T. (2012). Metal–Organic Framework Materials as Chemical Sensors. *Chem. Rev.* 112, 1105–1125. doi:10.1021/cr200324t

Kresse, G., and Hafner, J. (1993). Ab Initio molecular Dynamics for Liquid Metals. *Phys. Rev. B* 47, 558–561. doi:10.1103/physrevb.47.558

Larsen, A. H., Mortensen, J. J., Blomqvist, J., Castelli, I. E., Christensen, R., Du\lak, M., et al. (2017). The Atomic Simulation Environment—A Python Library for Working with Atoms. *J. Phys. Condensed Matter* 29, 273002. doi:10.1088/1361-648X/aa680e

Lee, C., Yang, W., and Parr, R. G. (1988). Development of the Colle-Salvetti Correlation-Energy Formula into a Functional of the Electron Density. *Phys. Rev. B* 37, 785–789. doi:10.1103/PhysRevB.37.785

Li, H., Eddaoudi, M., Groy, T. L., and Yaghi, O. M. (1998). Establishing Microporosity in Open Metal-Organic Frameworks:Gas Sorption Isotherms for Zn (BDC)(BDC= 1 ,4-Benzenedicarboxylate). *J. Am. Chem. Soc.* 120, 8571–8572. doi:10.1021/ja981669x

Li, H., Eddaoudi, M., O'Keeffe, M., and Yaghi, O. M. (1999). Design and Synthesis of an Exceptionally Stable and Highly Porous Metal-Organic Framework. *Nature* 402, 276–279. doi:10.1038/46248

Li, H., Wang, K., Sun, Y., Lollar, C. T., Li, J., and Zhou, H. C. (2018). Recent Advances in Gas Storage and Separation Using Metal–Organic Frameworks. *Mater. Today* 21, 108–121. doi:10.1016/j.mattod.2017.07.006

Li, J. R., Tao, Y., Yu, Q., and Bu, X. H. (2007). A Pcu-type Metal–Organic Framework with Spindle [Zn 7 (OH) 8] 6+ Cluster as Secondary Building Units. *Chem. Commun.* 0, 1527–1529. doi:10.1039/B615675D

Lin, J. D., Rong, C., Lv, R. X., Wang, Z. J., Long, X. F., Guo, G. C., et al. (2018). A 3D Metal-Organic Framework with a Pcu Net Constructed from Lead(II) and Thiophene-2, 5-dicarboxylic Acid: Synthesis, Structure and Ferroelectric Property. *J. Solid State. Chem.* 257, 34–39. doi:10.1016/j.jssc.2017.09.001

Lin, R. B., Xiang, S., Zhou, W., and Chen, B. (2020). Microporous Metal-Organic Framework Materials for Gas Separation. *Chem* 6, 337–363. doi:10.1016/j.chempr.2019.10.012

Liu, B., Ma, M., Zacher, D., Bétard, A., Yusenko, K., Metzler-Nolte, N., et al. (2011). Chemistry of SURMOFs: Layer-Selective Installation of Functional Groups and Post-synthetic Covalent Modification Probed by Fluorescence Microscopy. *J. Am. Chem. Soc.* 133, 1734–1737. doi:10.1021/ja1109826

Liu, B., Shekhah, O., Arslan, H. K., Liu, J., Wöll, C., and Fischer, R. A. (2012). Enantiopure Metal–Organic Framework Thin Films: Oriented SURMOF Growth and Enantioselective Adsorption. *Angew. Chem. Int. Edition* 51, 807–810. doi:10.1002/anie.201104240

Liu, J., and Wöll, C. (2017). Surface-supported Metal–Organic Framework Thin Films: Fabrication Methods, Applications, and Challenges. *Chem. Soc. Rev.* 46, 5730–5770. doi:10.1039/C7CS00315C

Liu, X., Kozlowska, M., Okkali, T., Wagner, D., Higashino, T., Brenner-Weifss, G., et al. (2019). Photoconductivity in Metal-Organic Framework (MOF) Thin Films. *Angew. Chemie-International Edition* 58, 9590–9595. doi:10.1002/anie.201904475

Luo, T. Y., Liu, C., Eliseeva, S. V., Muldoon, P. F., Petoud, S., and Rosi, N. L. (2017). Rare Earth Pcu Metal–Organic Framework Platform Based on RE4(*M*3-OH)4(COO)62+ Clusters: Rational Design, Directed Synthesis, and Deliberate Tuning of Excitation Wavelengths. *J. Am. Chem. Soc.* 139, 9333–9340. doi:10.1021/jacs.7b04532

Lyu, H., Ji, Z., Wuttke, S., and Yaghi, O. M. (2020). Digital Reticular Chemistry. *Chem* 6, 2219–2241. doi:10.1016/j.chempr.2020.08.008

Mailman, A., Leitch, A. A., Yong, W., Steven, E., Winter, S. M., Claridge, R. C. M., et al. (2017). The Power of Packing: Metallization of an Organic Semiconductor. *J. Am. Chem. Soc.* 139, 2180–2183. doi:10.1021/jacs.6b12814

Mancuso, J. L., Mroz, A. M., Le, K. N., and Hendon, C. H. (2020). Electronic Structure Modeling of Metal–Organic Frameworks. *Chem. Rev.* 120, 8641–8715. doi:10.1021/acs.chemrev.0c00148

Marcus, R. A. (1993). Electron Transfer Reactions in Chemistry. Theory and experiment. *Rev. Mod. Phys.* 65, 1111–1121. doi:10.1002/anie.199311113

Martin, R. L., and Haranczyk, M. (2014). Construction and Characterization of Structure Models of Crystalline Porous Polymers. *Cryst. Growth Des.* 14, 2431–2440. doi:10.1021/cg500158c

Moosavi, S. M., Nandy, A., Jablonka, K. M., Ongari, D., Janet, J. P., Boyd, P. G., et al. (2020). Understanding the Diversity of the Metal-Organic Framework Ecosystem. *Nat. Commun.* 11, 4068. doi:10.1038/s41467-020-17755-8

Mostaghimi, M., Rêgo, C., Wenzel, W., and Kozlowska, M. (2020). *PCU-MOF Workflow*. Genève, Switzerland: Zenodo. doi:10.5281/zenodo.5893325

Narayan, T. C., Miyakai, T., Seki, S., and Dincă, M. (2012). High Charge Mobility in a Tetrathiafulvalene-Based Microporous Metal–Organic Framework. *J. Am. Chem. Soc.* 134, 12932–12935. doi:10.1021/ja3059827

Nefedov, A., Haldar, R., Xu, Z., Kühner, H., Hofmann, D., Goll, D., et al. (2021). Avoiding the Center-Symmetry Trap: Programmed Assembly of Dipolar Precursors into Porous, Crystalline Molecular Thin Films. *Adv. Mater.* 33, 2103287. doi:10.1002/adma.202103287

Neumann, T., Liu, J., Wächter, T., Friederich, P., Symalla, F., Welle, A., et al. (2016). Superexchange Charge Transport in Loaded Metal Organic Frameworks. *ACS Nano* 10, 7085–7093. doi:10.1021/acsnano.6b03226

O'Boyle, N. M., Banck, M., James, C. A., Morley, C., Vandermeersch, T., and Hutchison, G. R. (2011). Open Babel: An Open Chemical Toolbox. *J. Cheminformatics* 3, 33. doi:10.1186/1758-2946-3-33

Öhrström, L. (2015). Let's Talk about MOFs—Topology and Terminology of Metal-Organic Frameworks and Why We Need Them. *Crystals* 5, 154–162. doi:10.3390/cryst5010154

O'Keeffe, M., Peskov, M. A., Ramsden, S. J., and Yaghi, O. M. (2008). The Reticular Chemistry Structure Resource (RCSR) Database of, and Symbols for, crystal Nets. *Acc. Chem. Res.* 41, 1782–1789. doi:10.1021/ar800124u

O'Keeffe, M., and Yaghi, O. M. (2012). Deconstructing the Crystal Structures of Metal–Organic Frameworks and Related Materials into Their Underlying Nets. *Chem. Rev.* 112, 675–702. doi:10.1021/cr200205j

Perdew, J. P., Burke, K., and Ernzerhof, M. (19961997). Generalized Gradient Approximation Made Simple. *Phys. Rev. Lett.Physical Rev. Lett.* 7778, 38651396–1396. doi:10.1103/PhysRevLett.78.1396

Plimpton, S., Kohlmeyer, A., Thompson, A., Moore, S., and Berger, R. (2020). *LAMMPS Stable Release 29 October 2020*. Genève, Switzerland: Zenodo. doi:10.5281/zenodo.4157471

Pohl, U. W. (2020). "Electronic Properties of Organic Semiconductors," in *Epitaxy of Semiconductors: Physics and Fabrication of Heterostructures*. Editor U. W. Pohl (Cham, Switzerland: Springer International Publishing), 177–205. Graduate Texts in Physics. doi:10.1007/978-3-030-43869-2_5

Qiao, J., Liu, X., Liu, X., Liu, X., Zhang, L., and Liu, Y. (2020). Two Urea-Functionalized Pcu Metal–Organic Frameworks Based on a Pillared-Layer Strategy for Gas Adsorption and Separation. *Inorg. Chem. Front.* 7, 3500–3508. doi:10.1039/D0QI00641F

Ricardo, C. (2021). *KIT-Workflows/DFT-VASP: DFT-VASP*. Genève, Switzerland: Zenodo. doi:10.5281/zenodo.5501531

Rowsell, J. L. C., and Yaghi, O. M. (2004). Metal–organic Frameworks: A New Class of Porous Materials. *Microporous Mesoporous Mater.* 73, 3–14. doi:10.1016/j.micromeso.2004.03.034

Schlöder, T., and Rêgo, C. (2022). Kit-workflows/dft-turbomole. Available at https://github.com/KIT-Workflows (accessed Sept 12, 2021).

Schoedel, A., and Yaghi, O. M. (2016). "Porosity in Metal-Organic Compounds," in *Macrocyclic and Supramolecular Chemistry*. Editor R. M. Izatt (Chichester, UK: John Wiley & Sons), 200–219. doi:10.1002/9781119053859.ch9

Stock, N., and Biswas, S. (2012). Synthesis of Metal-Organic Frameworks (MOFs): Routes to Various MOF Topologies, Morphologies, and Composites. *Chem. Rev.* 112, 933–969. doi:10.1021/cr200304e

Turcani, L., Tarzia, A., Szczypiński, F. T., and Jelfs, K. E. (2021). Stk: An Extendable Python Framework for Automated Molecular and Supramolecular Structure Assembly and Discovery. *J. Chem. Phys.* 154, 214102. doi:10.1063/5.0049708

Virkar, A., Mannsfeld, S., Oh, J. H., Toney, M. F., Tan, Y. H., Liu, G. Y., et al. (2009). The Role of OTS Density on Pentacene and C60 Nucleation, Thin Film Growth, and Transistor Performance. *Adv. Funct. Mater.* 19, 1962–1970. doi:10.1002/adfm.200801727

Wei, W., Wang, X., Zhang, K., Tian, C. B., and Du, S. W. (2019). Tuning the Topology from Fcu to Pcu: Synthesis and Magnetocaloric Effect of Metal–Organic Frameworks Based on a Hexanuclear Gd(III)-Hydroxy Cluster. *Cryst. Growth Des.* 19, 55–59. doi:10.1021/acs.cgd.8b01566

Weigend, F., and Ahlrichs, R. (2005). Balanced Basis Sets of Split Valence, Triple Zeta Valence and Quadruple Zeta Valence Quality for H to Rn: Design and Assessment of Accuracy. *Phys. Chem. Chem. Phys.* 7, 3297. doi:10.1039/b508541a

Xie, L. S., Skorupskii, G., and Dincă, M. (2020). Electrically Conductive Metal–Organic Frameworks. *Chem. Rev.* 120, 8536–8580. doi:10.1021/acs.chemrev.9b00766

Xing, J., Schweighauser, L., Okada, S., Harano, K., and Nakamura, E. (2019). Atomistic Structures and Dynamics of Prenucleation Clusters in MOF-2 and MOF-5 Syntheses. *Nat. Commun.* 10, 3608. doi:10.1038/s41467-019-11564-4

Xu, G., Guo, G. C., Yao, M. S., Fu, Z. H., and Wang, G. E. (2016). *The Chemistry of Metal–Organic Frameworks*. Hoboken, New Jersey, United States: John Wiley & Sons, 421–462. chap. 14. doi:10.1002/9783527693078.ch14Functional Linkers for Electron-Conducting MOFs

Yao, S. L., Tian, X. M., Li, L. Q., Liu, S. J., Zheng, T. F., Chen, Y. Q., et al. (2019). A CdII-Based Metal-Organic Framework with Pcu Topology as Turn-On Fluorescent Sensor for Al3+. *Chem. – Asian J.* 14, 3648–3654. doi:10.1002/asia.201900739

Yazdanparast, M. S., Day, V. W., and Gadzikwa, T. (2020). Hydrogen-Bonding Linkers Yield a Large-Pore, Non-catenated, Metal-Organic Framework with Pcu Topology. *Molecules* 25, 697. doi:10.3390/molecules25030697

Zhuang, J. L., Terfort, A., and Wöll, C. (2016). Formation of Oriented and Patterned Films of Metal–Organic Frameworks by Liquid Phase Epitaxy: A Review. *Coord. Chem. Rev.* 307, 391–424. doi:10.1016/j.ccr.2015.09.013

Zojer, E., and Winkler, C. (2021). Maximizing the Carrier Mobilities of Metal–Organic Frameworks Comprising Stacked Pentacene Units. *J. Phys. Chem. Lett.* 12, 7002–7009. doi:10.1021/acs.jpclett.1c01892

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers in Materials

ORIGINAL RESEARCH
published: 21 March 2022
doi: 10.3389/fmats.2022.851458

Fast All-Electron Hybrid Functionals and Their Application to Rare-Earth Iron Garnets

Matthias Redies[1,2,3], *Gregor Michalicek*[1], *Juba Bouaziz*[1], *Christian Terboven*[4], *Matthias S. Müller*[2,4], *Stefan Blügel*[1,2] *and Daniel Wortmann*[1]*

[1]Peter Grünberg Institut and Institute for Advanced Simulation, Forschungszentrum Jülich, Jülich, Germany, [2]JARA-CSD, Jülich, Germany, [3]Department of Physics, RWTH Aachen University, Aachen, Germany, [4]IT Center, RWTH Aachen University, Aachen, Germany

Virtual materials design requires not only the simulation of a huge number of systems, but also of systems with ever larger sizes and through increasingly accurate models of the electronic structure. These can be provided by density functional theory (DFT) using not only simple local approximations to the unknown exchange and correlation functional, but also more complex approaches such as hybrid functionals, which include some part of Hartree–Fock exact exchange. While hybrid functionals allow many properties such as lattice constants, bond lengths, magnetic moments and band gaps, to be calculated with improved accuracy, they require the calculation of a nonlocal potential, resulting in high computational costs, that scale rapidly with the system size. This limits their wide application. Here, we present a new highly-scalable implementation of the nonlocal Hartree-Fock-type potential into FLEUR—an all-electron electronic structure code that implements the full-potential linearized augmented plane-wave (FLAPW) method. This implementation enables the use of hybrid functionals for systems with several hundred atoms. By porting this algorithm to GPU accelerators, we can leverage future exascale supercomputers which we demonstrate by reporting scaling results for up to 64 GPUs and up to 12,000 CPU cores for a single **k**-point. As proof of principle, we apply the algorithm to large and complex iron garnet materials (YIG, GdIG, TmIG) that are used in several spintronic applications.

Keywords: Density Functional Theory (DFT), Rare-Earth Iron Garnets, High-Performance Computing (HPC), PBE0, Hybrid Functionals, YIG, GdIG, TmIG

OPEN ACCESS

Edited by:
Zhenyu Li,
University of Science and Technology of China, China

Reviewed by:
Honghui Shang,
Institute of Computing Technology (CAS), China
Mohan Chen,
Peking University, China
Wei Hu,
Lawrence Berkeley National Laboratory, United States

****Correspondence:***
Daniel Wortmann
d.wortmann@fz-juelich.de

Specialty section:
This article was submitted to Computational Materials Science, a section of the journal Frontiers in Materials

Received: *09 January 2022*
Accepted: *22 February 2022*
Published: *21 March 2022*

Citation:
Redies M, Michalicek G, Bouaziz J, Terboven C, Müller MS, Blügel S and Wortmann D (2022) Fast All-Electron Hybrid Functionals and Their Application to Rare-Earth Iron Garnets. Front. Mater. 9:851458. doi: 10.3389/fmats.2022.851458

1 INTRODUCTION

Materials science aims to understand and predict material properties ever more accurately, so that new sophisticated materials can be discovered to drive innovation in domains that rely on them. While materials science has been around for millennia, it was only at the beginning of the last century that the arrival of quantum mechanics enabled the exact description of the microscopic properties in materials. However, the cost of calculating the exact solution to the Schrödinger equation grows exponentially with the size of the system and is therefore limited to very small systems. Density functional theory (DFT) replaces the $3N$-dimensional wave function as the central quantity with the 3-dimensional ground-state density and thereby reduces the exponential computational cost to a polynomial one. While DFT is in principle exact, a key ingredient, the so-called exchange-correlation

energy, has no known analytical expression. The approximations used for this term determine the accuracy with which material properties can be predicted. While the most commonly used approximations, the local density approximation (LDA) and the generalized gradient approximation (GGA), can predict certain properties with a high precision at a very low computational cost, they fail to predict some essential electronic properties in particular of electronically complex materials (Alberi et al., 2018).

DFT is increasingly being used in the context of high-throughput calculations, where hundreds of thousands of material candidates are screened using automated workflows (Yan et al., 2017; Mounet et al., 2018; Rosen et al., 2019). However, all of these calculations are limited to material classes and properties for which the underlying exchange-correlation functionals have a good predictive power. In order to enhance these calculations with material classes and properties for which LDA and GGA fail, it is necessary to rely on more accurate methods producing high-quality results. One class of accurate methods are the hybrid exchange-correlation functionals which are particularly suited to predicting electronic properties such as the band gap, the degree of charge localization and the polarization in materials with a stronger electron correlation (Cramer and Truhlar, 2009; Zhang et al., 2011; Burke, 2012; Becke, 2014; Garza and Scuseria, 2016).

Hybrid exchange-correlation functionals, such as PBE0 (Perdew et al., 1996) or HSE06 (Krukau et al., 2006) functionals, mix a portion of an orbital dependent exact exchange with the electron correlation described by other approximations, such as LDA or GGA (Becke, 1993). Their reliance on the orbital dependent exact exchange makes them computationally considerably more expensive than LDA or GGA. While an LDA or a GGA calculation grows with the 3rd power of the number of atoms, a hybrid exchange-correlation functional calculation typically grows with the 4th power of the number of atoms. Additionally, the computational cost of a hybrid calculation grows quadratically with the number of **k**-points used to sample the Brillouin zone, whereas for an LDA or a GGA calculation it only grows linearly. This large computational cost has prohibited precise predictions for systems with large unit cells, including a number of interesting material classes such as garnets (Rodic et al., 1999; Nakamoto et al., 2017) or materials of interest for solid-state batteries (Yu et al., 2016).

This article focuses on the implementation of hybrid functionals in the full-potential linearized augmented-plane-wave (FLAPW) (Wimmer et al., 1981) method as it is implemented in the open-source code FLEUR (Fleur, 2021). Unlike approaches employing the pseudopotential approximation, the FLAPW method treats all electrons explicitly and does not employ any approximations to represent the potential or density. It is therefore well suited for a wide range of systems, including systems containing heavy atoms that have *d*- and *f*-electrons. It is considered to be one of the most accurate DFT methods and has been used as a benchmark for other methods and codes (Lejaeghere et al., 2016). More specifically we focus here on the efficient implementation of the Hartree-Fock type exact exchange based on the bare Coulomb kernel as relevant for the PBE0 functional. Functionals based on the screened Coulomb kernel as HSE06 can be always expressed in terms of the matrix elements of the bare Coulomb kernel subtracted by matrix elements of a smooth function (Schlipf et al., 2011), whose numerical evaluation is not time critical and is not further discussed here.

There have been significant advances in bringing hybrid functionals to systems with hundreds or even thousands of atoms in other approaches, such as the projector augmented wave method (PAW) (Barnes et al., 2017; Carnimeo et al., 2019), the s-MTACE METHOD (Mandal et al., 2021), gaussian basis functions (Guidon et al., 2008) and atomic-orbitals method (Hakala and Foster, 2013; Lin et al., 2021). Even some all-electron methods have demonstrated their capability to calculate large systems with hybrid functionals (Ihrig et al., 2015; Levchenko et al., 2015). However, hybrid functionals within FLAPW have been constrained to very small systems (Betzinger et al., 2010; Schlipf et al., 2011; Blaha et al., 2020). The work presented here enables FLEUR's hybrid functional implementation to run on the world's most advanced supercomputers and use their immense computational power to investigate these large and interesting systems containing hundreds of atoms. Building on the pioneering work previously done on hybrid functionals in the FLAPW basis and in FLEUR specifically (Betzinger et al., 2010; Schlipf et al., 2011), we analyzed the performance and bottlenecks of this legacy implementation, and explored algorithmic improvements needed to calculate hundreds of atoms with the accuracy that FLAPW and hybrid functionals offer.

2 METHODS

The FLAPW method, implemented by FLEUR, partitions the unit cell of volume Ω into two kinds of domains. In spherical regions MT_a centered around each atomic nucleus, a muffin-tin orbital basis (Andersen and Woolley, 1973) is used, relying on the products of spherical harmonics and radial functions. In between these spheres, in the so-called interstitial region (IS), a plane-wave basis is used. The resulting LAPW basis functions used to represent the wave functions are

$$\varphi^{\sigma}_{\mathbf{kG}}(\mathbf{r}) = \begin{cases} \frac{1}{\sqrt{\Omega}} \exp\left[i\left(\mathbf{k}+\mathbf{G}\right)\cdot\mathbf{r}\right] & \text{if } \mathbf{r} \in \mathrm{IS} \\ \sum_{lm}\left[\alpha^{a\sigma}_{lm}\left(\mathbf{k},\mathbf{G}\right)u^{a\sigma}_{l}\left(\|\mathbf{r}_a\|\right) + \beta^{a\sigma}_{lm}\left(\mathbf{k},\mathbf{G}\right)\dot{u}^{a\sigma}_{l}\left(\|\mathbf{r}_a\|\right)\right]Y_{lm}\left(\mathbf{e}_a\right) & \text{if } \mathbf{r} \in \mathrm{MT}_a \end{cases}. \quad (1)$$

Here **k** and **G** are the Bloch- and reciprocal lattice vectors, while σ indicates the spin. $\alpha^{a\sigma}_{lm}$ and $\beta^{a\sigma}_{lm}$ are coefficients chosen, such that the wave function is smooth and continuous at the boundary between the interstitial region and muffin-tin spheres. u and $\dot{u}$ are radial functions, where u is the solution to the radial Schrödinger equation for the spherically averaged muffin-tin potential and a fixed energy parameter and $\dot{u}$ is its energy derivative. a indicates the nucleus, l and m are the orbital- and magnetic quantum numbers of the spherical harmonic Y_{lm}. **r** denotes the position, while $\mathbf{r}_a := \mathbf{r} - \mathbf{R}_a$ is the position relative to the

center of the muffin-tin sphere and $\mathbf{e}_a := \mathbf{r}_a / \text{norm}\,\mathbf{r}_a$ is the unit vector in direction of $\mathbf{r}_a$.

In order to calculate the Hartree-Fock exact exchange, the Coulomb integral $\iint \frac{\varphi_i^*(\mathbf{r})\varphi_j^*(\mathbf{r}')\varphi_j(\mathbf{r})\varphi_i(\mathbf{r}')}{\mathbf{r}-\mathbf{r}'}$, containing four basis functions needs to be evaluated. It has previously been noted (Boys and Shavitt, 1959; Shavitt, 1959), that the product of the basis functions sharing the same argument $\varphi_i^*(\mathbf{r})\varphi_j(\mathbf{r})$ and $\varphi_j^*(\mathbf{r}')\varphi_i(\mathbf{r}')$ can be expressed in a more efficient way, since the basis function are already designed to be complete and the set of all products $\{\varphi_i^*(\mathbf{r})\varphi_j(\mathbf{r})\}$ makes a linear dependent set. In the case of the LAPW basis, this observation can be exploited by employing the so-called mixed-product basis. The mixed-product basis reduces the basis set for the muffin-tin regions separately from the interstitial region. In the muffin-tin regions, the overlap matrix of the products is calculated and diagonalized. The eigenvectors whose eigenvalues are above a certain threshold ν then provide a linear-independent representation of the product space. In the interstitial region products of plane-waves are also planes-waves, but with a higher cut-off $G'_{\max} = 2G_{\max}$, where $G_{\max}$ is the plane-wave cut-off of the LAPW basis. In practice reduced values of $G'_{\max}$ have proven to provide accurate results. While this new mixed-product basis is neither continuous nor smooth, it provides a significant reduction in computational effort compared to the naive evaluation of the Coulomb integral. A detailed description of the mixed-product basis can be found in (Friedrich et al., 2009).

2.1 Exact Exchange

Using this basis set the nonlocal exact exchange can be expressed as

$$V^{\text{exact}}_{\sigma,nn'}(\mathbf{k}) = -\sum_{n''}^{N_{\text{occ}}} \sum_{\mathbf{q}}^{\text{BZ}} \sum_{IJ} \langle \phi^{\sigma}_{n\mathbf{k}} | \phi^{\sigma}_{n''\mathbf{k}-\mathbf{q}} M_{\mathbf{q},I} \rangle \mathcal{C}_{IJ}(\mathbf{q}) \langle M_{\mathbf{q},J} \phi^{\sigma}_{n''\mathbf{k}-\mathbf{q}} | \phi^{\sigma}_{n'\mathbf{k}} \rangle, \tag{2}$$

where n, n' and n'' are band indices of the states $\phi^{\sigma}_{n\mathbf{k}}$, $\phi^{\sigma}_{n'\mathbf{k}}$ and $\phi^{\sigma}_{n''\mathbf{k}}$. I and J are indices enumerating the mixed-product basis and $\mathcal{C}$ is the Coulomb matrix expressed in this basis. A detailed derivation of M and $\mathcal{C}$ can be found in (Friedrich et al., 2009). Note that while the sum n'' only stretches over occupied states, n and n' cover all states. The square Coulomb matrix $\mathcal{C}$ with the dimensions of the size of the mixed-product basis is largely sparse, which allows for a significant reduction in the computational effort. The product $\mathcal{C}_{IJ}(\mathbf{q}) \langle M_{\mathbf{q},J} \phi^{\sigma}_{n''\mathbf{k}-\mathbf{q}} | \phi^{\sigma}_{n'\mathbf{k}} \rangle$ is referred to "Sparse matmul" in **Figure 1**; **Figures 3–5**. The projector matrix $P_{\sigma,n,\mathbf{k}}(n'', \mathbf{q}, I) = \langle \phi^{\sigma}_{n\mathbf{k}} | \phi^{\sigma}_{n''\mathbf{k}-\mathbf{q}} M_{\mathbf{q},I} \rangle$ has the dimensions of the size of the mixed-product basis and the number of states. The evaluation of this term is split into two components, one called "inters. wave-prod" and one called "MT wave-prod", referring to the evaluation of this term either within the interstitial region or the muffin-tins. In order to apply $V^{\text{exact}}_{\sigma,nn'}(\mathbf{k})$ to the Hamiltonian in the LAPW basis it is transformed from the eigenspace to the LAPW basis by applying the overlap matrix of the LAPW matrix and the eigenvector matrix. In **Figure 1**; **Figures 3–5** the full evaluation of the non-local potential, i.e., the setup of the Coulomb matrix, the evaluation of **Eq. 2** and the transformation into the LAPW basis together, is denoted as "non-local pot."

The numerical evaluation of **Eq. 2** represents the majority of the computational effort in a FLEUR calculation using hybrid functionals. In particular, the projection of the products of wave functions given in the LAPW basis set (**Eq. 1**) onto the mixed-product basis and the multiplication of these projections with the Coulomb matrix provide significant computational challenges. The implementation developed for this work relies on collecting data processed in the same way. It allows to exploit data parallelism on multiple levels, be it the use of a SIMD instruction set or an efficient and balanced use of multiple threads. Two significant changes have been made to the basic algorithm introduced in (Betzinger et al., 2010). First, contrary to the previous implementation, the projection onto the mixed-product basis in the interstitial region is now calculated by Fourier transforming the wave-functions into real-space and multiplying pairs of wave-functions there, before transforming them back into **G**-space. By employing fast Fourier transformations, this reduces the complexity of this calculation from $\mathcal{O}(n_{\mathbf{G}}{}^2)$ to $\mathcal{O}(n_{\mathbf{G}} \log(n_{\mathbf{G}}))$, where $n_{\mathbf{G}}$ is the number of LAPW basis functions. Second, rather than calculating all elements of $V^{\text{exact}}_{\sigma}(\mathbf{k})$ individually as vector-matrix-vector products of the Coulomb matrix and the mixed-product basis, the new implementation stacks groups of vectors of the mixed-product-basis into matrices and then calculates blocks of $V^{\text{exact}}_{\sigma}(\mathbf{k})$ as a single matrix-matrix-matrix product. While these operations are mathematically identical, this new block-wise implementation is twice as fast as the element-wise implementation even on a single CPU core, which is due to its better utilization of the core's vector units. Additionally, the element-wise evaluation of this term experiences almost no speedup if multiple cores are used, while the speedup of the modern implementation is shown in **Figure 1** in blue. Calculating the non-local potential on a single NVIDIA A100 GPU results in a speedup of 4× compared to an AMD EPYC 7742 CPU for the NaCl system with 64 atoms.

2.2 Shared Memory Parallelization

To enable the utilization of supercomputers with complex memory hierarchies, we rely on two classes of parallelization. We use shared memory parallelization to make full use of many-core CPUs or GPUs. While distributed memory parallelization is employed to distribute the calculation over hundreds of compute nodes. Shared memory parallelization is realized by utilizing libraries for standard math problems such as matrix-multiplications or Fourier transformations whenever possible. Code parts that do not fall within the mold of any standard math problem were parallelized using OpenMP on CPUs and OpenACC on GPUs. In **Figure 1** the strong scaling behavior on a single node is shown. For strong scaling a fixed-size problem is calculated with an increasing amount of resources and the resulting speedup is measured. Here, the speedup is defined as $S_n := \frac{T_{n_{\min}}}{T_n}$, which in the case of ideal scaling behaviour is $S_n^{\text{ideal}} = \frac{n}{n_{\min}}$, where T_n is the runtime with n cores, nodes or GPUs and $n_{\min}$ is the minimal value of n that was used in a calculation. This can be used to define the parallel efficiency as $\tau_n := \frac{S_n}{S_n^{\text{ideal}}}$.

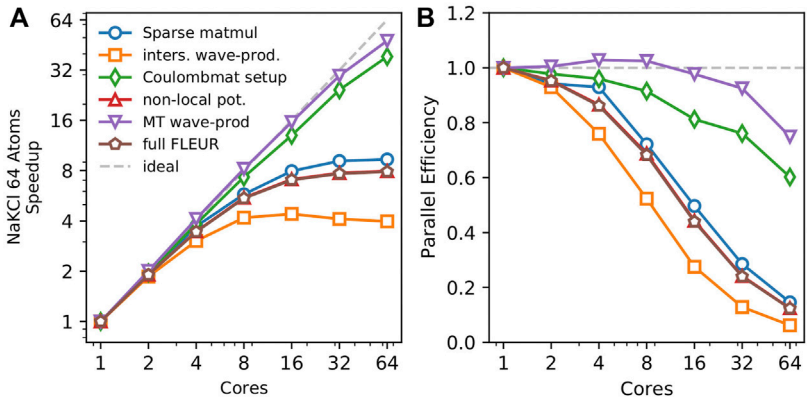


FIGURE 1 | Strong scaling behavior with OpenMP on a single AMD EPYC 7742 64 core processor. The overall FLEUR iteration is shown with brown pentagons, while the calculation of the nonlocal potential is shown in red triangles. The four remaining lines show the major parts of the nonlocal potential. The red triangles indicating the nonlocal potential largely coincide with the brown pentagons indicating the full runtime, making it difficult to see them. The parallel efficiency of slightly above 100% in the routine for the setup mixed-product basis for 4 and 8 nodes is explained by cache effects. Depending on the number of cores used the stride of the parallelized loops executed on each core is changed, which can reduce the number of cache misses if this stride coincides with certain array dimensions.

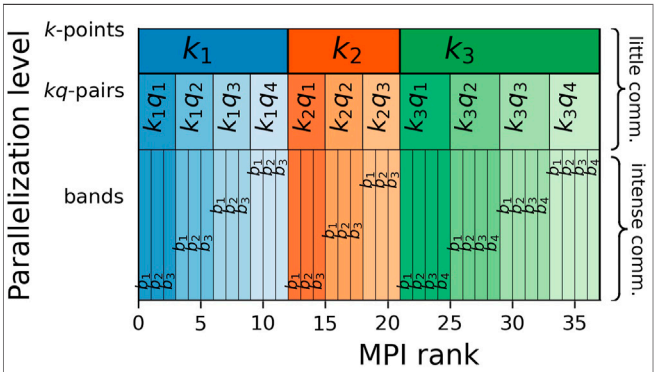


FIGURE 2 | The distributed-memory parallelization of **Eq. 2** is divided into three levels. For each **k**-point the exact exchange is calculated as an independent problem. At a **k**-point $\mathbf{k}_i$, the exact exchange is calculated as a sum over all **q**-points associated with $\mathbf{k}_i$, building the **kq**-pairs. These first two levels require very little communication, i.e., copying the final results to their destination for the **k**-points and a reduction within the sub-communicator of each **k**-point for the **q**-point sum. The third level of distributed-memory parallelization calculates groups of occupied bands n'' in parallel. Here, a lot of inter-dependencies create a much larger communication demand compared to the first two levels. In a typical calculation of the non-local potential the workload is not uniformly distributed: Some **k**-points have more **q**-points than others and some **kq**-pairs might have more associated bands than others. The algorithm attempts to compensate this by assigning more nodes to heavier calculations.

While some parts, such as the projection onto the mixed-product basis in the muffin-tins or the setup of the Coulomb matrix show excellent scaling, the speedup of the projection on the mixed-product basis in the interstitial exhibits a plateau around a speedup of 4 (see orange line). As mentioned previously, this algorithm relies on fast Fourier transformations, which have a low algorithmic intensity, meaning that only few calculations are performed compared to the number of load and store operations, making the algorithm more likely to be limited by the memory bandwidth rather than the available computational power, explaining the plateau in the speedup seen in **Figure 1**. Up until 8 cores, the FFT still benefits from the additional compute resources, beyond that the FFTs are not limited anymore by the computational power, but rather by the memory bandwidth, which does not increase with the number of assigned cores.

2.3 Distributed Memory Parallelization

The parallelism shown so far is limited to a single shared memory node and thus limited by the number of cores on a given node. Therefore, in order to scale the computational challenge posed by the hybrid exchange-correlation functionals to hundreds of nodes, we implemented three additional levels of distributed memory parallelism using MPI. The first two levels distribute the computations for different **k**- and **q**-points, while the third level parallelizes that of different occupied bands n''. The distributed memory parallelization scheme is shown in **Figure 2**.

The parallelization over **k**- and **q**-points requires very little communication and thus is very efficient, while the band-parallelization requires more communication. However, it allows us to limit the size of the largest matrix stored on a single node to $n_{\text{basis size}} \times n_{\text{total bands}}$, which then has a size on the order of $\mathcal{O}({n_{\text{atoms}}}^2)$. This turns out to be the bottleneck that determines the largest system we can calculate on a given computing platform. With 90 GB of memory per node we were able to calculate systems with up to 200 atoms.

Figure 3 singles out the strong scaling behavior of this 3rd MPI level for a single **k**- and **q**-point. All code parts except for the setup of the Coulomb matrix show a good scaling behavior with a parallel efficiency of over 50% on either 64 CPU nodes or 64 GPUs. The scaling behavior of the Coulomb-matrix setup is not critical, since it does not dominate the run time of the calculation even on 256 nodes. Additionally, it only scales linearly with the number of **k**-points whereas the other parts of the nonlocal potential scale quadratically with the number of **k**-points. To

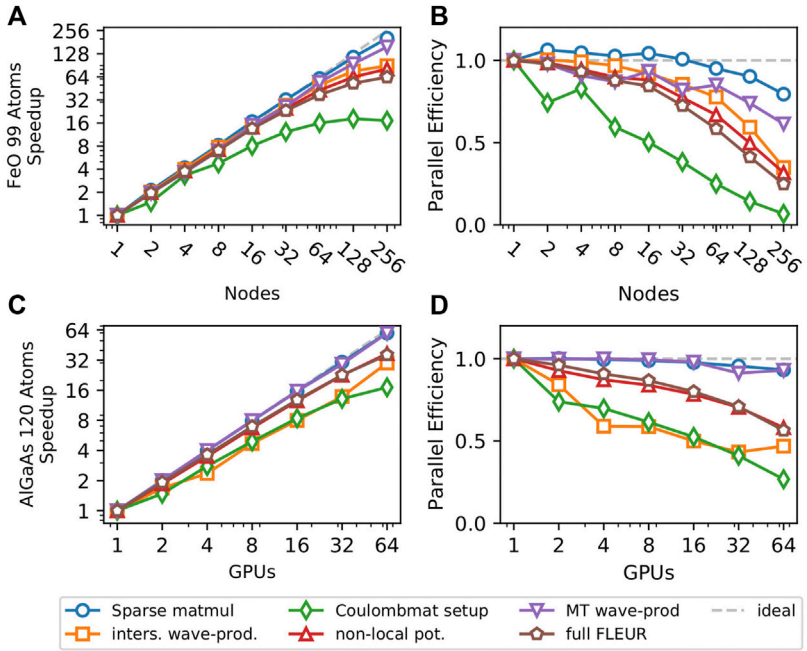


FIGURE 3 | Scaling behavior of systems with a single **k**-point on two types of architectures. Panels **(A)** and **(B)** show scaling of a 99-atom FeO supercell with a vacancy defect on the CPU-based SuperMUC-NG supercomputer, while **(C)** and **(D)** show the scaling behavior of a 120-atom GaAs supercell with an Al defect on JURECA's GPU partition. Panels **(A)** and **(C)** show the speedup, while **(B)** and **(D)** show the corresponding parallel efficiency.

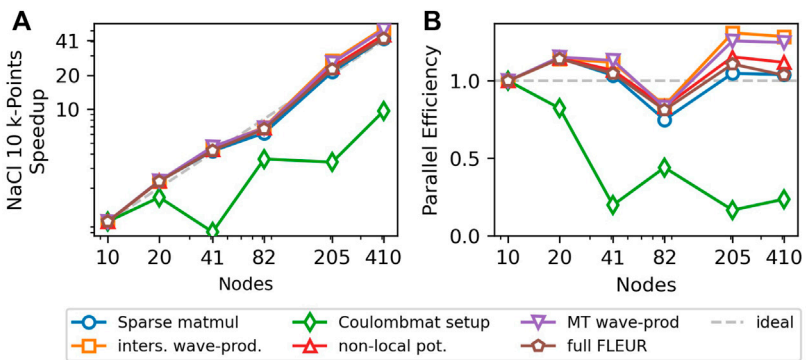


FIGURE 4 | Strong scaling behavior of multiple **k**-points for a 64 atom NaCl supercell with a potassium (K) defect. The system has 10 **k**-points and 205 **kq**-pairs. The super-scalar behavior is caused by the fact, that the 205 **kq**-pairs are not evenly distributed on 10 nodes (20 MPI). Some nodes are assigned more **kq**-pairs and therefore need longer while the others sit idle. This effect disappears for 205 or 410 nodes, which allow for a perfectly even and therefore more efficient distribution. For 41 nodes, the speedup and parallel efficiency of the Coulomb-matrix setup drop drastically. This is due to the fact, that the Coulomb-matrix setup does not have a **q**-dependence, while the number of nodes is chosen to be optimal for the evaluation of **Eq. 2**. For 10 nodes (20 MPI), all **k**-points can be calculated in parallel on 2 processes each, while for 41 nodes (82 MPI), it is only possible to calculate 2 **k**-points in parallel, so that each is distributed over 41 processes, leading to an inefficient parallelization. In practical calculations this is mitigated by including more nodes (e.g., 45), so that both **k**- and **kq**-parallelization are efficient. However, even with 41 nodes, the Coulomb-matrix setup only accounts for 6% of the total runtime.

investigate the performance of our algorithm with multiple **k**-points we show the strong scaling behavior for a NaCl supercell with 64 atoms, 10 **k**-points and 205 **kq**-pairs in **Figure 4**. Here the scaling behavior of the Coulomb-matrix setup is the weakest once again, but it only accounts for less than 10%, even with 410 nodes. All other code parts show nearly perfect scaling. This is due to the fact that each **kq**-pair represents a largely independent problem that only requires little communication and the 3rd MPI level is only used with 205 and 410 nodes, since the parallelization over the 205 **kq**-pairs is preferred. SuperMUC-NG has two CPUs per node and therefore we assigned two MPI ranks to each node, resulting in a better performance compared to a one rank-per-node setup.

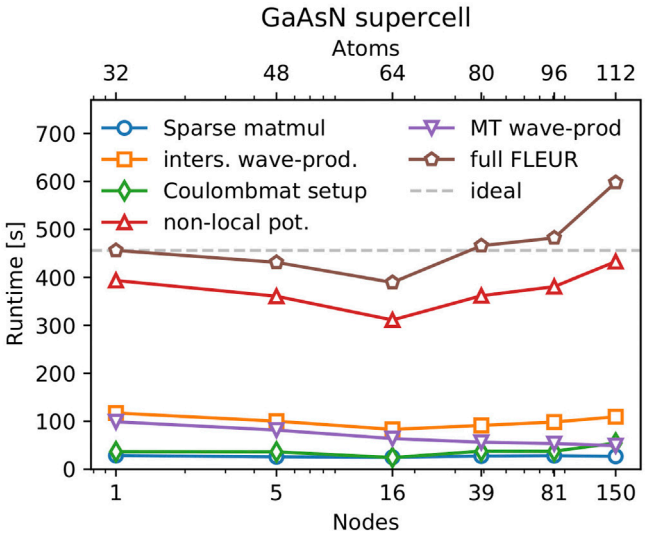


FIGURE 5 | Weak scaling behavior of FLEUR's hybrid functional calculations for a GaAs system which is scaled into a supercell with one Arsenic atom being substituted with Nitrogen. The *y*-axis shows the runtime for different code parts, while the bottom *x*-axis shows the number of nodes used. The top *x*-axis shows the number of atoms in that particular system.

2.4 Weak Scaling

While the meaning of strong scaling is very intuitive, it does not necessarily reflect real life applications. Being able to calculate a system with twenty atoms in a minute or less may not advance science significantly. Certainly, science is advanced by being able to calculate increasingly bigger, more inhomogeneous and more complex systems in a reasonable time frame. Weak scaling deals with the latter. As discussed in the introduction, the computational demand of a hybrid functional calculation scales with

$$\mathcal{O}(n_{\text{atom}}{}^4). \tag{3}$$

For simplicity and to focus on the ultimately limiting parallelization level, we use a single **kq**-pair and neglect the very efficient parallelization over different **k**- and **q**-vectors. In **Figure 5** a gallium arsenide (GaAs) setup was scaled into supercells with a single nitrogen defect. Then, they were calculated with the parallelization chosen such that

$$n_{\text{nodes}} = \left(\frac{n_{\text{atoms}}}{min(n_{\text{atoms}})}\right)^4, \tag{4}$$

where $\min(n_{\text{atoms}})$ is the number of atoms in the smallest supercell.

With ideal weak scaling behavior the runtime should be constant regardless of the size of the unit cell, since the computational cost in **Eq. 3** is canceled out by the additional compute resources chosen in **Eq. 4**. **Figure 5** shows that the hybrid functionals in FLEUR can be applied efficiently to a wide variety of system sizes. The time needed for the calculation of the nonlocal potential of the largest GaAs supercell is 9% larger compared to that of the smallest supercell and the full iteration runtime is 30% larger. The runtime does not monotonously increase as one would expect for the weak scaling of a simple algorithm performing a single task. In FLEUR, the situation is more complex, some parts of the code scale with $\mathcal{O}(n_{\text{atom}}{}^3)$, while others scale with $\mathcal{O}(n_{\text{atom}}{}^4)$: While the setup of the mixed-product basis in the muffin-tin spheres grows with $\mathcal{O}(n_{\text{atom}}{}^3)$, its counterpart in the interstitial region grows with $\mathcal{O}(n_{\text{atom}}{}^3 log(n_{\text{atom}}))$. In the Coulomb-matrix setup, some parts such as the MT-MT interaction grow with $\mathcal{O}(n_{\text{atom}}{}^2)$, while e.g., Γ-point correction for in the interstitial grows with $\mathcal{O}(n_{\text{atom}}{}^4)$. For larger systems terms with a bigger scaling-exponent will be dominant, but in small systems the parts with the smaller scaling-exponents dominate the runtime. In these cases the choice of **Eq. 4** is not suitable, because the compute resources are increased faster than the computational complexity grows, leading to the initial dip in the overall runtime in **Figure 5**.

3 APPLICATION TO RARE-EARTH IRON GARNETS

Yttrium iron garnet ($Y_3Fe_5O_{12}$ or short *YIG*) is a complex ferrimagnetic insulator with a number of remarkable properties and applications, in the fields of magnonics (Serga et al., 2010), ultra-low temperature physics (Demokritov et al., 2006) and quantum computing (Tabuchi et al., 2015). This success has sparked interest in a related class of materials, the so-called rare-earth-iron garnets (RIGs), where the yttrium atom in the YIG structure is replaced with an element of the lanthanide series. Here applications range from materials with giant magnetostriction (Sayetat, 1986) to spin Seebeck insulators (Uchida et al., 2010). Despite great interest in these materials there is only a limited number of theoretical studies of their electronic structure. This is most likely due to the large unit cells with 160 atoms in the conventional and 80 atoms in the primitive unit cell.

The typical unit cell of a garnet is shown in **Figure 6**. The iron atoms in this structure have two types of environments. They are either in the centre of an octahedron or a tetrahedron spanned by neighbouring oxygen atoms. These different iron environments have a strong effect on the electronic structure, which is discussed in detail later in this paper. YIG and most RIGs are ferrimagnets, such that the magnetic moments of the 8 octahedral iron atoms point in the opposite direction with respect to the 12 tetrahedral iron atoms, which, for the RIGs discussed here, are aligned in parallel with the rare-earth elements. Only a very minor magnetic moment is induced in the yttrium and oxygen atoms.

3.1 Electronic Structure

In order to understand how the choice of the exchange-correlation functional affects the electronic structure of YIG we calculated the density-of-states (DOS) with PBE and with PBE0. All calculations shown in the paper were performed on a $2 \times 2 \times 2$ **k**-point grid. We confirmed that the DOS is converged with this grid by comparing the PBE results to results on a denser **k**-point grid. We use a smearing of $\sigma = 0.136$ eV for all DOS calculations shown. The muffin-tin radii of Y, Gd, Tm, Fe and O

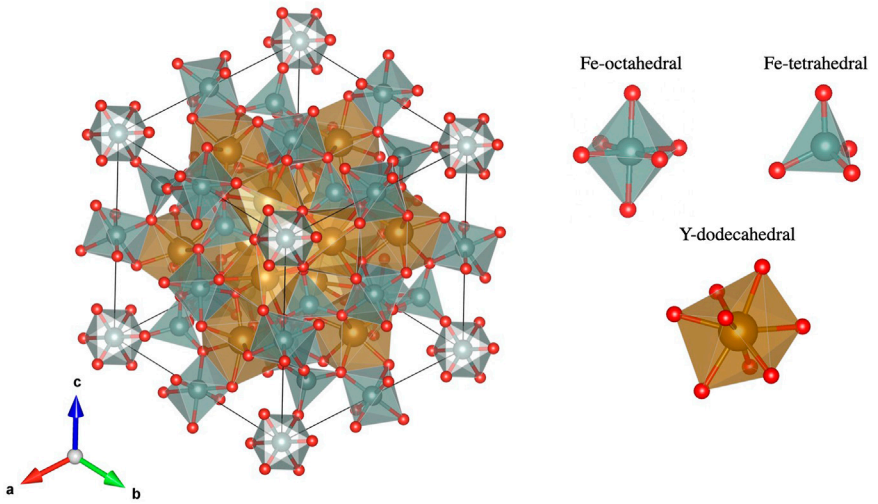


FIGURE 6 | Unit cell of a garnet. Oxygen atoms are shown in red, while iron atoms are shown inside the grey polyhedra. The rare-earth or yttrium atoms are shown inside the golden dodecahedra. While the yttrium or rare-earth atoms are all symmetry equivalent the iron is present in two different environments. Structure from (Y3Fe5O12 crystal structure, 2012) and plotted with (Momma and Izumi, 2011).

are chosen to be $r_{Y/Gd/Tm} = 2.8\ a_0$, $r_{Fe} = 2.14\ a_0$ and $r_O = 1.21\ a_0$. The structural information, e.g. the unit cell and the atomic positions used in this chapter are based on the experimental ones, exhibiting a $Ia\bar{3}d$-structure (Y3Fe5O12 crystal structure, 2012; Gd3FeO12 crystal structure, 2012; Tm3Fe5O12 crystal structure, 2012).

As expected, with a value of 0.44 eV, PBE massively underestimates the experimental band gap of 2.8 eV (Larsen and Metselaar, 1975), while PBE0 predicts an improved band gap of 1.83 eV. However, the experimental value relies on optical measurements, which are not sensitive to all transitions, potentially missing certain states and thus overestimating the real band gap.

In **Figure 7A**) the DOS of YIG is calculated using PBE as an exchange-correlation functional. In this figure, the antiferrimagnetic alignment of the iron atoms is visible: the occupied states associated with the tetrahedral iron atoms are mainly in the spin-up channel and the unoccupied ones are in the spin-down channel, while for the octahedral iron atoms the situation is reversed: below the Fermi level the octahedral iron states are mostly in the spin-down channel and above it in the spin-up one. Most states associated with the oxygen atoms are occupied, while the yttrium states are largely unoccupied. Below the Fermi level, the DOS in the interstitial region closely follows the oxygen DOS. Additionally, the DOS associated with both iron types also coincide with the oxygen and interstitial DOS. This indicates that the 2*p*-states of the oxygen and the 3*d*-states of iron hybridize for both iron environments. This analysis is supported by the number of valence electrons found in the different muffin-tin spheres, which are 6.5 and 6.2 electrons for iron atoms in the tetrahedral and octhedral environments, respectively, 1.1 valence electrons in the sphere of yttrium, and an average of 3.7 valence electrons in the spheres of oxygen. The large number of 164.1 electrons in the interstitial region additionally indicates a high degree of de-localization of these states. For the unoccupied octahedral iron states in contrast, we can see a clear signature of simple crystal field splitting of localized *d*-states: the three t_{2g}-states shift down and the two e_g-states shift up leading to two distinct peaks with the lower one containing three and the higher one containing two states. Similarly, for the unoccupied tetrahedral Fe *d*-levels the *e*-states are shifted down, while the t_2-states are shifted up. This separation however, is not so clear as the shifts are smaller, the peaks still overlap and another splitting due to next-nearest neighbors can be seen.

In **Figure 7B**) we show the DOS calculated using the hybrid exchange-correlation functional PBE0. The results are qualitatively different from the PBE results with the most significant change seen in the different behavior of the two types of Fe in the PBE results: While the occupied tetrahedral iron 3*d*-states still hybridize with the 2*p*-states of the surrounding oxygen atoms, most of the octahedral iron 3*d*-states are now strongly localized and form a double peak in the DOS at around −6.5 eV.

Such a localization effect of the *d*-states can also be reproduced in a PBE+U treatment (Chen et al., 2021). However, in these simulations the *d*-states of both Fe types show the same behavior. The different tendency to localize can also be seen in these simulation by the different values of *U* used for the different atoms to achieve the localization. Hence, the result strongly suggests that the local Coulomb interaction exhibits different strength in the two environments of Fe. This effect can be caused by the different initial localization of the *d*-states as well as by different interactions and screening effects of the surrounding. Such a difference can also be seen in the unoccupied spectrum which is again dominated by a crystal field splitting of the *d*-states. However, in the octahedral environment this effect is again much clearer while the tetrahedral *d*-states form a broad band with several peaks also indicating next-nearest neighbor effects.

Finally, we would like to point out that the octahedral Fe *d*-states show a rather complex sub-structure with a large peak at

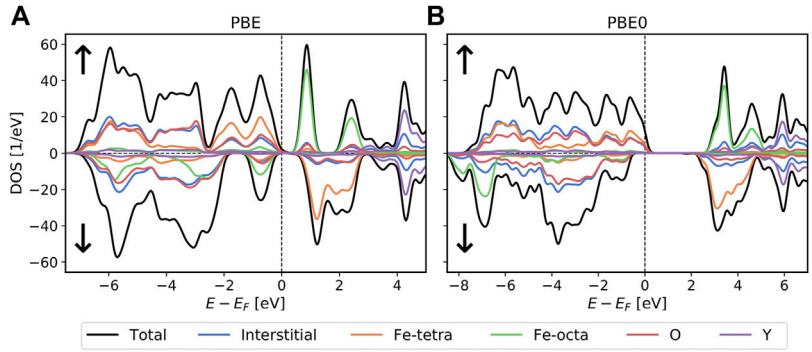


FIGURE 7 | DOS of YIG calculated with PBE in **(A)** and with PBE0 in **(B)** on a 2 × 2 × 2 **k**-grid. In both calculations we use a $K_{max} = 4.5a_0^{-1}$. The mixed-product basis for the calculation in **(B)** uses a eigenvalue threshold of $\nu = 10^{-4}$ and an $l_{MPB} = 16$ cutoff for the spherical harmonics.

TABLE 1 | The magnetic moments within the different muffin-tins of both Fe types in units of μ_B.

		Fe tetra. [μ_B]	Fe octa. [μ_B]
PBE		3.52	−3.64
PBE0		3.83	−4.01
PBE + U	Nakamoto et al. (2017)	4.10	−4.20
QSGW	Barker et al. (2020)	3.93	−4.17
exp. $R\bar{3}$	Rodic et al. (1999)	3.95	−4.01
exp. $Ia\bar{3}d$	Rodic et al. (1999)	5.37	−4.11

− 7 eV and a minor peak at − 8 eV. This is not a crystal field splitting but rather shows that different states with a different degree of localization are formed. While the lower peak is clearly separated from the O *p*-states, some remaining hybridization can be identified for the larger peak.

Further investigations of the consequences of these differences between the Fe atoms is beyond the scope of this paper, but we expect this electronic structure to have some influence, e.g. on magnetic interactions and the transition temperature.

3.2 Magnetic Moment

In the introduction we discussed that some key applications of YIG are related to its magnetic properties. Therefore, we want to investigate the precision of our predictions for magnetic properties with different exchange-correlation functionals. In **Table 1** we compare the magnetic moments predicted for the different iron atom types. We use the magnetic moment inside the muffin-tin sphere to assign the moment to a specific atom. Therefore, the magnetic moment depends on the choice of the radius of the muffin-tin sphere and, strictly speaking, is not uniquely defined. The magnetization calculated for the oxygen and yttrium atoms is negligible regardless of the computational method used. The total magnetic moment per unit formula was 5 μ_B for every functional. This agreement is expected, since YIG is a magnetic insulator, which constrains the total magnetization per unit cell to integer values.

While PBE predicts the magnetic moments of the two iron types within only ±0.5 μ_B of the experimental value for the $R\bar{3}$ crystal structure, the predictions by PBE0 are even closer to the experimental those results. This again can be understood by the observed tendency to localize the Fe *d*-states and compares very well to magnetic moments predicted by Barker et al. (Barker et al., 2020) obtained using QSGW, another highly precise electronic structure method. The slight difference in the magnetic moments between the QSGW and PBE0 approach we account to the different choice of the muffin-tin radii and the different degree of localization of the Fe *d*-states. Interestingly, in the comparison of PBE0 to QSGW in the case of this octahedral iron, the magnetic moment for this octahedral Fe agrees better with the experimental results in the $R\bar{3}$ crystal structure than the QSGW value, supporting our findings of a slightly stronger localized *d*-wave function in the case of the PBE0 functional. We note that all theoretical results were calculated for the cubic $Ia\bar{3}d$ crystal structure and the magnetic moments of Fe in the tetrahedral environment agree quite well with each other but also with the experimental values of Fe in the trigonal $R\bar{3}$ structure. On the other hand, the experimental Fe moment in the $Ia\bar{3}d$ symmetry is completely off. It shows a moment of 5.37 μ_B. This value seems unrealistic since it is higher than that of a free Fe^{3+} ion (~ 5 μ_B), while the presence of hybridization with oxygen is expected to lower the moment further. We conclude that further experimental efforts are needed to analyze the structure-magnetism relationship of YIG.

3.3 Rare-earth-iron Garnets

As two representatives of the rare-earth-iron garnet group, we chose to examine $Gd_3Fe_5O_{12}$ (GdIG) and $Tm_3Fe_5O_{12}$ (TmIG) more closely. We selected these materials, because a lot of interesting experimental (Fechine et al., 2008; Phan et al., 2009; Lassri et al., 2011; Lee et al., 2020; Vilela et al., 2020; Vu et al., 2020) and even some theoretical work using the FLAPW method (Lassri et al., 2011) has been published for these materials.

In **Figure 8** we present the density of states for GdIG and TmIG calculated with the PBE0 exchange-correlation functional. Reaching numerical self-consistency for TmIG was challenging with PBE, which is the starting point for any PBE0 calculation. We achieved self-consistency by using a few hundred straight mixing iterations with a low mixing parameter, followed by a set of Anderson mixing iterations until convergence was reached. With a converged PBE as a

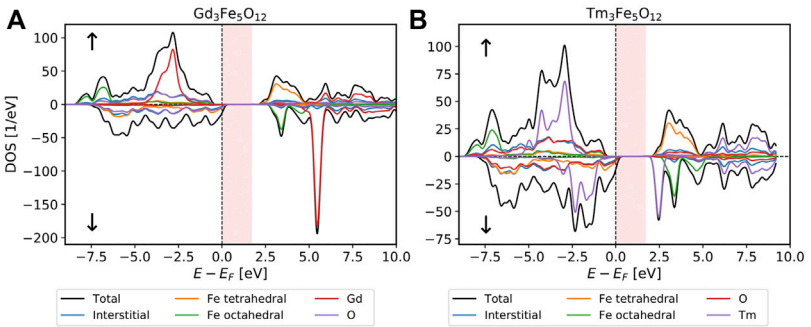


FIGURE 8 | The density-of-states is calculated for GdIG in **(A)** and for TmIG in **(B)** using PBE0 on a 2 × 2 × 2 **k**-point grid. Both calculations were performed with a $K_{max} = 4.5a_0^{-1}$ and the mixed-product basis was setup using $\nu = 10^{-4}$ and $l_{MPB} = 16$. Both band gaps are 1.7 eV and marked in red. The Gd states are fully occupied for the majority spin-channel and fully unoccupied for the minority spin-channel. The Tm spin-up channel is also fully occupied while the minority spin channel is only partially occupied.

TABLE 2 | Predicted magnetic moments of GdIG and TmIG for each atom type, given in units of [μ_B].

		Fe tetra [μ_B]	Fe octa [μ_B]	Gd/Tm [μ_B]	O [μ_B]
$Gd_3Fe_5O_{12}$	PBE	−3.54	3.69	6.88	−0.06
	PBE0	−3.85	4.04	6.94	−0.06
	PBE + U (Nakamoto et al., 2017)	−4.1	4.2	7.0	–
$Tm_3Fe_5O_{12}$	PBE	−3.298	3.59	1.61	−0.04
	PBE0	−3.82	4.01	1.93	−0.05
	PBE + U (Nakamoto et al., 2017)	−4.1	4.2	1.9	–

starting density the convergence of PBE0 is straight forward. This difficult convergence is caused by the metallic behavior of TmIG with PBE as a functional. After the non-local potential is included, a gap opens up and all later density convergence cycles do not exhibit this problematic metallic behavior. GdIG converged without problems both for PBE and PBE0.

For GdIG the band gap was calculated to be 1.7 eV with PBE0. Literature values obtained using PBE + U suggest a gap of 1.6 eV (Nakamoto et al., 2017). For TmIG we also predict a band gap of 1.7 eV using PBE0. To our knowledge, this is the first prediction for the band gap of TmIG. We are not aware of any experimental results regarding the band gap in either system.

The electronic structure of these two systems has a few striking similarities with that of YIG. The 3*d*-states of both types of iron atoms hybridize with the oxygen 2*p*-states in PBE, while with PBE0 the octahedral iron states show localization and a strong shift to lower energies. This again highlights the difference of the tetrahedral and octahedral oxygen environment of the iron atoms causing different effective interactions at these atoms and casting doubt on simple PBE+U predictions for these garnet systems. For the unoccupied octahedral iron states we can see the typical signature of crystal field splitting and in the tetrahedral case this signature is weaker. The additional 4*f*-states of the rare-earth elements in the spin-up channel are strongly localized in PBE. In PBE0 they show a slightly larger bandwidth, indicating increased hybridization with the oxygen 2*p*-states which could be understood due to the decrease of hybridization of these states with the octahedral Fe *d*-states. As expected, Gd has no occupied 4*f*-states in the spin-down channel, while the 4*f*-states of Tm are partially occupied, causing a metallic behavior in PBE. In PBE0 the increased interaction provided by the exchange term opens a gap in the Tm 4*f*-band.

In **Table 2** the magnetic moments of all atom types are given. For GdIG we predict a total magnetization per formula unit of 16.0 μ_B and for TmIG we predict 1.75 μ_B for PBE as well as PBE0. Notice, that the formula unit contains 20 atoms, while the primitive unit cell contains 80. This means, while the magnetic moment per formula unit is not integer, it is integer per unit cell.

The predicted total magnetic moments are in exact agreement with experimental results for GdIG (Geller et al., 1965), while they are in good agreement with the experimental value of 1.2 μ_B for the TmIG. This experimental value would correspond to a total magnetic moment of 4.8 μ_B for the primitive unit cell. PBE + U shows a tendency to predict larger magnetic moments for almost all atoms: 4.2 μ_B for the octahedral iron, − 4.1 μ_B for the tetrahedral iron, 7.0 μ_B for Gd and 1.9 μ_B for Tm (Nakamoto et al., 2017).

4 CONCLUSION

In this article we presented a highly scalable implementation of hybrid exchange-correlation functionals in the LAPW basis. In this work we focused on the scalable implementation of the Hartree-Fock exact exchange, which corresponds to the implementation of the PBE0 functional, but screened functionals like HSE06 are related by an additional fast computation of a smooth

function. The combination of shared and distributed memory parallelization allows to calculate a broad range of systems with high efficiency. Combining all three MPI levels gives us an outlook on the scaling potential of this algorithm. If we were for example to calculate the GaAs system with 120 atoms and we would use 8 **k**-points we would get 125 **kq**-pairs. **Figure 3** shows that for this system a single **kq**-pair has a good parallel performance even if distributed over more than 32 GPUs. Therefore, it is reasonable to assume that the calculation of the nonlocal potential for a system with 8 **k**-points would still have good scaling with $32\,\frac{\text{GPUs}}{\mathbf{kq}-\text{pair}} \times 125\,\text{GPUs} = 4000$ GPUs, which is ~ 250 more than the 44 PetaFLOP JUWELS Booster Module has to offer. This not only allows the code to run on the supercomputers currently available, it also gives us confidence that our code can make good practical use of future exascale machines. Here, making good practical use of a supercomputer does not necessarily mean sending jobs which queue for weeks-on-end and then scale to every single core which the machine has to offer, but rather that we can efficiently use significant portions of the machine to investigate interesting and meaningful systems.

Using the new implementation of the hybrid functional code, we performed simulations of the electronic structure of iron based garnet materials. The significant improvement in the obtained band gap as well as the changes in the electronic structures discussed in detail demonstrate the significance and power of this treatment for these technological relevant material class. Our results suggests an experimental reevaluation of the structure-magnetism relation of the yttrium iron garnet (YIG), $Y_3Fe_5O_{12}$.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

MR, DW, and GM performed and analyzed the initial performance measurements. MR, DW, and GM designed the 3-Level MPI parallelism with feedback form CT and MM. CT and MM suggested solutions for the performance bottlenecks in MPIs one-sided communication. MR, DW, GM implemented shared-memory parallelism using OpenMP, while CT contributed memory blocking for certain OpenMP kernels. MR performed the performance measurements and the electronic structure calculation for the garnet materials. JB and SB helped with the analysis and understanding of the electronic structure of the garnet materials. JB created **Figure 5**.

FUNDING

The authors gratefully acknowledge the computing time granted through JARA on the supercomputer JURECA at Forschungszentrum Jülich as well as the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu). for funding this project by providing computing time on the SuperMUC-NG Supercomputer. The work is funded by the JARA-CSD School for Simulation and Data Science (SSD) and by the MaX Center of Excellence funded by the EU through the H2020-INFRAEDI-2018 (project: GA 824143).

ACKNOWLEDGMENTS

We would like to acknowledge fruitful discussions with Christoph Friedrich.

REFERENCES

Alberi, K., Nardelli, M. B., Zakutayev, A., Mitas, L., Curtarolo, S., Jain, A., et al. (2018). The 2019 Materials by Design Roadmap. *J. Phys. D: Appl. Phys.* 52, 013001. doi:10.1088/1361-6463/aad926

Andersen, O. K., and Woolley, R. G. (1973). Muffin-Tin Orbitals and Molecular Calculations: General Formalism. *Mol. Phys.* 26, 905–927. doi:10.1080/00268977300102171

Barker, J., Pashov, D., and Jackson, J. (2020). Electronic Structure and Finite Temperature Magnetism of Yttrium Iron Garnet. *Electron. Struct.* 2, 044002. doi:10.1088/2516-1075/abd097

Barnes, T. A., Kurth, T., Carrier, P., Wichmann, N., Prendergast, D., Kent, P. R. C., et al. (2017). Improved Treatment of Exact Exchange in Quantum Espresso. *Computer Phys. Commun.* 214, 52–58. doi:10.1016/j.cpc.2017.01.008

Becke, A. D. (1993). A New Mixing of Hartree-Fock and Local Density-Functional Theories. *J. Chem. Phys.* 98, 1372–1377. doi:10.1063/1.464304

Becke, A. D. (2014). Perspective: Fifty Years of Density-Functional Theory in Chemical Physics. *J. Chem. Phys.* 140, 18A301. doi:10.1063/1.4869598

Betzinger, M., Friedrich, C., and Blügel, S. (2010). Hybrid Functionals within the All-Electron FLAPW Method: Implementation and Applications of PBE0. *Phys. Rev. B* 81, 195117. doi:10.1103/PhysRevB.81.195117

Blaha, P., Schwarz, K., Tran, F., Laskowski, R., Madsen, G. K. H., and Marks, L. D. (2020). Wien2k: An APW+lo Program for Calculating the Properties of Solids. *J. Chem. Phys.* 152, 074101. doi:10.1063/1.5143061

Boys, S. F., and Shavitt, I. (1959). Technical Report WIS-AF-13, University of Wisconsin.

Burke, K. (2012). Perspective on Density Functional Theory. *J. Chem. Phys.* 136, 150901. doi:10.1063/1.4704546

Carnimeo, I., Baroni, S., and Giannozzi, P. (2019). Fast Hybrid Density-Functional Computations Using Plane-Wave Basis Sets. *Electron. Struct.* 1, 015009. doi:10.1088/2516-1075/aaf7d4

Chen, J., Hsu, H.-S., and Lo, F.-Y. (2021). Spin-Dependent Optical Transitions in Yttrium Iron Garnet. *Mater. Res. Express* 8, 026101. doi:10.1088/2053-1591/abe013

Cramer, C. J., and Truhlar, D. G. (2009). Density Functional Theory for Transition Metals and Transition Metal Chemistry. *Phys. Chem. Chem. Phys.* 11, 10757–10816. doi:10.1039/B907148B

Demokritov, S. O., Demidov, V. E., Dzyapko, O., Melkov, G. A., Serga, A. A., Hillebrands, B., et al. (2006). Bose-Einstein Condensation of Quasi-Equilibrium Magnons at Room Temperature under Pumping. *Nature* 443, 430–433. doi:10.1038/nature05117

Fechine, P. B. A., Moretzsohn, R. S. T., Costa, R. C. S., Derov, J., Stewart, J. W., Drehman, A. J., et al. (2008). Magnego-Dielectric Properties of the Y3Fe5O12 and Gd3Fe5O12 Dielectric Ferrite Resonator Antennas. *Microw. Opt. Technol. Lett.* 50, 2852–2857. doi:10.1002/mop.23824

Fleur (2021). Forschungszentrum Jülich - PGI-1 & IAS-1. Available at: https://www.flapw.de.

Friedrich, C., Schindlmayr, A., and Blügel, S. (2009). Efficient Calculation of the Coulomb Matrix and its Expansion Around k=0 within the FLAPW Method. *Computer Phys. Commun.* 180, 347–359. doi:10.1016/j.cpc.2008.10.009

Garza, A. J., and Scuseria, G. E. (2016). Predicting Band Gaps with Hybrid Density Functionals. *J. Phys. Chem. Lett.* 7, 4165–4170. doi:10.1021/acs.jpclett.6b01807

Gd3FeO12 crystal structure (2012). *Datasheet from "Pauling File Multinaries Edition – 2012" in Springermaterials* (Japan: Springer-Verlag Berlin Heidelberg & Material Phases Data System (MPDS), Switzerland & National Institute for Materials Science NIMS). Available at: https://materials.springer.com/isp/crystallographic/docs/sd_0308674. Copyright 2016 .

Geller, S., Remeika, J. P., Sherwood, R. C., Williams, H. J., and Espinosa, G. P. (1965). Magnetic Study of the Heavier Rare-Earth Iron Garnets. *Phys. Rev.* 137, A1034–A1038. doi:10.1103/PhysRev.137.A1034

Guidon, M., Schiffmann, F., Hutter, J., and VandeVondele, J. (2008). Ab Initio Molecular Dynamics Using Hybrid Density Functionals. *J. Chem. Phys.* 128, 214104. doi:10.1063/1.2931945

Hakala, M. H., and Foster, A. S. (2013). Computationally Efficient Implementation of Hybrid Functionals in SIESTA. *Tech. Rep.*

Ihrig, A. C., Wieferink, J., Zhang, I. Y., Ropo, M., Ren, X., Rinke, P., et al. (2015). Accurate Localized Resolution of Identity Approach for Linear-Scaling Hybrid Density Functionals and for Many-Body Perturbation Theory. *New J. Phys.* 17, 093020. doi:10.1088/1367-2630/17/9/093020

Krukau, A. V., Vydrov, O. A., Izmaylov, A. F., and Scuseria, G. E. (2006). Influence of the Exchange Screening Parameter on the Performance of Screened Hybrid Functionals. *J. Chem. Phys.* 125, 224106. doi:10.1063/1.2404663

Larsen, P. K., and Metselaar, R. (1975). Defects and the Electronic Properties of Y3Fe5O12. *J. Solid State. Chem.* 12, 253–258. doi:10.1016/0022-4596(75)90315-1

Lassri, H., Hlil, E. K., Prasad, S., and Krishnan, R. (2011). Magnetic and Electronic Properties of Nanocrystalline Gd3Fe5O12 Garnet. *J. Solid State. Chem.* 184, 3216–3220. doi:10.1016/j.jssc.2011.09.034

Lee, A. J., Ahmed, A. S., Flores, J., Guo, S., Wang, B., Bagués, N., et al. (2020). Probing the Source of the Interfacial Dzyaloshinskii-Moriya Interaction Responsible for the Topological Hall Effect in Metal/Tm3Fe5O12 Systems. *Phys. Rev. Lett.* 124, 107201. doi:10.1103/PhysRevLett.124.107201

Lejaeghere, K., Bihlmayer, G., Björkman, T., Blaha, P., Blügel, S., Blum, V., et al. (2016). Reproducibility in Density Functional Theory Calculations of Solids. *Science* 351, aad3000. doi:10.1126/science.aad3000

Levchenko, S. V., Ren, X., Wieferink, J., Johanni, R., Rinke, P., Blum, V., et al. (2015). Hybrid Functionals for Large Periodic Systems in an All-Electron, Numeric Atom-Centered Basis Framework. *Computer Phys. Commun.* 192, 60–69. doi:10.1016/j.cpc.2015.02.021

Lin, P., Ren, X., and He, L. (2021). Efficient Hybrid Density Functional Calculations for Large Periodic Systems Using Numerical Atomic Orbitals. *J. Chem. Theor. Comput.* 17, 222–239. doi:10.1021/acs.jctc.0c00960

Mandal, S., Kar, R., Kloeffel, T., Meyer, B., and Nair, N. N. (2021). Improving the Scaling and Performance of Multiple Time Stepping Based Molecular Dynamics with Hybrid Density Functionals. *ArXiv*, 07670. doi:10.1002/jcc.26816

Momma, K., and Izumi, F. (2011). VESTA 3 for Three-Dimensional Visualization of Crystal, Volumetric and Morphology Data. *J. Appl. Cryst.* 44, 1272–1276. doi:10.1107/S0021889811038970

Mounet, N., Gibertini, M., Schwaller, P., Campi, D., Merkys, A., Marrazzo, A., et al. (2018). Two-Dimensional Materials from High-Throughput Computational Exfoliation of Experimentally Known Compounds. *Nat. Nanotech* 13, 246–252. doi:10.1038/s41565-017-0035-5

Nakamoto, R., Xu, B., Xu, C., Xu, H., and Bellaiche, L. (2017). Properties of Rare-Earth Iron Garnets from First Principles. *Phys. Rev. B* 95, 024434. doi:10.1103/PhysRevB.95.024434

Perdew, J. P., Ernzerhof, M., and Burke, K. (1996). Rationale for Mixing Exact Exchange with Density Functional Approximations. *J. Chem. Phys.* 105, 9982–9985. doi:10.1063/1.472933

Phan, M. H., Morales, M. B., Chinnasamy, C. N., Latha, B., Harris, V. G., and Srikanth, H. (2009). Magnetocaloric Effect in Bulk and Nanostructured Gd3Fe5O12 Materials. *J. Phys. D: Appl. Phys.* 42, 115007. doi:10.1088/0022-3727/42/11/115007

Rodic, D., Mitric, M., Tellgren, R., Rundlof, H., and Kremenovic, A. (1999). True Magnetic Structure of the Ferrimagnetic Garnet Y3Fe5O12 and Magnetic Moments of Iron Ions. *J. Magnetism Magn. Mater.* 191, 137–145. doi:10.1016/S0304-8853(98)00317-5

Rosen, A. S., Notestein, J. M., and Snurr, R. Q. (2019). Identifying Promising Metal-Organic Frameworks for Heterogeneous Catalysis via High-Throughput Periodic Density Functional Theory. *J. Comput. Chem.* 40, 1305–1318. doi:10.1002/jcc.25787

Sayetat, F. (1986). Huge Magnetostriction in Tb3Fe5O12, Dy3Fe5O12, Ho3Fe5O12, Er3Fe5O12 Garnets. *J. Magnetism Magn. Mater.* 58, 334–346. doi:10.1016/0304-8853(86)90456-7

Schlipf, M., Betzinger, M., Friedrich, C., Ležaić, M., and Blügel, S. (2011). HSE Hybrid Functional within the FLPAW Method and its Application to GdN. *Phys. Rev. B* 84, 125142. doi:10.1103/PhysRevB.84.125142

Serga, A. A., Chumak, A. V., and Hillebrands, B. (2010). YIG Magnonics. *J. Phys. D: Appl. Phys.* 43, 264002. doi:10.1088/0022-3727/43/26/264002

Shavitt, I. (1959). A Calculation of the Rates of the Ortho-Para Conversions and Isotope Exchanges in Hydrogen. *J. Chem. Phys.* 31, 1359–1367. doi:10.1063/1.1730599

Tabuchi, Y., Ishino, S., Noguchi, A., Ishikawa, T., Yamazaki, R., Usami, K., et al. (2015). Coherent Coupling between a Ferromagnetic Magnon and a Superconducting Qubit. *Science* 349, 405–408. doi:10.1126/science.aaa3693

Tm3Fe5O12 crystal structure (2012). *Datasheet from "Pauling File Multinaries Edition – 2012" in Springermaterials* (Japan: Springer-Verlag Berlin Heidelberg & Material Phases Data System (MPDS), Switzerland & National Institute for Materials Science NIMS). Available at: https://materials.springer.com/isp/crystallographic/docs/sd_0312594. Copyright 2016 .

Uchida, K., Xiao, J., Adachi, H., Ohe, J., Takahashi, S., Ieda, J., et al. (2010). Spin Seebeck Insulator. *Nat. Mater* 9, 894–897. doi:10.1038/nmat2856

Vilela, G. L. S., Abrao, J. E., Santos, E., Yao, Y., Mendes, J. B. S., Rodríguez-Suárez, R. L., et al. (2020). Magnon-Mediated Spin Currents in Tm3Fe5O12/Pt with Perpendicular Magnetic Anisotropy. *Appl. Phys. Lett.* 117, 122412. doi:10.1063/5.0023242

Vu, N. M., Meisenheimer, P. B., and Heron, J. T. (2020). Tunable Magnetoelastic Anisotropy in Epitaxial (111) Tm3Fe5O12 Thin Films. *J. Appl. Phys.* 127, 153905. doi:10.1063/1.5142856

Wimmer, E., Krakauer, H., Weinert, M., and Freeman, A. J. (1981). Full-Potential Self-Consistent Linearized-Augmented-Plane-Wave Method for Calculating the Electronic Structure of Molecules and surfaces: O2 molecule. *Phys. Rev. B* 24, 864–875. doi:10.1103/PhysRevB.24.864

Y3Fe5O12 crystal structure (2012). *Datasheet from "Pauling File Multinaries Edition – 2012" in Springermaterials* (Japan: Springer-Verlag Berlin Heidelberg & Material Phases Data System (MPDS), Switzerland & National Institute for Materials Science NIMS). Available at: https://materials.springer.com/isp/crystallographic/docs/sd_0310635. Copyright 2016.

Yan, Q., Yu, J., Suram, S. K., Zhou, L., Shinde, A., Newhouse, P. F., et al. (2017). Solar Fuels Photoanode Materials Discovery by Integrating High-Throughput Theory and Experiment. *Proc. Natl. Acad. Sci. USA* 114, 3040–3043. doi:10.1073/pnas.1619940114

Yu, S., Schmidt, R. D., Garcia-Mendez, R., Herbert, E., Dudney, N. J., Wolfenstine, J. B., et al. (2016). Elastic Properties of the Solid Electrolyte Li7La3Zr2O12 (LLZO). *Chem. Mater.* 28, 197–206. doi:10.1021/acs.chemmater.5b03854

Zhang, C., Donadio, D., Gygi, F., and Galli, G. (2011). First Principles Simulations of the Infrared Spectrum of Liquid Water Using Hybrid Density Functionals. *J. Chem. Theor. Comput.* 7, 1443–1449. doi:10.1021/ct2000952

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers | Frontiers in Materials

ORIGINAL RESEARCH
published: 11 April 2022
doi: 10.3389/fmats.2022.850610

Simulation Study on Internal Short Circuits in a Li-Ion Battery Depending on the Sizes, Quantities, and Locations of Li Dendrites

Suhwan Kim[1†], Jihun Song[1†], Hyobin Lee[1], Seungwon Jung[1], Joonam Park[2], Hongkyung Lee[1,2] and Yong Min Lee[1,2*]*

[1]Department of Energy Science and Engineering, Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu, South Korea, [2]Energy Science and Engineering Research Center, Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu, South Korea

OPEN ACCESS

Edited by:
Wolfgang Wenzel, Karlsruhe Institute of Technology (KIT), Germany

Reviewed by:
Maria Helena Braga, University of Porto, Portugal
Rachel Carter, Naval Research Laboratory, United States

***Correspondence:**
Hongkyung Lee
hongkyung.lee@dgist.ac.kr
Yong Min Lee
yongmin.lee@dgist.ac.kr

[†]*These authors have contributed equally to this work*

Specialty section:
This article was submitted to Computational Materials Science, a section of the journal Frontiers in Materials

Received: *07 January 2022*
Accepted: *14 March 2022*
Published: *11 April 2022*

Citation:
Kim S, Song J, Lee H, Jung S, Park J, Lee H and Lee YM (2022) Simulation Study on Internal Short Circuits in a Li-Ion Battery Depending on the Sizes, Quantities, and Locations of Li Dendrites. Front. Mater. 9:850610. doi: 10.3389/fmats.2022.850610

The internal short circuit caused by the Li dendrite is well known to be a major cause for fire or explosion accidents involving state-of-the-art lithium-ion batteries (LIBs). However, post-mortem analysis cannot identify the most probable cause, which is initially embedded in the cell, because the original structure of the cell totally collapses after the accident. Thus, multiphysics modeling and simulation must be an effective solution to investigate the effect of a specific cause in a variety of conditions. Herein, we reported an electrochemical-thermal model to simulate the internal short circuit depending on Li dendrite's sizes (1, 3, 5, 7, and 9 μm), quantities (1–9), relative locations (0, 25, 50, 100, and 150 μm), and external temperature (–10, 10, 30, and 50°C). Through monitoring the temperature change affected by the joule and reaction heats for each case, we suggested critical conditions that led to unavoidable thermal runaway. Thus, this model can be a steppingstone in understanding the correlation between internal short circuits and Li dendrites.

Keywords: internal short circuit, Li dendrite, Li-ion battery, simulation, safety

INTRODUCTION

A lithium-ion battery (LIB) is an energy storage device widely used from small portable electronics to large electric vehicles (EVs) and energy storage systems (ESSs). However, there are still safety risks in state-of-the-art LIBs because of flammable liquid electrolytes. To manage this risk systematically, the European Council for Automotive R&D (EUCAR) has already set the hazard level for LIB cells, modules, and packs depending on the failure type (Josefowitz et al., 2005). According to this criterion, severe battery failures such as fire, fracture, and explosion are categorized into levels 5, 6, and 7, respectively. Regardless of many efforts to prevent those serious accidents, various reasons such as BMS malfunctions (Ye et al., 2016; Ren et al., 2017; Wang Z. et al., 2021), external shock (Feng et al., 2015; Wang et al., 2017), and high temperature exposure (Kim et al., 2007; Feng et al., 2018a) have been reported ceaselessly. Furthermore, since the energy density of advanced LIBs increases, it is not easy to lower the risk of fire or explosion without understanding fundamental reasons at the cell level (Liu et al., 2018; Wang et al., 2019). Among them, an internal short circuit (ISC) must be the most frequently mentioned and highlighted reason for LIB accidents. Unfortunately, the ISC has not been studied systematically and extensively, owing to difficulties in not only detecting the ISC reliably

but also reproducing a similar phenomenon resulting from the same reason. This is why the ISC remains an unconquered research area to date.

To emphasize the severe impact of the ISC in fully charged LIBs, as an example, with an electrochemical model having an LIB electrode pair (loading level = 1.9 mAh cm^{-2} and N/P ratio = 1.1), we could simulate how much current can flow once the Li dendrite with a radius of 5 μm short-circuits the cell. In this simulation, the current density was calculated simply based on Ohm's law without taking both reversible electrode reactions and irreversible side reactions into consideration (Qi et al., 2021). As a result, it is estimated that the current density can increase to 1.97×10^6 mA cm^{-2}, which is about a million times higher than that in a 50-Ah pouch cell at a 1C-rate (1.715 mA cm^{-2}). Thus, a huge amount of IR heat is generated to rapidly increase the temperature near the dendrite inside the battery. In particular, when the internal temperature increases beyond around 100°C, thermal decomposition of the solid electrolyte interphase (SEI) initiates, and subsequent exothermic reactions occur in series (Feng et al., 2018b). Depending on the difference between heat generation and dissipation rates, the temperature can reach the separator collapse limit or not, but once the temperature increases beyond the limit, the direct contact area of the two electrodes gets enlarged, thereby reaching the ignition temperature of organic solvents and resulting in a thermal runaway. In other words, the combination of flammable organic electrolytes as a fuel, exothermal side reactions as a heat source, and the decomposition of lithium transition metal oxide as an oxygen source accelerate the continuous fire or explosion that is hardly extinguished. Due to this high risk of fires or explosion caused by the ISC, many researchers have attempted to clearly understand the causes of the ISC, the mechanisms of heat generation and accumulation, and the temperature rising behavior (Feng et al., 2018b; Zhang et al., 2021).

Among them, many studies have focused on analyzing the temperature rise depending on ISC types, where they can be classified into electrical abuse, internal defects, mechanical abuse, and thermal abuse (Guo et al., 2015; Zhang et al., 2017a; Wang et al., 2019; Foroozan et al., 2020; Zhang et al., 2021). Also, depending on the contact area, ISCs can be divided into the point contact and surface contact (Zhang et al., 2021). Regardless of many previous studies showing threats caused by various types of ISCs, however, it was almost impossible to reproduce or repeat experimental results under the same conditions. Moreover, owing to significantly improved operando analysis methods (Guo et al., 2015; Foroozan et al., 2020), although the size or location of Li dendrites starts to be unveiled in real time, it is still challenging to fabricate model cell systems for dendrite-based ISC studies.

Nonetheless, Zhang et al., (2017a) realized the internal short circuit in a pouch cell using a millimeter-sized shape memory alloy, which shapes a sharp tip at a certain temperature, with high reproducibility. But their model cells could not mimic the small contact area of actual micrometer-sized Li dendrites. Also, the quantities and locations of ISC points were not controlled in their cell systems. In particular, considering the necessity to track some variables such as the internal temperature, local current density and potential, and lithium ion concentrations for better understanding ISCs, various simulations have been attempted with 2D models with micrometer-sized ISCs (Zavalis et al., 2012; Zhang et al., 2017b) or 3D models with millimeter-sized ISCs (Feng et al., 2016; Wang J. et al., 2021). However, as an actual point of view of Li dendrites, 3D models simulating micrometer-sized ISCs are strongly needed to understand the thermal behavior of actual dendrites and to make battery systems under control.

For this purpose, in this study, we built a 3D cell model having various micrometer-sized ISCs in a domain of 1 mm × 1 mm × 0.177 mm. In particular, to secure the reliability of parameters in this model, we primarily constructed a model of a 50 Ah-level NMC442/graphite pouch cell and compared simulated data with measured ones from electrochemical and thermal experiments. First, we simulated the maximum temperatures caused by a single Li dendrite under various external temperatures. Subsequently, while changing both the number of Li dendrites and their distances, the maximum temperature behaviors were also investigated to figure out their interplay. Finally, through analyzing each case study, we estimated the dominant factors to determine the maximum temperatures over time.

MODEL DEVELOPMENTS

Mining Parameters and Building a Model for the 50 Ah-Level Pouch Cell

In order to obtain parameters for developing a reliable 3D ISC model, a 50 Ah-level pouch cell, which has an electrode chemistry of NMC442/graphite and an electrolyte of 1.15 M $LiPF_6$ in ethylene carbonate and ethylmethyl carbonate (EC/EMC = 3/7, v/v) mixture with a size of 247 mm × 227 mm × 8.02 mm, was used (Top in **Figure 1A**). Based on the design parameters of the pouch cell and some literature values on materials, a P2D electrochemical—3D thermal model of the pouch cell was initially built using COMSOL Multiphysics 5.5 (COMSOL Inc., USA) (Bottom of **Figure 1A**; **Supplementary Figure S1**; **Supplementary Table S2**). After that, the initial model was modified better by fitting model parameters through filling the gap between simulated data and experimental ones obtained from the rate capability evaluation [constant current discharging at 0.5, 1, 2, and 5C-rate at 25°C with a 300-A max current cycler (PNE Solution Co., Ltd., Republic of Korea)] (**Figures 1B, C**). In particular, the surface temperature profiles of the 50 Ah-level cell were also obtained through a thermocouple while changing the C-rate. Also, as shown in **Figure 1D**, depending on the C-rate, the model could estimate the surface temperature of the actual pouch cell because not only the heat generation originated from the electrochemical reaction but also the heat dissipation to the surroundings was well simulated in our model. In this simulation, the internal resistance of the tab was not considered because the tab welding conditions of the pouch cell were not accurately known. As shown in **Figure 1D**, the simulation results for the temperature distribution of the pouch cell are locally inaccurate, but it can be seen that at the center of the pouch

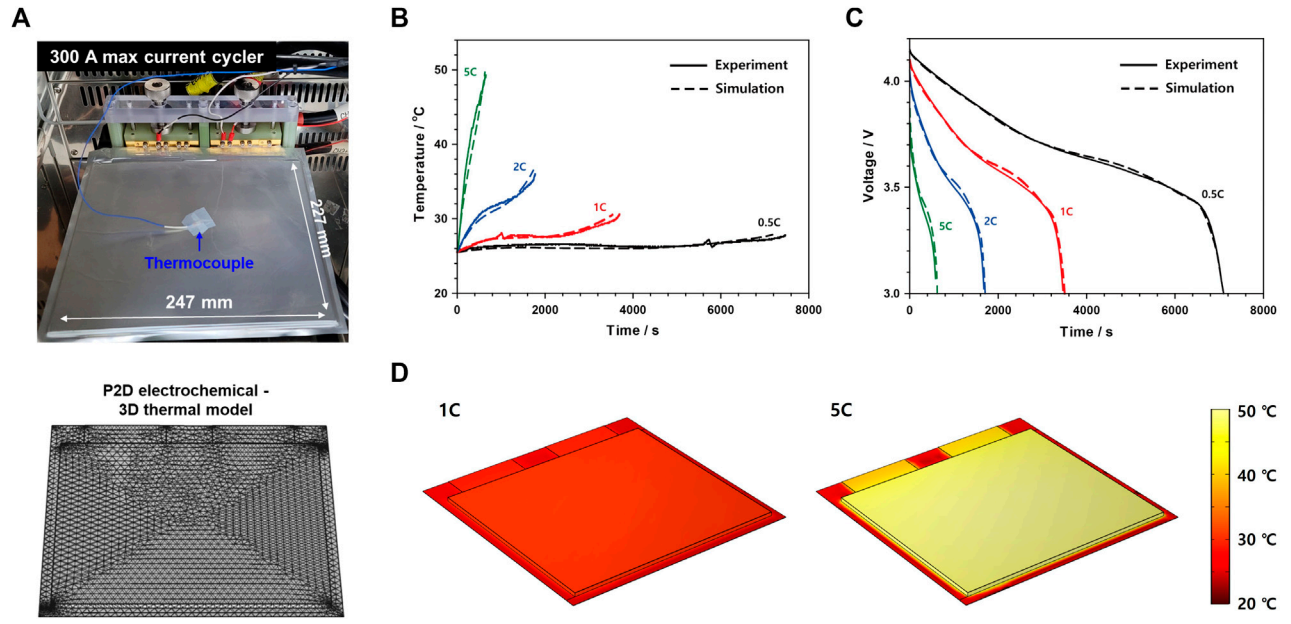


FIGURE 1 | (A) Real image (top) and 3D model structure (bottom) of a 50 Ah-level NMC442/graphite LIB pouch cell. **(B)** Temperature profiles and **(C)** discharge voltage profiles of the pouch cell at various discharge C-rates (0.5, 1, 2, and 5C) along with simulated ones. **(D)** Simulated surface temperatures of the pouch cell in a fully discharged state at different discharge C-rates (1 and 5C).

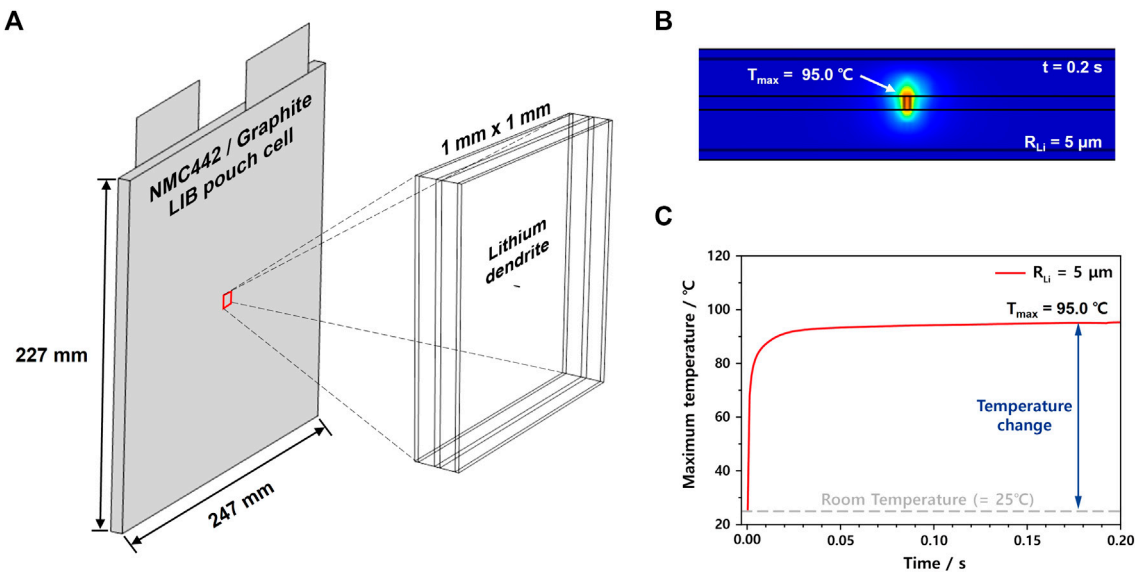


FIGURE 2 | (A) Actual 3D domain size of the ISC model compared to the 50 Ah-level pouch cell. **(B)** Temperature changes in the cross-sectional domain and **(C)** maximum temperature changes with time (from 0 to 0.2 s) of the ISC model having a Li dendrite of a radius of 5 μm.

cell, the temperature increase caused by the electrochemical reaction and the electrode resistance was well calculated. Thus, final model parameters, including the chemical diffusion coefficient and exchange current density, were set in **Supplementary Table S3**, following the aforementioned fitting processes.

3D ISC Model Development

Using the parameters obtained from the cell model (**Supplementary Table S3**), the 3D electrochemical-thermal model for simulating the ISC was developed using COMSOL Multiphysics 5.5. This model is designed to simulate the ISC between two electrodes induced by the formation of

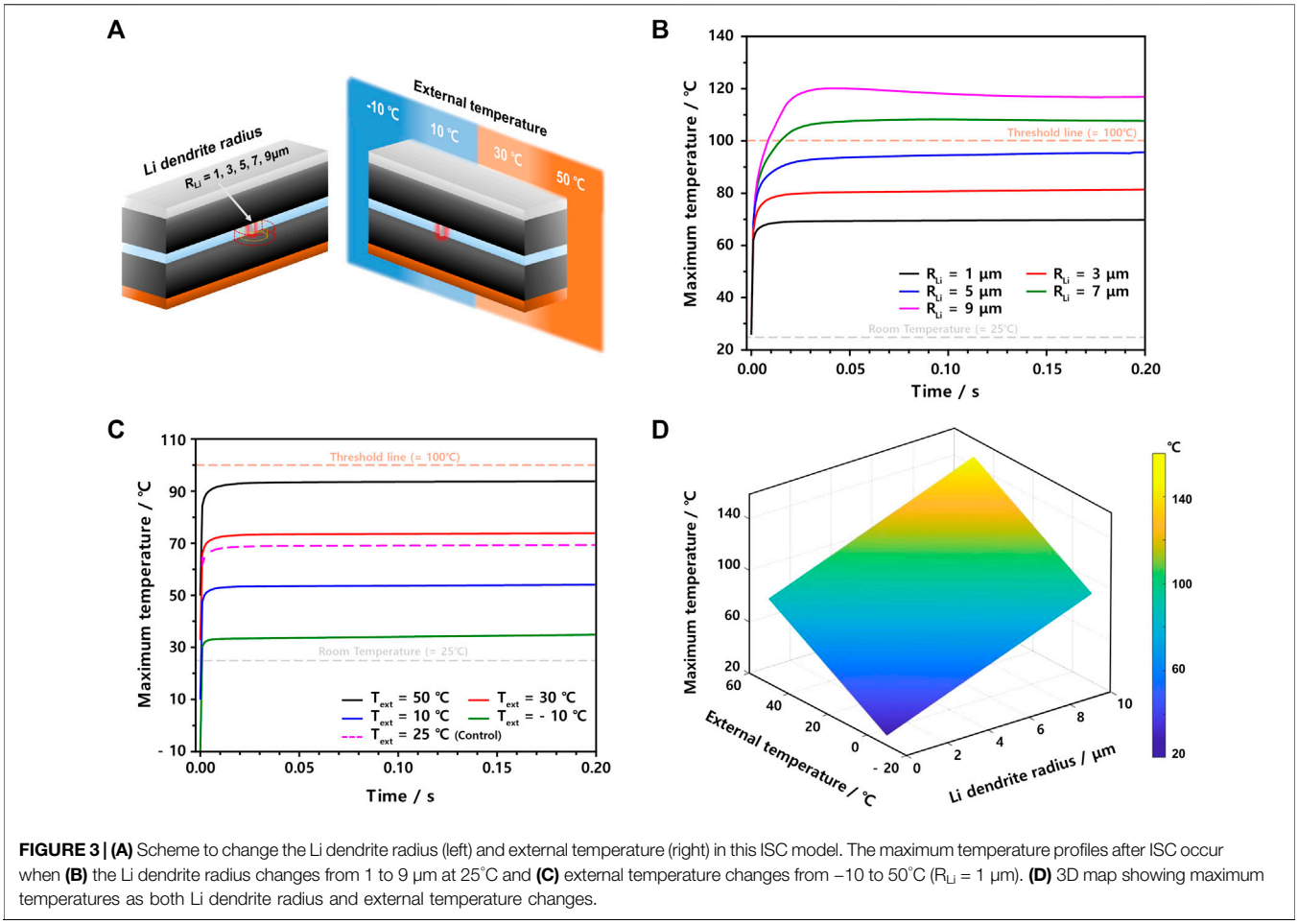


FIGURE 3 | (A) Scheme to change the Li dendrite radius (left) and external temperature (right) in this ISC model. The maximum temperature profiles after ISC occur when **(B)** the Li dendrite radius changes from 1 to 9 μm at 25°C and **(C)** external temperature changes from −10 to 50°C (R_{Li} = 1 μm). **(D)** 3D map showing maximum temperatures as both Li dendrite radius and external temperature changes.

micrometer-sized Li dendrites locally within the pouch cell. Considering both calculation time and sufficiently large domain size compared to Li dendrites, this 3D model was designed in a volume of 1 mm × 1 mm × 0.177 mm with the same parameters and boundary conditions as the entire pouch cell. Furthermore, considering the excellent thermal conductivity and heat capacity of the current collector and the larger value of in-plane thermal conductivity of the electrode layer than through-plane (Zhang et al., 2020) thermal conductivity, we did not have to consider thermal transfer from any layer to the peripheral layer (**Figure 2A**). As described in the middle of the 3D model, the cylindrical Li dendrite with a radius of 5 μm looks like just a thin line in the simulated domain. This is because Li dendrites are known to be grown through the pores in the separator as cylindrical shapes rather than fractals, which can be readily formed in the liquid electrolyte under no physical barriers (Jana et al., 2015; Mu et al., 2019; Jungjohann et al., 2021).For convenience, in this work, the shape of Li dendrites was assumed to be a cylinder having the same height as the separator thickness. The range of the Li dendrite radius was initially controlled from 1 to 5 μm based on previous works showing actual Li dendrite sizes (Frenck et al., 2019; Guo et al., 2020; Liu et al., 2020) and then increased to 9 μm to simulate catastrophic situations. This model can simulate the thermal behavior only caused by the current flow through the Li dendrite, where the corresponding current value is dependent on the amount and state of active materials in both electrodes. In other words, this model does not consider the thermal behavior related to side reactions such as SEI breakdown, separator melting, or active material decomposition. Thus, for instance, we can estimate temperature changes around Li dendrites, as shown in **Figure 2B**, or maximum temperature changes as a function of time, as shown in **Figure 2C**.

RESULTS AND DISCUSSION

To systematically investigate the effects of Li dendrites on temperature changes in internal short-circuited LIB cells, we utilized our 3D ISC model while changing the Li dendrite radius from 1 to 9 μm and the external temperature from −10 to 50°C (**Figure 3A**). In the case of the Li dendrite radius study, the external temperature was set to 25°C. As depicted in **Figure 3B**, the radius of the Li dendrite governs the maximum temperature. In more detail, when the radius is as small as 1 μm, the temperature just increases to 69°C within 0.03 s. On the other hand, when the radius becomes greater than or equal to 7 μm, the maximum temperature passes a threshold point of around 100°C, where SEI begins to be decomposed. In particular, as soon as the ISC

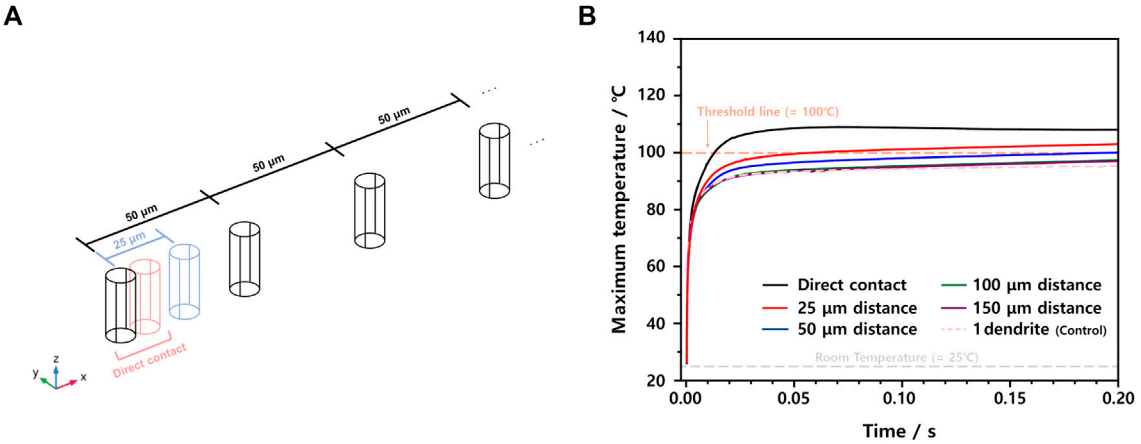


FIGURE 4 | (A) Scheme describing the relative distance of two Li dendrites of 5 μm radius and **(B)** maximum temperature profiles of ISC models depending on the distance of 10 μm (direct contact), 25, 50, 100, and 150 μm.

occurs, the temperature rapidly increases to a saturated value within 0.03 s for all cases because a huge current flows within a very short period as the adiabatic condition. However, when the amount of heat transfer becomes almost the same as that of joule heating, the maximum temperature is maintained at its specific point regardless of the Li dendrite radius. When the effect of the external temperatures on the maximum temperature was also simulated while fixing the Li dendrite radius to 1 μm, as shown in **Figure 3C**, the maximum temperature changed similarly, as in **Figure 3B**. However, although the maximum temperature is strongly dependent on the external temperature, the temperature difference before and after the ISC seems quite similar at around 44°C. Specifically, in this case, when the external temperature is greater than or equal to 50°C, the maximum temperature passes the first threshold point, just like the case of the Li dendrite radius of 7 μm. Similar to this analysis, we need to check both the Li dendrite radius and external temperature simultaneously, as shown in **Figure 3D**, where safe and dangerous regions are colored by the combination of blue and yellow with different ratios. Thus, our ISC model can provide us with a platform to understand the impact of the Li dendrite size and external temperature on the thermal runaway of LIB cells.

As the next step, we need to consider some ISC situations having more than a single Li dendrite by designing their relative distances, as in **Figure 4A**, where the base distance of 50 μm is set by assuming that another Li dendrite of a radius of 5 μm can exist ten times farther away. Also, although the probability is low, two Li dendrites that are contacted or located nearby are also considered. Based on the simulated data summarized in **Figure 4B**, when the distance between two Li dendrites is sufficiently far, e.g., 100 μm, the maximum temperatures in two Li dendrites cases are almost the same as those in a single Li dendrite one. In the region of less than 100 μm distance, of course, the maximum temperature tends to increase as the distance becomes shorter. However, even in the direct contact case, the maximum temperature additionally increased by 12.7°C to be 107.7°C while 95.0°C was reached in the reference single Li dendrite case (**Supplementary Table S4**). Thus, when more than one Li dendrite is formed simultaneously, the risk of thermal runaway of LIBs does not increase proportionally. In other words, when Li dendrites are formed relatively far away enough not to interplay, the probability of encountering catastrophic events becomes lower.

To confirm the effect of the distance between Li dendrites on the maximum temperature, we simulated the thermal behaviors of two ISC models: one has a big Li dendrite of 3 μm radius, and the other has nine small Li dendrites of 1 μm radius, where they are located by 200 μm, while their total areas are identical to be 9π μm^2 (**Figures 5A,B**). As readily estimated from the results in **Figures 3**, **4**, the maximum temperature is mainly governed by the radius of the Li dendrite regardless of Li dendrite numbers only if each dendrite is far away. However, as observed in **Figure 5C**, the maximum temperature of nine small Li dendrites tends to gradually increase with time, thereby reaching a little higher value of around 4°C after 0.20 s. This continuous increase can be ascribed to the average temperature of 29.9°C, which is even higher than the single big Li dendrite case (**Figure 5D**; **Supplementary Table S5**). In other words, when there are many heat sources, the heat dissipation rate reduces with time, and both the maximum and average temperature increase steadily.

To investigate the influence of the Li dendrite (R_{Li} = 5 μm) number on the maximum temperature in more detail, we built four ISC models having one, three, five, and nine Li dendrites with 200 μm distances, as shown in **Figure 6A**, where the temperature distribution of each case is expressed through different colors at 0.2 s after ISC occurs. As already observed in **Figures 4**, **5**, the maximum temperature in the initial state, e.g.,t = 0.03 s, is not dependent on the number of Li dendrites. However, the temperature increase rate is largely affected by the Li dendrite number, as shown in **Figure 6B**. More specifically, while the ISC model with one Li dendrite reaches the maximum temperature of 95.0°C at 0.2 s, the maximum temperatures of five and nine Li dendrite models exceed the critical temperature of 100°C, i.e., 101.6 and 109.0°C, respectively (**Supplementary Table**

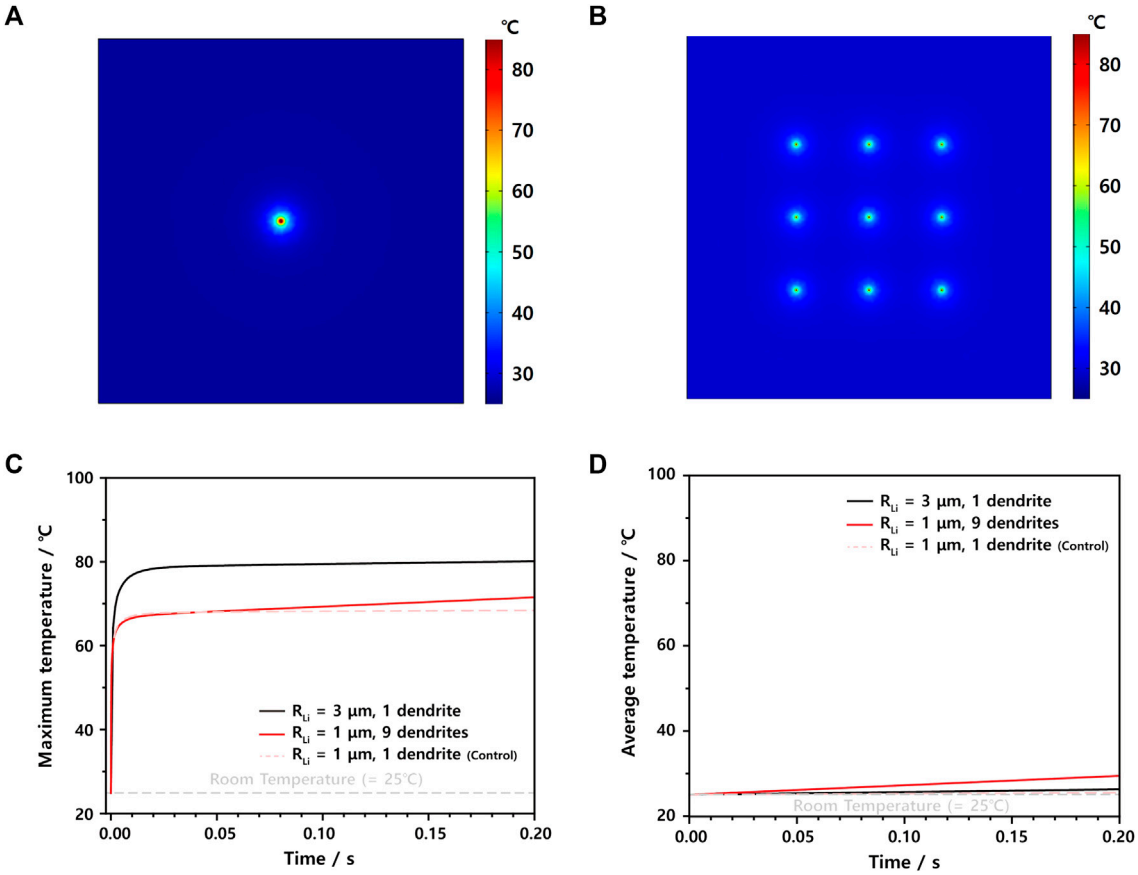


FIGURE 5 | Thermal 2D color maps at 0.2 s of ISC models having **(A)** one large Li dendrite of 3 μm radius and **(B)** nine small Li dendrites of 1 μm radius. **(C)** Maximum temperature profiles and **(D)** average temperature profiles of the aforementioned two ISC models, with the control case having one small Li dendrite of 1 μm.

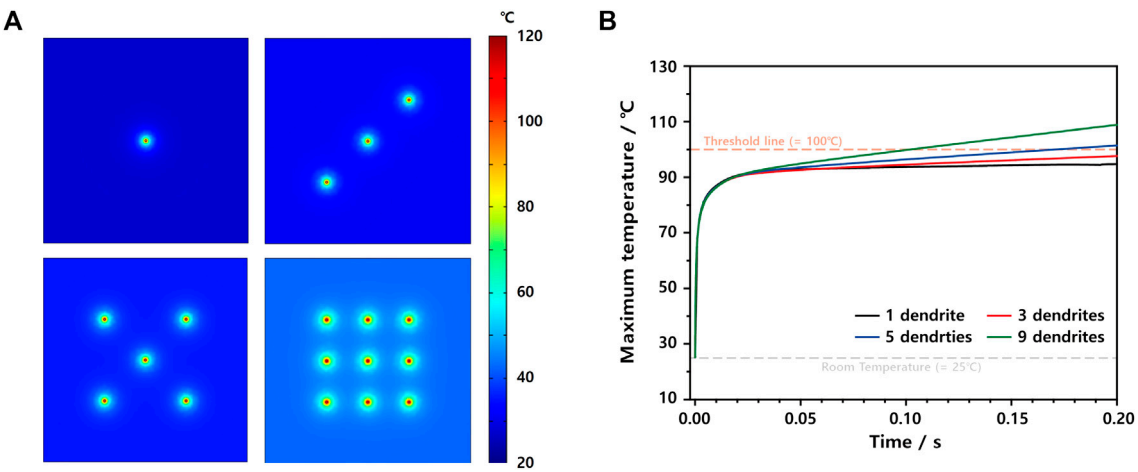


FIGURE 6 | (A) Thermal 2D color map at 0.2 s of ISC models having one Li dendrite, three Li dendrites, five Li dendrites, and nine Li dendrites. **(B)** Maximum temperature profiles of ISC models depending on the number of Li dendrites.

S6). Thus, the maximum temperature in the initial state is mainly dependent on the Li dendrite size, but the number of Li dendrites significantly affects the temperature increase rate. With this ISC model, we can provide a good platform to investigate the influence of Li dendrite sizes and distribution on the thermal behaviors of LIB cells in 3D domains.

CONCLUSION

We developed a 3D electrochemical–thermal model to simulate the ISC caused by the Li dendrites, which considers external temperature, Li dendrite size, number, and relative distances simultaneously. With assumptions such as no side reactions and a simplified Li dendrite shape, the maximum temperature at the start point (0 s) is determined by the external temperature, and within a short period (from 0 to 0.03 s), the maximum temperature increase is governed by the size of the Li dendrite. After a steep temperature increase (after 0.03 s), the maximum temperature is affected by the number of Li dendrites. Therefore, these conditions affecting the thermal behaviors of the LIB cell should be considered simultaneously. Additionally, using our model, the relative distance effect of Li dendrites was investigated. The Li dendrites did not affect each other's maximum temperature rise when they were sufficiently far away. This result shows that even if many Li dendrites are generated simultaneously, the risk of thermal runaway does not increase unconditionally.

Based on this advanced analysis using a 3D ISC model, the limitation of dendrite-based ISC evaluation, which is almost impossible to obtain experimental reproducibility , was addressed, and governing factors for maximum temperature behavior over time were determined. We believe that our model can provide additional insights on the correlation between dendrite-based ISCs and thermal behavior of LIB cells, which can be the basis for further investigating and understanding the thermal runaway.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding authors.

AUTHOR CONTRIBUTIONS

SK: formal analysis, investigation, methodology, software, conceptualization, data curation, and writing—original draft. JS: conceptualization, investigation, methodology, and software. HL: investigation. SJ: investigation. JP: investigation. HL: conceptualization and methodology. YL: supervision, conceptualization, methodology, formal analysis, and writing—review and editing.

FUNDING

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2020R1A4A4079810) and the Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government (20ZB1200, Development of ICT Materials, Components, and Equipment Technologies).

ACKNOWLEDGMENTS

We are also very grateful for the support from the DGIST Supercomputing and Big Data Center.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fmats.2022.850610/full#supplementary-material

REFERENCES

Feng, X., He, X., Ouyang, M., Wang, L., Lu, L., Ren, D., et al. (2018a). A Coupled Electrochemical-Thermal Failure Model for Predicting the Thermal Runaway Behavior of Lithium-Ion Batteries. *J. Electrochem. Soc.* 165, A3748–A3765. doi:10.1149/2.0311816jes

Feng, X., Ouyang, M., Liu, X., Lu, L., Xia, Y., and He, X. (2018b). Thermal Runaway Mechanism of Lithium Ion Battery for Electric Vehicles: A Review. *Energ. Storage Mater.* 10, 246–267. doi:10.1016/j.ensm.2017.05.013

Feng, X., Sun, J., Ouyang, M., Wang, F., He, X., Lu, L., et al. (2015). Characterization of Penetration Induced thermal Runaway Propagation Process within a Large Format Lithium Ion Battery Module. *J. Power Sourc.* 275, 261–273. doi:10.1016/j.jpowsour.2014.11.017

Feng, X., Weng, C., Ouyang, M., and Sun, J. (2016). Online Internal Short Circuit Detection for a Large Format Lithium Ion Battery. *Appl. Energ.* 161, 168–180. doi:10.1016/j.apenergy.2015.10.019

Foroozan, T., Sharifi-Asl, S., and Shahbazian-Yassar, R. (2020). Mechanistic Understanding of Li Dendrites Growth by in- Situ/operando Imaging Techniques. *J. Power Sourc.* 461, 228135. doi:10.1016/j.jpowsour.2020.228135

Frenck, L., Sethi, G. K., Maslyn, J. A., and Balsara, N. P. (2019). Factors that Control the Formation of Dendrites and Other Morphologies on Lithium Metal Anodes. *Front. Energ. Res.* 7. doi:10.3389/fenrg.2019.00115

Guo, S., Wang, L., Jin, Y., Piao, N., Chen, Z., Tian, G., et al. (2020). A Polymeric Composite Protective Layer for Stable Li Metal Anodes. *Nano Convergence* 7, 21. doi:10.1186/s40580-020-00231-w

Guo, Z., Zhu, J., Feng, J., and Du, S. (2015). Direct *In Situ* Observation and Explanation of Lithium Dendrite of Commercial Graphite Electrodes. *RSC Adv.* 5, 69514–69521. doi:10.1039/c5ra13289d

Jana, A., Ely, D. R., and García, R. E. (2015). Dendrite-separator Interactions in Lithium-Based Batteries. *J. Power Sourc.* 275, 912–921. doi:10.1016/j.jpowsour.2014.11.056

Josefowitz, W., Kranz, H., Macerata, D., Soczka-Guth, T., Mettlach, H., Porcellato, D., et al. (2005). "Assessment and Testing of Advanced Energy Storage Systems for Propulsion–European Testing Report," in Proceedings of the 21 worldwide battery, hybrid and fuel cell electric vehicle symposium & exhibition. Monte Carlo, Monaco: EVS21 Organization, 2–6.

Jungjohann, K. L., Gannon, R. N., Goriparti, S., Randolph, S. J., Merrill, L. C., Johnson, D. C., et al. (2021). Cryogenic Laser Ablation Reveals Short-Circuit Mechanism in Lithium Metal Batteries. *ACS Energ. Lett.* 6, 2138–2144. doi:10.1021/acsenergylett.1c00509

Kim, G.-H., Pesaran, A., and Spotnitz, R. (2007). A Three-Dimensional thermal Abuse Model for Lithium-Ion Cells. *J. Power Sourc.* 170, 476–489. doi:10.1016/j.jpowsour.2007.04.018

Liu, K., Liu, Y., Lin, D., Pei, A., and Cui, Y. (2018). Materials for Lithium-Ion Battery Safety. *Sci. Adv.* 4, eaas9820. doi:10.1126/sciadv.aas9820

Liu, L., Feng, X., Zhang, M., Lu, L., Han, X., He, X., et al. (2020). Comparative Study on Substitute Triggering Approaches for Internal Short Circuit in Lithium-Ion Batteries. *Appl. Energ.* 259, 114143. doi:10.1016/j.apenergy.2019.114143

Mu, W., Liu, X., Wen, Z., and Liu, L. (2019). Numerical Simulation of the Factors Affecting the Growth of Lithium Dendrites. *J. Energ. Storage* 26, 100921. doi:10.1016/j.est.2019.100921

Qi, X., Liu, B., Pang, J., Yun, F., Wang, R., Cui, Y., et al. (2021). Unveiling Micro Internal Short Circuit Mechanism in a 60 Ah High-Energy-Density Li-Ion Pouch Cell. *Nano Energy* 84, 105908. doi:10.1016/j.nanoen.2021.105908

Ren, D., Feng, X., Lu, L., Ouyang, M., Zheng, S., Li, J., et al. (2017). An Electrochemical-thermal Coupled overcharge-to-thermal-runaway Model for Lithium Ion Battery. *J. Power Sourc.* 364, 328–340. doi:10.1016/j.jpowsour.2017.08.035

Wang, H., Lara-Curzio, E., Rule, E. T., and Winchester, C. S. (2017). Mechanical Abuse Simulation and thermal Runaway Risks of Large-Format Li-Ion Batteries. *J. Power Sourc.* 342, 913–920. doi:10.1016/j.jpowsour.2016.12.111

Wang, J., Mei, W., Cui, Z., Dong, D., Shen, W., Hong, J., et al. (2021a). Investigation of the thermal Performance in Lithium-Ion Cells during Polyformaldehyde Nail Penetration. *J. Therm. Anal. Calorim.* 145, 3255–3268. doi:10.1007/s10973-020-09853-y

Wang, Q., Mao, B., Stoliarov, S. I., and Sun, J. (2019). A Review of Lithium Ion Battery Failure Mechanisms and Fire Prevention Strategies. *Prog. Energ. Combustion Sci.* 73, 95–131. doi:10.1016/j.pecs.2019.03.002

Wang, Z., Yuan, J., Zhu, X., Wang, H., Huang, L., Wang, Y., et al. (2021b). Overcharge-to-thermal-runaway Behavior and Safety Assessment of Commercial Lithium-Ion Cells with Different Cathode Materials: A Comparison Study. *J. Energ. Chem.* 55, 484–498. doi:10.1016/j.jechem.2020.07.028

Ye, J., Chen, H., Wang, Q., Huang, P., Sun, J., and Lo, S. (2016). Thermal Behavior and Failure Mechanism of Lithium Ion Cells during Overcharge under Adiabatic Conditions. *Appl. Energ.* 182, 464–474. doi:10.1016/j.apenergy.2016.08.124

Zavalis, T. G., Behm, M., and Lindbergh, G. (2012). Investigation of Short-Circuit Scenarios in a Lithium-Ion Battery Cell. *J. Electrochem. Soc.* 159, A848–A859. doi:10.1149/2.096206jes

Zhang, G., Wei, X., Tang, X., Zhu, J., Chen, S., and Dai, H. (2021). Internal Short Circuit Mechanisms, Experimental Approaches and Detection Methods of Lithium-Ion Batteries for Electric Vehicles: A Review. *Renew. Sustain. Energ. Rev.* 141, 110790. doi:10.1016/j.rser.2021.110790

Zhang, L., Zhao, P., Xu, M., and Wang, X. (2020). Computational Identification of the Safety Regime of Li-Ion Battery Thermal Runaway. *Appl. Energy* 261, 114440.

Zhang, M., Du, J., Liu, L., Stefanopoulou, A., Siegel, J., Lu, L., et al. (2017a). Internal Short Circuit Trigger Method for Lithium-Ion Battery Based on Shape Memory Alloy. *J. Electrochem. Soc.* 164, A3038–A3044. doi:10.1149/2.0731713jes

Zhang, M., Liu, L., Stefanopoulou, A., Siegel, J., Lu, L., He, X., et al. (2017b). Fusing Phenomenon of Lithium-Ion Battery Internal Short Circuit. *J. Electrochem. Soc.* 164, A2738–A2745. doi:10.1149/2.1721712jes

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers | Frontiers in Materials

ORIGINAL RESEARCH
published: 27 April 2022
doi: 10.3389/fmats.2022.868248

Optimal Data-Generation Strategy for Machine Learning Yield Functions in Anisotropic Plasticity

Ronak Shoghi[†] *and Alexander Hartmaier*[*†]

ICAMS, Ruhr-Universität Bochum, Bochum, Germany

Trained machine learning (ML) algorithms can serve as numerically efficient surrogate models of sophisticated but numerically expensive constitutive models of material behavior. In the field of plasticity, ML yield functions have been proposed that serve as the basis of a constitutive model for plastic material behavior. If the training data for such ML flow rules is gained by micromechanical models, the training procedure can be considered as a homogenization method that captures essential information of microstructure-property relationships of a given material. However, generating training data with micromechanical methods, as for example, the crystal plasticity finite element method, is a numerically challenging task. Hence, in this work, it is investigated how an optimal data-generation strategy for the training of a ML model can be established that produces reliable and accurate ML yield functions with the least possible effort. It is shown that even for materials with a significant plastic anisotropy, as polycrystals with a pronounced Goss texture, 300 data points representing the yield locus of the material in stress space, are sufficient to train the ML yield function successfully. Furthermore, it is demonstrated how data-oriented flow rules can be used in standard finite element analysis.

Keywords: plasticity, data-driven methods, machine learning, data generation, uniform distribution, hypersphere, homogenization

OPEN ACCESS

Edited by:
Norbert Huber,
Helmholtz-Zentrum Hereon, Germany

Reviewed by:
Francisco Chinesta,
Arts et Metiers Institute of Technology, France
Christian Johannes Cyron,
Hamburg University of Technology, Germany

***Correspondence:**
Alexander Hartmaier
alexander.hartmaier@rub.de

[†]*These authors have contributed equally to this work*

Specialty section:
This article was submitted to Computational Materials Science, a section of the journal Frontiers in Materials

Received: *02 February 2022*
Accepted: *22 March 2022*
Published: *27 April 2022*

Citation:
Shoghi R and Hartmaier A (2022) Optimal Data-Generation Strategy for Machine Learning Yield Functions in Anisotropic Plasticity. Front. Mater. 9:868248. doi: 10.3389/fmats.2022.868248

INTRODUCTION

The finite element method is one of the most popular methods used in solid mechanics to solve the nonlinear partial differential equations describing mechanical equilibrium of a solid under arbitrary boundary conditions. The solution of a solid mechanics problem must satisfy equations of equilibrium, compatibility of strains and displacements, and obey the constitutive laws for the materials represented in the model (Chen and Saleeb, 1994; Pian and Wu, 2005). These constitutive equations describe the relationships between stresses and strains, i.e., they quantify the material response under given distortions. In the simplest case of elastic materials, this relationship assumes a linear form in which stress and strain are proportional to each other, as described by Hooke's law. For nonlinear and irreversible material behavior, as it must be considered for plastic materials, elastic and plastic strains need to be treated by separate constitutive models (Chen and Saleeb, 1994; Bonet and Wood, 1997). While for elastic strains, Hooke's law is still valid, plastic strains need to be calculated in a way that is consistent with the yield strength and the flow stress observed in tensile tests of the given material. To accomplish this, a yield function is formulated that indicates whether a given stress state results in a linear-elastic material response or whether plastic deformation needs to be considered. In the linear elastic regime, the yield function takes negative values, and it reaches the value zero for those stress tensors for which plastic yielding starts. Hence, the zeros of the yield function can be represented

by a hypersurface in stress space, which embodies the convex hull of the stress tensors leading to a linear-elastic material response. In the Voigt notation, the six-dimensional (6D) stress space itself is spanned by the six independent components of the stress tensor, including normal and shear components. In the case of ideal plasticity, this hypersurface in stress space, the so-called yield-locus, remains unchanged by plastic deformation (Chen and Saleeb, 1994). In contrast, in the case of strain hardening, it can change its volume (isotropic hardening), location (kinematic hardening), or shape (distortive hardening) (Helling and Miller, 1987; Kurtyka and Życzkowski, 1996).

This concept of the yield locus as geometrical representation of the yield criterion can also be applied in cases of anisotropic plastic deformation, as it occurs, for example, in polycrystalline metals with specific crystallographic textures. This plastic anisotropy of polycrystals can be described with the crystal plasticity finite element method (CPFEM) (Roters et al., 2011), for an overview, in which plastic slip on the discrete crystallographic slip planes of each grain is considered. Such methods also allow us to study the evolution of textures under large plastic strains (Peranio et al., 2010), where certain crystallographic planes tend to rotate concerning the direction of the highest principal strain and, thus, cause a dynamic change of the crystallographic texture and the resulting plastic anisotropy, which is the origin of distortive hardening in these materials.

While CPFEM methods represent a fundamental way of describing plastic anisotropy of materials, they cause a high numerical effort, limiting their application to relatively small volumes. In the work of Vajragupta et al. (2017), it has been shown that the results of CPFEM calculations can be homogenized by fitting the parameters of an anisotropic yield criterion to data obtained from CPFEM calculation of specific textures. Various formulations of such anisotropic yield criteria have been published in the literature (Karafillis and Boyce, 1993; Cazacu and Barlat, 2001; Banabic et al., 2004). In this work, we refer to the formulation of Hill (Hill, 1948) with six parameters describing the state of anisotropy, which has been suggested as a generalization of the isotropic yield criterion after Huber-Von Mises (Huber, 1904; v Mises, 1913), based on the second invariant of the stress deviator (J2). The corresponding equivalent stress can be compared to the scalar yield stress of the material determined in a uniaxial test. Barlat et al. (2005) suggested a yield function that contains 18 parameters (Yld 2004-18p) based on a linear transformation of the stress deviator.

In more recent approaches, data-driven computational methods have been suggested in the literature to describe anisotropic material behavior. In the data-driven method suggested by Kirchdoerfer and Ortiz (2016), instead of constitutive models for finite element analysis, experimental material data can be used directly to satisfy the required constraints and conservation laws for mechanical equilibrium. In a similar approach, Eggersmann et al. (2019), Eggersmann et al. (2021) have shown that a model-free data-driven formulation of mechanical problems can be achieved. In the work of Chinesta et al. (2017) the data-driven strategy was extended for nonlinear material behavior, which includes not only the plastic strain rate and the rate of accumulated plastic deformation but also the kinematic hardening rate. Another popular approach to develop a data-driven material model is using machine learning algorithms that are capable of handling large data sets. At the same time, they provide the possibility of describing arbitrary mathematical functions, thus relieving the restrictions for closed-form mathematical descriptions of anisotropic yield criteria. Liu et al. (2018) employ a clustering technique to solve the equilibrium equation on clusters of material points with similar mechanical responses. In another approach suggested by Ibañez et al. (2018), manifold learning methods were used to define the constitutive manifold of a given material, allowing the extraction of relevant information directly from large experimental data sets. In Linka et al. (2021), a machine learning-based hyperelasticity model is suggested using a feed-forward neural network trained using experimental data. (Linka et al. (2021) introduced constitutive artificial neural networks to describe hyperelastic material behavior.

Following the method developed in (Hartmaier, 2020), the present work uses Support Vector Classification (SVC) of the elastic and plastic domains in the stress space as a data-oriented yield function. The SVC algorithm is trained by using input data in the form of critical stresses that mark the onset of plastic yielding, thus representing the yield locus in a data-based manner. In this way, a machine learning (ML) yield function is obtained to determine whether a given stress state lies inside or outside the material's elastic domain. Getting a proper data-based representation of the yield locus as a hypersurface in the 6D stress space can be challenging. This holds in particular if the training data is generated by numerically expensive methods as CPFEM. Hence, in this work, we will use simpler methods for the generation of various data sets that are the basis for the development of an optimal strategy to distribute the training data points over the entire yield locus with as small as possible data sets. Based on these training data, an accurate ML yield locus, i.e., the hypersurface in the stress space on which plastic deformation occurs, can be reconstructed from the SVC in the form of a convolutional sum over a kernel function, from which the gradient on this yield locus can be conveniently calculated. Therefore, the standard formulations of continuum plasticity, as the return mapping algorithm, can be applied in the usual way to finite element analysis (FEA). Thus, it is demonstrated that the new ML yield function can replace conventional FEA flow rules.

METHODS

Machine Learning Flow Rule

The elastic-plastic deformation of a material can be described using stress and strain tensors denoted with $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$, respectively. The stress tensor describes the force acting on the surface of a material, and the strain tensor describes the deformation of the material.

In the Voigt notation, the symmetric tensor is defined by its six independent components as

$$\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6) \tag{1}$$

where $\sigma_1 = \sigma_{11}, \sigma_2 = \sigma_{22}, \sigma_3 = \sigma_{33}, \sigma_4 = \sigma_{23}, \sigma_5 = \sigma_{13}, \sigma_6 = \sigma_{12}$. The yield function of a material is defined as

$$f(\boldsymbol{\sigma}) = \sigma_{eq}(\boldsymbol{\sigma}) - \sigma_y \tag{2}$$

and plastic deformation sets in at $f = 0$, i.e., when the equivalent stress σ_{eq} equals the yield strength σ_y of the material. The equivalent stress used here follows the definition of Hill for anisotropic materials in the form

$$\sigma_{eq}(\boldsymbol{\sigma}) = \frac{1}{\sqrt{2}}\big[H_1(\sigma_1-\sigma_2)^2 + H_2(\sigma_2-\sigma_3)^2 + H_3(\sigma_3-\sigma_1)^2 + 6H_4\sigma_4^2 + 6H_5\sigma_5^2 + 6H_6\sigma_6^2\big]^{1/2} \tag{3}$$

In this yield function, the anisotropy of the material's flow behavior is described in a Hill-like approach, where the parameters $H_1, \ldots, H_6$ control the anisotropic flow behavior of the material. Note that these parameters correspond to the parameters used in the conventional Hill yield criterion (Hill, 1948).

$$E(\sigma_2-\sigma_3)^2 + G(\sigma_3-\sigma_1)^2 + H(\sigma_1-\sigma_2)^2 + 2L\sigma_4^2 + 2M\sigma_5^2 + 2N\sigma_6^2 = 1 \tag{4}$$

when $H = H_1\sigma_y^2, E = H_2\sigma_y^2, G = H_3\sigma_y^2, L = 3H_4\sigma_y^2, M = 3H_5\sigma_y^2$, $N = 3H_6\sigma_y^2$. The special case $H_1 = H_2 = \ldots = H_6 = 1$ describes isotropic plastic behavior, where the equivalent stress defined here is identical to the von Mises (J2) equivalent stress.

The gradient of the yield function with respect to the stress components is needed for calculating the plastic strain increments in the return mapping algorithm of continuum plasticity and can be evaluated analytically as

$$\frac{\partial f}{\partial \sigma_1} = \frac{\partial \sigma_{eq}}{\partial \sigma_1} = \frac{(H_1+H_3)\sigma_1 - H_1\sigma_2 - H_3\sigma_3}{2\sigma_{eq}} \tag{5}$$

$$\frac{\partial f}{\partial \sigma_2} = \frac{\partial \sigma_{eq}}{\partial \sigma_2} = \frac{(H_2+H_1)\sigma_2 - H_1\sigma_1 - H_2\sigma_3}{2\sigma_{eq}} \tag{6}$$

$$\frac{\partial f}{\partial \sigma_3} = \frac{\partial \sigma_{eq}}{\partial \sigma_3} = \frac{(H_3+H_2)\sigma_3 - H_3\sigma_1 - H_2\sigma_2}{2\sigma_{eq}} \tag{7}$$

$$\frac{\partial f}{\partial \sigma_4} = \frac{\partial \sigma_{eq}}{\partial \sigma_4} = 3H_4\frac{\sigma_4}{\sigma_{eq}} \tag{8}$$

$$\frac{\partial f}{\partial \sigma_5} = \frac{\partial \sigma_{eq}}{\partial \sigma_5} = 3H_5\frac{\sigma_5}{\sigma_{eq}} \tag{9}$$

$$\frac{\partial f}{\partial \sigma_6} = \frac{\partial \sigma_{eq}}{\partial \sigma_6} = 3H_6\frac{\sigma_6}{\sigma_{eq}} \tag{10}$$

In the case of isotropic plasticity, i.e., $H_1 = H_2 = \ldots = H_6 = 1$ the gradient takes the simple form

$$\frac{\partial f}{\partial \boldsymbol{\sigma}} = \frac{3}{2}\frac{\boldsymbol{\sigma}^{dev}}{\sigma_{eq}} \tag{11}$$

Where $\boldsymbol{\sigma}^{dev} = (\sigma_1 - p, \sigma_2 - p, \sigma_3 - p, \sigma_4, \sigma_5, \sigma_6)$ is the deviatoric stress tensor and $p = (\sigma_1+\sigma_2+\sigma_3)/3$ is the hydrostatic stress.

Data-Oriented Yield Function

In the data-oriented approach followed in this work, the yield function $f_{ML}(\boldsymbol{\sigma})$ is described in the form of a machine learning (ML) algorithm, which uses Support Vector Classification (SVC) for categorizing any given stress tensor $\boldsymbol{\sigma}$ into the categories "elastic" ($f_{ML}(\boldsymbol{\sigma}) = -1$) and "plastic" ($f_{ML}(\boldsymbol{\sigma}) = +1$) (Hartmaier, 2020). The purpose is to find the optimal hypersurface which separates these two regions from each other. This hypersurface is the yield locus defined by the zeros of the yield function. Based on the SVM algorithm, the optimal hypersurface is the one in which the margin between training data points of the respective classes "elastic" and "plastic" is maximum. This margin is defined as the distance between the separator and the closest data points to it from both classes. These data points in the vicinity of the separator are called support vectors. Given a training set of N data points $\{y_i, \boldsymbol{\sigma}_i\}_{i=1}^N$ where $\boldsymbol{\sigma_i} \varepsilon \mathbb{R}^6$ is the ith input stress and $y_i = f_{ML}(\boldsymbol{\sigma}_i)$ is the ith output term required for the supervised training algorithm. Note that this data-oriented yield function f_{ML} can be considered as the signum function of the physical yield function defined in **Eq. 2**, thus $f_{ML}(\boldsymbol{\sigma}) = \mathrm{sgn}(f(\boldsymbol{\sigma}))$. This has the advantage that the input data can be given in terms of critical stresses marking the onset of plastic yielding. Each of these stresses can then simply be scaled proportionally into the elastic or plastic region of the stress space during the training procedure. Furthermore, once the categorial yield function $f_{ML}(\sigma)$ is known, the value of the true yield function can be reconstructed by calculating the distance of the given stress tensor to the yield locus in stress space.

The support vector method aims to construct a classifier in form

$$f_{ML}(\boldsymbol{\sigma}) = \left[\sum_{k=1}^{N_{SV}} \alpha_k y_k \psi(\boldsymbol{\sigma}, \boldsymbol{\sigma}_k) + b\right] \tag{12}$$

where N_{SV} is the number of the support vectors $\boldsymbol{\sigma}_k$ and $\alpha_k y_k$ are the so-called dual coefficients and b is an offset. These parameters are defined during the training procedure of the SVC. For nonlinear problems $\psi(\boldsymbol{\sigma}, \boldsymbol{\sigma}_k)$ should be chosen as the radial basis function (RBF) kernel, which is defined as (Suykens and Vandewalle, 1999).

$$\psi(\boldsymbol{\sigma}, \boldsymbol{\sigma}_k) = \exp\big[-\gamma\|\boldsymbol{\sigma} - \boldsymbol{\sigma}_k\|^2\big] \tag{13}$$

Since the ML yield function is defined as convolution sum over support vectors, the gradient to the SVC decision function can be calculated as

$$\frac{\partial f_{ML}(\boldsymbol{\sigma})}{\partial \boldsymbol{\sigma}} = \left[\sum_{k=1}^{N_{SV}} -2\gamma\alpha_k y_k \exp\big(-\gamma\|\boldsymbol{\sigma} - \boldsymbol{\sigma}_k\|^2\big)(\boldsymbol{\sigma} - \boldsymbol{\sigma}_k)\right] \tag{14}$$

When using the RBF kernel for training, γ and C are the two most important hyperparameters to be defined. γ indicates the width of the kernel function and, thus, the extent to which a single training point has an impact. A smaller value of γ leads to a short-ranged influence. Any misclassified data point is penalized by parameter C. When C is small, the penalty for misclassified points is also small such that a wide-margin decision function on the boundary is chosen at the expense of a larger number of misclassifications. If C is large, the training algorithm limits the number of misclassified cases by using a high penalty and a smaller decision boundary. Thus, a higher value of C produces a "softer" boundary of the classifier, i.e., a function with more

undulations and an irregular gradient, whereas a smaller value of C results in a "stiffer" classifier function, i.e., in a rather straight boundary, which might, however, be less accurate.

A high training quality relies on providing sufficient training points on the yield locus, as the SVM algorithm creates support vectors only in the areas covered by training points. While in previous work (Hartmaier, 2020) only the three-dimensional space of principal stresses was considered, where the coverage of the yield locus with data points is a rather trivial task, it is quite a challenge to create data points efficiently in the full 6D stress space. An optimal strategy to create as few as possible data points representing the yield locus in the best possible way is necessary, particularly when data points are created with numerically expensive methods such as CPFEM.

Data-Generation Model

Creating a set of yield stresses in the full stress space that serves as the ground truth is required to train the ML yield function. In a first step, unit stresses need to be generated that define the load cases, i.e., the directions in which the load is applied. Then, each unit stress is proportionally increased until the value of the yield function $f(\boldsymbol{\sigma})$ defined in **Eq. 2** is zero, indicating the start of plastic yielding for the stress tensor $\boldsymbol{\sigma}$.

Finally, the full set of stress tensors at the onset of plastic yielding represents the ground truth for the training of the ML yield function (Hartmaier, 2020). The task of finding the zeros of the yield function, i.e., the critical stress tensor at the onset of plastic yielding, for each load case needs to be done by a fundamental method, such as CPFEM or mechanical testing that captures the yielding behavior of the material under investigation. Hence, the effort for funding this ground truth is a considerable task that represents the major effort for training an ML flow rule. Thus, finding an optimal strategy for creating the unit stresses is the primary goal of this work. The starting point is creating different distributions of unit stresses on the surface of a unit sphere in a full 6D stress space. After this, the quality of the ML yield criterion resulting from the training with these data points is verified to identify the optimal strategy for data generation.

There are different methods for distributing points on the surface of a unit sphere, but not all of them can be extended to dimensions higher than 3. Different algorithms have been proposed to solve the problem of distributing points uniformly over the surface of a unit sphere in higher dimensional spaces because it is a common but highly non-trivial task with many applications in science and engineering. In many proposed solutions, the initial idea is to uniformly distribute points over a rectangular area that is then mapped to a sphere using the cylindrical projection (Hannay and Nye, 2004). In the next part, four different methods are introduced and the dimensions where they can be used are investigated.

One of the most common and simple methods for distributing points on a 3-dimensional sphere is based on the Fibonacci lattice. Based on the work of Marques et al. (2013), Fibonacci lattice points are constructed through a mapping process from the unit square to the unit sphere. A Fibonacci lattice in the unit square is a set Q_m of F_m points (x, y) defined as

$$\begin{aligned} x_j &= \left\{ j \frac{F_{m-1}}{F_m} \right\} \\ y_j &= \frac{j}{F_m} \\ 0 &\le j < F_m \end{aligned} \tag{15}$$

where F_m and F_{m-1} are the two last members of the Fibonacci sequence for a given $m > 1$, as defined by the recurrence equation $F_m = F_{m-1} + F_{m-2}$, with the starting numbers $F_0 = 0$ and $F_1 = 1$. In this equation, $\{x\} = x - [x]$ is the fractional part for non-negative real numbers x, where $[x]$ is the integer part of x (Marques et al., 2013). Mapping this lattice to the unit sphere is done based on the Lambert mapping or equal area north pole projection which is visualized in **Figure 1**. In this projection, the points

$$\begin{aligned} \theta_j &= \arcsin\left(2j/2F_m + 1\right) \\ \phi_j &= 2\pi \left\{ j \frac{F_{m-1}}{F_m} \right\} \end{aligned} \tag{16}$$

given by the polar angle θ_j and the azimuthal angle ϕ_j are first moved to $(\frac{1}{2}\theta, \phi)$ on the northern hemisphere and then projected perpendicularly on to the equatorial plane. It can be shown that equal areas on the sphere's surface transform into equal areas on the projection. The south pole becomes the whole circumference (Hannay and Nye, 2004). The resulting equatorial plane is shown in **Figure 1** with 378 points using the Fibonacci number $F_{15} = 377$.

As m increases, the Fibonacci ratio F_m/F_{m+1} approaches the golden ratio $\varphi = (1 + \sqrt{5})/2$ and, as a result, the asymptotic azimuthal angle.

$$\lim_{m \to \infty} \phi_j = 2j\pi\varphi^{-1} \tag{17}$$

is obtained due to the periodicity of the spherical coordinates (Marques et al., 2013). This relation can be exploited to release the requirement that the number of points has to be a Fibonacci number by defining the coordinates of a spherical point set with an arbitrary, but sufficiently large number N of points as:

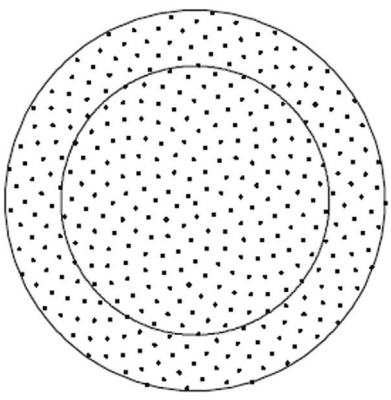


FIGURE 1 | Equal area north pole projection of points on the surface of a sphere (Hannay and Nye, 2004).

$$\begin{aligned}\theta_j &= \arcsin\left(2j/2N+1\right)\\ \phi_j &= 2j\pi\varphi^{-1}\\ 0 &\le j < N\end{aligned} \tag{18}$$

Based on these angles, the cartesian coordinates of the set of equally distributed points on the 3D unit sphere can be calculated as

$$\left(x_j, y_j, z_j\right) = \left(\cos\theta_j \sin\phi_j,\ \sin\theta_j \sin\phi_j,\ \cos\phi_j\right) \tag{19}$$

The even arrangement of points divides the sphere into equal-area spherical rings due to the area-preserving property of the Lambert map. In this arrangement, each ring contains a single lattice node (Swinbank and James Purser, 2006; González, 2010; Marques et al., 2013). A major disadvantage of this formulation for finding equally distributed points on a sphere based on Fibonacci lattice points is that it cannot be extended directly to dimensions higher than 3.

To overcome this limitation, Marsaglia (1972) has suggested a method that allows the selection of uniformly distributed points on the surface of a 4D-sphere. This method is conducted by picking two points as (x_1, x_2) and (x_3, x_4) and rejecting any points if $(x_1^2 + x_2^2) \ge 1$ and $(x_3^2 + x_4^2) \ge 1$. In this case the points

$$\begin{aligned}x &= x_1\\ y &= x_2\\ z &= x_3\sqrt{\frac{1 - x_1^2 - x_2^2}{x_3^2 + x_4^2}}\\ w &= x_4\sqrt{\frac{1 - x_1^2 - x_2^2}{x_3^2 + x_4^2}}\end{aligned} \tag{20}$$

have a uniform distribution on the surface of a 4D hypersphere. However, this method does not generalize to higher dimensions like the Fibonacci lattice.

Another method that can be used even in higher dimensions is the spherical code problem. In this method, the main goal is distributing N points on the unit sphere $\mathbb{S}^{d-1}$ in a way that the minimal distance between any two points is maximized. Any set of points on the unit sphere is called a spherical code (Nurmela, 1995). In the literature, different solutions for the problem of maximizing the mutual distance between any two points have been suggested based on energy minimization techniques. In the work of Buddenhagen and Kottwitz (2001) up to 90 points were suggested in three dimensions. Nurmela (1995) suggested proper spherical codes up that distribute uniformly on unit spheres up to five-dimensional space. Sloane et al. (2000) collected the most extensive spherical codes in various dimensions. The major restriction with spherical codes is their restriction to specific solutions with fixed coordinates and can only be calculated for a specific number of points in different dimensions. Although there are spherical codes that give the coordinates of a uniform distribution of points on the surface of a 6D unit sphere, the limitation in the available number of points makes them unsuited to select points for the training process.

The most promising method which can be used for distributing points on the d-dimensional surface of the sphere $\mathbb{S}^d$ embedded in d + 1 dimensions is an inverse sampling technique that can generate samples from any distribution (Kroese and Rubinstein, 2012). This method can be used for uniform sampling and relies on repeated random sampling and statistical analysis to compute the result. Based on this theory, a random variable that is uniformly distributed in the range (0,1) can be used to generate a value of a desired random variable with the given distribution. To sample a function uniformly, the first step is to find the PDF (probability distribution function) of that function and then to compute its cumulative probability distribution function (CDF). As the final step, the inverse function of the CDF must be calculated. This method can generate random points on the surface of a unit sphere, but it can also be implemented in combination with the Fibonacci lattice concept described before to generate uniform points. In this case when a random ensemble is generated instead of taking a random independent point from the hypersphere, Fibonacci-like points can be selected to have uniform even distribution on the d-dimensional surface of the sphere $\mathbb{S}^d$.

For introducing this method, a polar coordinate system is defined by $(r, \theta^1, \theta^2, \ldots, \theta^d)$, which can be converted to cartesian coordinates $(x^1, x^2, \ldots, x^{d+1})$ with the relations

$$\begin{aligned}x^{d+1} &= r\cos\theta^d\\ x^d &= r\sin\theta^d\cos\theta^{d-1}\\ &\vdots\\ x^1 &= r\sin\theta^d\cos\theta^1\end{aligned} \tag{21}$$

For calculating the PDF, we assume that the data drawn from a particular distribution are independent and identically distributed. If we consider a vector $\boldsymbol{\theta}$, containing all angles, as the parameter vector for ρ which is the probability density function, the PDF is denoted as ρ_θ (Raychaudhuri, 2008). Since the points are on $\mathbb{S}^d$ and r = 1, the distribution over $\boldsymbol{\theta}$ coordinates must introduce a uniform surface measure which can be written as

$$\rho_\theta(\boldsymbol{\theta}) = \rho_{\theta^d}\left(\theta^d\right)\rho_{\theta^{d-1}|\theta^d}\left(\theta^{d-1}\right)\ldots\rho_{\theta^1|\theta^2\ldots\theta^d}\left(\theta^1\right) \tag{22}$$

Considering **Eqs 21**, **22** and based on the assumption that the angles are independently distributed, the d-dimensional PDF can be written as (Raychaudhuri, 2008).

$$\rho_\theta(\boldsymbol{\theta}) = \prod_{\alpha=1}^{d}\rho_\alpha(\theta^\alpha) \tag{23}$$

Based on the work of Cai et al. (2013), when the variable α is fixed, the density function or ρ_α is given by

$$\begin{aligned}\rho_\alpha(\theta^\alpha) &= \frac{1}{\sqrt{\pi}}\frac{\Gamma\left(\frac{\alpha+1}{2}\right)}{\Gamma\left(\frac{\alpha}{2}\right)}\sin^{\alpha-1}(\theta^\alpha)\\ \theta^\alpha &\in [0, \pi]\end{aligned} \tag{24}$$

Once the distribution is known for each θ^α the next step is to calculate their respective CDFs and finally find their inverse functions.

If we consider X as the continuous random variate that we want to generate, it will follow $\rho(\theta)$ as PDF. The CDF for the

variant F is continuous and increasing in (0,1), which can be seen from its definition

$$Y^{\alpha}(\theta^{\alpha}) = \int_0^{\theta^{\alpha}} \rho_{\alpha}(u)du \tag{25}$$

which satisfies $Y^{\alpha}(0) = 0$ and $Y^{\alpha}(\pi) = 1$.

If Y^{α} is a random number generated from a continuous uniform distribution between 0 and 1, then X is a random number from a distribution with a CDF F, and can be defined as

$$X = F^{-1}(Y^{\alpha}) \tag{26}$$

where F^{-1} is the inverse of the CDF (Raychaudhuri, 2008). As the final step, mapping from Y hypercube to hypersphere should be done based on the Lambert mapping as discussed before. Now a random point ensemble on the sphere has been established. To make it uniform, instead of taking random independent points from the hypercube of Y's, we generate points $Y(n)$ as

$$\begin{aligned} Y_n^d &= \frac{n}{N+1} \\ Y_n^{\alpha-1} &= \{na_1\} \\ Y_n^{\alpha-2} &= \{na_2\} \\ &\cdot \\ &\cdot \\ Y_n^1 &= \{na_{d-1}\} \end{aligned} \tag{27}$$

where a_i can be irrational numbers like the Fibonacci sequence that satisfies $\frac{a_i}{a_j} \notin \mathbb{Q}\ \forall i \neq j$. Using the inverse transform method in combination with the Fibonacci lattice concept, in the case of 4-dimensional space $(d = 4)$ the points are generated as

$$\begin{aligned} w &= cos\,\theta^1 \\ z &= \sin\theta^1 cos\theta^2 \\ y &= \sin\theta^1 \sin\theta^2 \cos\theta^3 \\ x &= \sin\theta^1 \sin\theta^2 \sin\theta^3 \end{aligned} \tag{28}$$

And the angles are given by

$$\begin{aligned} &F(x) \equiv x - \frac{1}{2}\sin 2x,\ \theta_n^1 = F^{-1}\left(\frac{n\pi}{N+1}\right) \\ &\theta_n^2 = arccos\left(1 - 2\{n\sqrt{2}\}\right) \\ &\theta_n^3 = 2\pi\{n\sqrt{3}\} \end{aligned} \tag{29}$$

This method can be used for recursively creating uniformly distributed points in 6- or even higher-dimensional spaces. The points are distributed such that the distance to the nearest neighbors for each point is maximum. Since the points are constructed to lie on the surface of a unit sphere, the distance of each point to the center of the sphere is unity.

Validation of Uniform Distribution of Data Points

The inverse transform method mentioned in the previous section has been implemented in a python code that is freely available within the Python package "pyLabFEA" (Hartmaier et al., 2022) from a public repository. Using this algorithm, any desired number of points can be distributed uniformly on the surface of a d-dimensional sphere. In **Figure 2**, 400 points were distributed uniformly on the surface of a unit sphere in 3D space.

To test the uniformity of this data generation method, 400 points were distributed on the surface of a hypersphere in 6D space. The average distance from each point to its five nearest neighbors was calculated. The distribution of these distances can be seen in **Figure 3A**. For comparison, 400 points were generated randomly, and the same average neighboring distance was calculated and is shown in **Figure 3B**. The random points were generated using the random function available in the NumPy (Harris et al., 2020) package in python.

It is seen that the method laid out here can create any number of data points that are uniformly distributed on a surface of a d-dimensional unit sphere in the sense that the mutual distance of any two points is maximized and the average distances for K nearest neighbors are at the same range. This method will be used in the following to create the data points for training an ML yield criterion.

OPTIMAL STRATEGY FOR DATA GENERATION

In order to find an optimal strategy for selecting a set of training data points, in the following, various data sets will be created with uniform and random distributions and for different sub-spaces of the full stress space. The training results for the different sets are evaluated concerning different error measures, and thus, a strategy to find the smallest possible training data set that provides the desired accuracy of the result is developed.

Since the training data points need to be generated as stress tensors lying on the yield locus of material, we define a reference

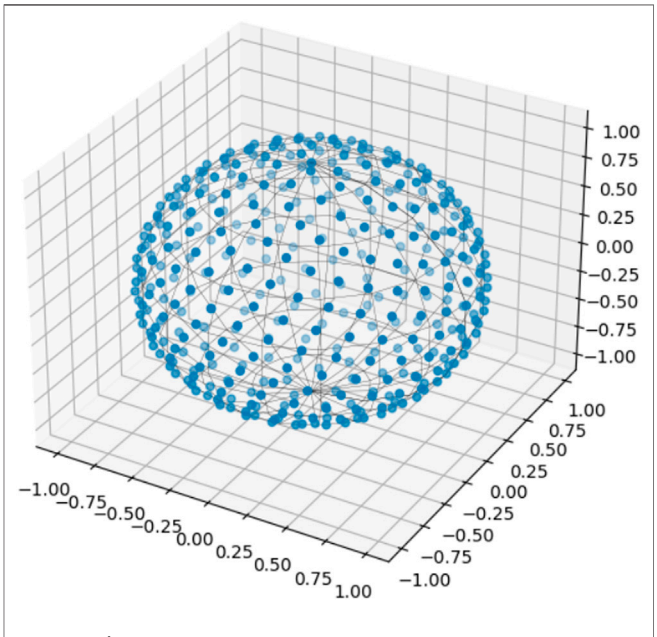


FIGURE 2 | 400 points generated uniformly on the surface of a sphere in 3-dimensional space using the combination of Monte Carlo theory and Fibonacci spiral principle.

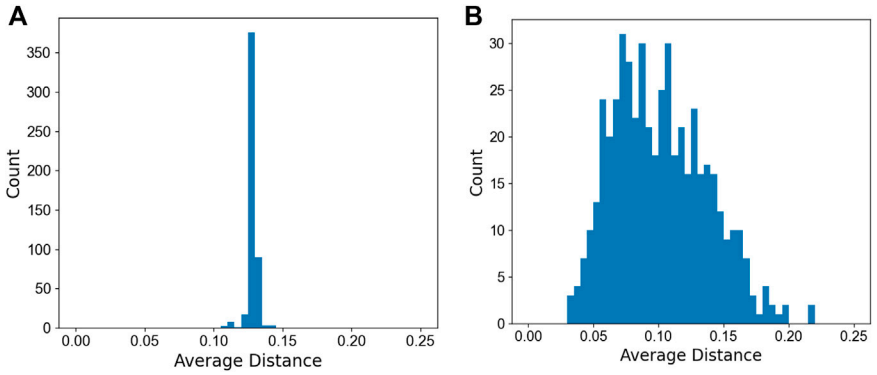


FIGURE 3 | Distribution of average distances for five nearest neighboring points of **(A)** uniform and **(B)** randomly distributed points on the surface of the unit sphere in 6D space. The total number of points is 400.

material with Hill-type anisotropy with the parameters given in **Table 1**. Using such a relatively simple material model with a simple yield criterion defined in **Eq. 2**, allows us to generate the training data sets with a minimum effort. This is beneficial for developing an optimal strategy for generating data sets for materials with significant plastic anisotropy. Afterward, it is verified that the training strategy also produces good results for more severe cases of anisotropy as they can be seen in CPFEM results or described by Barlat-type yield criteria to demonstrate the general applicability of the developed method.

The described reference material is used for creating training data for machine learning algorithms. This is accomplished by first creating unit stresses according to various schemes. Then each unit stress is increased proportionally until the yield function of this stress tensor is zero, i.e., when plastic yielding starts for this specific load case. The full set of stress tensors at the onset of plastic yielding represents the yield function in a data-oriented way and, thus, forms the ground truth for the training of the ML yield function. After this, the ML yield function in form of a Support Vector Classifier (SVC) is trained based on this data set.

As will be seen later, it is necessary to create unit stresses that combine load cases in the full 6D stress space and load cases with purely normal stresses, representing a 3D subspace of the full stress space. For training sets with only 6D load cases, the normal stress space is grossly under-represented with yields a very poor training result. This reflects the fact that the Voigt representation of the symmetric (3×3) stress tensor as vector of its six independent components does not fully represent the tensorial properties of the stress as, for example, the existence of a transformation into a diagonal tensor by a rotation of the coordinate system. Hence, the combination of 3D and 6D stresses reflects the tensorial properties of the stress tensor better. The optimal combination of these load cases will be investigated in the second step together with the optimization of the size of the training set.

Uniform Versus Random Training Data Sets

In the first step to develop an optimal training strategy, we compare the results from training with uniformly and randomly distributed training points. For training, 400 load cases have been genertaed, including 350 load cases in full stress space and 50 load cases in principal space. Since the success of the training procedure depends critically on the hyperparameters, the optimal hyperparameters for the SVC were selected using a grid search algorithm for each data set.

TABLE 1 | Elastic and plastic material parameters define the reference material with Hill-like anisotropy in plastic flow behavior. For simplicity, ideal plasticity with no work hardening is considered in this work.

Quantity	Symbol	Value
Yield strength	σ_y	50 MPa
Young's modulus	E	200 GPa
Poisson's ratio	ν	0.3
Hill parameters	H_1, H_2, H_3, H_4, H_5,H_6	1.4, 1, 0.7, 1.3, 0.8, 1

For uniformly distributed load cases, the result of SVC training with the hyperparameters $C = 5$ and $\gamma = 2.5$ is shown in **Figure 4**, where the J2 equivalent stress of the stress tensors at the onset of plastic yielding is plotted over the polar angle of the stress tensor in the π-plane, i.e., the space of principal deviatoric stresses, **Appendix A** for the definition of these quantities. A good agreement between trained ML function (black line) and the Hill yield locus (blue line) can be observed. In this figure, the support vectors identified during the training procedure are also represented with a color indicating their location in the elastic or plastic domain of the stress space. It is seen that all support vectors lie in the vicinity of the yield locus. At first sight, it appears that many support vectors are misclassified, i.e., are lying in the wrong domain. However, this is a mere artifact from projecting the 6D stresses onto the π-plane for a material with a significant plastic anisotropy.

For comparison, the training was done with the same number of load cases, i.e., a total of 400 with 350 in 6D stress space and 50 in normal stress space, that were generated from a random distribution. The optimal hyperparameters for this case were identified as $C = 12$ and $\gamma = 2.5$ by a grid search algorithm. The result of the training is shown in **Figure 5**, where it can be seen that the ML yield locus resulting from training data points

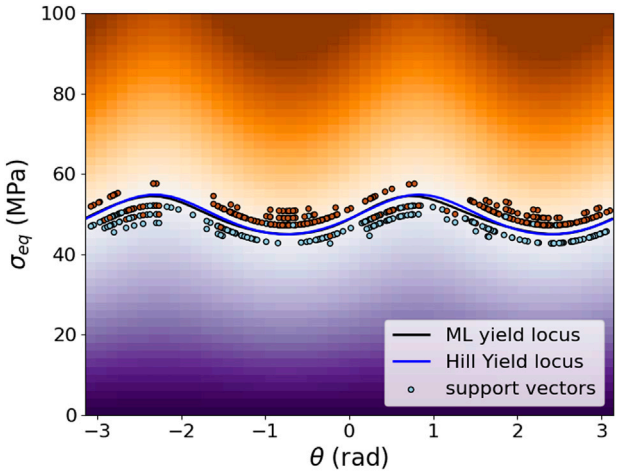


FIGURE 4 | The plot of trained SVM classification with uniformly distributed training data in cylindrical coordinates on the π-plane (space of principal deviatoric stresses). Orange colors represent positive yield function values, i.e., they indicate plastic yielding. Purple colors represent negative values of the yield function, where the stress lies in the elastic regime. No color scheme is given because the absolute value of the ML yield function has no physical meaning. The blue line indicates the stresses where the yield function is zero for the reference material with an analytically formulated Hill-like yield criterion and a black line for the ML yield criterion. The support vectors identified during the training procedure are represented as open symbols.

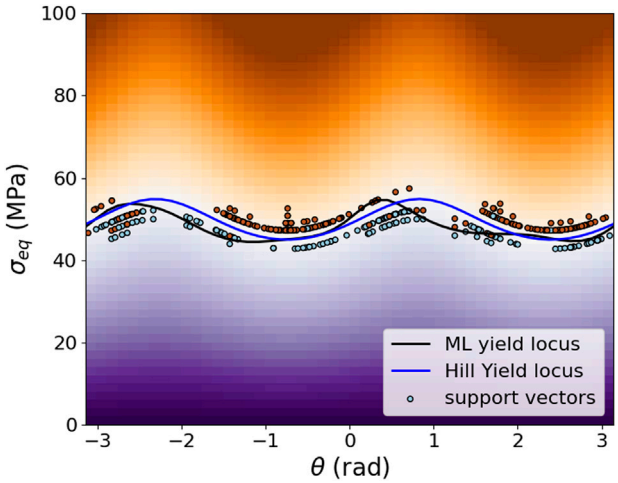


FIGURE 5 | The plot of trained SVM classification with randomly selected training data in cylindrical coordinates on the π-plane (space of principal deviatoric stresses). The figure annotations are identical to **Figure 4**.

representing randomly selected load cases does not have a good agreement with that of the reference material.

To further quantify the results of the training procedure, both ML yield functions are verified by comparison with the analytical yield function for random stress tensors in the vicinity of the yield locus. The confusion matrices in **Figures 6**, **7** summarize the classification performance of both yield functions concerning the

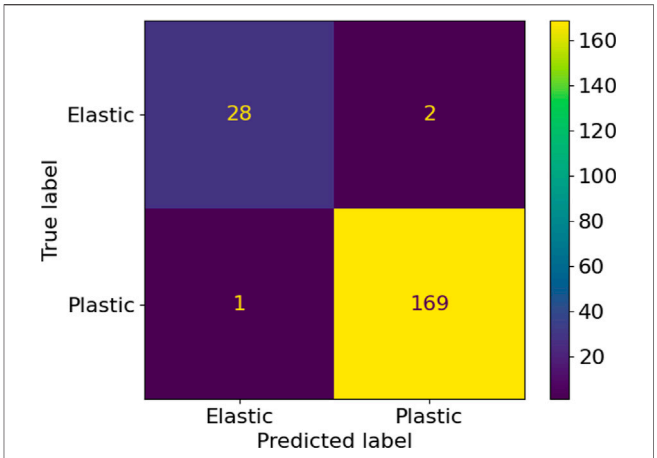


FIGURE 6 | Confusion matrix for uniformly distributed load cases.

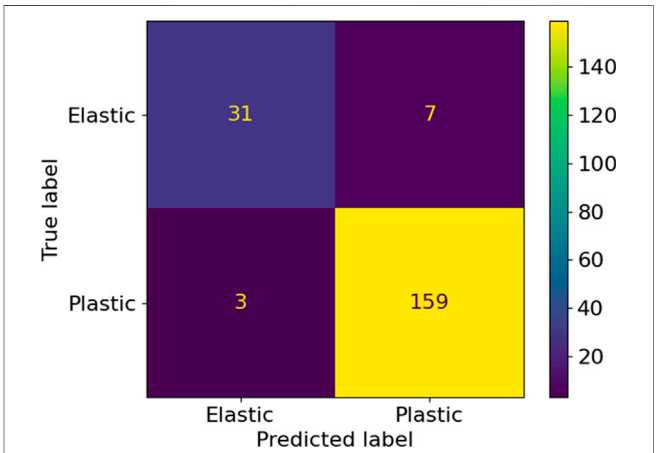


FIGURE 7 | Confusion matrix for randomly distributed load cases.

same test data in the form of a two-dimensional matrix, indexed in one dimension by the object's true class and in the other dimension by the class assigned by the classifier. In this context, the four cells of the matrix are designated as true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) (Shultz et al., 2011).

The classification performance can further be quantified in terms of four classification results, as

$$\begin{aligned} \text{Recall} &= TP/(TP + FN) \\ \text{Precision} &= TP/(TP + FP) \\ \text{Accuracy} &= (TP + TN)/(TP + FP + FN + TN) \\ \text{F1 Score} &= 2 * Precision * Recall/(Precision + Recall) \end{aligned} \tag{30}$$

The summary of metrics for uniform and randomly distributed load cases are given in **Table 2**. It is concluded that uniformly distributed training data results in a significantly higher quality of the training of the ML yield function. Hence, in the remainder of this work, only this method will be further investigated.

TABLE 2 | Summary of the metrics for uniform and random load cases indicate the quality of training.

Metrics	Uniform	Random
Precision	0.9885	0.9578
Accuracy	0.985	0.95
Recall	0.9942	0.9815
F1 Score	0.9913	0.9695

Optimal Number and Structure of Load Cases

After finding the proper strategy for creating unit stresses uniformly distributed in the stress space, the next step is to find the optimal size for it. Besides the quality, the quantity of training data plays a vital role in the performance of any machine learning model. Small training data sets result in low training accuracy and lack of precision, and a very big training size may lead to overfitting and the problem that the model cannot generalize well to new data. In this context, it is also important to find the optimal ratio of unit stresses in the full 6D stress space and purely normal stresses in the 3D subspace.

In the first step for finding the proper ratio between 6D and 3D load cases, training was done for a fixed number of 300 load cases and different ratios of 1:2, 1:1, 1.5:1, 2:1, 2.5:1, and 3:2. and the confusion matrix and the training metrics were compared in each case. The corresponding plots can be seen in **Figure 8**. After training, the precision and mean absolute error were calculated and compared in different ratios of unit load cases in full stress space and the extra cases at principal stress space, the ratio of 2:1 for 6D:3D load cases have highest precision and the lowest mean absolute error (MAE). Mean Absolute error measures the average magnitude of error which quantifies the difference between prediction and the actual observation which can be defined as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - x_i\right| \tag{31}$$

At the next step for finding the optimal training size, the training was done for different total numbers of uniformly distributed load cases with a fixed ratio of 2:1 for 6D to 3D load cases. The comparison is again made based on the precision calculated from the confusion matrix and the MAE, **Figure 9**. It is seen that a number of 300 is the optimal size for the set of training data with the lowest mean absolute error and a high precision after training. It can be seen that an increasing number of data points leads to an increasing MAE, possibly due to overfitting. Furthermore, the precision does not have a uniform trend with an increasing number of data points such that we conclude that 300 data points is the optimal value, which still allows an efficient data generation process.

The resulting ML yield function, trained with the optimized training set of 300 uniformly distributed load cases, is visualized in **Figure 10**. The optimal hyperparameters for the SVC algorithm have been determined as $C = 15$ and $\gamma = 3$ using grid search. A comparison to **Figure 4**, where a total of 400 data points have been used in the training set, clearly reveals the importance of the optimization of the ratio between 6D and 3D stresses in the training data, as the optimized result is more accurate with 25% less training data.

The quality of the ML flow rule training with the optimized data set is further verified by comparing the results of the ML yield function to the known reference values for random stresses in the vicinity of the yield locus. The confusion matrix and the performance metrics were calculated and shown in **Figure 11**, and **Table 3**. Comparing to **Figures 6**, **7** and **Table 2** again indicates a better performance after optimization.

APPLICATION OF TRAINED MACHINE LEARNING YIELD FUNCTIONS IN FINITE ELEMENT ANALYSIS

Hill-Type Anisotropy

To demonstrate the capabilities and the accuracy of the trained ML yield function, in the first step a finite element

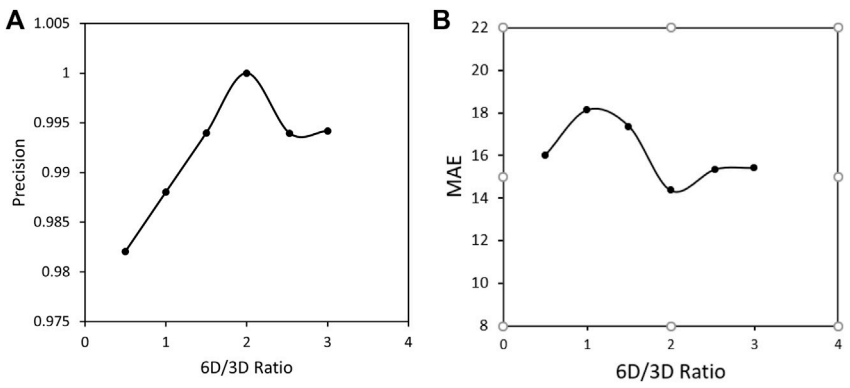


FIGURE 8 | (A) Precision and **(B)** Mean Absolute error after training with total number of 300 load cases in different ratios of load cases in full stress space and extra cases in principal space.

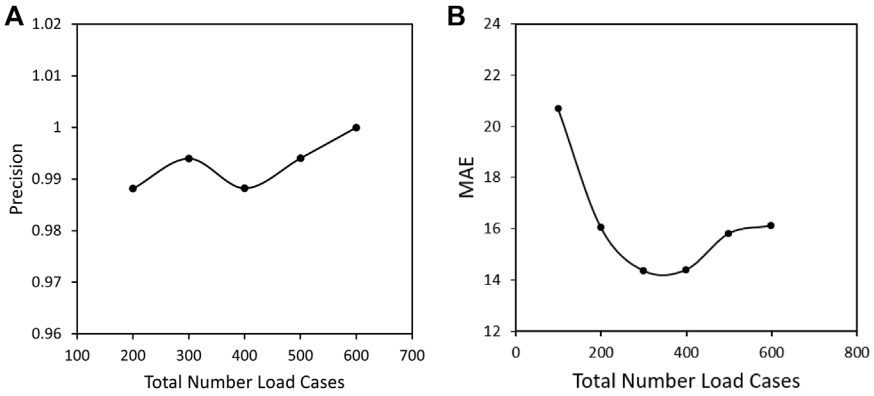


FIGURE 9 | (A) Precision and **(B)** Mean Absolute error after training with different number of load cases and a fixed ratio of 2:1 for 6D to 3D load cases in full stress space and extra cases in principal space.

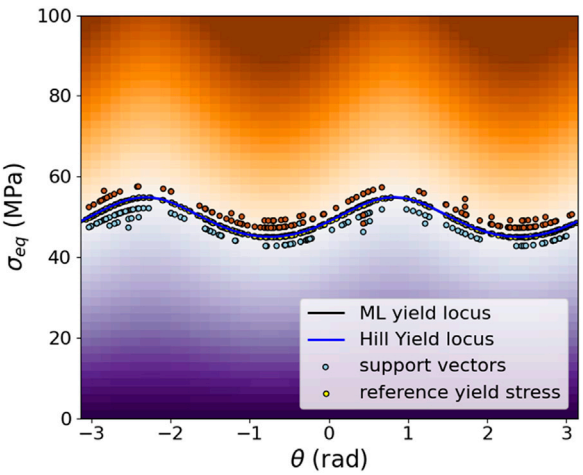


FIGURE 10 | Plot of trained SVM classification with optimized training size and ratio in cylindrical coordinates on the π-plane (space of principal deviatoric stresses). The figure annotations are identical to **Figure 5**.

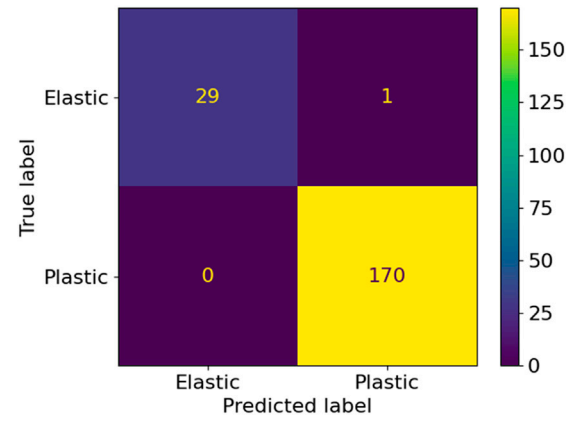


FIGURE 11 | Confusion matrix for 300 load cases and the ratio of 2:1 between 6D and 3D load cases.

TABLE 3 | Summary of the metrics for 300 with the ratio of 2 between 3D and 6D load cases.

Metrics	Uniform
Precision	0.994
Accuracy	0.995
Recall	1
F1 Score	0.997

TABLE 4 | Yield stress obtained for a Hill-type reference material with analytical yield function and a material with an ML yield function trained to data from the reference material under four specified load cases.

Load case	Hill yield stress (MPa)	ML yield stress (MPa)	Rel. difference (%)
Uniaxial stress, horizontal	48.744	48.795	0.10
Uniaxial stress, vertical	45.596	45.644	0.11
Equibiaxial strain	54.171	54.233	0.11
Pure shear strain	45.283	45.33	0.10

analysis (FEA) of a simple 4-element 2D plane-stress model under four load cases has been performed: 1) uniaxial stress in the vertical direction, 2) uniaxial stress in the horizontal direction, 3) equibiaxial strain, and 3) pure shear strain. The numerical simulations shown in this work have been completed with the open-source package pyLabFEA (Hartmaier et al., 2022). All examples of this work are also provided as python code and a Juypter notebook in this public repository. The return mapping algorithm used to calculate the plastic strain increments based on the ML yield function and its gradient, defined in **Eqs 12**, **14**, respectively, has been described in detail in (Suykens and Vandewalle, 1999). In that work, also the full details of the finite element model to evaluate the given load cases are provided. To estimate the accuracy of the ML yield function, its results are compared to

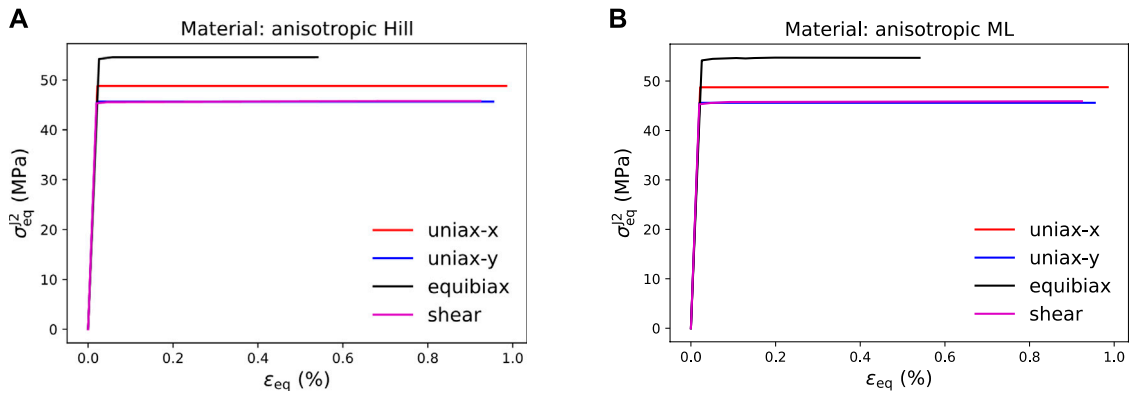


FIGURE 12 | Stress-strain curves obtained for elastic-ideal plastic material behavior under the loading conditions specified in the legend. **(A)** Equivalent J2 stress vs. equivalent total strain for Hill-type yield function, **(B)** Equivalent J2 stress vs. equivalent total strain for ML yield function.

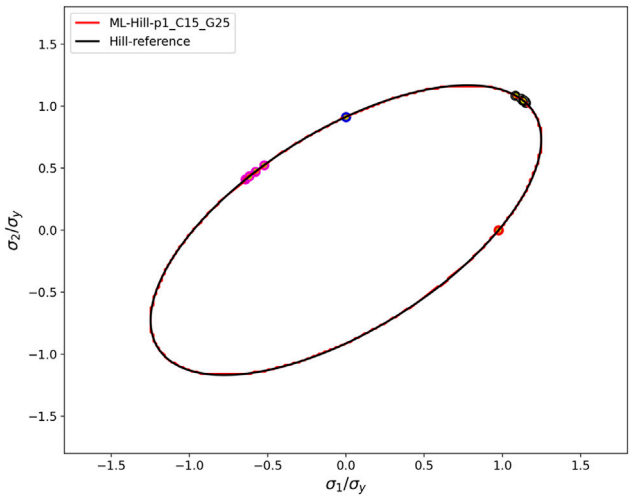


FIGURE 13 | The two non-zero principal values of the flow stresses (colored circles) and yield loci of the trained ML flow rule (red line) and the Hill-type reference material (black line) are plotted for four different load cases of the 2D plane stress model. The large colored circles represent the flow stresses resulting from the ML yield function, and the small yellow circles show flow stresses from the analytical Hill-type yield function.

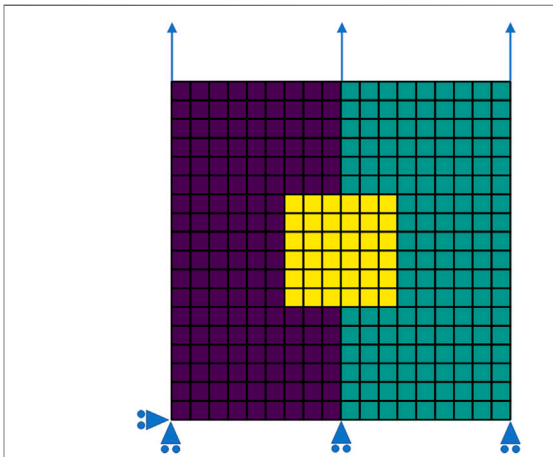


FIGURE 14 | A plane strain model with three sections: (i) linear elastic (yellow), (ii) elastic, ideal-plastic reference material with analytic Hill-type yield function (purple), and (iii) elastic, ideal-plastic material with ML yield function (green). The elements forming a regular mesh are indicated as well as the imposed displacements at the boundary.

the ones obtained from the same model with a Hill-type reference material with elastic-ideal plastic behavior and an analytically defined yield function, **Eq. 2**. The material parameters for this reference material are given in **Table 1**. As described above, the ML yield function has been trained to a data set of 300 stress tensors lying on the yield locus of this reference material. The equivalent yield stresses resulting from the FEA under the four given load cases are summarized in **Table 4**, where it is seen that the error of the ML yield function is only on the order of 0.1%. The resulting curves for equivalent stress versus equivalent strain for each material under the different load cases are plotted in **Figure 12**, where the elastic-ideal plastic material behavior of both materials is verified for total equivalent strains of up to 1%. Note that the equivalent J2 stress is plotted for both materials because the Hill-type equivalent stress defined in **Eq. 3** depends on material parameters that are typically unknown when working with data-based yield functions.

During the plastic deformation of materials with ideal plasticity, it is expected that the flow stresses remain on the yield locus, as the material does not support higher equivalent stresses. In **Figure 13**, the flow stresses for Hill and ML cases are plotted in the space of the non-zero principal stresses of the plane stress model, together with the yield locus in this slice of the stress space. It is seen that all flow stresses, in fact, remain on the yield locus. Furthermore, it is seen that under stress-controlled boundary conditions, i.e., uniaxial stress in the horizontal or vertical direction, the flow stresses remain constant, whereas, under strain-controlled boundary conditions, the response stress of the anisotropic material is subject to changes. A comparison of

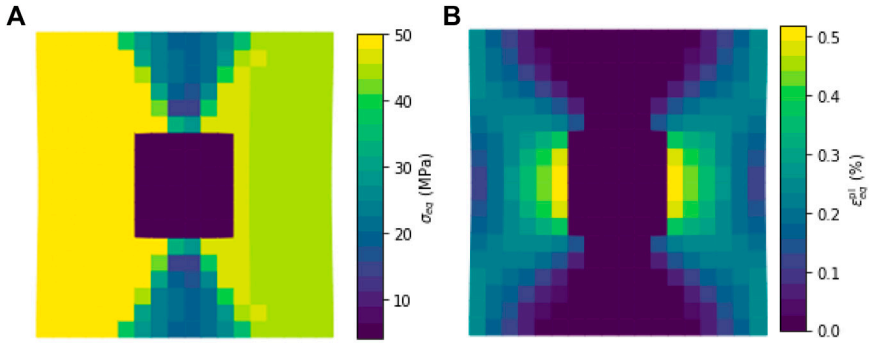


FIGURE 15 | Resulting equivalent stress **(A)** and equivalent plastic strain **(B)** within the three sections of the model given in **Figure 14**.

this evolution of the flow stress tensor under strain-controlled deformation for the analytical Hill-type yield function and the ML yield function exhibits that both solutions are in very good agreement, demonstrating the accuracy and numerical stability of the ML flow rule even under finite plastic strains.

In a second step, the ML yield function is tested under a more complex state of deformation. This is accomplished by performing the FEA of a plane-strain model under uniaxial strain with laterally free boundaries. The three sections of the model are given by a square-shaped elastic inclusion in the middle of a matrix that is vertically split up into the reference material and the ML material, **Figure 14**. As the ML material is trained to have identical material properties as the reference material, the matrix is expected to show a mirror-symmetric state of deformation. The linear-elastic inclusion is rather compliant, with Young's modulus of $E_{inc} = 1$ GPa and a Poisson ratio of $\upsilon_{inc} = 0.27$. The boundary conditions at the top surface are such that a total strain of 0.2% in the vertical direction is applied at the end of the load step.

Figure 14A shows the resulting equivalent stress for each finite element. It is apparent that qualitatively a symmetrical stress state in the different materials is reached. However, quantitatively, the equivalent stresses differ because in the region with the Hill-type material (left-hand-side), the equivalent stress as defined in **Eq. 3** is plotted, whereas in the region with the ML material (right-hand-side) the J2 equivalent stress is plotted, which can also be calculated from **Eq. 3** when the Hill parameters are set $H_1 = H_2 = \ldots = H_6 = 1$. Note that for a data-based constitutive model, properties like the Hill parameters are typically not known and also not necessary for the training process. Hence, it is best to use a material independent definition for the equivalent stress, as the J2 equivalent stress. In **Figure 15B**, the equivalent plastic strains are plotted, demonstrating a completely symmetric deformation between both regions, as expected for materials with identical plastic properties. Thus, this example verifies that the ML yield function can, in fact, be trained to possess the same plastic properties as the Hill-type reference material and produce the same plastic strain increments even in complex loading situations.

Barlat-Type Anisotropy

After verifying the accuracy and robustness of the ML yield function for cases of Hill-type plastic anisotropy, it is tested here under a more demanding kind of anisotropic behavior, given by a material with a Barlat-type yield function (Yld 2004-18p) (Barlat et al., 2005). The material parameters given in **Table 5** have been chosen to mimic a polycrystal with a strong Goss texture, which represents a rather severe case of anisotropic yielding behavior.

Following the workflow developed above, in the first step, a set of 300 stress tensors on the yield locus of the Barlat-type reference model is generated. These training stresses are again produced by generating 200 unit stresses as load cases that are uniformly distributed in the 6D stress space and 100 load cases uniformly distributed in the 3D sub-space of normal stresses. Then, these unit stresses are increased proportionally until the zero of the Barlat yield function Yld 2004-18p (Barlat et al., 2005) is reached. The python code for this example is also provided in the pyLabFEA package (Hartmaier et al., 2022). With this training data set, representing the yield locus of the Barlat-type reference

TABLE 5 | Material parameters for the Barlat-type reference material, mimicking the plastic anisotropy of a Goss-textured polycrystal.

Quantity	Symbol	Value
Yield strength	σ_y	46.76 MPa
Young's modulus	E	151.22 GPa
Poisson's ratio	ν	0.3
Barlat parameters	$B_1, B_2, \ldots, B_6$	0.818, −0.364, 0.312, 0.843, −0.018, 0.832
	$B_7, B_8, \ldots, B_{12}$	0.360, 0.081, 1.293, 1.096, 0.909, 0.277
	$B_{13}, B_{14}, \ldots, B_{18}$	1.090, 1.183, −0.019, 0.905, 1.883, 0.013

material, the ML yield function is trained with the same hyperparameters as given above, i.e., $C = 15$ and $\gamma = 2.5$. The training result is shown in **Figure 16** by projecting yield stresses and support vectors to the π-plane, i.e., the plane of principal deviatoric stresses, **Appendix A** for a definition of the plotted quantities. It is seen that even though the Barlat-type yield function is much more irregular than the Hill-type yield function, the trained ML yield function provides a very accurate representation of its yield locus.

The quality of the ML yield function training is further quantified by comparing the signs of the values of the analytical yield function and the ML yield function for 200 random stress tensors in the vicinity of the yield locus. The corresponding confusion matrix is plotted in **Figure 17,** and the summary of the metrics indicating the high quality of the training result is summarized in **Table 6**. This analysis confirms that the ML yield function describes the Baralat-type reference material with very high accuracy.

With this trained ML yield function, the same FEA cases as above have been performed to demonstrate the stability of the ML yield function even for a more severe plastic anisotropy. After training, the stress-strain curves are plotted as J2 equivalent stress vs. equivalent total strain in four load **Figure 18A** cases as shown in **Figure 18A**. Also, the flow stresses resulting from FEA are plotted in **Figure 18B**, and they all lie on the ML yield locus as expected for ideal plasticity. Furthermore, it is seen again that for the strain-controlled boundary conditions, the stress tensors change during the plastic deformation. This behavior has already been observed for the Hill-type plastic anisotropy, but it is even more pronounced in this case. Note that for the case of equibiaxial strain, the stresses evolve into a "corner" of the yield locus, which causes a significant increase in the equivalent stress that is also seen in the corresponding stress strain curve for this load case.

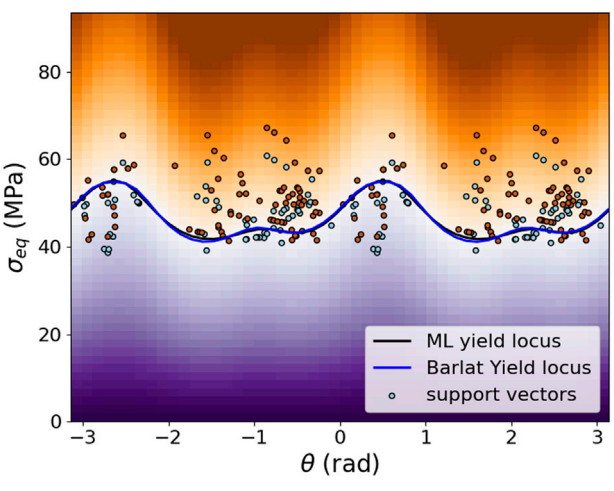


FIGURE 16 | Plot of trained ML yield function (black line), the analytic Barlat yield function (blue line) and the support vectors (circles) in cylindrical coordinates on the π-plane. The same color scheme as in **Figure 5** has been applied.

TABLE 6 | Summary of the metrics for 300 with a ratio of 2 between 3D and 6D load cases for Trained ML yield function.

Metrics	Uniform
Precision	0.9939
Accuracy	0.995
Recall	1
F1 Score	0.9969

Furthermore, in **Figure 18**, the equivalent stress **Figure 18C** and the equivalent plastic strain **Figure 18D** for a model with a square-shaped elastic inclusion and an elastic-ideal plastic matrix represented by the ML yield function are illustrated to demonstrate the numerical stability of the ML flow rule even in cases of heterogeneous deformation patterns.

CONCLUSION

In this work, an optimized procedure to generate a data-based description of the yield function of an arbitrary material has been developed. Conventionally, the yield function is based on the concept of the equivalent stress and indicates whether the material response to an applied stress tensor results in linear elastic material response, indicated by a negative value of the yield function, or rather in plastic yielding of the material when the yield function is zero or positive. Mathematically, the zeros of the yield function constitute a hypersurface in stress space, the so-called yield locus, that separates elastic and plastic domains. Since the stress space is 6-dimensional and spanned by the six independent normal and shear components of the stress tensor, it is essential to find an optimal way for sampling this hypersurface with as few as possible data points. Each data point represents a stress tensor at which the material starts to yield plastically, and generating such data requires either numerically

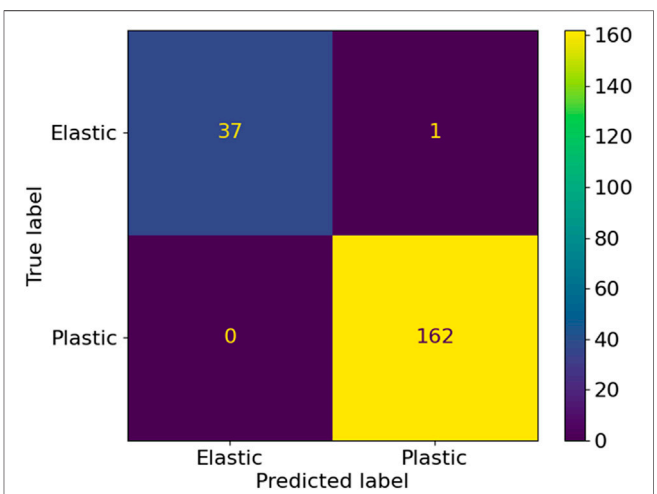


FIGURE 17 | Confusion matrix comparing the trained ML yield function and the reference Barlat-type yield function for 200 random stress tensors in the vicinity of the yield locus.

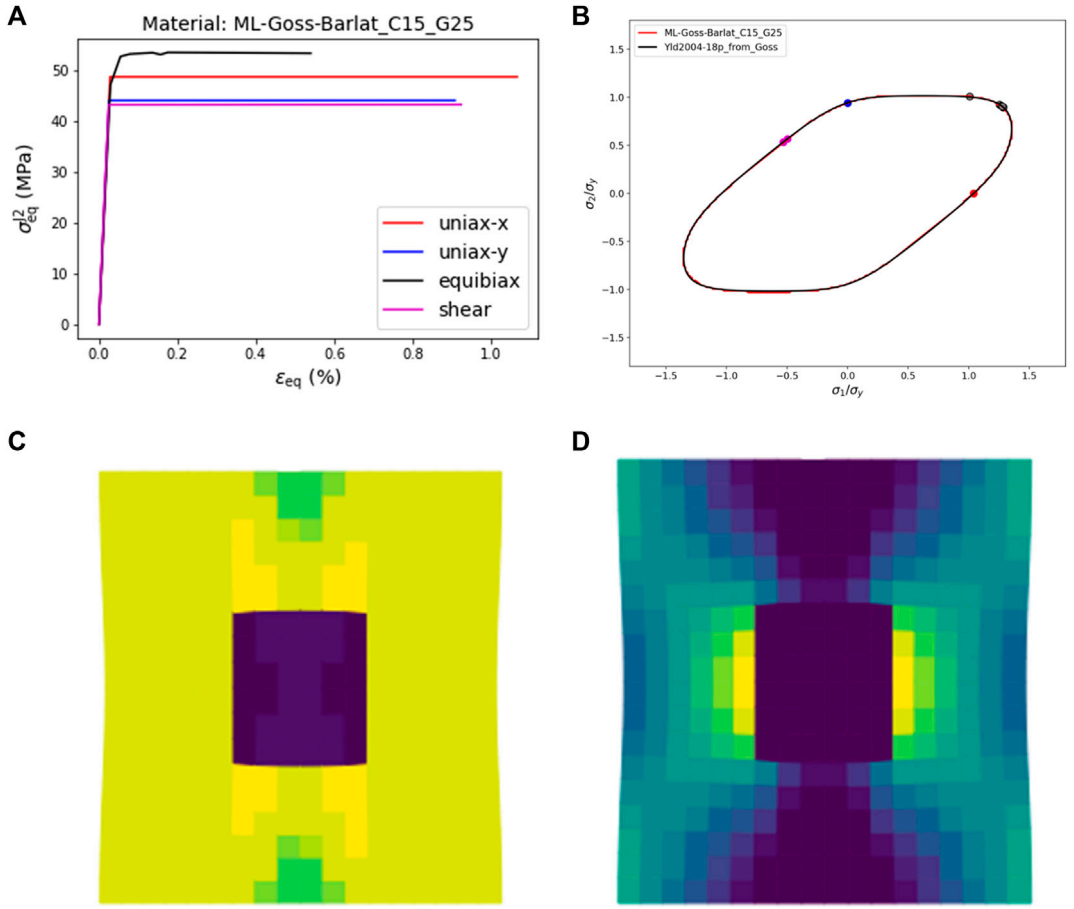


FIGURE 18 | (A) Equivalent J2 stress vs. equivalent total strain curves obtained for elastic-ideal plastic material behavior under the loading conditions specified in the legend for the rained ML yield function; **(B)** yield loci of ML and Barlat-type materials and the flow stresses obtained from the load cases in **(A)** plotted as principal stresses; **(C)** equivalent J2 stress; and **(D)** equivalent plastic strain resulting from the 2D plane strain model with a square-shaped elastic inclusion in the center.

expensive approaches such as crystal plasticity finite element methods or laborious mechanical testing under multi-axial loading conditions or a combination of both.

In a first step, a numerical method has been introduced that allows a uniform sampling of the yield locus in a six-dimensional stress space. The main idea behind this method is to proportionally increase uniformly distributed unit stresses until the criterion for plastic yielding is reached for each loading direction. Finding a uniform distribution of arbitrarily many points on the surface of a unit sphere in more than three dimensions is, however, a highly non-trivial task. The solution suggested in this work combines ideas of Monte Carlo sampling and choosing regularly distributed sampling points in higher-dimensional spaces based on the Fibonacci sequence. To sample a higher dimensional unit sphere uniformly, the probability distribution function of its surface needs to be defined in Cartesian coordinates, and then the cumulative distribution function and its inverse need to be computed. Based on this inverse function, a mapping algorithm is defined by which random numbers from the unit interval can be distributed uniformly on the hypersphere. If Fibonacci-like points are mapped accordingly, instead of random numbers, it can be shown that the mutual distance between any two sampling points on the surface of the hypersphere is rather constant and maximized compared to randomly distributed points. It is demonstrated in this work that such a uniform distribution of stress tensors in the 6-dimensional stress space is superior to purely random sampling.

In the next step, this data-oriented description of the yield locus is used as the basis for training of a support vector classifier (SVC) that takes an arbitrary stress tensor as input and predicts whether the material response to this stress is elastic or plastic. In earlier work (Hartmaier, 2020), it has been shown for purely normal stresses how plasticity in the framework of finite element analysis (FEA) can be described based on such trained SVC. Here, this formulation is generalized to the full 6-dimensional stress space, and it is demonstrated that SVC can be trained to predict the behavior of classical yield functions, like Hill or Barlat-type yield functions, with very high accuracy, even for a very significant plastic anisotropy.

As for any machine learning (ML) algorithm, providing high-quality training data is of great importance, as training with smaller sets of proper data will result in a better training

success than larger sets of poor data. After finding a suitable way to distribute training data uniformly in stress space, it was found that an appropriate ratio of normal stresses and stresses in the full 6-dimensional stress space is necessary because otherwise, the normal stresses are under-represented. It is demonstrated that a ratio of 1:2 for normal stresses and full stresses represents the tensorial properties of the yield stress in the best way. It is also concluded that 300 data points in the form of stress tensors on the yield locus are sufficient to train the ML yield function with high accuracy, even in cases of severe plastic anisotropy. The thus-trained ML yield functions have been shown to produce accurate and stable numerical solutions in FEA.

To further expand the applicability of ML yield functions in FEA, microstructural parameters like crystallographic texture or grain size and morphology can be included in the input data for the training of the ML yield function, besides the purely mechanical data used in this work. Furthermore, it is crucial to develop a proper data-oriented formulation of work hardening and history-dependent material behavior in the next step.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding author.

AUTHOR CONTRIBUTIONS

Both authors have contributed equally to the work.

ACKNOWLEDGMENTS

RS gratefully acknowledges helpful discussions on the mathematical problems with Erik Brinkman. AH gratefully acknowledges funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—Project-ID 190389738—TRR 103.

REFERENCES

Banabic, D., Kuwabara, T., Balan, T., and Comsa, D. S. (2004). An Anisotropic Yield Criterion for Sheet Metals. *J. Mater. Process. Technol.* 157-158, 462–465. doi:10.1016/j.jmatprotec.2004.07.106

Barlat, F., Aretz, H., Yoon, J. W., Karabin, M. E., Brem, J. C., and Dick, R. E. (2005). Linear Transfomation-Based Anisotropic Yield Functions. *Int. J. Plasticity* 21, 1009–1039. doi:10.1016/j.ijplas.2004.06.004

Bonet, J., and Wood, R. D. (1997). *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge: Cambridge University Press. Available at: https://books.google.de/books?id=ORmLdrq1fI8C.

Buddenhagen, J., and Kottwitz, D. A. (2001). Multiplicity and Symmetry Breaking in (Conjectured) Densest Packings of Congruent Circles on a Sphere. *Preprint*. Available at: https://www.researchgate.net/publication/248871642_Multiplicity_and_Symmetry_Breaking_in_Conjectured_Densest_Packings_of_Congruent_Circles_on_a_Sphere.

Cai, T., Fan, J., and Jiang, T. (2013). Distributions of Angles in Random Packing on Spheres. *J. Mach. Learn. Res.* 14, 1837–1864.

Cazacu, O., and Barlat, F. (2001). Generalization of Drucker's Yield Criterion to Orthotropy. *Maths. Mech. Sol.* 6, 613–630. doi:10.1177/108128650100600603

Chen, W. F., and Saleeb, A. F. (1994). *Constitutive Equations for Engineering Materials*. Amsterdam, Netherlands: Elsevier. Available at: https://books.google.de/books?id=P91IAQAAIAAJ.

Chinesta, F., Ladeveze, P., Ibanez, R., Aguado, J. V., Abisset-Chavanne, E., and Cueto, E. (2017). Data-driven Computational Plasticity. *Proced. Eng.* 207, 209–214. doi:10.1016/j.proeng.2017.10.763

Eggersmann, R., Kirchdoerfer, T., Reese, S., Stainier, L., and Ortiz, M. (2019). Model-free Data-Driven Inelasticity. *Comput. Methods Appl. Mech. Eng.* 350, 81–99. doi:10.1016/j.cma.2019.02.016

Eggersmann, R., Stainier, L., Ortiz, M., and Reese, S. (2021). Model-free Data-Driven Computational Mechanics Enhanced by Tensor Voting. *Comput. Methods Appl. Mech. Eng.* 373, 113499. doi:10.1016/j.cma.2020.113499

González, Á. (2010). Measurement of Areas on a Sphere Using Fibonacci and Latitude–Longitude Lattices. *Math. Geosci.* 42, 49–64.

Hannay, J. H., and Nye, J. F. (2004). Fibonacci Numerical Integration on a Sphere. *J. Phys. A: Math. Gen.* 37, 11591–11601. doi:10.1088/0305-4470/37/48/005

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., et al. (2020). Array Programming with NumPy. *Nature* 585, 357–362. doi:10.1038/s41586-020-2649-2

Hartmaier, A. (2020). Data-oriented Constitutive Modeling of Plasticity in Metals. *Materials* 13, 1600. doi:10.3390/ma13071600

Hartmaier, A., Menon, S., and Shoghi, R. (2022). *Python Laboratory for Finite Element Analysis (PyLabFEA)*. doi:10.5281/zenodo.5913365

Helling, D. E., and Miller, A. K. (1987). The Incorporation of Yield Surface Distortion into a Unified Constitutive Model, Part 1: Equation Development. *Acta Mechanica* 69, 9–23. doi:10.1007/bf01175711

Hill, R. (1948). A Theory of the Yielding and Plastic Flow of Anisotropic Metals. *Proc. R. Soc. Lond. Ser. A. Math. Phys. Sci.* 193, 281–297.

Huber, M. T. (1904). Przyczynek Do Podstaw Wytorymalosci. *Czasop Techn* 22, 81.

Ibañez, R., Abisset-Chavanne, E., Aguado, J. V., Gonzalez, D., Cueto, E., and Chinesta, F. (2018). A Manifold Learning Approach to Data-Driven Computational Elasticity and Inelasticity. *Arch. Computat Methods Eng.* 25, 47–57. doi:10.1007/s11831-016-9197-9

Karafillis, A. P., and Boyce, M. C. (1993). A General Anisotropic Yield Criterion Using Bounds and a Transformation Weighting Tensor. *J. Mech. Phys. Sol.* 41, 1859–1886. doi:10.1016/0022-5096(93)90073-o

Kirchdoerfer, T., and Ortiz, M. (2016). Data-driven Computational Mechanics. *Comput. Methods Appl. Mech. Eng.* 304, 81–101. doi:10.1016/j.cma.2016.02.001

Kroese, D. P., and Rubinstein, R. Y. (2012). Monte Carlo Methods. *Wires Comp. Stat.* 4, 48–58. doi:10.1002/wics.194

Kurtyka, T., and Życzkowski, M. (1996). Evolution Equations for Distortional Plastic Hardening. *Int. J. Plasticity* 12, 191–213. doi:10.1016/s0749-6419(96)00003-4

Linka, K., Hillgärtner, M., Abdolazizi, K. P., Aydin, R. C., Itskov, M., and Cyron, C. J. (2021). Constitutive Artificial Neural Networks: A Fast and General Approach to Predictive Data-Driven Constitutive Modeling by Deep Learning. *J. Comput. Phys.* 429, 110010. doi:10.1016/j.jcp.2020.110010

Liu, Z., Kafka, O. L., Yu, C., and Liu, W. K. (2018). "Data-driven Self-Consistent Clustering Analysis of Heterogeneous Materials with crystal Plasticity," in *Adv. Comput. Plast.* (Berlin, Germany: Springer), 221–242. doi:10.1007/978-3-319-60885-3_11

Marques, R., Bouville, C., Ribardière, M., Santos, L. P., and Bouatouch, K. (2013). Spherical Fibonacci point Sets for Illumination Integrals. *Comput. Graphics Forum* 32, 134–143. doi:10.1111/cgf.12190

Marsaglia, G. (1972). Choosing a point from the Surface of a Sphere. *Ann. Math. Statist.* 43, 645–646. doi:10.1214/aoms/1177692644

Nurmela, K. J. (1995). *Constructing Spherical Codes by Global Optimization Methods*. Helsinki: Helsinki University of Technology.

Peranio, N., Li, Y. J., Roters, F., and Raabe, D. (2010). Microstructure and Texture Evolution in Dual-phase Steels: Competition between Recovery, Recrystallization, and Phase Transformation. *Mater. Sci. Eng. A* 527, 4161–4168. doi:10.1016/j.msea.2010.03.028

Pian, T. H. H., and Wu, C. C. (2005). *Hybrid and Incompatible Finite Element Methods*. Abingdon-on-Thames, Oxfordshire, UK: Taylor & Francis. Available at: https://books.google.de/books?id=t595AgAAQBAJ.

Raychaudhuri, S. (2008). "Introduction to Monte Carlo Simulation," in *2008 Winter Simul. Conf.* (Piscataway, NJ, USA: IEEE), 91–100. doi:10.1109/wsc.2008.4736059

Roters, F., Eisenlohr, P., Bieler, T. R., and Raabe, D. (2011). *Crystal Plasticity Finite Element Methods: In Materials Science and Engineering*. Hoboken, NJ, USA: Wiley. Available at: https://books.google.de/books?id=WFSNYRvc-ZYC.

Shultz, T. R., Fahlman, S. E., Craw, S., Andritsos, P., Tsaparas, P., Silva, R., et al. (2011). Confusion Matrix. *Encycl. Mach. Learn.* 61, 209. doi:10.1007/978-0-387-30164-8_157

Sloane, N. J. A., Hardin, R. H., and Smith, W. D. (2000). *Spherical Codes*. Available at: http//www2.Res.Att.Com/~Njas/Packings.

Suykens, J. A. K., and Vandewalle, J. (1999). Least Squares Support Vector Machine Classifiers. *Neural Process. Lett.* 9, 293–300.

Swinbank, R., and James Purser, R. (2006). Fibonacci Grids: A Novel Approach to Global Modelling. *Q.J.R. Meteorol. Soc.* 132, 1769–1793. doi:10.1256/qj.05.227

v Mises, R. (1913). Mechanik der festen Körper im plastisch-deformablen Zustand, Nachrichten von Der Gesellschaft Der Wissenschaften Zu Göttingen. *Math. Klasse.* 1913, 582–592.

Vajragupta, N., Ahmed, S., Boeff, M., Ma, A., and Hartmaier, A. (2017). Micromechanical Modeling Approach to Derive the Yield Surface for BCC and FCC Steels Using Statistically Informed Microstructure Models and Nonlocal crystal Plasticity. *Phys. Mesomech.* 20, 343–352. doi:10.1134/s1029959917030109

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

APPENDIX A

Since in most metals hydrostatic stress components do not play a significant role in plastic deformation, it is useful to analyze their flow stresses in the deviatoric stress space. To accomplish this, oftentimes a transform to principal stresses is applied. All deviatoric principal stresses lie on a plane in the stress space, the so-called π-plane. In this work, we use a cylindrical coordinate system to plot stresses in this plane, where the equivalent stress σ_{eq} is plotted along the y-axis and the polar angle θ is plotted along the x-axis. This projection is best introduced via defining a complex-valued stress deviator based on the vector of deviatoric principal stresses $\boldsymbol{\sigma}' = (\sigma_1 - p,\ \sigma_2 - p,\ \sigma_3 - p)$, where $\sigma_1, \sigma_2, \sigma_3$ are the principal stresses and p is the hydrostatic stress. Furthermore, it is necessary to specify two arbitrary orthogonal directions in the π-plane, for which in this work $\boldsymbol{a} = (2, -1, -1)/\sqrt{6}$ and $\boldsymbol{b} = (0,\ 1,\ -1)/\sqrt{2}$ are chosen. Note that these unit vectors span the plane normal to the hydrostatic axis $\boldsymbol{c} = (1, 1, 1)/\sqrt{3}$. Following the method developed in previous work (Hartmaier, 2020), the complex-valued deviatoric stress is defined as

$$\sigma_c^{'} = \boldsymbol{\sigma} \cdot \boldsymbol{a} + i\boldsymbol{\sigma} \cdot \boldsymbol{b} = \sqrt{2/3} \cdot \sigma_{eq} \cdot e^{i\theta} \tag{32}$$

In this definition, i is the imaginary unit and the polar angle θ can be evaluated by

$$\theta = -i \ln \frac{\boldsymbol{\sigma} \cdot \boldsymbol{a} + i\boldsymbol{\sigma} \cdot \boldsymbol{b}}{\sqrt{2/3}\sigma_{eq}} \tag{33}$$

This mapping of the stress is used in **Figures 4**, **5**, **10**, **16**.

frontiers | Frontiers in Materials

ORIGINAL RESEARCH
published: 31 May 2022
doi: 10.3389/fmats.2022.877597

SimStack: An Intuitive Workflow Framework

Celso R. C. Rêgo[1]*, *Jörg Schaarschmidt*[1], *Tobias Schlöder*[1], *Montserrat Penaloza-Amion*[1], *Saientan Bag*[1], *Tobias Neumann*[2], *Timo Strunk*[2] and *Wolfgang Wenzel*[1]

[1]Karlsruhe Institute of Technology (KIT), Institute of Nanotechnology, Eggenstein-Leopoldshafen, Germany, [2]Nanomatch GmbH, Karlsruhe, Germany

Establishing a fundamental understanding of the nature of materials via computational simulation approaches requires knowledge from different areas, including physics, materials science, chemistry, mechanical engineering, mathematics, and computer science. Accurate modeling of the characteristics of a particular system usually involves multiple scales and therefore requires the combination of methods from various fields into custom-tailored simulation workflows. The typical approach to developing patchwork solutions on a case-to-case basis requires extensive expertise in scripting, command-line execution, and knowledge of all methods and tools involved for data preparation, data transfer between modules, module execution, and analysis. Therefore multiscale simulations involving state-of-the-art methods suffer from limited scalability, reproducibility, and flexibility. In this work, we present the workflow framework SimStack that enables rapid prototyping of simulation workflows involving modules from various sources. In this platform, multiscale- and multimodule workflows for execution on remote computational resources are crafted via drag and drop, minimizing the required expertise and effort for workflow setup. By hiding the complexity of high-performance computations on remote resources and maximizing reproducibility, SimStack enables users from academia and industry to combine cutting-edge models into custom-tailored, scalable simulation solutions.

Keywords: SimStack, workflows, Materials Design, Multiscale modelling, WaNos

OPEN ACCESS

Edited by:
Zhenyu Li,
University of Science and Technology of China, China

Reviewed by:
Kesong Yang,
University of California, San Diego, United States
Thomas Hammerschmidt,
Ruhr University Bochum, Germany

***Correspondence:**
Celso R. C. Rêgo
celso.rego@kit.edu

Specialty section:
This article was submitted to Computational Materials Science, a section of the journal Frontiers in Materials

Received: *16 February 2022*
Accepted: *11 April 2022*
Published: *31 May 2022*

Citation:
Rêgo CRC, Schaarschmidt J, Schlöder T, Penaloza-Amion M, Bag S, Neumann T, Strunk T and Wenzel W (2022) SimStack: An Intuitive Workflow Framework. Front. Mater. 9.877597. doi: 10.3389/fmats.2022.877597

1 INTRODUCTION

In the Industry 4.0 context (Lasi et al., 2014), digital twins are an essential tool for companies based who's R&D is based on scientific innovation (Posada et al., 2018). The digitalization of a system or a process provides vital information about the real-world scenario in real-time. This enables the efficient identification of bottlenecks, thereby speeding up the product development cycle (Zhu and Geng, 2013; Müller and Däschle, 2018) and, in consequence, saving R&D costs and shortening time-to-market (Mathew et al., 2017). In terms of the physical-chemical processes, the development of digital twins is gaining mainstream attention in the scientific community, especially in materials design (Wu et al., 2020; Ngandjong et al., 2021). In this field considerable efforts are made to build digital twins in order to screen and discover new functional materials e.g., for solar cells (Kim et al., 2021; Octavio de Araujo et al., 2021), batteries (Ponce et al., 2017; Bölle et al., 2019), thermoelectricity (Madsen, 2006; Yao et al., 2021), and catalysis (Mamun et al., 2019; Mamun et al., 2020). A prerequisite for building a useful digital

twin is the availability of predictive simulation protocols. During the designing process of digital twins, a high level of complexity induces a significant interdisciplinary challenge, especially when the material characteristics need to be described by different scales of materials behavior, demanding, in many cases, multiscale methods. Schaarschmidt et al. (2021) show that workflow frameworks can address those challenges in computational materials design.

In addition to technical challenges, computational modeling of a complex physico-chemical process requires knowledge of different methods. These include, amongst others, density functional theory (DFT), molecular dynamics (MD), kinetic Monte-Carlo (KMC), and or finite element methods (FEM). Different techniques are employed depending on the studied phenomena and the scale at which the system is represented. However, in each method, models typically have many parameters and often require a meticulous manual setup to generate meaningful results. Many applications requiring multiscale or high-throughput calculations for a given system also include executing a large number of simulations. Handling these complex computational protocols with script-based approaches is challenging, especially in simulations where the numerical errors embedded in the codes need to be carefully controlled. Usually, multiple steps are required to handle errors, yet, these are often poorly documented and not standardized, making it challenging to keep track of, even for experienced computational experts, thereby limiting the reuse of these computational protocols even within the same group. Therefore, scientific workflows have been proposed to address these shortcomings and inefficiencies by providing automation, complexity reduction, high-performance computing (HPC) readiness, data reusability, data provenance, and reliability and resilience of formalized workflows. Workflows can describe a complex simulation protocol while only exposing a predefined set of relevant computational parameters to the end-user. Therefore, the general aim of workflow frameworks is to allow the end-users to focus on the science instead of spending time setting up and monitoring individual calculations. Several such frameworks have been proposed to leverage the scientific workflow benefits in the last decade. These include free and commercial solutions such as Fireworks (Jain et al., 2015), AiiDA (Pizzi et al., 2016; Huber et al., 2020; Uhrin et al., 2021), KNIME (Berthold et al., 2008), Pipeline Pilot (Warr, 2012), MyQueue (Hjorth Larsen et al., 2017; Mortensen et al., 2020), Pyiron (Janssen et al., 2019), or AFLOW (Curtarolo et al., 2012), and to name a few.

Next to reducing complexity, another major benefit of workflow frameworks is the improved reproducibility of formalized workflows. Reproducibility is a huge challenge for the scientific community: In 2016, researchers from fields such as biology, medicine, physics, chemistry, and engineering largely failed to reproduce their previously published experiments (Baker, 2016). The transition from *theory* + *experiment* to the *theory* + *experiment* + *computer simulation* paradigm (Rodrigues et al., 2021) imposes increasing challenges on the experimental and computational research. The advent of computer simulation in the theoretical sciences introduced further challenges regarding the reproducibility of scientific studies that are not present in purely analytical methods (Rodrigues et al., 2021). In a computational simulation study, five groups were asked to perform the same simulation tasks using eight codes with the same force fields. The initial results were highly inconsistent between the groups and simulation codes. Only after some iterations the outcomes started to become consistent (Schappals et al., 2017). This simple experiment shows that incorrect usage is, in most cases, the source of errors in simulations (Wong-ekkabut and Karttunen, 2016; DeFever et al., 2021). Thus, describing the full simulation in a formalized workflow ensures correct usage and consistency among identical and similar simulations.

As one approach to overcome the issue of reproducibility and leverage the advantages of reusability, transferability, and flexibility concepts, we discuss the SimStack workflow framework here. SimStack enables the rapid prototyping of complex simulation workflows with computational modules from various sources. The transfer of re-usable workflows between groups and researchers allow scientists to perform particular predefined simulations with the same quality as the computational expert who conceived and implemented it.

In this work, we present four workflow applications where SimStack has been employed. These workflows combine typical state-of-the-art methods of materials design to solve and deal with real problems and issues representative of those commonly encountered by researchers in the simulation field using a comprehensive range of methods. The SimStack concepts and their usage, features, and applicability to various fields are illustrated by the selected examples covering Umbrella Sampling, Exciton Dynamics, Dihedral Scan, and Emission spectra of organic molecules. The documentation of those workflows shows additional details on applying the SimStack framework features.

2 THE WORKFLOW FRAMEWORK SIMSTACK

The main goal of all major workflow frameworks is to capture the elements of a complex protocol and automate its execution. Depending on the implementation and target user group, expert knowledge is often required for using the framework, setting it up on local or remote compute resources, or incorporating new simulation methods. In many cases, easy-to-use frameworks are often limited in their flexibility and, therefore, hard to extend to the needs of a specific problem, while flexible frameworks are hard-to-use for inexperienced users. Here we introduce the SimStack framework https://simstack.de/, which addresses the issue by providing an easy-to-use flexible drag and drop graphical user interface (GUI), which is automatically generated for a given set of exposed parameters from a simple file in Extensible Markup Language (XML) format. The usage of the XML description of the user input coupled with a simple templating language enables computational experts and non-experts to provide a GUI for a particular application in a matter of minutes. SimStack connects to remote

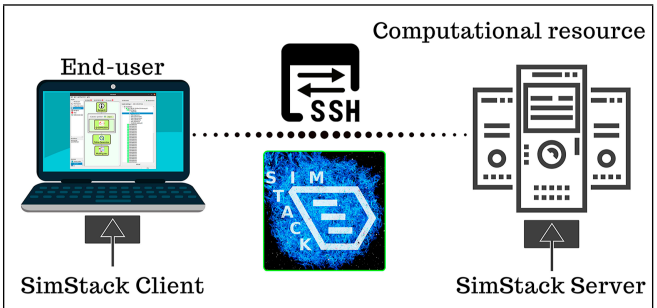


FIGURE 1 | The SimStack workflow framework is based on a client-server concept connected via the secure shell (SSH) protocol. The end-user designs and sets up the workflow within the SimStack client on a local machine. The workflow and all required input are transferred to the HPC resource via SSH upon submission. The SimStack server process on the HPC resource subsequently generates a single job for each step of the workflow and manages the execution of these jobs through the local scheduling software.

high-performance computing (HPC) resources and automates data transfer and execution of the entire workflow within the HPC environment. Thus, it facilitates the efficient implementation, adoption, and execution of complex and extensive simulation workflows and enables fast uptake of modeling techniques for advanced materials by researchers in academia and industry. SimStack is developed in a joint project by Nanomatch GmbH and the Karlsruhe Institute of Technology (KIT).

2.1 SimStack Concept

As shown in **Figure 1**, the SimStack workflow framework is based on a lightweight client-server concept. The client provides a GUI for the end-user to construct, modify, and configure the workflows, submit the workflow to the server component on remote HPC resources, monitor submitted workflows, and browse and retrieve the generated data. Each workflow comprises various building blocks with predefined control elements for a given computational task. The tasks represent discrete steps in the execution of the workflow and are called **W**orkflow **A**ctive **No**des (WaNos) within SimStack. The core component of a WaNo is an XML file describing the expected input, configurable parameters, the output generated by the WaNo, and the code to be executed. By drag and drop, the end-user can quickly create a new workflow from the available building blocks or adapt existing workflows to generate a custom-tailored solution for a scientific problem. In order to incorporate the user input, SimStack employs the templating engine Jinja (https://jinja.palletsprojects.com). With this templating approach, specific parameters can be exposed

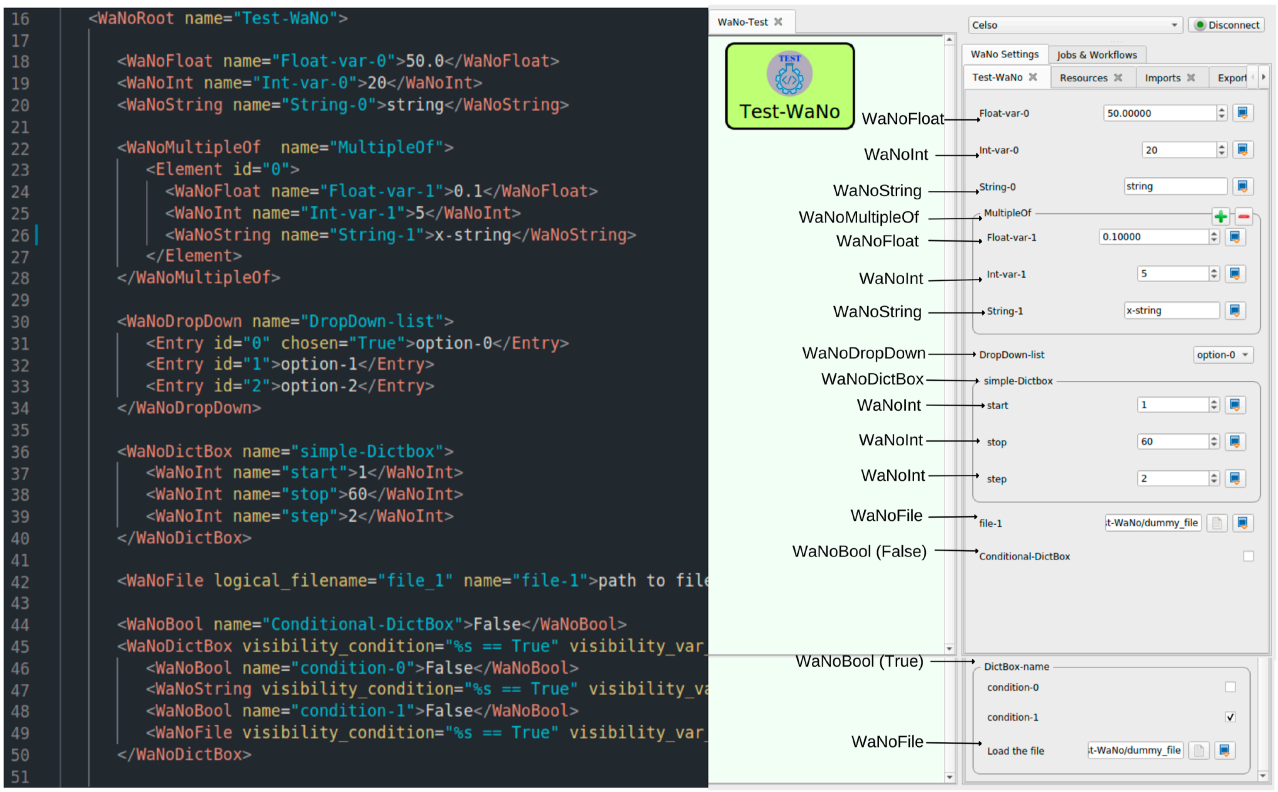


FIGURE 2 | This WaNo example shows the XML file and its correspondent GUI. The right side displays the XML file with the tags available within the SimStack workflow framework. On the left side, the arrows associate the tags used to generate the field and variable types of the GUI. The visibility of the second *WaNoDictBox* in the XML is coupled to the Boolean variable *Conditional-DictBox*. In this example, the executable is the python script test-script.py.

easily via the GUI and included as command line parameters or into script and input file templates, turning a static script into a user-configurable building block with a graphical interface within minutes. This concept enables the simple incorporation of any arbitrary software or script routinely used on HPC resources. Multiple compute backends can be configured within the client. Upon submission of a workflow, the client transfers the data *via* an SSH connection to the SimStack server on the connected remote machine. The SimStack server subsequently processes the workflow and coordinates the submission of the individual tasks *via* the local scheduling software and data transfer between the workflow elements. In order to provide broad compatibility with HPC backends, the SimStack server can communicate with all major schedulers and runs as a process of the individual user. Consequently, SimStack can be deployed and used without administrative access to the compute resource.

SimStack aims to be as simple as possible. Every workflow on the HPC resource is processed within a dedicated directory labeled with a submission timestamp and the workflow name. Every WaNo inside this workflow is referenced with a unique path in this directory and is also referenced by a unique ID (UID). The generated data remains at the HPC resource. From within the client, the user can browse this data on the remote resource within a hierarchical structure, view images, text files, and download specific files to his local machine if needed. Besides this, each WaNo can include an automated report in HyperText Markup Language (HTML), providing a concise summary of each workflow step.

2.2 SimStack Documentation

In order to guide users, documentation is made available, continuously updated, and extended at https://simstack.readthedocs.io. The documentation includes instructions on Client installation and configuration, a tutorial exploring the main SimStack features and functionalities like branching workflows and parallelizing high-throughput tasks. The developer section guides how software can be integrated into SimStack via WaNos to build custom workflows or make its own developments available to the community as a SimStack component. It furthermore provides a reference guide for the WaNo XML syntax and available tags. Beyond the documentation, the user can also find an exemplary WaNo available at https://github.com/KIT-Workflows/Test-WaNo. **Figure 2** illustrates how the XML tags of this WaNo are translated into input fields of the GUI of SimStack.

Workflows designed and pre-configured by experts can be shared with non-expert users, enabling those to conduct high-level simulations with the same quality as expert users. Additionally, all data generated can be made discoverable and accessible either in public or private repositories meeting the FAIR principles (Wilkinson et al., 2016). Finally, all the WaNos and workflows built in the SimStack framework can be extensible, locally tested, shared between researchers, and made transparent regarding their dependencies (Thompson et al., 2020). These features minimize the barrier to transferring scientific state-of-the-art modeling approaches from experts (e.g., academic researchers) to non-experts (e.g., industrial users), thereby boosting the uptake of virtual design approaches.

3 WORKFLOWS

This section illustrates the application of the SimStack framework with different workflows implemented within SimStack. Four different exemplary workflows were selected to demonstrate the broad applicability of Simstack and its main features and concepts. The Umbrella Sampling workflow computes the binding free energy of the adsorption of a molecule on surfaces by chaining structure builders, MD code, umbrella sampling, and weighted histogram analysis methods. In the Exciton Dynamics workflow, we present a multiscale simulation approach combining DFT, forcefield-based molecular modeling, and KMC approaches to generate a digital twin of OLED devices. In this workflow, we translate molecular properties to the device scale to determine their impact on the efficiency and lifetime of OLED devices. The Dihedral Scan workflow calculates the dihedral energy potential obtained from MC and DFT calculations, which can be used to parametrize forcefields for MC and MD simulations. The Emission spectra of organic molecules workflow computes fluorescent, phosphorescent, and Thermally Activated Delayed Fluorescence (TADF) molecules to determine their emission wavelength by combing DFT and TDDFT methods.

3.1 Umbrella Sampling

Knowledge about the binding free energy of molecules to different surfaces is of enormous importance in a great variety of applications from natural and engineering sciences (Wagner et al., 2021; Rauwolf et al., 2021; Bag et al., 2021). Umbrella Sampling (US) simulation (Wagner et al., 2021; Rauwolf et al., 2021; Kästner, 2011; Bag et al., 2020; Suyetin et al., 2022) is one of the widely used methods for this purpose. However, performing a US simulation to evaluate the binding free energy (to a given surface) of an arbitrary small molecule requires a complicated simulation routine as depicted in **Figure 3A**. Starting with a molecular model, one needs to first generate the forcefield parameters for the molecule. The molecular model has to be combined with the predefined surface model thereafter and the combined system has to be solvated and charge neutralized. After equilibrating this system, one has to make many copies of the system for different distances (reaction coordinate) of the molecule from the surface. Each individual system will then be subjected to an equilibration and a production run and the histograms of the reaction coordinates will be collected. In the end, all these histograms have to be analyzed using the Weighted Histogram Analysis Method (WHAM) (Kumar et al., 1992) to get the binding free energy. Therefore, we designed a workflow using the SimStack framework features, to implement the complicated US simulation routine for the calculation of binding free energy of arbitrary small molecules to predefined (silica/graphene) surfaces. The structure of the SimStack Workflow is illustrated in **Figure 3B**.

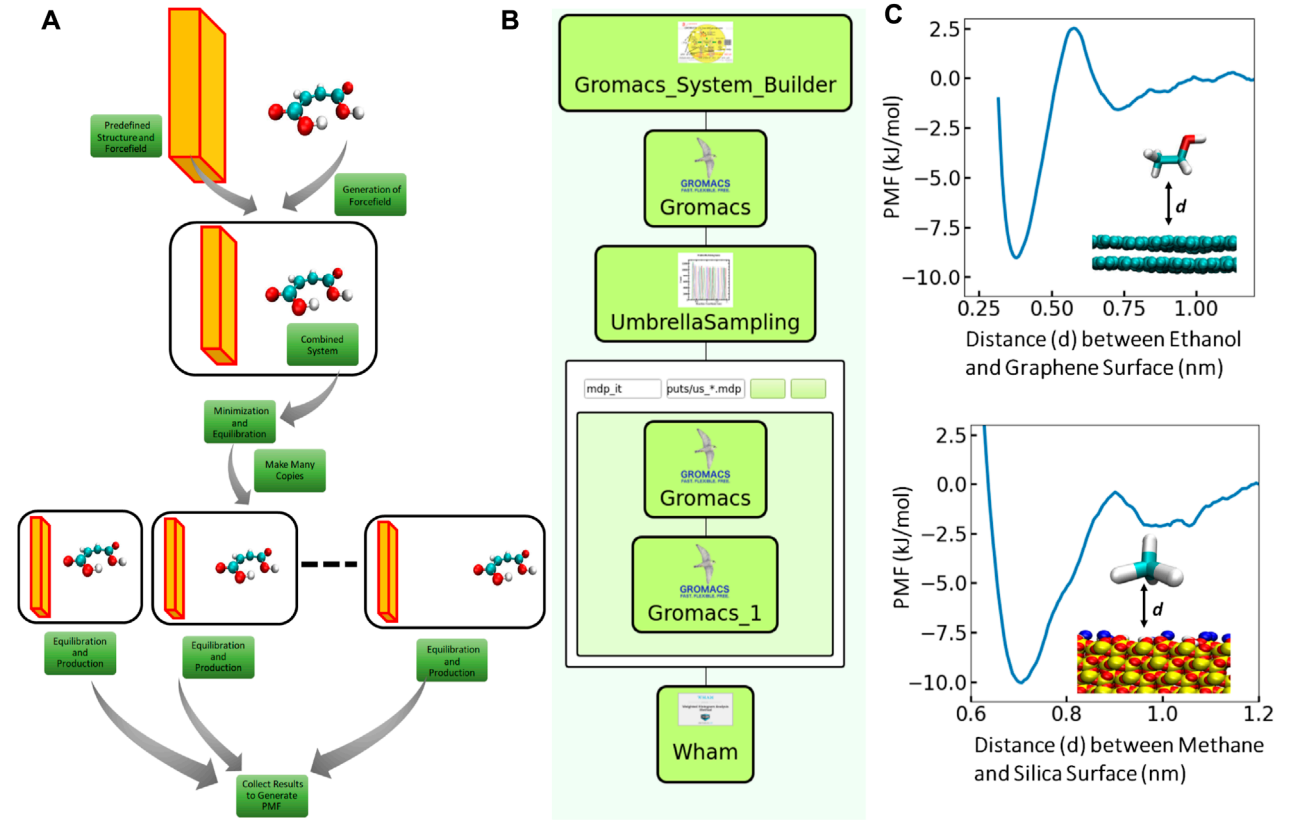


FIGURE 3 | (A) Schematic diagram illustrating potential of mean force (PMF) calculation of an arbitrary small molecule to a given surface using US. **(B)** Structure of the SimStack workflow designed to accomplish the complex PMF calculation described in **Figure 3A**. We combined four different WaNo to build the complete workflow. **(C)** PMF profiles calculated using the workflow. PMF profile of ethanol adsorbtion on Graphene are shown in the top panel and the PMF profile between methane and Silica are shown in the bottom panel. Here we have used two layers of Graphene to model the Graphene surface. The atomistic models of the small molecules (ethanol and methane) and the surfaces are shown in the inset.

Here, we combine four different WaNos: 1) GromacsSystemBuilder, 2) Umbrella Sampling 3) Gromacs and 4) Wham. The features and function of the different WaNos in this workflow are described as follows:

1) GromacsSystemBuilder: The WaNo prepares the necessary input files for a Gromacs run (Van Der Spoel et al., 2005). It takes the "pdb" file of the small molecule as input and combines it with the predefined graphene/Silica surface. To generate the forcefield (FF) parameters for the small molecule, the WaNo uses the AmberTools software package (Case et al., 2016). The FF parameters for the surface are also preloaded along with their structure. The combined system is further solvated in water and charge neutralised using standard Gromacs commands (Van Der Spoel et al., 2005). In the end all necessary input files for the Gromacs run are generated. Input: pdb (*pdb) of the small molecule. Output: Gromacs input files (*gro, *top, and *ndx).
2) Umbrella Sampling: The WaNo generates the snippet of the specific gromacs run parameter file for all the US windows. This snippet can be read by the Gromacs WaNo and run the US. The users are supposed to provide the description of the reaction coordinates as input and the WaNo creates all the Windows for the US run. Input: Description of reaction coordinates and umbrella specification. Output: All Umbrella sampling windows (with all the Gromacs input files) and their specific MD run parameter (*mdp) file.
3) Gromacs: This is simply a WaNo to run Gromacs (Van Der Spoel et al., 2005). Input: i) *gro file, ii) *top file, iii) *ndx file, iv) The Gromacs MD run parameters, v) If the gromacs run is an umbrella sampling run then the custom umbrella sampling inputs, vi) Custom forcefield files called in the *top file. Output: i) The binary run parameter file for gromacs (*tpr), ii) The equilibrated system Geometry (*gro). iii) In case of US run, the additional files for the histogram (pullf/pullx files).
4) Wham: This WaNo collects the output from the US run and generates the potential of mean force (PMF). Input: files for Histogram (pullf/pullx files) generated after US. Output: Free energy Curve.

We further use this developed workflow to calculate the free energy of binding of various small molecules to the surfaces (Graphene and Silica). In **Figure 3C** we show the PMF profile

from two such calculations: ethanol binding on Graphene and methane binding on Silica. The free energy of binding of ethanol to the Graphene is ~ 8 kJ/mol while the corresponding free energy between methane and Silica is ~ 10 kJ/mol. Although the free energy of binding is very similar for both the systems, the PMF profile for methane (to the Silica) is wider around minima which indicates strong binding affinity of methane in comparison to ethanol. It is evident from 3 (c) that ethanol can come much closer to the Graphene than methane can come to the Silica. Both of the surfaces show essentially no interaction when the molecules are more than 1 nm away from the surface.

3.2 Exciton Dynamics

Modern organic light emitting diodes (OLED) consist of multiple layers of small organic molecules (Li et al., 2017; Wong and Zysman-Colman, 2017; Lee et al., 2019; Zou et al., 2020). To achieve optimal performance and long lifetime of these devices, molecular properties of materials used in a single OLED need to be carefully aligned. While the vast chemical spaces opens the prospect of employing "perfect" material combinations in an OLED, the identification of suitable material pairings via experimental trial and error is time consuming and costly, and especially in the area of blue pixels, OLEDs have to date not been able to exploit their full potential (Scholz et al., 2015; Song and Lee, 2017). One fundamental reason for this shortcoming is the complexity connected to tuning charge carrier and exciton dynamics in OLEDs, which in turn determine efficiency and lifetime: The full system dynamics is a complex consequence of a multitude of microscopic processes (charge hops between molecules, formation of excitons, and excitonic loss processes, etc.) that are determined by microscopic molecular properties (Friederich et al., 2016; Friederich et al., 2017). Further, these properties change when molecules are embedded in thin films, depending on their exact environment, and are therefore hardly accessible experimentally (Bag et al., 2019; Li et al., 2019). To support experimental R&D by deriving fundamental understanding of how microscopic properties determine device performance by triggering and balancing a zoo of microscopic processes, we developed a multiscale simulation approach translating molecular properties to the device scale. This workflow consists of four basic steps, illustrated in **Figure 4A**. In the first step, customized force-fields are derived for all molecules involved. Subsequently, we run a simulation protocol mimicking physical vapor deposition to generate digital twins of thin films with atomistic resolution. In a third step we perform a full quantum chemical electronic structure analysis of molecules in the thin film morphology to compute molecular properties required for the simulation of charge carrier and exciton dynamics, taking into account environmental effects. Ultimately, we conduct KMC simulations in LightForge, resulting in all-particle trajectories for further analysis of the system dynamics.

To enable the efficient analysis of a variety of OLEDs with different layer setups and materials we integrated all simulation modules in the workflow platform SimStack. The full workflow for a specific OLED is constructed via drag and drop and may be saved for later re-use. **Figure 4B** depicts the workflow exemplified for a three-layer OLED, comprising a hole-transport layer (HTL), an emission layer consisting of a host material and an emitter, and an electron-transport layer (ETL): In the first layer we compute customized forcefields for all four materials using "parallel" panels. In addition to the Parametrizer module, we use the DihedralParametrizer module to account for flexibility of molecules. The outputs of each parallel panel (i.e., the forcefield files of a single material) are then passed to the respective deposit modules, where we first deposit the HTL (*Deposit3*), followed by the deposition of host and emitter of the EML (*Deposit3_1*) and the deposition of the ETL (*Deposit3_2*). In each deposition step we define the molecular input from the respective DihedralParametrizer module(s), the size of the simulation box, number of molecules to be deposited and, in the case of the EML, concentrations of the molecular mix, along with certain simulation parameters. Note that each deposited morphology is passed to the next deposition step as a substrate so that the output of the last deposition is a three-layer morphology. Using this three-layer morphology as input, we conduct two independent (and therefore parallel) *QuantumPatch* computations: We compute electronic couplings in the left panel and energy level distributions in the right panel. Both are required by LightForge to compute rate distributions for microscopic processes. For simplicity, other key quantities such as transition dipoles and further input for quenching rates are set manually in LightForge.

An output of a corresponding parametric simulation using a phosphorescent emitter is exemplified in **Figure 4C**. The left panel depicts the spatial distribution of major excitonic events, i. e. the count of exciton formation and quenching events over the device cross section. Here we see that most excitons are, as expected, generated in the EML ("recombination"). Further, we find that the major loss channel in the EML is triplet-triplet annihilation (TTA). As this process occurs at high exciton densities, we can derive from this simulation that a reduction of emitter concentration in the EML may increase efficiency. The left panel of **Figure 4C** depicts the averaged exciton life-cycle for this system. Read from the inside out, we find that almost all singlets (generated by "recombination S1") undergo a triplet conversion ("spin flip exc") before they are quenched by triplet- or polaron-quenching ("TTA" and "move chg" respectively).

In this study we implemented a multiscale workflow to simulate charge-carrier and exciton dynamics in multilayer OLEDs in the workflow platform SimStack. This workflow consists of 14 simulation modules with models for different time and length scales. A corresponding manual execution of this workflow via manual file transfer and submission of each individual module would eliminate the advantage that computer simulations pose in OLED design, as it would be time consuming and prone to errors. Instead, the implementation via SimStack provides a re-usable solution that can be adapted within minutes to various OLED setups (different number of layers, layer thicknesses, materials and material combinations, etc.) to maximize the impact of virtual design in OLED development. The exemplified output of this workflows demonstrates how this type

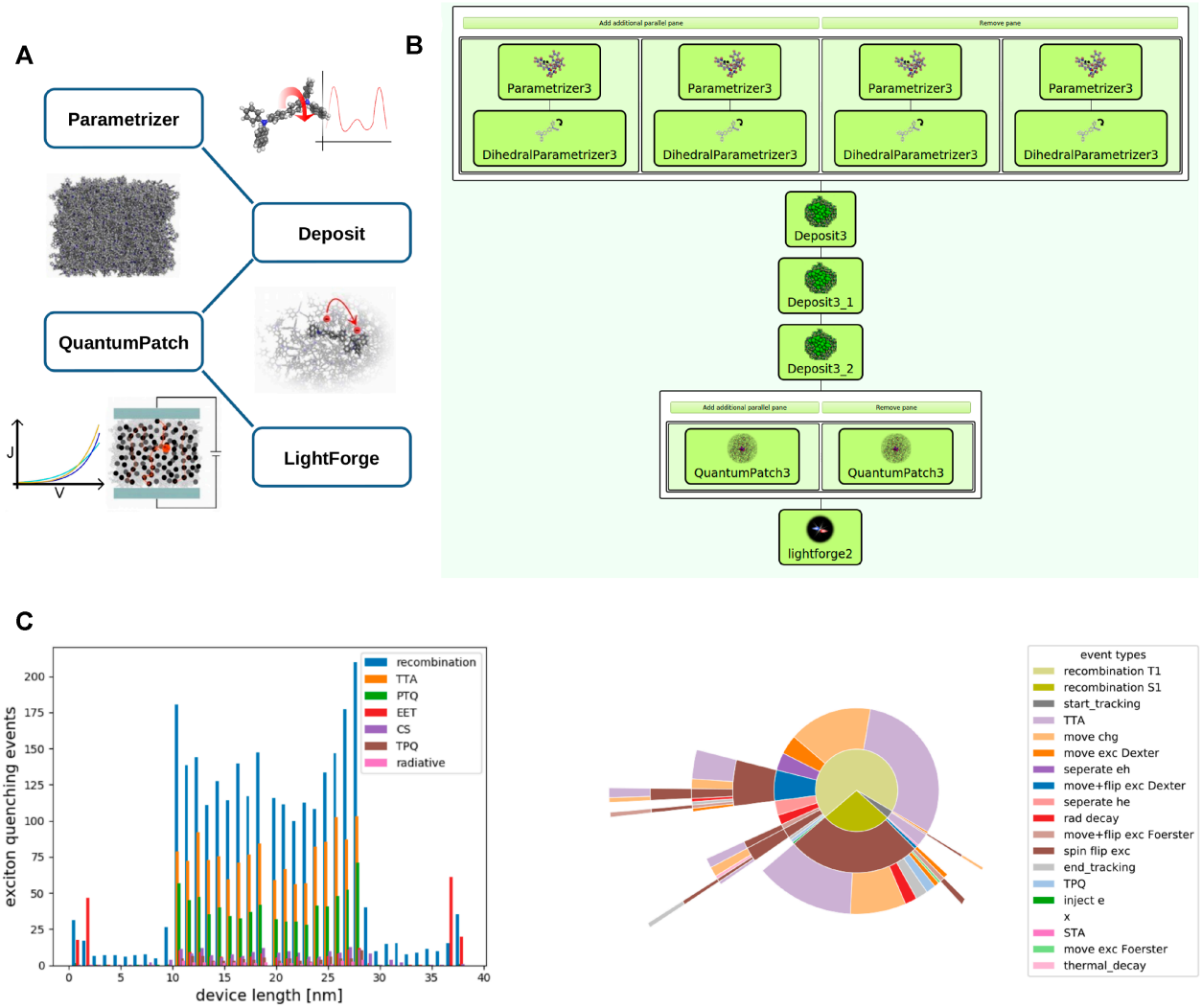


FIGURE 4 | (A) Illustration of the multiscale-OLED workflow: Parametrizer: computation of customized molecule specific force-fields; Deposit: simulation of physical vapor deposition to generate digital twins of thin films with atomistic resolution. QuantumPatch: full quantum chemical electronic structure analysis of molecules in the thin film morphology; LightForge: KMC simulations of the system dynamics. **(B)** The simulation workflow for a three-layer OLED consisting of HTL, EML (mixed guest-host system) and ETL. **(C)** Exemplified output of LightForge. Left panel: spatial distribution of major excitonic events in the three-layer OLED indicating triplet-triplet annihilation (TTA) as major loss channel. Right panel: Exciton lifecycles in this OLED.

of simulation can aid experimental R&D by deriving design rules, in this case reducing emitter concentration.

3.3 Dihedral Scan

It is imperative to perform preliminary optimization steps to generate reliable atomic models and then calculate Physico-chemical properties by applying Molecular Dynamics or Monte Carlo simulations. While it is frequent to use Quantum calculations such as DFT to obtain molecular conformations with high accuracy, depending on the molecule complexity, this approach could lead to local energy conformations. In many molecules, such as conjugated compounds, the most critical term that governs their energetic profiles are their dihedral movements, which configurations could influence their optical absorption and emission properties and their performance during MD simulations (Wildman et al., 2016). Studying different torsions for a given molecule is sensible before performing any parametrization. Dihedral scans using low-level theory calculations can determine global and local energy configurations before applying a final higher-level refinement calculation and reducing the computational cost in search of desired structures. In our recent paper (Penaloza-Amion et al., 2022) we report how the study of dihedrals using DFT scan calculations on a dimer of poly cis-transoid (4-carboxyphenyl) acetylene gave structural insights regarding the clockwise and counterclockwise helical screw-sense.

Following our previous approach, we created the Dihedral-Scan workflow (Montserrat Penaloza-Amion, 2022) (https://

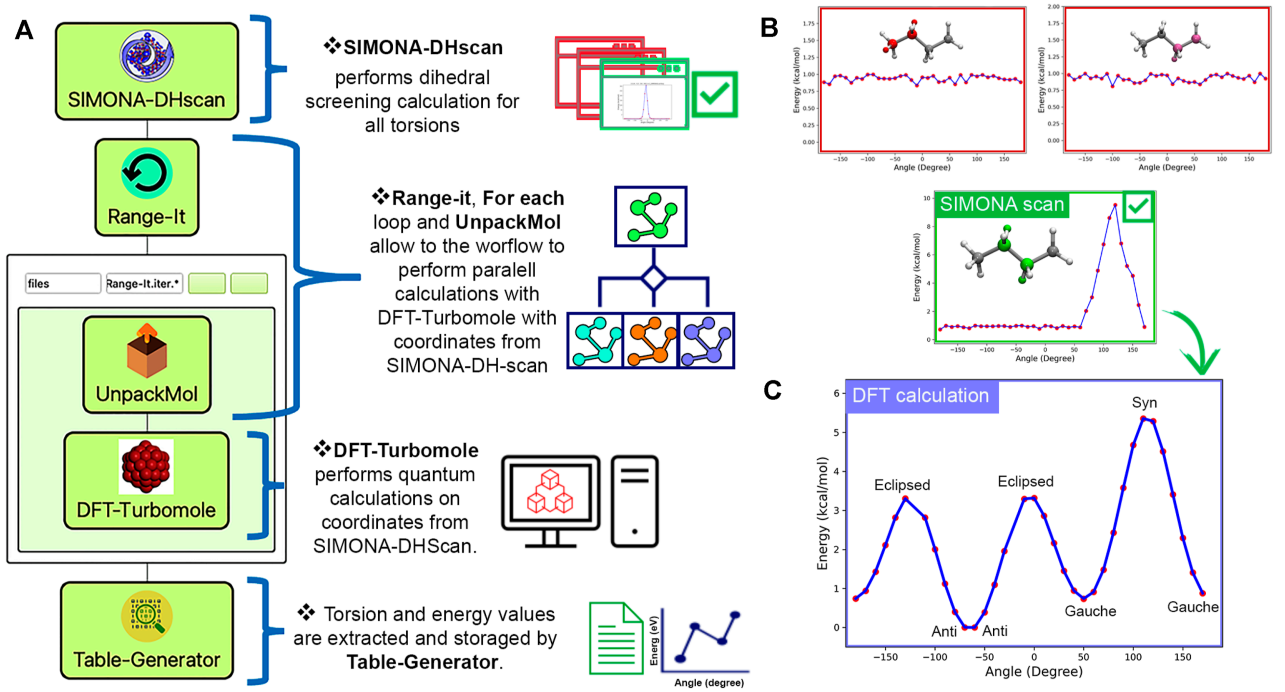


FIGURE 5 | (A) Scheme representing the dihedral scan workflow using SIMONA-DHscan, Range-It, ForEach loop, UnpackMol, DFT-Turbomole, and Table-Generator WaNos. **(B)** Dihedral profiles of n-butane calculated with SIMONA-DHscan. The best profile is highlighted in green, and the negative scores are red. **(C)** Dihedral profile output obtained after using Dihedral-Scan workflow for n-butane showing syn, eclipsed, gauche, and anti configurations.

github.com/KITWorkflows/Dihedral-Scan) to support the study of torsions for all-atom molecule models as a preliminary step for further studies such as MD or MC simulations. Our workflow consists of the following WaNos: 1) SIMONA-DHscan (Penaloza-Amion, 2022), 2) Range-It, 3) For each loop, 4)UnpackMol, 5) DFT-Turbomole and 6) Table-Generator. As shown in **Figure 5A**, the first step is to perform a dihedral screening with SIMONA-DHscan. SMILE code or structure coordinates in PDB format are allowed. Using SIMONA (Strunk et al., 2012; Penaloza-Amion et al., 2021) all possible torsions are identified, and dihedral scans on all dihedrals are performed individually. Each scan consists of the arbitrary rotation of the torsion selected and optimizing adjacent torsions using the metropolis MC algorithm. The calculation of the total energy of each configuration generated is based on the Coulomb and Lenard-Jones terms from the General Amber forcefield (GAFF) (Ozpinar et al., 2010). Parameters such as molecule net charge and rotation steps are provided manually. Finally, each scan calculation is scored by the energy difference of the maximum and minimum energies to reveal which torsion has the most significant energy influence in the tested molecule. The outputs provided are 1) scoring list and plots of all torsion profiles calculated, 2) compressed file with all the configurations for the best-scored torsion, and 3) input list with atoms Ids for best dihedral scored for further DFT calculations. The next step is to perform a high-level calculation with DFT-Turbomole 5) on all the structures provided by SIMONA-DHscan. Steps (2), (3), and 4) are needed to support the workflow in extracting the structures inputs and performing parallel calculations of each point for the dihedral profile. Finally, output data can be collected with Table-Generator WaNo to generate an out file containing the data needed to plot the final energy profile, as can be seen in **Figure 5C**.

To illustrate the Dihedral-Scan workflow, we calculated the dihedral energy profile for n-butane (**Figure 5B,C**). The n-butane structure is generated by providing a SMILE code in SIMONA-DHscan WaNo. SIMONA identified three dihedrals, providing their respective dihedral profiles (5 (b)). Each SIMONA simulation is performed using the dihedral scan protocol explained before. After the identification of the best-scored torsion (5 (b, green)), the coordinates used to generate the SIMONA dihedral profile are used to feed the quantum calculation using Range-It, UnpackMol, *ForEach* loop, and DFT-Turbomole WaNos. Each configuration was optimized using the hybrid B3LYP functional (Becke, 1993a; Becke, 1993b) and def2-SV(P) basis set (Zheng et al., 2011). The data obtained after using Table-Generator to extract the angle and total energy values indicate that our workflow can identify the torsion that has the biggest influence on the configuration of n-butane. Additionally, after the refinement calculations using DFT, the energy profile of n-butane reveals the syn, eclipsed, gauche, and anti configurations (5 (c)). Our results showed that Dihedral-Scan could identify torsions, score them, and perform quantum calculations that support future MD or MC simulations.

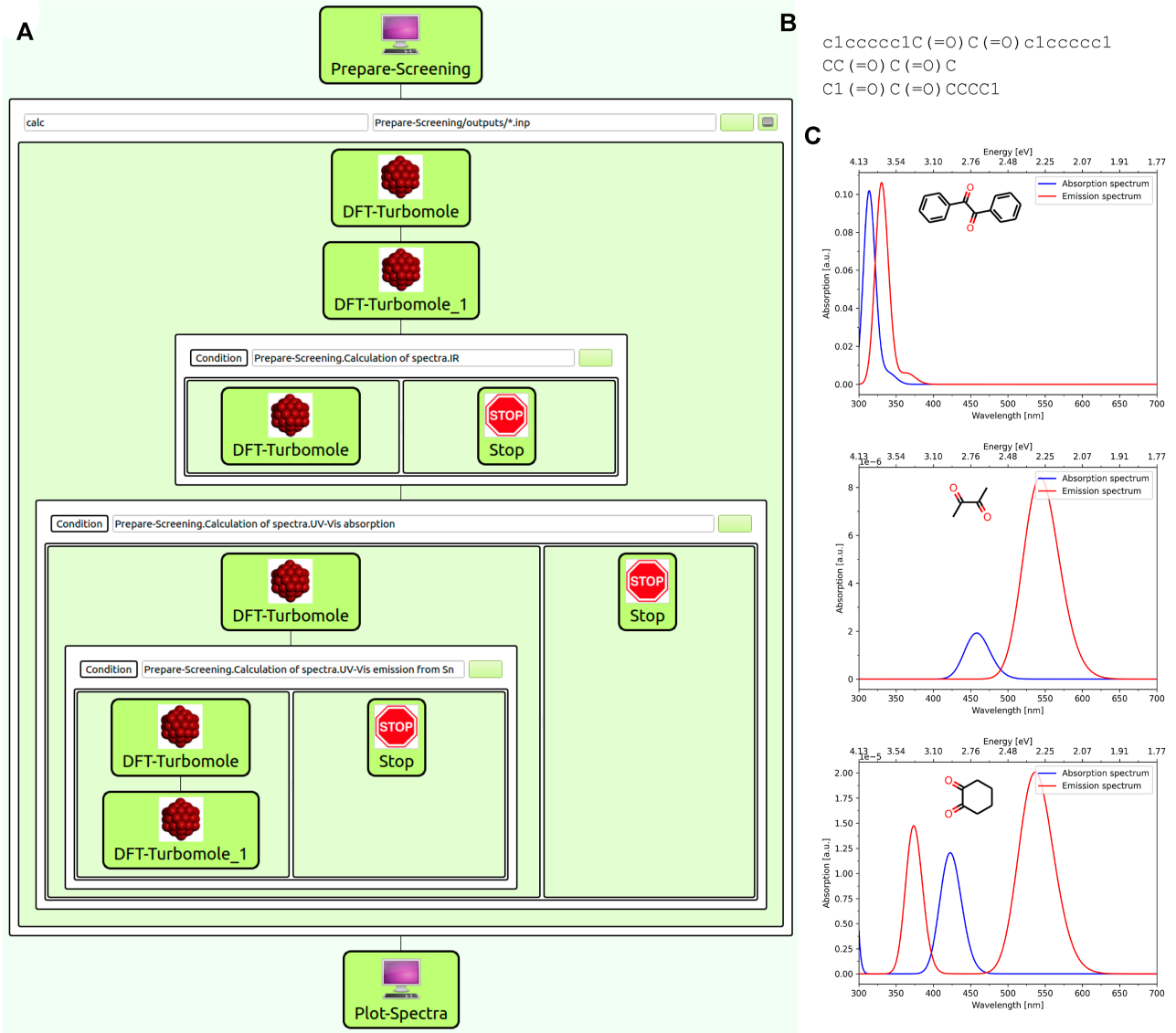


FIGURE 6 | (A) Structure of the SimStack workflow for the calculation of UV/Vis emission spectra: The "Prepare-Screening" WaNo allows to choose the types of calculated spectra and creates input files for each molecule. The Following "DFT-Turbomole" WaNos perform the actual DFT and TDDFT calculation after which all information is gathered by the final "Plot-Spectra" WaNo which creates png files for each calculated IR or UV/Vis spectrum. **(B)** SMILES input file. **(C)** Calculated UV/Vis spectra at B3LYP/def2-TZVP level.

3.4 Emission Spectra of Organic Molecules

Luminescent molecules have found widespread applications as emitter molecules in OLED devices in which the recombination of electrons an holes leads to the formation of exciton which can—after migration to an emitter molecule—relax to the ground state by emitting a photon. Several types of emitter molecules exist in so far three generation of OLEDs based on fluorescent, phosphorescent and TADF molecule with their respective advantages and drawbacks. When designing new emitter molecules, one important factor (next to other equally important ones as for example the accessibility and stability) is the emission wavelength which corresponds to the color of the molecules.

The computational procedure to determine the emission wavelength of a molecule consists of several DFT and TDDFT calculation steps including structure optimizations of the ground and first excited state as well as the calculation of electronic excitations for both optimized structures. While for most molecules, this task is a routine one for an expert on the underlying DFT code, this is in general not the case for the average user. Furthermore, the repetition of this task for a large set of molecules is time-consuming and prone to errors when done manually even by an experienced user.

We therefore developed a workflow for the execution of this procedure which requires nothing more than the structure of the molecules as an input while other parameters of the (TD) DFT calculation such as e.g., the functional, the basis set or the type of excited states can be easily adjusted if necessary. The workflow is able to loop over a large number of molecules for screening purposes, and also gives the additional option of calculating the ground state IR spectrum of each molecule.

The workflow is structured as follows: The first WaNo ("Prepare-Screening") creates input files in Gaussian style from a given list of SMILES codes or an archive file containing several structure files. These input files have the advantage over a simple xyz format of containing the desired charge and multiplicity of the molecule and therefore allow to easily calculate spectra for ions as well. After this preparatory WaNo which furthermore gives the options of choosing the type of calculated spectrum, a sequence of DFT calculations is performed using the "DFT-Turbomole" WaNo for each structure file. The first two steps consist of a preoptimization of the structures BP86/def-SV(P) level followed by an optimization at the B3LYP-D3/def2-TZVP level of theory which will be used throughout all the following calculations of the workflow. Depending on the choices made in the initial WaNo, the workflows continues with a DFT frequency analysis, the calculation of the electronic excitation spectrum, and finally an optimization of the first (n^{th}) excited state followed by an electronic excitation spectrum for the structure of the excited state. The final WaNo in the workflow ("Plot-Spectra") reads in the results from the previous Turbomole calculations which are saved in yml format and plots the Spectra. **Figure 6** shows the structure of the workflow (a) as well as an example input file containing the SMILES codes for the three geminal diones Benzil, Biacetyl and 1,2-Cyclohexanedione (b) which was used to generate the UV/Vis-spectrum plots (c).

4 CONCLUSION AND PERSPECTIVES

The presented workflow framework SimStack enables rapid prototyping of multi-module simulation workflows to design, implement, and test simulation protocols for various applications. The workflow design steps are carried out interactively via an easy-to-use flexible GUI. Simulation modules from any source are incorporated into SimStack as a simple file XML format, exposing a limited set of application-specific parameters to the end-user. This format enables computational experts and non-experts to provide a GUI for a particular application in a matter of minutes. SimStack connects to remote HPC resources and automates data transfer and execution of the simulation to and from the HPC environment. Pre-defined workflows can be saved for later re-use and transferred among users, enabling a high level of reproducibility and transferability of simulation protocols. This enables the transfer of state-of-the-art scientific simulation approaches from experts to non-experts, boosting the uptake of multiscale modeling approaches.

To show the usage of Simstack features, we presented four workflows tackling different scientific problems. These are real problems frequently found by researchers when designing simulation protocols. Those workflows make heavy use of the *ForEach*, *AdvancedFor*, *If*, and *Parallel* controls, which are responsible for branching the workflow and parallelize the simulation in the HPC resources. Currently, several WaNos and workflows using standard codes and libraries are supported in SimStack. Such as; SIMONA https://github.com/KIT-Workflows/Dihedral-Scan, VASP (Kresse and Joubert, 1999) https://github.com/KIT-Workflows/DFT-VASP, Quantum Espresso (Giannozzi et al., 2017) https://github.com/KIT-Workflows/DFT-QE, Turbomole (Ahlrichs et al., 1989) https://github.com/KIT-Workflows/DFT-Turbomole, ORCA (Neese, 2011), DFTB + (Hourahine et al., 2020) https://github.com/KIT-Workflows/DFTB-Neural-Net, Fenics (Alnaes et al., 2015) https://github.com/KIT-Workflows/Fenics-mesh, ASE (Hjorth Larsen et al., 2017), PymatGen (Ong et al., 2013), https://github.com/KIT-Workflows/Format-Converter and many in-house solutions https://github.com/KIT-Workflows/Table-Generator. All the WaNos where the same methods are embedded are code interoperable, e.g., workflow running periodic DFT simulations can run with VASP or Quantum Espresso.

Next to the available features and capabilities of SimStack, the software is continuously updated and extended. One of the main upcoming features is the capability to fully or partially restart a workflow.

DATA AVAILABILITY STATEMENT

Additional information on the workflows are available: (Umbrella Sampling) (https://github.com/KIT-Workflows/Umbrella-Sampling), (Exciton Dynamics) (https://github.com/KIT-Workflows/Exciton-Dynamics), (Dihedral Scan) (https://github.com/KIT-Workflows/Dihedral-Scan), and (Emission spectra of organic molecules) (https://github.com/KIT-Workflows/Spectrum-Screening). Documentation related to installing and tutorials of SimStack client are available on (https://www.simstack.de/) and (https://simstack.readthedocs.io/). SimStack is available at (https://www.simstack.de/) in the downloads section.

AUTHOR CONTRIBUTIONS

TS, TN, JS, WW, and CR: SimStack development team; TN and TS: WaNo development (Parametrizer3, DihedralParametrizer3, Deposit3, QuatumPatch3, and lightforge2), researching and writing of the paper; SB and JS: WaNo development (Gromacs_System_Builder, Gromacs, UmbrellaSampling, and Wham), researching and writing of the paper. TS WaNo and Workflow development (Prepare-Screening, DFT-Turbomole, Plot-Spectra, and Spectrum-Screening), writing of the paper; MP-A and CR: WaNo development (SIMONA-DHscan, Range-It, and UnpackMol), researching and writing. CR, JS, and WW: concepts organization, researching, funding acquisition, paper writing, and review.

FUNDING

JS, WW, and CR thank the German Federal Ministry of Education and Research (BMBF) for financial support of the project Innovation-Platform MaterialDigital (www.materialdigital.de) through project funding FKZ no: 13XP5094A. MP-A acknowledge the financial support by the Deutscher Akademischer Austauschdienst (DAAD) within the scholarship for Doctoral Research (91683960). TS further acknowledges funding by the "Virtual materials design" (VirtMat) initiative at KIT within the Joint Lab VMD funded by the Helmholtz society.

REFERENCES

Ahlrichs, R., Bär, M., Häser, M., Horn, H., and Kölmel, C. (1989). Electronic Structure Calculations on Workstation Computers: The Program System Turbomole. *Chem. Phys. Lett.* 162, 165–169. doi:10.1016/0009-2614(89)85118-8

Alnaes, M. S., Blechta, J., and Wells, G. N. (2015). The FEniCS Project Version 1.5. *Archive Numer. Softw.* 3, 20553. doi:10.11588/ans.2015.100.20553

Bag, S., Friederich, P., Kondov, I., and Wenzel, W. (2019). Concentration Dependent Energy Levels Shifts in Donor-Acceptor Mixtures Due to Intermolecular Electrostatic Interaction. *Sci. Rep.* 9. doi:10.1038/s41598-019-48877-9

Bag, S., Rauwolf, S., Schwaminger, S. P., Wenzel, W., and Berensmeier, S. (2021). Dna Binding to the Silica: Cooperative Adsorption in Action. *Langmuir* 37, 5902–5908. doi:10.1021/acs.langmuir.1c00381

Bag, S., Rauwolf, S., Suyetin, M., Schwaminger, S. P., Wenzel, W., and Berensmeier, S. (2020). Buffer Influence on the Amino Acid Silica Interaction. *ChemPhysChem* 21, 2347–2356. doi:10.1002/cphc.202000572

Baker, M. (2016). 1,500 Scientists Lift the Lid on Reproducibility. *Nature* 533, 452–454. doi:10.1038/533452a

Becke, A. D. (1993). A New Mixing of Hartree-Fock and Local Density-functional Theories. *J. Chem. Phys.* 98, 1372–1377. doi:10.1063/1.464304

Becke, A. D. (1993). Density-functional Thermochemistry. III. The Role of Exact Exchange. *J. Chem. Phys.* 98, 5648–5652. doi:10.1063/1.464913

Berthold, M. R., Cebron, N., Dill, F., Gabriel, T. R., Kötter, T., Meinl, T., et al. (2008). "KNIME: The Konstanz Information Miner," in *Data Analysis, Machine Learning and Applications* (Berlin: Springer Berlin Heidelberg), 319–326. doi:10.1007/978-3-540-78246-9_38

Bölle, F. T., Mathiesen, N. R., Nielsen, A. J., Vegge, T., Lastra, J. M. G., and Castelli, I. E. (2019). Autonomous Discovery of Materials for Intercalation Electrodes. *Batteries & Supercaps* 3. doi:10.26434/chemrxiv.9971054.v1

Case, D., Betz, R., Botello-Smith, W., Cerutti, D., Cheatham, T.III, Darden, T., et al. (2016). *Ambertools 16*. San Francisco: University of California.

Curtarolo, S., Setyawan, W., Hart, G. L. W., Jahnatek, M., Chepulskii, R. V., Taylor, R. H., et al. (2012). Aflow: An Automatic Framework for High-Throughput Materials Discovery. *Comput. Mater. Sci.* 58, 218–226. doi:10.1016/j.commatsci.2012.02.005

DeFever, R. S., Matsumoto, R. A., Dowling, A. W., Cummings, P. T., and Maginn, E. J. (2021). MoSDeF Cassandra: A Complete python Interface for the Cassandra Monte Carlo Software. *J. Comput. Chem.* 42, 1321–1331. doi:10.1002/jcc.26544

Friederich, P., Coehoorn, R., and Wenzel, W. (2017). Molecular Origin of the Anisotropic Dye Orientation in Emissive Layers of Organic Light Emitting Diodes. *Chem. Mater.* 29, 9528–9535. doi:10.1021/acs.chemmater.7b03742

Friederich, P., Meded, V., Poschlad, A., Neumann, T., Rodin, V., Stehr, V., et al. (2016). Molecular Origin of the Charge Carrier Mobility in Small Molecule Organic Semiconductors. *Adv. Funct. Mater.* 26, 5757–5763. doi:10.1002/adfm.201601807

Giannozzi, P., Andreussi, O., Brumme, T., Bunau, O., Buongiorno Nardelli, M., Calandra, M., et al. (2017). Advanced Capabilities for Materials Modelling with Quantum Espresso. *J. Phys. Condens. Matter* 29, 465901. doi:10.1088/1361-648x/aa8f79

Hjorth Larsen, A., Jørgen Mortensen, J., Blomqvist, J., Castelli, I. E., Christensen, R., Dułak, M., et al. (2017). The Atomic Simulation Environment-A Python Library for Working with Atoms. *J. Phys. Condens. Matter* 29, 273002. doi:10.1088/1361-648x/aa680e

Hourahine, B., Aradi, B., Blum, V., Bonafé, F., Buccheri, A., Camacho, C., et al. (2020). DFTB+, a Software Package for Efficient Approximate Density Functional Theory Based Atomistic Simulations. *J. Chem. Phys.* 152, 124101. doi:10.1063/1.5143190

Huber, S. P., Zoupanos, S., Uhrin, M., Talirz, L., Kahle, L., Häuselmann, R., et al. (2020). AiiDA 1.0, a Scalable Computational Infrastructure for Automated Reproducible Workflows and Data Provenance. *Sci. Data* 7, 300. doi:10.1038/s41597-020-00638-4

Jain, A., Ong, S. P., Chen, W., Medasani, B., Qu, X., Kocher, M., et al. (2015). FireWorks: a Dynamic Workflow System Designed for High-throughput Applications. *Concurrency Computat.: Pract. Exper.* 27, 5037–5059. doi:10.1002/cpe.3505

Janssen, J., Surendralal, S., Lysogorskiy, Y., Todorova, M., Hickel, T., Drautz, R., et al. (2019). Pyiron: An Integrated Development Environment for Computational Materials Science. *Comput. Mater. Sci.* 163, 24–36. doi:10.1016/j.commatsci.2018.07.043

Kästner, J. (2011). Umbrella Sampling. *Wires Comput. Mol. Sci.* 1, 932–942. doi:10.1002/wcms.66

Kim, D., Muckley, E. S., Creange, N., Wan, T. H., Ann, M. H., Quattrocchi, E., et al. (2021). Exploring Transport Behavior in Hybrid Perovskites Solar Cells via Machine Learning Analysis of Environmental-Dependent Impedance Spectroscopy. *Adv. Sci.* 8, 2002510. doi:10.1002/advs.202002510

Kresse, G., and Joubert, D. (1999). From Ultrasoft Pseudopotentials to the Projector Augmented-Wave Method. *Phys. Rev. B* 59, 1758–1775. doi:10.1103/physrevb.59.1758

Kumar, S., Rosenberg, J. M., Bouzida, D., Swendsen, R. H., and Kollman, P. A. (1992). The Weighted Histogram Analysis Method for Free-Energy Calculations on Biomolecules. I. The Method. *J. Comput. Chem.* 13, 1011–1021. doi:10.1002/jcc.540130812

Lasi, H., Fettke, P., Kemper, H.-G., Feld, T., and Hoffmann, M. (2014). Industry 4.0. *Bus Inf. Syst. Eng.* 6, 239–242. doi:10.1007/s12599-014-0334-4

Lee, J.-H., Chen, C.-H., Lee, P.-H., Lin, H.-Y., Leung, M.-k., Chiu, T.-L., et al. (2019). Blue Organic Light-Emitting Diodes: Current Status, Challenges, and Future Outlook. *J. Mater. Chem. C* 7, 5874–5888. doi:10.1039/C9TC00204A

Li, J., Duchemin, I., Roscioni, O. M., Friederich, P., Anderson, M., Da Como, E., et al. (2019). Host Dependence of the Electron Affinity of Molecular Dopants. *Mater. Horiz.* 6, 107–114. doi:10.1039/C8MH00921J

Li, Y., Liu, J.-Y., Zhao, Y.-D., and Cao, Y.-C. (2017). Recent Advancements of High Efficient Donor-Acceptor Type Blue Small Molecule Applied for OLEDs. *Mater. Today* 20, 258–266. doi:10.1016/j.mattod.2016.12.003

Madsen, G. K. H. (2006). Automated Search for New Thermoelectric Materials: The Case of LiZnSb. *J. Am. Chem. Soc.* 128, 12140–12146. doi:10.1021/ja062526a

Mamun, O., Winther, K. T., Boes, J. R., and Bligaard, T. (2019). High-throughput Calculations of Catalytic Properties of Bimetallic alloy Surfaces. *Sci. Data* 6, 76. doi:10.1038/s41597-019-0080-z

Mamun, O., Winther, K. T., Boes, J. R., and Bligaard, T. (2020). A Bayesian Framework for Adsorption Energy Prediction on Bimetallic alloy Catalysts. *Npj Comput. Mater.* 6. doi:10.1038/s41524-020-00447-8

Mathew, K., Montoya, J. H., Faghaninia, A., Dwarakanath, S., Aykol, M., Tang, H., et al. (2017). Atomate: A High-Level Interface to Generate, Execute, and Analyze Computational Materials Science Workflows. *Comput. Mater. Sci.* 139, 140–152. doi:10.1016/j.commatsci.2017.07.030

[Dataset] Montserrat Penaloza-Amion, C. R. C. R. (2022). Kit-workflows/dihedral. Available at: https://github.com/kit-workflows/dihedral-scan.

Mortensen, J., Gjerding, M., and Thygesen, K. (2020). Myqueue: Task and Workflow Scheduling System. *Joss* 5, 1844. doi:10.21105/joss.01844

Müller, J. M., and Däschle, S. (2018). Business Model Innovation of Industry 4.0 Solution Providers towards Customer Process Innovation. *Processes* 6, 260. doi:10.3390/pr6120260

Neese, F. (2011). The ORCA Program System. *Wires Comput. Mol. Sci.* 2, 73–78. doi:10.1002/wcms.81

Ngandjong, A. C., Lombardo, T., Primo, E. N., Chouchane, M., Shodiev, A., Arcelus, O., et al. (2021). Investigating Electrode Calendering and its Impact on Electrochemical Performance by Means of a New Discrete Element Method Model: Towards a Digital Twin of Li-Ion Battery Manufacturing. *J. Power Sourc.* 485, 229320. doi:10.1016/j.jpowsour.2020.229320

Octavio de Araujo, L., Sabino, F. P., Rêgo, C. R. C., and Guedes-Sobrinho, D. (2021). Bulk Rashba Effect Splitting and Suppression in Polymorphs of Metal Iodine Perovskites. *J. Phys. Chem. Lett.* 12, 7245–7251. doi:10.1021/acs.jpclett.1c02048

Ong, S. P., Richards, W. D., Jain, A., Hautier, G., Kocher, M., Cholia, S., et al. (2013). Python Materials Genomics (Pymatgen): A Robust, Open-Source python Library for Materials Analysis. *Comput. Mater. Sci.* 68, 314–319. doi:10.1016/j.commatsci.2012.10.028

Ozpinar, G. A., Peukert, W., and Clark, T. (2010). An Improved Generalized AMBER Force Field (GAFF) for Urea. *J. Mol. Model.* 16, 1427–1440. doi:10.1007/s00894-010-0650-7

Penaloza-Amion, M., C Rego, C. R., and Wenzel, W. (2022). Local Electronic Charge Transfer in the Helical Induction of Cis-Transoid Poly (4-carboxyphenyl) Acetylene by Chiral Amines. *J. Chem. Inf. Model.* 62, 544. doi:10.1021/acs.jcim.1c01347

[Dataset] Penaloza-Amion, M. (2022). Montserratamion/SIMONA-DHscan. Available at: https://github.com/montserratamion/simona-dhscan.

Penaloza-Amion, M., Sedghamiz, E., Kozlowska, M., Degitz, C., Possel, C., and Wenzel, W. (2021). Monte-carlo Simulations of Soft Matter Using Simona: A Review of Recent Applications. *Front. Phys.* 9, 83. doi:10.3389/fphy.2021.635959

Pizzi, G., Cepellotti, A., Sabatini, R., Marzari, N., and Kozinsky, B. (2016). AiiDA: Automated Interactive Infrastructure and Database for Computational Science. *Comput. Mater. Sci.* 111, 218–230. doi:10.1016/j.commatsci.2015.09.013

Ponce, V., Galvez-Aranda, D. E., and Seminario, J. M. (2017). Analysis of a Li-Ion Nanobattery with Graphite Anode Using Molecular Dynamics Simulations. *J. Phys. Chem. C* 121, 12959–12971. doi:10.1021/acs.jpcc.7b04190

Posada, J., Zorrilla, M., Dominguez, A., Simoes, B., Eisert, P., Stricker, D., et al. (2018). Graphics and media Technologies for Operators in Industry 4.0. *IEEE Comput. Grap. Appl.* 38, 119–132. doi:10.1109/mcg.2018.053491736

Rauwolf, S., Bag, S., Rouqueiro, R., Schwaminger, S. P., Dias-Cabral, A. C., Berensmeier, S., et al. (2021). Insights on Alanine and Arginine Binding to Silica with Atomic Resolution. *J. Phys. Chem. Lett.* 12, 9384–9390. doi:10.1021/acs.jpclett.1c02398

Rodrigues, J. F., Florea, L., de Oliveira, M. C. F., Diamond, D., and Oliveira, O. N. (2021). Big Data and Machine Learning for Materials Science. *Discov. Mater.* 1, 12. doi:10.1007/s43939-021-00012-0

Schaarschmidt, J., Yuan, J., Strunk, T., Kondov, I., Huber, S. P., Pizzi, G., et al. (2021). Workflow Engineering in Materials Design within the BATTERY 2030 + Project. *Adv. Energ. Mater.* 2021, 2102638. doi:10.1002/aenm.202102638

Schappals, M., Mecklenfeld, A., Kröger, L., Botan, V., Köster, A., Stephan, S., et al. (2017). Round Robin Study: Molecular Simulation of Thermodynamic Properties from Models with Internal Degrees of freedom. *J. Chem. Theor. Comput.* 13, 4270–4280. doi:10.1021/acs.jctc.7b00489

Scholz, S., Kondakov, D., Lüssem, B., and Leo, K. (2015). Degradation Mechanisms and Reactions in Organic Light-Emitting Devices. *Chem. Rev.* 115, 8449–8503. doi:10.1021/cr400704v

Song, W., and Lee, J. Y. (2017). Degradation Mechanism and Lifetime Improvement Strategy for Blue Phosphorescent Organic Light-Emitting Diodes. *Adv. Opt. Mater.* 5, 1600901. doi:10.1002/adom.201600901

Strunk, T., Wolf, M., Brieg, M., Klenin, K., Biewer, A., Tristram, F., et al. (2012). Simona 1.0: An Efficient and Versatile Framework for Stochastic Simulations of Molecular and Nanoscale Systems. *J. Comput. Chem.* 33, 2602–2613. doi:10.1002/jcc.23089

Suyetin, M., Bag, S., Anand, P., Borkowska-Panek, M., Gußmann, F., Brieg, M., et al. (2022). Modelling Peptide Adsorption Energies on Gold Surfaces with an Effective Implicit Solvent and Surface Model. *J. Colloid Interf. Sci.* 605, 493–499. doi:10.1016/j.jcis.2021.07.090

Thompson, M. W., Gilmer, J. B., Matsumoto, R. A., Quach, C. D., Shamaprasad, P., Yang, A. H., et al. (2020). Towards Molecular Simulations that Are Transparent, Reproducible, Usable by Others, and Extensible (TRUE). *Mol. Phys.* 118, e1742938. doi:10.1080/00268976.2020.1742938

Uhrin, M., Huber, S. P., Yu, J., Marzari, N., and Pizzi, G. (2021). Workflows in AiiDA: Engineering a High-Throughput, Event-Based Engine for Robust and Modular Computational Workflows. *Comput. Mater. Sci.* 187, 110086. doi:10.1016/j.commatsci.2020.110086

Van Der Spoel, D., Lindahl, E., Hess, B., Groenhof, G., Mark, A. E., and Berendsen, H. J. C. (2005). Gromacs: Fast, Flexible, and Free. *J. Comput. Chem.* 26, 1701–1718. doi:10.1002/jcc.20291

Wagner, R., Bag, S., Trunzer, T., Fraga-García, P., Wenzel, W., Berensmeier, S., et al. (2021). Adsorption of Organic Molecules on Carbon Surfaces: Experimental Data and Molecular Dynamics Simulation Considering Multiple Protonation States. *J. Colloid Interf. Sci.* 589, 424–437. doi:10.1016/j.jcis.2020.12.107

Warr, W. A. (2012). Scientific Workflow Systems: Pipeline Pilot and KNIME. *J. Comput. Aided Mol. Des.* 26, 801–804. doi:10.1007/s10822-012-9577-7

Wildman, J., Repiščák, P., Paterson, M. J., and Galbraith, I. (2016). General Force-Field Parametrization Scheme for Molecular Dynamics Simulations of Conjugated Materials in Solution. *J. Chem. Theor. Comput.* 12, 3813–3824. doi:10.1021/acs.jctc.5b01195

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., et al. (2016). The FAIR Guiding Principles for Scientific Data Management and Stewardship. *Sci. Data* 3, 18. doi:10.1038/sdata.2016.18

Wong, M. Y., and Zysman-Colman, E. (2017). Purely Organic Thermally Activated Delayed Fluorescence Materials for Organic Light-Emitting Diodes. *Adv. Mater.* 29, 1605444. doi:10.1002/adma.201605444

Wong-ekkabut, J., and Karttunen, M. (2016). The Good, the Bad and the User in Soft Matter Simulations. *Biochim. Biophys. Acta (Bba) - Biomembranes* 1858, 2529–2538. doi:10.1016/j.bbamem.2016.02.004

Wu, B., Widanage, W. D., Yang, S., and Liu, X. (2020). Battery Digital Twins: Perspectives on the Fusion of Models, Data and Artificial Intelligence for Smart Battery Management Systems. *Energy and AI* 1, 100016. doi:10.1016/j.egyai.2020.100016

Yao, M., Wang, Y., Li, X., Sheng, Y., Huo, H., Xi, L., et al. (2021). Materials Informatics Platform with Three Dimensional Structures, Workflow and Thermoelectric Applications. *Sci. Data* 8, 236. doi:10.1038/s41597-021-01022-6

Zheng, J., Xu, X., and Truhlar, D. G. (2011). Minimally Augmented Karlsruhe Basis Sets. *Theor. Chem. Acc.* 128, 295–305. doi:10.1007/s00214-010-0846-z

Zhu, Q., and Geng, Y. (2013). Drivers and Barriers of Extended Supply Chain Practices for Energy Saving and Emission Reduction Among Chinese Manufacturers. *J. Clean. Prod.* 40, 6–12. doi:10.1016/j.jclepro.2010.09.017

Zou, S.-J., Shen, Y., Xie, F.-M., Chen, J.-D., Li, Y.-Q., and Tang, J.-X. (2020). Recent Advances in Organic Light-Emitting Diodes: toward Smart Lighting and Displays. *Mater. Chem. Front.* 4, 788–820. doi:10.1039/C9QM00716D

Conflict of Interest: TN, TS, and WW are shareholders of Nanomatch GmbH, which intends to commercialize SimStack for commercial usage.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers | Frontiers in Materials

ORIGINAL RESEARCH
published: 31 May 2022
doi: 10.3389/fmats.2022.879614

Crossing Scales: Data-Driven Determination of the Micro-scale Behavior of Polymers From Non-homogeneous Tests at the Continuum-Scale

Víctor J. Amores[1], Francisco J. Montáns[1,2], Elías Cueto[3] and Francisco Chinesta[4]*

[1]E.T.S. de Ingeniería Aeronáutica y del Espacio, Universidad Politécnica de Madrid, Madrid, Spain, [2]Department of Mechanical and Aerospace Engineering, Herbert Wertheim College of Engineering, University of Florida, Gainesville, FL, United States, [3]Aragon Institute of Engineering Research (I3A), Universidad de Zaragoza, Zaragoza, Spain, [4]ESI GROUP Chair & PIMM Laboratory, ENSAM Institute of Technology, Paris, France

We propose an efficient method to determine the micro-structural entropic behavior of polymer chains directly from a sufficiently rich non-homogeneous experiment at the continuum scale. The procedure is developed in 2 stages: First, a Macro-Micro-Macro approach; second, a finite element method. Thus, we no longer require the typical stress-strain curves from standard homogeneous tests, but we use instead the applied/reaction forces and the displacement field obtained, for example, from Digital Image Correlation. The approach is based on the P-spline local approximation of the constituents behavior at the micro-scale (*a priori* unknown). The sought spline vertices determining the polymer behavior are first pushed up from the micro-scale to the integration point of the finite element, and then from the integration point to the element forces. The polymer chain behavior is then obtained immediately by solving a linear system of equations which results from a least squares minimization error, resulting in an inverse problem which crosses material scales. The result is physically interpretable and directly linked to the micro-structure of the material, and the resulting polymer behavior may be employed in any other finite element simulation. We give some demonstrative examples (academic and from actual polymers) in which we demonstrate that we are capable of recovering "unknown" analytical models and spline-based constitutive behavior previously obtained from homogeneous tests.

Keywords: hyperelasticity, data-driven modeling, polymers, digital image correlation, machine learning, splines

OPEN ACCESS

Edited by:
Norbert Huber,
Helmholtz-Zentrum Hereon, Germany

Reviewed by:
Alexander Lion,
Munich University of the Federal Armed Forces, Germany
Kerstin Weinberg,
University of Siegen, Germany

****Correspondence:***
Víctor J. Amores
victorjesus.amores@upm.es

Specialty section:
This article was submitted to Computational Materials Science, a section of the journal Frontiers in Materials

Received: *19 February 2022*
Accepted: *22 March 2022*
Published: *31 May 2022*

Citation:
Amores VJ, Montáns FJ, Cueto E and Chinesta F (2022) Crossing Scales: Data-Driven Determination of the Micro-scale Behavior of Polymers From Non-homogeneous Tests at the Continuum-Scale. Front. Mater. 9:879614. doi: 10.3389/fmats.2022.879614

1 INTRODUCTION

Modern applications and the easiness of 3D printing of polymers even at the micro-scale (e.g., *via* dual-photon polymerization), have renewed the interest in large deformation modeling of these entropic materials. Polymeric materials can now be found in a wide range of biomedical applications (stents, sutures, spinal cages, soft tissue implants, and tissue engineering scaffolds, …), see Bergström and Hayman (2016). Even most human soft biological tissues, which are made of a matrix (elastine, proteoglycans) plus fibers (collagen),

withstand large reversible deformations within the physiological range, and therefore, use hyperelasticity as ground for more complex aspects; Chagnon et al. (2015), Chagnon et al. (2017). As the simplest procedure to guarantee true elasticity (reversible, non-dissipative processes) the cornerstone in hyperelasticity is the free energy function, the state function, from which the stresses are uniquely derived from the strains (or vice-versa), regardless of path. Since the 3D strain energy function cannot be measured directly, the classical approach in constitutive modeling establishes a predefined form for the free energy. This function typically contains some parameters that are adjusted according to the experimentally (stress-strain) observed material behavior. Although it is relatively simple to tune model parameters to predict (up to a desirable precision) a single experimental curve, determining the parameters that produce accurate results for different modes of deformation is not trivial, as it is apparent from the unaccountable number of hyperelastic models available; Volokh (2016). Theoretically, if the proposed model is correct, this set of parameters should exist and although determined for specific tests they should predict well other modes of deformation. In practice, when parameters are obtained from a single experimental curve, they fail to generalize to other deformation states; this is the reason why in practice multiple tests are recommended to determine the parameters of the free energy function (Marckmann and Verron, 2006, 3, p.12). Using multiple tests to calibrate the parameters alleviates the deviations of the model for other modes of deformation (at the cost of accuracy for a given test), but at the same time it raises the question of whether the proposed form for the free energy really captures the physical phenomena behind experimental data or this assumed form is just a complex interpolation scheme that adapts its parameters to fit the curves used during calibration. We remark that if the physics behind were accurately represented, a single curve should be sufficient to capture the general multiaxial behavior of isotropic, incompressible polymers under reversible deformations.

This kind of problems has encouraged many researchers to pursue different approaches. One of them is the model-free data-driven computing paradigm. In this approach, basic conservation laws and essential constraints are satisfied but the constitutive laws are eliminated in the benefit of data; Kirchdoerfer and Ortiz (2016), Kirchdoerfer and Ortiz (2018), Eggersmann et al. (2019), Ibañez et al. (2017), Ibañez et al. (2018). Regarding the leading role of data for some of these references, works that address the efficient handling of data have also been published, see Zheng et al. (2020), Korzeniowski and Weinberg (2021). On the other hand, other approaches attempt to surrogate the constitutive law with an input-output relation through Artificial neural networks (ANNs), Nguyen-Thanh et al. (2020), Liu et al. (2020). Both approaches (model-free data-driven and surrogate-like ones) show promising results, however, the predictive capability of models that *just* rely on data is strongly dependent on the amount and quality of data being employed. In addition, since there is no expression for the free energy function most of those models are very difficult to interpret from a physical standpoint. It seems clear that an approach solely based on data might not be the best option for this kind of problems (the more the model needs to learn, the more data is required). This need for introducing physics information in full data-driven models has led to other works based on Physics-informed neural networks (PINNs) Liu et al. (2020) and on thermodynamically consistent data-driven approaches; González et al. (2018). Still, the increased generality of those approaches increase the amount of data required when compared to classical constitutive modeling techniques and their interpretability is much less direct. To summarize, an optimal approach for the constitutive modeling of polymers should: 1) include information about the physical equations without assuming a fixed given form for the free energy function; 2) use data to complement what we know about the physical phenomena and fill in the gaps in our knowledge, but without using more data than actually needed; 3) interpretability is also very important because understanding the solution and being able to identify its physical meaning avoids many pitfalls allowing us to search for the answer within a smaller solution space and identify spurious solutions. Interpretability also facilitates the imposition of desired (physics-based) requirements to the sought solution, for example, smoothness, monotonic increase or decrease, isotropy, etc.

With all those requirements in mind we developed the WYPiWYG (What-You-Prescribe is What-You -Get) approach to constitutive modelling, Latorre and Montáns (2013), based on some seminal ideas from the Sussman-Bathe model for isotropic, incompressible materials, Sussman and Bathe (2009). The WYPiWYG approach determines the free energy function or its contributions, but in contrast to classical phenomenological models, which presume a form for the energy function and fit the model parameters to the experimental data, our approach starts with some basic fundamental assumptions about the material behavior (isotropy/anisotropy, Valanis-Landel decomposition, invariant-based contributions to the energy function) and then obtains numerically the constitutive equation from equilibrium using a local approximation scheme based on splines. This local approximation philosophy is similar to the way shape functions in finite elements interpolate the displacement field, instead of computing coefficients of predefined analytical functions as in the Navier and Rayleigh methods. The generality of this approach is demonstrated on the models elaborated for anisotropic materials, Latorre and Montáns (2014), auxetic materials Crespo and Montáns (2018) and models for the active and passive response of skeletal muscle, Moreno et al. (2020). So far phenomenological WYPiWYG hyperelasticity circumvents the need to prescribe the shape of the energy function while maintaining the same model interpretability that the phenomenological models have. However, the amount of data required to characterize the behavior of polymeric materials is similar to the amount of data required by other parametric phenomenological models like the Ogden Model, Ogden (1972).

Other alternative to phenomenological models are those based on the micro-structure which employ additional information about the structure of polymers to get better predictive capabilities with fewer data. Most micro-structural models assume that the polymer is fully entropic and thus all the work employed in its deformation directly translates into a variation of

its entropy. This physical insight has been exploited by researches, leading to well known models as the Neo-Hookean model, Flory and Rehner (1943); Gent (1989); Treloar (1975), and the 8-chain model, Arruda and Boyce (1993). The expressions of those models depend just on the first invariant of the Green-Cauchy tensor, I_1^C and some material parameters, but in contrast to the phenomenological ones, the material parameters are linked to the micro-structure resulting in some additional physical insight. Although they were conceived to be characterized from a simple extension test, the results on other modes of deformation are not satisfactory, even with the additional information about the microstructural behavior. This fall from expectations added up to the conclusions of Mooney and its Mooney plots Mooney (1940), which showed that I_1^C was not the right (or at least the unique) variable to describe the polymer behavior due to the controversial slope C_2 that consistently appeared on simple extension experimental data.

The extension of the WYPiWYG approach to microstructural modelling with the aim of overcoming those dificulties resulted in a Macro-Micro-Macro (MMM) approach to obtain the polymer constituents behavior directly from experimental data with no assumptions about its analytical form or parameters to calibrate; Amores et al. (2020). Just some basic assumptions were made: 1) homogenization of the chains free energy to obtain the free energy of the continuum $\Psi(\lambda_1,\lambda_2,\lambda_3) = \int_S \psi_{ch}(\lambda_{ch})\, dS/S$, and 2) the computation of the micro-stretch variable is obtained from the continuum stretch tensor, $\lambda_{ch} = \boldsymbol{r}\cdot\boldsymbol{U}\cdot\boldsymbol{r}$ (a non-affine measure of deformation in agreement with the lack of relevant contribution of chains orientation change in the entropy reduction). The MMM approach predicts well any general deformation mode requiring just a single experimental curve to characterize the chain behavior, see (Amores et al., 2020, Figure 3). Since a similar Data-Driven MMM framework using the affine deformation measure $(\lambda_{ch}^C)^2 = \boldsymbol{r}\cdot\boldsymbol{C}\cdot\boldsymbol{r}$ was not able to offer the same results, in Amores et al. (2021) we questioned the affine micro stretch assumption from theoretical grounds, which seems to be the most popular in micro-structural models; Treloar (1975), Arruda and Boyce (1993), Alastrué et al. (2009), Sáez et al. (2011), Khiêm and Itskov (2016). In that same work the non-affine measure of deformation $\lambda_{ch} = \boldsymbol{r}\cdot\boldsymbol{U}\cdot\boldsymbol{r}$ did show to be in accordance with the "controversial" C_2 slope observed by Mooney that up to the date could not be successfully explained from the classical statistical theory.

The framework presented in Amores et al. (2020) seems to be in accordance with both the chain statistical theory and experimental results, but needs homogeneous tests to characterize the chain behavior. Hence, our work here is to pursue a more general approach by employing arbitrary continuum non-homogeneous tests and using Digital Image Correlation (DIC), crossing scales from the continuum to the polymer constituent macromolecules.

The procedure consists of linking two stages. One is the previously introduced MMM method, and the other one is to link that method to a finite element analysis of non-homogeneous continuum problems continuum problems, see (Cite to **Figure 1**) outline. We assume in the latter that the non-homogeneous field of displacements (*via* DIC), plus the test loads (*via* load cell) are known, the input data could be either 1D, 2D or 3D depending on the case, but it is important to note that in 2D and 1D, it should be possible to employ reasonable assumptions to determine the principal stretches and stresses in the eliminated directions (incompressibility plus plane stress allow to determine both the stretch and the stress out of the plane just from the plane information). Then, the polymer chain behavior is modeled by P-splines, which vertices are to be determined—P-splines are penalyzed interpolating B-splines to guarantee smoothness; see Eilers and Marx (2021). That structure (the unknown vertices) are transferred to the continuum scale *via* integration in all the material directions and the result attached to the finite element integration point (the continuum constitutive behavior). Hence, the nodal forces of the finite element are set as a direct, explicit function of the unknown P-spline vertices of the polymer chains and the prescribed deformation gradient. By a least squares formulation, a *linear* system of equations is established, which allows for the immediate determination (i.e., simply solving a linear system of equations) of the P-spline vertices of the chain behavior from the macroscopic loads in the specimen and the macroscopic field of deformation. The physics equations present on the procedure include at the FEM level the compatibility equations (computing the strain quantities from the displacement field) and the equilibrium equations (null force residual), incompressible hyperelasticity with volumetric-deviatoric decoupling at the integration point level, properly including the micro-macro connections (energy homogenization and affine micro-stretch) at the chain level.

In the following sections we introduce the procedure, first using a continuum hyperelastic formulation and then the micromechanical one. We also demonstrate the applicability through an academic example and an example using the well-known Treloar's rubber.

2 METHODOLOGY

2.1 Detailed Procedure Description

One way to obtain the displacement-based finite element formulation is through the principle of virtual work, which for the quasi-static case reads $\delta W = \delta W_{\text{int}} - \delta W_{\text{ext}} = 0$. Considering a conventional FEM discretization and interpolation of the displacement field, both virtual works (internal and external) could also be expressed in terms of the internal and external nodal forces, $\delta W = \delta\boldsymbol{u}\cdot(\boldsymbol{f}_{\text{int}} - \boldsymbol{f}_{\text{ext}}) = 0$ or equivalently $\boldsymbol{f}_{\text{int}} = \boldsymbol{f}_{\text{ext}}$. To conclude, a weighted integration of the equilibrium equation leads to an alternative expression for the virtual work, see (**Eq. 1**), that if compared with $\delta W = \delta\boldsymbol{u}\cdot(\boldsymbol{f}_{\text{int}} - \boldsymbol{f}_{\text{ext}}) = 0$ provides an expression for the internal and external nodal forces. The expression for the weighted integration of the equilibrium equation in the reference configuration can be expressed as:

$$\int_\Omega [\delta\boldsymbol{u}\otimes\nabla] : \boldsymbol{P}(\boldsymbol{u})\, d\Omega = \int_\Omega \delta\boldsymbol{u}\cdot\boldsymbol{b}\, d\Omega + \int_\Gamma \delta\boldsymbol{u}\cdot\boldsymbol{t}\, d\Gamma \qquad (1)$$

where $\delta\boldsymbol{u}$ and $\boldsymbol{u}$ are the virtual displacement field, and the displacement field, respectively, $\boldsymbol{P}$, the 1st PK (First Piola-Kirchhoff) stress tensor, $\boldsymbol{b}$ the volumetric forces and $\boldsymbol{t}$ the

surface forces. The symbol ⊗ represents the dyadic production, so $[\delta \boldsymbol{u} \otimes \nabla] : \boldsymbol{P}(\boldsymbol{u}) = \boldsymbol{P} : \nabla \delta \boldsymbol{u}$ is the internal virtual work density.

2.1.1 Internal Force Term

As it has already been mentioned, a FEM discretization for the internal force term, $\int_\Omega \boldsymbol{P} : \nabla \delta \boldsymbol{u} d\Omega$, and the interpolation of the displacement field in the reference unit element, $\delta \boldsymbol{u} = \sum_{a=1}^{n_n} h^a\left({}^0\boldsymbol{\xi}\right) \delta \boldsymbol{u}^a$ leads to the expression for the nodal internal forces, see (**Eq. 13**). Note that $h^a\left({}^0\boldsymbol{\xi}\right)$ are the shape functions, $\delta \boldsymbol{u}^a$ the virtual displacement vector for the node a, and that in δu_i^a, $a = 1, \ldots, n_n$ and i indicates the spatial dimension ($i = 1, 2$ in 2D). The expressions for the internal forces are described with depth underneath, (**Eq. 2**) and (**Eq. 3**):

$$\begin{aligned}\int_\Omega \boldsymbol{P} : \nabla \delta \boldsymbol{u} d\Omega &= \sum_{e=1}^{n_{el}} \int_{\Omega e} \boldsymbol{P} : \nabla \delta \boldsymbol{u} d\Omega_e \\ &= \sum_{e=1}^{n_{el}} \int_{\square} \boldsymbol{P} : \nabla \delta \boldsymbol{u} J^e d\square \\ &= \sum_{e=1}^{n_{el}} \sum_{j=1}^{n_{qp}} \boldsymbol{P}_j : \nabla \delta \boldsymbol{u}_j J_j^e w_j \\ &= \sum_{e=1}^{n_{el}} \sum_{j=1}^{n_{qp}} \sum_{a=1}^{n_n} \boldsymbol{P}_j : \left(\delta \boldsymbol{u}^a \otimes \nabla h^a\left({}^0\boldsymbol{\xi}_j\right)\right) J_j^e w_j \end{aligned} \quad (2)$$

where all the variables with subscript j are computed in the integration point j. The previous equations can be rewritten doing the sum over the DOF of the element ($n_{dofs} = n_n \times 2$ in the 2D solid elements) instead of doing the sum over the nodes:

$$\int_\Omega \boldsymbol{P} : \nabla \delta \boldsymbol{u} d\Omega = \sum_{e=1}^{n_{el}} \sum_{j=1}^{n_{qp}} \sum_{i=1}^{n_{\text{dofs}}} \delta u_i \boldsymbol{P}_j : \nabla \boldsymbol{h}_j^i J_j^e w_j, \quad (3)$$

In **Eq. 3**, i is the index that runs through the local degrees of freedom in the element, for the cases studied here (2D plane stress problems) $i = 1, \ldots, 2n_n$, δu_i is the virtual displacement at the local degree of freedom i and $\nabla \boldsymbol{h}_j^i$ is a second order tensor that projects the contribution of $\boldsymbol{P}_j$ to the *ith* local degree of freedom. Regarding $\nabla \boldsymbol{h}_j^i$, for $i = 2$, $\nabla \boldsymbol{h}_j^2$ is the projector for the second local degree of freedom of the element, this DOF corresponds to the first node of the element $a = 1$ and the second dimension 2, therefore, $\nabla \boldsymbol{h}_j^2 = \boldsymbol{e}_2 \otimes \nabla h^1\left({}^0\boldsymbol{\xi}_j\right)$. If $i = 3$ instead, $\nabla \boldsymbol{h}_j^3$ is the projector for the third local degree of freedom of the element, which corresponds to the second node of the element $a = 2$ and the first dimension 1, therefore, $\nabla \boldsymbol{h}_j^3 = \boldsymbol{e}_1 \otimes \nabla h^2\left({}^0\boldsymbol{\xi}_j\right)$. Now **Eq. 3** is rewritten in matrix form to identify the components of the internal force vector:

$$\begin{aligned}\sum_{e=1}^{n_{el}} \sum_{j=1}^{n_{qp}} \sum_{i=1}^{n_{\text{dofs}}} \delta u_i \boldsymbol{P}_j : \nabla \boldsymbol{h}_j^i J_j^e w_j &= \sum_{e=1}^{n_{el}} \begin{bmatrix} \left(f_{\text{int}}^e\right)_1 & \cdots & \left(f_{\text{int}}^e\right)_{n_{\text{dofs}}} \end{bmatrix} \begin{bmatrix} \delta u_1^e \\ \vdots \\ \delta u_{n_{\text{dofs}}}^e \end{bmatrix} \\ &= \sum_{e=1}^{n_{el}} \boldsymbol{f}_{\text{int}}^e \cdot \delta \boldsymbol{u}^e \end{aligned} \quad (4)$$

where

$$\left(f_{\text{int}}^e\right)_i = \sum_{j=1}^{n_{qp}} \boldsymbol{P}_j : \nabla \boldsymbol{h}_j^i J_j^e w_j \quad (5)$$

Since the PK1 tensor is a two-leg tensor placed in 2 configurations at the same time (material in the right and spatial in the left), it might be more suitable to rewrite the term $\boldsymbol{P}_j : \nabla \boldsymbol{h}_j^i$ in terms of the PK2 (Second Piola–Kirchhoff) stress tensor, $\boldsymbol{S}$, which lies completely in material configuration, $\boldsymbol{P} = \boldsymbol{X}\boldsymbol{S}$, being $\boldsymbol{X} = \partial^t \boldsymbol{x} / \partial^0 \boldsymbol{x}$ the deformation gradient:

$$\begin{aligned}\boldsymbol{P}_j : \nabla \boldsymbol{h}_j^i = \left(\boldsymbol{X}_j \boldsymbol{S}_j\right) : \nabla \boldsymbol{h}_j^i = \boldsymbol{S}_j : \boldsymbol{X}_j^T \nabla \boldsymbol{h}_j^i = \boldsymbol{S}_j : \text{sym}\left(\boldsymbol{X}_j^T \nabla \boldsymbol{h}_j^i\right) \\ = \text{sym}\left(\boldsymbol{X}_j^T \nabla \boldsymbol{h}_j^i\right) : \boldsymbol{S}_j \end{aligned} \quad (6)$$

In order to simplify the notation we define $\boldsymbol{l}_j^i := \text{sym}\left(\boldsymbol{X}_j^T \nabla \boldsymbol{h}_j^i\right)$, on the other hand the double contraction will be computed using the Voigt notation. Note that $S_3 = 0$ since we are considering the case of plane stress and that the components in the Voigt notation are with respect to the principal directions of deformation/stress. Then, we write in principal directions (denoted as the X_{ppal} system of representation)

$$\boldsymbol{P}_j : \nabla \boldsymbol{h}_j^i = \begin{bmatrix} l_{11} & l_{22} & 2l_{12} \end{bmatrix}_{j,X_{\text{ppal}}}^i \begin{bmatrix} S_1 \\ S_2 \\ 0 \end{bmatrix}_{j,X_{\text{ppal}}} \quad (7)$$

To compute the principal stress components of a polymeric and quasi-incompressible material, we assume that the volumetric and deviatoric contributions of the energy can be separated (a typical assumption in quasi-incompressibility), $\Psi(\boldsymbol{A}) = \mathcal{U}(J) + \mathcal{W}\left(\lambda_1^d, \lambda_2^d, \lambda_3^d\right)$, where the volumetric term $\mathcal{U}(J)$ is just a penalization function, being $J = \det(\boldsymbol{X}) = d^t V / d^0 V$ the volume ratio, and $\boldsymbol{A} = 1/2\left(\boldsymbol{X}^T \boldsymbol{X} - \boldsymbol{I}\right)$ the Green-Lagrange strain tensor. A typical choice for the penalization could be $\mathcal{U}(J) = \kappa/2\,(J-1)^2$ with the bulk modulus, κ, selected such that the stress produced by a volumetric deformation grows rapidly. If an estimation for the shear modulus of the material, μ, is known, $\kappa/\mu \sim 10^4$, might suffice to ensure the satisfaction of the quasi-incompressibility condition $J = \lambda_1 \cdot \lambda_2 \cdot \lambda_3 \approx 1$. The decoupling of the energy in a volumetric and deviatoric energy results in a similar decoupling for the PK2 stress tensor, $\boldsymbol{S} = \boldsymbol{S}^v + \boldsymbol{S}^d$. We note that $\boldsymbol{S}^v$ and $\boldsymbol{S}^d$ are not the volumetric and deviatoric part, respectively, of a second order tensor $\boldsymbol{S}$ (as obtained from the respective mathematical operators), but $\boldsymbol{S}^v$ is the contribution to $\boldsymbol{S}$ that comes from the volumetric energy $\mathcal{U}(J)$, and $\boldsymbol{S}^d$ is the contribution to $\boldsymbol{S}$ that comes from the deviatoric energy, $\mathcal{W}(\lambda_1^d, \lambda_2^d, \lambda_3^d)$:

$$\boldsymbol{S} = \frac{d\Psi}{d\boldsymbol{A}} = \frac{d\mathcal{U}}{dJ}\frac{dJ}{d\boldsymbol{A}} + \sum_{i=1}^{3} \frac{\partial \mathcal{W}}{\partial \lambda_i^d}\frac{d\lambda_i^d}{d\boldsymbol{A}} \quad (8)$$

$$= \boldsymbol{S}^v + \boldsymbol{S}^d = pJ\boldsymbol{C}^{-1} + \sum_i \sum_k \frac{\mathcal{W}_k J^{-1/3}}{\lambda_i^2}\left(\lambda_i \delta_{ik} - \frac{1}{3}\lambda_k\right)\boldsymbol{N}_i \otimes \boldsymbol{N}_i \quad (9)$$

where $\boldsymbol{N}_i$ are the principal referential directions of deformation, $\mathcal{W}_k = \partial \mathcal{W} / \partial \lambda_k^d$ and $\lambda_i^d = \lambda_i J^{-1/3}$, the isochoric stretches. As we have already mentioned the test cases are plane stress and therefore, $S_3 = 0 = \boldsymbol{S} : (\boldsymbol{N}_3 \otimes \boldsymbol{N}_3)$:

$$S_3 = \frac{pJ}{\lambda_3^2} + \frac{2}{3}\frac{\mathcal{W}_3 J^{-1/3}}{\lambda_3} - \frac{1}{3}\left(\frac{\mathcal{W}_1 J^{-1/3}}{\lambda_3^2}\lambda_1 + \frac{\mathcal{W}_2 J^{-1/3}}{\lambda_3^2}\lambda_2\right) = 0 \quad (10)$$

From the previous equation, the pressure term could be isolated and introduced in the equations for S_1 and S_2:

$$S_j = J^{-2/3}\frac{1}{\lambda_j^d}\left(\frac{\partial \mathcal{W}}{\partial \lambda_j^d} - \frac{\lambda_3^d}{\lambda_j^d}\frac{\partial \mathcal{W}}{\partial \lambda_3^d}\right) \quad \text{for } j = 1,2 \tag{11}$$

Once the principal stretches are known for a particular integration point, the principal PK2 stress components in that same point could be determined through evaluation of the deviatoric contribution derivatives. When the stress tensor has been computed in all the integration points of the element, the internal nodal forces for that element are computed using (**Eq. 4**). On the other hand, if the functions for the deviatoric contribution derivatives are unknown, a cubic P-Spline local approximation can be employed, see Amores et al. (2019), Eilers and Marx (2021). The B-splines (or its penalized version, P-splines) are one of the approaches for expressing any general function $y(x)$ as the product of a set of known basis functions (cubic in this case), $B_i(x)$ and it's *a priori* unknown corresponding weights (also called vertices), $\hat{v}_i$, if the number of vertices of the B-Spline is n_{vert}, the expression for the unknown function is a scalar product, $y(x) = \sum_{i=1}^{n_{\text{vert}}} B_i(x)\,\hat{v}_i = \boldsymbol{B}(x)\cdot\hat{\boldsymbol{v}}$. In contrast to the basis of features proposed in Flaschel et al. (2021), where a basis of preassumed functions is used, the local B-Spline basis is general with no assumptions about the possible function expressions (the only assumption would be that locally the function is at most a cubic function when cubic B-Splines are used). Additionally, the local P-Spline approximation in this work is typically performed directly on the derivatives so there is no need to compute derivatives of the approximated function. With the mentioned P-Spline for the derivative of the energy function, the principal stresses could then be represented as $S_i\left(\lambda_1^d,\lambda_2^d\right) = \boldsymbol{S}_{\text{row}}^i\left(\lambda_1^d,\lambda_2^d\right)\cdot\hat{\boldsymbol{v}}$ where $\boldsymbol{S}_{\text{row}}^i$ is a vector defined with 1D P-Splines basis vector ($\boldsymbol{B}\left(\lambda_1^d\right)$, $\boldsymbol{B}\left(\lambda_2^d\right)$), see **Section 3.1** and **Section 3.2** for specific expressions of this vector. Therefore, depending on the approach that is used to compute the continuum principal stresses, the unknown vertices, $\hat{\boldsymbol{v}}$, could represent the derivative of the energy either on the macro-structure or in the micro-structure:

$$\begin{aligned}
\boldsymbol{P}_j : \nabla\boldsymbol{h}_j^i &= \text{sym}\left(\boldsymbol{X}_j^T\nabla\boldsymbol{h}_j^i\right) : \\
\boldsymbol{S}_j &= \begin{bmatrix} l_{11} & l_{22} & 2l_{12}\end{bmatrix}_{j,X_{\text{ppal}}}^{i}\begin{bmatrix}\left[\boldsymbol{S}_{\text{row}}^1\left(\lambda_1^d,\lambda_2^d\right)\right]^T \\ \left[\boldsymbol{S}_{\text{row}}^2\left(\lambda_1^d,\lambda_2^d\right)\right]^T \\ [0]^T\end{bmatrix}_j[\hat{\boldsymbol{v}}] \\
&= \left(l_{11}\boldsymbol{S}_{\text{row}}^1\left(\lambda_1^d,\lambda_2^d\right) + l_{22}\boldsymbol{S}_{\text{row}}^2\left(\lambda_1^d,\lambda_2^d\right)\right)_j^i\cdot\hat{\boldsymbol{v}} \\
&= \left({}^*\boldsymbol{S}_{\text{row}}\right)_j^i\left(\lambda_1^d,\lambda_2^d\right)\cdot\hat{\boldsymbol{v}}
\end{aligned} \tag{12}$$

and for the element local degree of freedom i

$$\left(f_{\text{int}}^e\right)_i = \left[\sum_{j=1}^{n_{qp}} J_j^e w_j \left({}^*\boldsymbol{S}_{\text{row}}\right)_j^i\left(\lambda_1^d,\lambda_2^d\right)\right]\cdot\hat{\boldsymbol{v}} \tag{13}$$

which means that every component of the elemental nodal force vector is obtained as the product of a known row multiplied by the unknown vertices, or expressed in a different manner:

$$\boldsymbol{f}_{\text{int}}^e = \text{F}_{\text{int}}^e\cdot\hat{\boldsymbol{v}} \tag{14}$$

where F_{int}^e is a known matrix, given in square brackets in **Eq. 13**.

2.1.2 External Forces Term

In case the body forces and the tractions are not zero, the external force vector for each element, $\boldsymbol{f}_{\text{ext}}^e$ has also to be computed and assembled into $\boldsymbol{f}_{\text{ext}}$, looking at the right hand side of (**Eq. 1**).

$$\left(f_{\text{ext}}^e\right)_i = \sum_{j=1}^{n_{qp}}\boldsymbol{h}_j^i\cdot\boldsymbol{b}J_j^e w_j + \sum_{s=1}^{n_{sbound}^e}\sum_{j=1}^{n_{qps}}\overline{\boldsymbol{h}}_j^i\cdot\boldsymbol{t}J_j^s w_j^s \tag{15}$$

$$\boldsymbol{f}_{\text{ext}} = \bigwedge_{e=1}^{nel}\boldsymbol{f}_{\text{ext}}^e \tag{16}$$

On the previous equations, the reaction forces on the boundary where displacements are imposed are accounted in the traction term, for the sake of simplicity we are going to consider that those nodal reaction forces are known, while this is not typically the case in a experimental setting, instead, the total reaction forces are known rather than the nodal forces. To deal with this fact, 2 different approaches can be followed: 1) take another artificial boundary far enough from the original one in which we can suppose that the reaction force is evenly distributed according to the Saint-Venant's principle or 2) consider two independent set of equations one for the free dofs and other for the DOF with fixed displacements, in the fixed displacement DOF the resultant of the internal forces equals the reaction force at each of the boundaries.

2.1.3 System of Linear Equations for the P-Spline Vertices

The problem to solve is an overdetermined linear system of equations $\text{F}_{\text{int}}\cdot\hat{\boldsymbol{v}} = \boldsymbol{f}_{\text{ext}}$, where $\hat{\boldsymbol{v}}$ are the unknown vertices of the P-Spline that approximates the derivative of the energy function at the macro-scale (phenomenological energy function) or the derivative of the energy function of the constituents at the micro-scale. The solution process consists of finding the vertices, $\hat{\boldsymbol{v}}$, that minimizes the mean square error, MSE, in the force residual:

$$MSE(\hat{\boldsymbol{v}}) = \left(\text{F}_{\text{int}}\cdot\hat{\boldsymbol{v}} - \boldsymbol{f}_{\text{ext}}\right)\cdot\left(\text{F}_{\text{int}}\cdot\hat{\boldsymbol{v}} - \boldsymbol{f}_{\text{ext}}\right) \tag{17}$$

The solution for the previous minimization problem is analytical and result in the mentioned linear system of equations:

$$\hat{\boldsymbol{v}} = \boldsymbol{A}_{\text{sys}}^{-1}\boldsymbol{b}_{\text{sys}};\quad \boldsymbol{A}_{\text{sys}} = \text{F}_{\text{int}}^T\text{F}_{\text{int}},\ \boldsymbol{b}_{\text{sys}} = \text{F}_{\text{int}}^T\boldsymbol{f}_{\text{ext}} \tag{18}$$

As we have already mentioned, one of the advantages of using the P-Spline-based local approximation is that although we do not assume the form of the energy function, additional requirements can be added to the solution, a typical one is smoothness, which can be translated into a penalization on the second order finite differences of the P-spline vertices, if the solution obtained is not monotonically increasing, this property could also be imposed

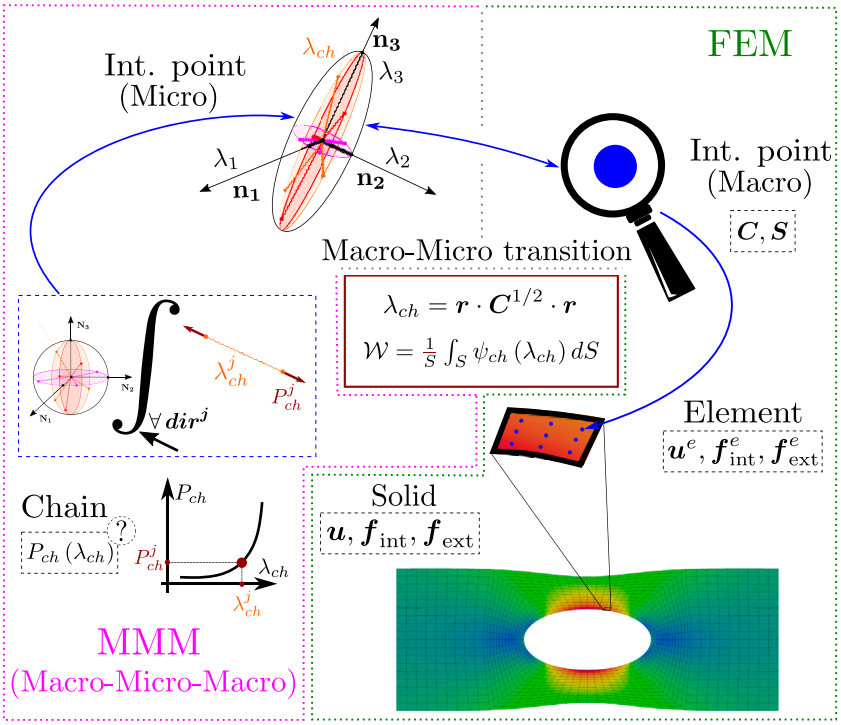


FIGURE 1 | On the right side the FEM part (green contour), from bottom to top there are different levels, first at the solid level, the nodal displacement vector, $\boldsymbol{u}$, the nodal internal forces, $\boldsymbol{f}_{\text{int}}$, and the external nodal forces, $\boldsymbol{f}_{\text{ext}}$, are encountered. Second, at the element level, the elemental nodal displacement vector, $\boldsymbol{u}^e$, the elemental nodal internal forces, $\boldsymbol{f}^e_{\text{int}}$, and the elemental external nodal forces, $\boldsymbol{f}^e_{\text{ext}}$. Finally, the integration point (at the macro-scale) characterized by the Right Green Cauchy tensor $\boldsymbol{C}$, and the PK2 stress tensor, $\boldsymbol{S}$, at this point. On the left hand side the Macro-Micro-Macro approach (magenta contour), the general idea would be "zooming" (hence the magnifying glass) on the integration point and consider that for polymeric materials, the chains oriented in a certain direction (representative direction) suffer a deformation described by the continuum stretches ellipsoid (upper left). The *a priori* unknown chain mechanical behavior, $P_{ch}(\lambda_{ch})$ (bottom left), will be evaluated for each direction and the effect in each representative direction will be integrated for all the representative directions on the reference unit micro-sphere (center left) to obtain the continuum principal stress components. Finally, the Macro-Micro transition (center of the figure) dictates how the continuum energy is obtained from the constituents free energy and how the micro-structural stretches (non-affine chain stretch) are obtained from the continuum deformation measures. From a more practical perspective: 1) The Macro-Micro-Macro approach expresses the continuum principal stresses in terms of the unknown P-Spline vertices for the constituents mechanical behavior, $\boldsymbol{S}(\hat{\boldsymbol{v}})$, and 2) A classical FEM assembly process is employed to write nodal equilibrium equations from the information at the integration points, $\mathrm{F}_{\text{int}} \cdot \hat{\boldsymbol{v}} = \boldsymbol{f}_{\text{ext}}$. The substeps involved are: 1.1) The continuum strain, $\boldsymbol{C}$, tensor defines the micro-structural deformations, λ_{ch} on the representative directions; 1.2) The micro-mechanical behavior (unknown *a priori*) is evaluated in all the representative directions in terms of the unknown vertices $\hat{\boldsymbol{v}}$, $P_{ch}(\lambda_{ch}) = \boldsymbol{B}_{row}(x) \cdot \hat{\boldsymbol{v}}$; 1.3) The constituents mechanical behavior in all directions is integrated over the unit micro-sphere to obtain the continuum principal stresses at the integration point, $\boldsymbol{S}(\hat{\boldsymbol{v}})$; 2.1) The integration point stress $\boldsymbol{S}(\hat{\boldsymbol{v}})$ is integrated over the element volume to obtain the elemental nodal internal forces, $\boldsymbol{f}^e_{\text{int}}(\hat{\boldsymbol{v}})$; 2.2) The elemental nodal internal forces are assembled to obtain the nodal internal forces of the solid, $\boldsymbol{f}_{\text{int}}(\hat{\boldsymbol{v}})$; 2.3) The unknown vertices are determined from equilibrium, $\boldsymbol{f}_{\text{int}}(\hat{\boldsymbol{v}}) = \mathrm{F}_{\text{int}} \cdot \hat{\boldsymbol{v}} = \boldsymbol{f}_{\text{ext}}$.

by iterative penalization on the vertices that do not satisfy the condition see Amores et al. (2019), Eilers and Marx (2021). With all the penalizations, the general system of equations to solve would be:

$$\begin{aligned} \hat{\boldsymbol{v}}^k &= \left(\boldsymbol{A}^k_{\text{sys}}\right)^{-1} \boldsymbol{b}_{\text{sys}}; \\ \boldsymbol{A}^k_{\text{sys}} &= \mathrm{F}^T_{\text{int}} \boldsymbol{W} \mathrm{F}_{\text{int}} + \boldsymbol{D}_2^T \boldsymbol{\Omega}_2 \boldsymbol{D}_2 + \boldsymbol{D}_1^T \boldsymbol{\Omega}_1^k \boldsymbol{D}_1, \\ \boldsymbol{b}_{\text{sys}} &= \mathrm{F}^T_{\text{int}} \boldsymbol{W} \boldsymbol{f}_{\text{ext}}, \end{aligned} \tag{19}$$

where $\boldsymbol{W}$ is a diagonal matrix that weights the relative importance of the equations in F_{int}, $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ are the first and second differences matrices respectively, $\boldsymbol{\Omega}_2$ and $\boldsymbol{\Omega}_1$, are also diagonal weight matrices for the penalizations in the second and first differences of the vertices, note that k indicates the step of the iterative penalization for monotonic smoothing. When all the intervals are monotonically increasing the weighting matrix $\boldsymbol{\Omega}_1^k = 0$ and that penalization automatically disappears. In contrast to the overdetermined system of linear equations produced by piecewise approximation presented in this work, the methodology in Flaschel et al. (2021) employs a basis of functions which extend over the whole function domain, although this procedure seems similar to the local one, the global support typically leads to dense solutions which are less interpretable. For the sake of interpretability Flaschel et al. (2021) proposes a sparsity promotion with ℓ_p regularization, but at the price of requiring a fixed point iteration procedure to obtain the solution and introducing p as an hyperparameter that has also to be determined in the solution process. On the other hand, the additional matrices added to the system matrix in our work ensure smoothness of the solution (making the solution more

robust against noise) and allows penalization in monotonicity if required (this penalization has not been required for the examples presented in this article).

3 RESULTS

3.1 Toy Example: Recovering an Analytical Neo-Hookean With a P-Spline for the Continuum Free Energy

In this section we are going to demonstrate just the FEM part of the methodology using a P-Spline based approximation for the functions that form the deviatoric continuum free energy. The experimental data is a virtual test (FEM results) of an analytical quasi-incompressible Neo-Hookean model with parameters $\mu = 3.5\text{MPa}$ and $\kappa = 1\times10^5\text{MPa}$. The FEM model employed for the virtual tests is a 2D plate with a hole on plane stress conditions. The solicitation is an imposed displacement in the upper border with $u = 4$. The dimensions of the plate and the mesh employed in the simulation is shown in **Figure 2**, the elements are quadratic quadrilaterals with nine integration points. The formulation employed for plane stress in large deformations is detailed in **Supplementary Appendix SA**, the reader can also find the details about the Neo-Hookean material model in **Supplementary Appendix SB**. Regarding the software employed, Julia programming language, Bezanson et al. (2017) has been used, in particular the package FerriteFem.jl, see Carlsson and Ekre (2021), for the FEM simulations and Amores (2022) for the P-Splines functionality.

A typical assumption made on the phenomenological approach for isotropic incompressible solids is the Valanis Landel decomposition, $\mathcal{W}\left(\lambda_1^d,\lambda_2^d,\lambda_3^d\right) = \omega\left(\lambda_1^d\right) + \omega\left(\lambda_2^d\right) + \omega\left(\lambda_3^d\right)$. With that assumption, the expressions for the principal PK2 stresses can be obtained in terms of an unknown function $\omega'(x)$:

$$S_j = J^{-2/3}\frac{1}{\lambda_j^d}\left(\omega'\left(\lambda_j^d\right) - \frac{\lambda_3^d}{\lambda_j^d}\omega'\left(\lambda_3^d\right)\right) \quad \text{for } j = 1,2 \qquad (20)$$

Once $\omega'(x)$ is obtained, the principal stresses are determined for any deformation state under the plane stress hypothesis or the contribution to the principal stresses coming from the deviatoric energy for any general case (not in plane stress), see (**Eq. 8**), this is all that can be determined from a constitutive point of view because for an incompressible solid, the pressure does not come from the constitutive equation but from equilibrium conditions. Using P-splines, the unknown function can be written in terms of some known basis functions and some unknown vertices in a similar manner that it is done in FEM formulation with the displacement field:

$$\omega'(x) = \sum_{i=1}^{n_{\text{vert}}} B_i(x)\,\hat{\omega}_i = \begin{bmatrix} B_1(x) & \dots & B_{n_{\text{vert}}}(x)\end{bmatrix}\begin{bmatrix}\hat{\omega}_1 \\ \vdots \\ \hat{\omega}_{n_{\text{vert}}}\end{bmatrix} = \boldsymbol{B}_{row}(x)\cdot\hat{\boldsymbol{\omega}}' \qquad (21)$$

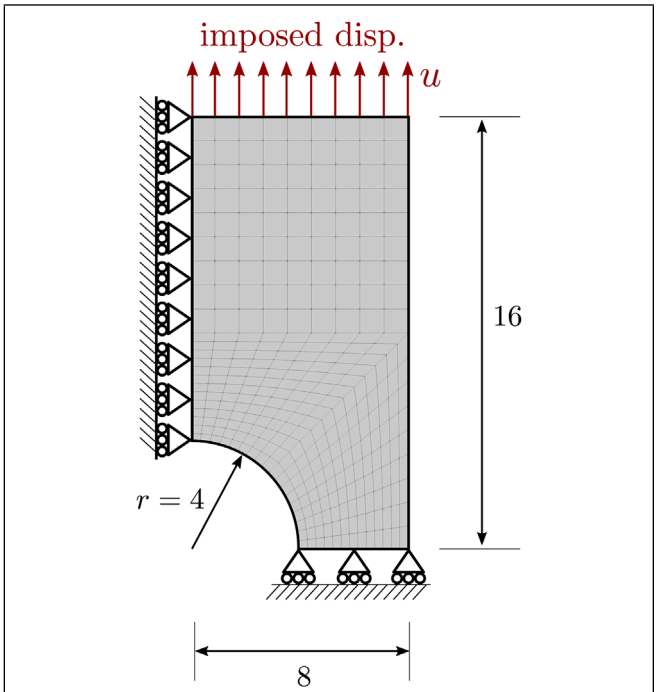


FIGURE 2 | FEM model used during virtual tests. The elements employed are quadratic quadrilateral with nine integration points.

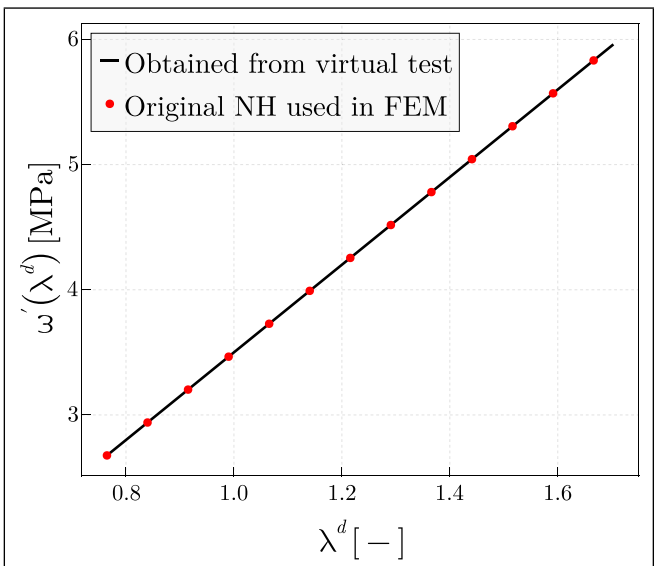


FIGURE 3 | Comparison of the Neo-Hookean $\omega'_{\text{NH}}\left(\lambda^d\right)$ and the $\omega'\left(\lambda^d\right)$ obtained from the non-homogeneous virtual test.

With the previous expression, the principal stresses could be written as:

$$S_j = J^{-2/3}\frac{1}{\lambda_j^d}\left(\boldsymbol{B}_{row}\left(\lambda_j^d\right) - \frac{\lambda_3^d}{\lambda_j^d}\boldsymbol{B}_{row}\left(\lambda_3^d\right)\right)\cdot\hat{\boldsymbol{\omega}}' = \boldsymbol{S}^j_{\text{row}}\left(\lambda_j^d,\lambda_3^d\right)\cdot\hat{\boldsymbol{\omega}}' \quad \text{for } j = 1,2 \qquad (22)$$

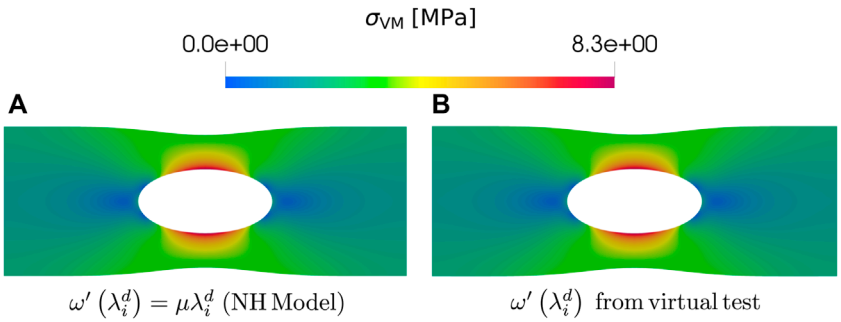


FIGURE 4 | Plot comparing the results of σ_{VM} for the reference Neo-Hookean continuum model and the reverse-engineered one. **(A)** σ_{VM} using the Neo-Hookean model. **(B)** σ_{VM} using the $\omega'\left(\lambda_i^d\right)$ reverse-engineered from the non-homogeneous virtual test.

$$\begin{bmatrix} S_1 \\ S_2 \\ 0 \end{bmatrix}_{j,X_{\text{ppal}}} = \begin{bmatrix} \left[\boldsymbol{S}^1_{\text{row}}\left(\lambda_1^d,\lambda_3^d\right)\right]^T \\ \left[\boldsymbol{S}^2_{\text{row}}\left(\lambda_2^d,\lambda_3^d\right)\right]^T \\ [0]^T \end{bmatrix}_j [\hat{\boldsymbol{\omega}}']^T \tag{23}$$

$$\begin{aligned} \boldsymbol{P}_j : \nabla \boldsymbol{h}_j^i = \text{sym}\left(\boldsymbol{X}_j^T \nabla \boldsymbol{h}_j^i\right) : \boldsymbol{S}_j &= \begin{bmatrix} l_{11} & l_{22} & 2l_{12} \end{bmatrix}^i_{j,X_{\text{ppal}}} \begin{bmatrix} S_1 \\ S_2 \\ 0 \end{bmatrix}_{j,X_{\text{ppal}}} \\ &= \left(l_{11} \cdot \boldsymbol{S}^1_{\text{row}}\left(\lambda_1^d,\lambda_3^d\right) + l_{22} \cdot \boldsymbol{S}^2_{\text{row}}\left(\lambda_2^d,\lambda_3^d\right)\right)^i_j \cdot \hat{\boldsymbol{\omega}}' \\ &= \left({}^*\boldsymbol{S}_{\text{row}}\right)^i_j\left(\lambda_1^d,\lambda_2^d,\lambda_3^d\right) \cdot \hat{\boldsymbol{\omega}}' \end{aligned} \tag{24}$$

Referring again to **Eq. 13**, internal nodal forces of the element can be written as the product of a known matrix by a vector of unknown vertices:

$$\boldsymbol{f}^e_{\text{int}} = \text{F}^e_{\text{int}} \cdot \hat{\boldsymbol{\omega}}' \tag{25}$$

Doing the assembly of the internal nodal forces of the elements in the mesh:

$$\boldsymbol{f}_{\text{int}} = \bigwedge_{e=1}^{nel} \boldsymbol{f}^e_{\text{int}} = \left(\bigwedge_{e=1}^{nel} \text{F}^e_{\text{int}}\right) \cdot \hat{\boldsymbol{\omega}}' = \text{F}_{\text{int}} \cdot \hat{\boldsymbol{\omega}}' \tag{26}$$

From the overdetermined linear system of equations $\text{F}_{\text{int}} \cdot \hat{\boldsymbol{\omega}}' = \boldsymbol{f}_{\text{ext}}$ we can solve for $\hat{\boldsymbol{\omega}}'$:

$$\begin{aligned} (\hat{\boldsymbol{\omega}}')^k &= \left(\boldsymbol{A}^k_{\text{sys}}\right)^{-1} \boldsymbol{b}_{\text{sys}}; \\ \boldsymbol{A}^k_{\text{sys}} &= \text{F}^T_{\text{int}} \boldsymbol{W} \text{F}_{\text{int}} + \boldsymbol{D}_2^T \boldsymbol{\Omega}_2 \boldsymbol{D}_2 + \boldsymbol{D}_1^T \boldsymbol{\Omega}_1^k \boldsymbol{D}_1, \\ \boldsymbol{b}_{\text{sys}} &= \text{F}^T_{\text{int}} \boldsymbol{W} \boldsymbol{f}_{\text{ext}} \end{aligned} \tag{27}$$

The number of vertices has to be enough to capture the complexity of the curve, typically $n_{\text{vert}} = 14$ suffice, but additional vertices can be added. If more and more vertices are added, the number of equations required to determine them increases and the problem can become ill-conditioned, this is solved by the smoothing term which adds the information of smooth transition between vertices and links the vertices to its neighbours. In the homogeneous case, just a single equation is obtained for every load/displacement step, therefore, in order to obtain information for the function on the considered domain (from the minimum principal stretch to the maximum principal stretch on the simulation), it would be necessary to sweep a whole range of load/displacement steps. On the other hand, for the non-homogeneous case, in principle, it would be possible to employ just a single load/displacement step if the step under consideration is rich enough (in this case, just the last step, $u = 4$ was used). In case that additional information is needed to determine the function on the considered range, it is also possible to add more steps between $u = 0$ and $u = 4$. Since the initial solution was directly monotonically increasing, $\Omega_1 = \boldsymbol{0}$ and it was not necessary to follow an iterative process, $\hat{\boldsymbol{\omega}}'$ was directly obtained from the simple initial system of equations:

$$(\hat{\boldsymbol{\omega}}') = \left(\boldsymbol{A}_{\text{sys}}\right)^{-1} \boldsymbol{b}_{\text{sys}}; \quad \boldsymbol{A}_{\text{sys}} = \text{F}^T_{\text{int}} \boldsymbol{W} \text{F}_{\text{int}} + \boldsymbol{D}_2^T \boldsymbol{\Omega}_2 \boldsymbol{D}_2, \ \boldsymbol{b}_{\text{sys}} = \text{F}^T_{\text{int}} \boldsymbol{W} \boldsymbol{f}_{\text{ext}} \tag{28}$$

With the vertices obtained, the P-Spline could be reconstructed and compared to the original one, see **Figure 3**.

Since the function $\omega'(x)$ has been reverse-engineered exactly, the values for the free energy partial derivatives for any state of deformation can be determined, and will be exactly equal to those obtained with the original Neo-Hookean model. If desired, the FEM simulations can be run again with the reversed-engineered derivatives of the energy function and the σ_{VM} plot could be compared with the one obtained for the original analytical Neo-Hookean, see **Figure 4**. As it can be seen in **Figure 4** the plots are exactly the same as it might be expected from having obtained the exact same ω' as the one from the original Neo-Hookean.

In this particular example, the matrix of the system $\boldsymbol{A}_{\text{sys}} \cdot \hat{\boldsymbol{\omega}}' = \boldsymbol{b}_{\text{sys}}$, presents a null eigenvalue with its corresponding associated eigenvector, i.e., there is a certain subspace of possible solutions $\hat{\boldsymbol{\omega}}'_0$ (containing just the direction of the eigenvector associated to the null eigenvalue) such that added to the solution, $\hat{\boldsymbol{\omega}}'$, produce null effect on the independent term, $\boldsymbol{b}_{\text{sys}}$ and therefore $\hat{\boldsymbol{\omega}}' + \alpha\hat{\boldsymbol{\omega}}'_0$, for an arbitrary scalar α, is also a solution of the system. Looking at **Eq. 20**, to not produce any effect, such function has to satisfy the condition $\omega'\left(\lambda_i^d\right) - \lambda_3^d/\lambda_i^d \cdot \omega'\left(\lambda_3^d\right) = 0 \quad \forall\, \lambda_1^d,\lambda_2^d,\lambda_3^d$. Regarding the functions that satisfy this condition, there is a specific

form of function that complies (which at the same time must be smooth due to the regularization) $\omega'\left(\lambda_i^d\right)\lambda_i^d = \omega'\left(\lambda_3^d\right)\lambda_3^d \quad \forall\, \lambda_1^d,\lambda_2^d,\lambda_3^d$. The only way two functions of different independent variables can be equal $\forall\, \lambda_1^d,\lambda_2^d,\lambda_3^d$ is that those functions are constant $\omega'\left(\lambda_i^d\right)\lambda_i^d = \omega'\left(\lambda_3^d\right)\lambda_3^d = \beta_0 \rightarrow \omega'(\lambda) = \beta_0/\lambda$. Since the shape of the eigenvector associated to the null eigenvalue exactly matches the shape of the function, $\omega'(\lambda) = \beta_0/\lambda$, we would like to understand physically the origin of this null contribution to the system. Looking at **Eq. 8**, we can see that the pressure term is of the form $p(\lambda_1,\lambda_2,\lambda_3)/\lambda_i^2$, the same form of the term resulting from introducing an additional β_0/λ_i^d to the function $\omega'\left(\lambda_i^d\right)$. From a more general perspective we can see that this indetermination comes from the split of the energy into a volumetric and a deviatoric contribution:

$$\boldsymbol{S} = \boldsymbol{S}^v + \boldsymbol{S}^d; \quad \boldsymbol{S}^d = 2\frac{d\mathcal{W}}{d\boldsymbol{C}} = 2J^{-2/3}\mathbb{P} : \frac{d\mathcal{W}}{d\boldsymbol{C}^d} = J^{-2/3}\mathbb{P} : \boldsymbol{S}^{|d} \tag{29}$$

where $\boldsymbol{C} = \boldsymbol{X}^T\boldsymbol{X}$ is the right Cauchy-Green tensor and $\mathbb{P}$ is the fourth order projector tensor, see **Supplementary Appendix SB**. Since $\mathbb{P} : \left(\alpha\boldsymbol{C}^{-1}\right) = 0$, from (**Eq. 29**) it can be seen that there are multiple $\boldsymbol{S}^{|d} = \partial\mathcal{W}/\partial\boldsymbol{C}^{\boldsymbol{d}}$ that produce the same $\boldsymbol{S}^d$ and the undetermined part has the form of a volumetric-like PK2 stress tensor. Any contribution of this form has no effect on the solution since it is going to be finally eliminated by the projector $\mathbb{P}$.

3.2 Micro-Mechanical Approach Based on the Non-Affine Deformation Chain Model

In this section the complete methodology described in **Section 2** (Macro-Micro-Macro first, FEM second) is demonstrated. In contrast to the procedure described in **Section 3.1**, here the P-Spline will approximate the derivative of the chain energy function $P_{ch}\left(\lambda_{ch}\right) = d\psi_{ch}/d\lambda_{ch}$ (mechanical behavior of the material constituents at the micro-scale) rather than the continuum contributions to the energy. The experimental data is obtained from a virtual (FEM simulation) using the analytical structure-based material model in Amores et al. (2020). To be more precise, the $P_{ch}\left(\lambda_{ch}\right)$ that will be employed to perform the virtual test is the one obtained for the Treloar test data for unfilled rubber, Treloar (1944), that function is displayed in (Amores et al., 2020, Figure B.1). The FEM model employed for the virtual tests is again the 2D plate with a hole on plane stress conditions with an imposed displacement in the upper border ($u = 8$). All the required dimensions and the mesh employed in the simulation is shown in **Figure 2**. The reader can find further details about the non-affine deformation chain material model in **Supplementary Appendix SC** and Amores et al. (2020), a discussion about the suitability of the non-affine stretch employed in the model (free-fluctuating network assumption) is presented in Amores et al. (2021).

In Amores et al. (2020) a way was established to compute the strain energy function of the continuum with an homogenization of the micro-structural chain free energy function in polymeric like materials under the assumption of incompressibility $(\lambda_1 \cdot \lambda_2 \cdot \lambda_3 = 1)$:

$$\mathcal{W}(\lambda_1,\lambda_2,\lambda_3) = \frac{1}{S}\int_S \psi_{ch}(\lambda_{ch})\, dS \tag{30}$$

For FEM simulations even pure incompressible solids are simulated with the quasi-incompressible material framework:

$$\Psi(\boldsymbol{A}) = \mathcal{U}(J) + \mathcal{W}\left(\lambda_1^d,\lambda_2^d,\lambda_3^d\right)$$

in which $\mathcal{U}(J)$ is a penalization function that leads to incompressibility for $\kappa \rightarrow \infty$ like $\mathcal{U}(J) = \frac{1}{2}k(J-1)^2$ and where $\mathcal{W}\left(\lambda_1^d,\lambda_2^d,\lambda_3^d\right)$ is the same as the one used in pure incompressibility but with the deviatoric stretches instead of the total ones:

$$\begin{aligned} \mathcal{W}\left(\lambda_1^d,\lambda_2^d,\lambda_3^d\right) &= \frac{1}{S}\int_S \psi_{ch}\left(\lambda_{ch}^d\right) dS; \\ \lambda_{ch}^d = \boldsymbol{U}^d : (\boldsymbol{r}\otimes\boldsymbol{r}) &= \lambda_1^d r_1^2 + \lambda_2^d r_2^2 + \lambda_3^d r_3^2 \end{aligned} \tag{31}$$

$$\mathcal{W}_k = \frac{\partial\mathcal{W}}{\partial\lambda_k^d} = \frac{1}{S}\int_S \frac{d\psi_{ch}\left(\lambda_{ch}^d\right)}{d\lambda_{ch}^d}\frac{d\lambda_{ch}^d}{d\lambda_k^d}dS = \frac{1}{S}\int_S P_{ch}\left(\lambda_{ch}^d\right) r_k^2 dS \tag{32}$$

The unknown function that will be approximated using P-Splines is $P_{ch}\left(\lambda_{ch}^d\right)$, as described above, the P-Splines representation expresses the unknown function in terms of some known basis vector multiplied by a vector of unknown vertices:

$$\begin{aligned} P_{ch}(x) &= \sum_{i=1}^{n_{\text{vert}}} B_i(x)\,\hat{P}_{chi} \\ &= \begin{bmatrix} B_1(x) & \dots & B_{n_{\text{vert}}}(x) \end{bmatrix} \begin{bmatrix} \hat{P}_{chi} \\ \vdots \\ \hat{P}_{chn_{\text{vert}}} \end{bmatrix} \\ &= \boldsymbol{B}_{row}(x)\cdot\hat{\boldsymbol{P}}_{ch} \end{aligned} \tag{33}$$

Introducing the expression for $P_{ch}(x)$ in (**Eq. 32**) and then all the derivatives of the form $\mathcal{W}_k$ in (**Eq. 11**) a expression for the principal stresses in terms of the vertices that define the Polymeric-chain response function can be obtained:

$$\begin{aligned} S_j &= J^{-2/3}\frac{1}{\lambda_j^d}\left\{\frac{1}{S}\int_S \boldsymbol{B}_{\text{row}}\left(\lambda_{ch}^d\right)\left(r_j^2 - \frac{\lambda_3^d}{\lambda_j^d}r_3^2\right)dS\right\}\cdot\hat{\boldsymbol{P}}_{ch} \\ &= \boldsymbol{S}_{\text{row}}^j\left(\lambda_1^d,\lambda_2^d\right)\cdot\hat{\boldsymbol{P}}_{ch} \quad \text{for } j = 1,2 \end{aligned} \tag{34}$$

Note that the previous integral in the microsphere is computed by a numerical quadrature, $\int_S f(\boldsymbol{r})\,dS = \sum_{j=1}^{n_{qs}} f(\boldsymbol{r})\,w_j^S$ an that although λ_{ch}^d depends on λ_1^d, λ_2^d and λ_3^d, since $\lambda_1^d \cdot \lambda_2^d \cdot \lambda_3^d = 1$, there are just 2 independent variables. Again a procedure similar to the one in the toy example, **Section 3.1**, is followed:

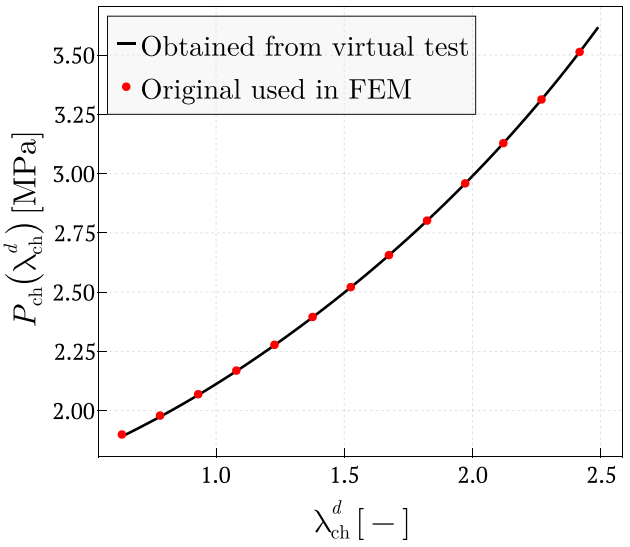


FIGURE 5 | Plot comparing $P_{ch}\left(\lambda^d_{ch}\right)$ extracted from (Amores et al., 2020, Figure B.1) and the reverse-engineered $P_{ch}\left(\lambda^d_{ch}\right)$ obtained with the methodology described in the article.

$$\begin{bmatrix} S_1 \\ S_2 \\ 0 \end{bmatrix}_{j,X_{\text{ppal}}} = \begin{bmatrix} \left[\boldsymbol{S}^1_{\text{row}}\left(\lambda^d_1,\lambda^d_2\right)\right]^T \\ \left[\boldsymbol{S}^1_{\text{row}}\left(\lambda^d_1,\lambda^d_2\right)\right]^T \\ [0]^T \end{bmatrix}_j [\hat{\boldsymbol{P}}_{ch}] \tag{35}$$

$$\begin{aligned} \boldsymbol{P}_j : \nabla \boldsymbol{h}^i_j &= \text{sym}\left(\boldsymbol{h}^T_j \nabla \boldsymbol{h}^i_j\right) : \boldsymbol{S}_j \\ &= \left[l_{11} \quad l_{22} \quad 2l_{12}\right]^i_{j,X_{\text{ppal}}} \begin{bmatrix} S_1 \\ S_2 \\ 0 \end{bmatrix}_{j,X_{\text{ppal}}} \\ &= \left(l_{11} \cdot \boldsymbol{S}^1_{\text{row}}\left(\lambda^d_1,\lambda^d_2\right) + l_{22} \cdot \boldsymbol{S}^2_{\text{row}}\left(\lambda^d_1,\lambda^d_2\right)\right)^i_j \cdot \hat{\boldsymbol{P}}_{ch} \\ &= \left({}^*\boldsymbol{S}_{\text{row}}\right)^i_j\left(\lambda^d_1,\lambda^d_2\right) \cdot \hat{\boldsymbol{P}}_{ch} \end{aligned} \tag{36}$$

Looking at the previous matrix equation and using again (**Eq. 13**), it is straightforward to write the internal vector force for an element as

$$\boldsymbol{f}^e_{\text{int}} = \text{F}^e_{\text{int}} \cdot \hat{\boldsymbol{P}}_{ch} \tag{37}$$

where F^e_{int} is a matrix. Now in the same way that the components that the components of $\boldsymbol{f}^e_{\text{int}}$ can be assembled to obtain the global internal forces vector, $\boldsymbol{f}_{\text{int}}$. The rows of F^e_{int} can be assembled to obtain a global internal force matrix F_{int}.

$$\boldsymbol{f}_{\text{int}} = \bigwedge_{e=1}^{nel} \boldsymbol{f}^e_{\text{int}} = \left(\bigwedge_{e=1}^{nel} \text{F}^e_{\text{int}}\right) \cdot \hat{\boldsymbol{P}}_{ch} = \text{F}_{\text{int}} \cdot \hat{\boldsymbol{P}}_{ch} \tag{38}$$

At that point it is important to note that if compared to the procedure followed in **Section 3.1**, now the internal force term is linked with the unknown vertices that correspond to the constituents behavior at the micro-scale. With that approach we show that scales can be crossed and information in one scale can be pushed up to other scales, that of course taking into account that the macro to micro connection has already been established.

From the overdetermined linear system of equations $\text{F}_{\text{int}} \cdot \hat{\boldsymbol{P}}_{ch} = \boldsymbol{f}_{\text{ext}}$ a solution can be obtained for $\hat{\boldsymbol{P}}_{ch}$:

$$\begin{aligned} \hat{\boldsymbol{P}}^k_{ch} &= \left(\boldsymbol{A}^k_{\text{sys}}\right)^{-1} \boldsymbol{b}_{\text{sys}}; \\ \boldsymbol{A}^k_{\text{sys}} &= \text{F}^T_{\text{int}} \boldsymbol{W} \text{F}_{\text{int}} + \boldsymbol{D}^T_2 \boldsymbol{\Omega}_2 \boldsymbol{D}_2 + \boldsymbol{D}^T_1 \boldsymbol{\Omega}^k_1 \boldsymbol{D}_1, \\ \boldsymbol{b}_{\text{sys}} &= \text{F}^T_{\text{int}} \boldsymbol{W} \boldsymbol{f}_{\text{ext}} \end{aligned} \tag{39}$$

The discussion about the number of vertices and steps of load/displacement required in **Section 3.1** is also applicable here. In this particular case, just the last step of deformation ($u = 8$) was used. Again, in case that additional information is needed to determine the function on the considered range, it is also possible to add more steps between $u = 0$ and $u = 8$. Since the initial solution was directly monotonically increasing, $\Omega_1 = \mathbf{0}$ and it was not necessary to follow an iterative process, $\hat{\boldsymbol{P}}_{ch}$ was directly obtained from the simple initial system of equations:

$$\hat{\boldsymbol{P}}_{ch} = \left(\boldsymbol{A}_{\text{sys}}\right)^{-1} \boldsymbol{b}_{\text{sys}}; \quad \boldsymbol{A}_{\text{sys}} = \text{F}^T_{\text{int}} \boldsymbol{W} \text{F}_{\text{int}} + \boldsymbol{D}^T_2 \boldsymbol{\Omega}_2 \boldsymbol{D}_2, \; \boldsymbol{b}_{\text{sys}} = \text{F}^T_{\text{int}} \boldsymbol{W} \boldsymbol{f}_{\text{ext}} \tag{40}$$

With the vertices obtained, the P-Spline could be reconstructed and compared it to the original one, see **Figure 5**.

As it is shown in **Figure 5**, the $P_{ch}\left(\lambda^d_{ch}\right)$ extracted from (Amores et al., 2020, Figure B.1) and the one obtained through the methodology described in the paper are practically identical. Looking closely, there are some small deviations close to $\lambda^d_{ch} = 1$

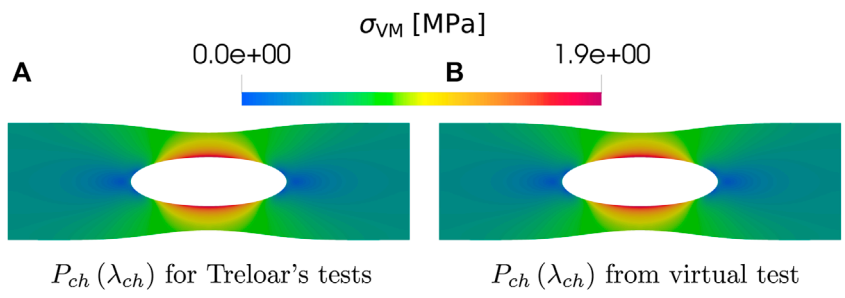


FIGURE 6 | Plot comparing the results of σ_{VM} for the reference micro-mechanical model in Amores et al. (2020) that uses (Amores et al., 2020, Figure B.1) and the reverse-engineered one described in the article. **(A)** σ_{VM} using the $P_{ch}\left(\lambda_{ch}\right)$ obtained for unfilled rubber, Treloar (1944) extracted from (Amores et al., 2020, Figure B.1). **(B)** σ_{VM} using the $P_{ch}\left(\lambda_{ch}\right)$ reverse-engineered from the non-homogeneous virtual test using the methodology described in this article.

on the compression range. As it has already been mentioned, the origin of those deviations is the eigenvector linked to an almost zero eigenvalue on the reduced matrix of the system, this matter has been carefully justified at the end of **Section 3.1**. With the obtained $P_{ch}\left(\lambda_{ch}^{d}\right)$ the partial derivatives of the free energy can be determined for any state of deformation, with them, it is possible to re-run the FEM simulations and compare for example, the σ_{VM} plot. **Figure 6** compares the σ_{VM} with $P_{ch}\left(\lambda_{ch}^{d}\right)$ extracted from (Amores et al., 2020, Figure B.1) and the $P_{ch}\left(\lambda_{ch}^{d}\right)$ with the methodology described in this paper. As we can see, although the functions are slightly different, that discrepancy has no effect on the final stresses, this is so because the additional spurious term that appears in the solution corresponds to a pressure term (in the cauchy stress sense) and therefore, it will not appear in the final computed stress due to the projector $\mathbb{P}$.

4 CONCLUSION

We have proposed a numerical method for determining both the continuum free energy and the polymer macromolecules behavior from arbitrary non-homogeneous DIC-based tests at the continuum scale. The procedure consists of a combination of our Macro-Micro-Macro approach and a Finite Element model. As a novel contribution of this approach, we show that by crossing scales transferring the microscale unknowns to the finite element formulation it is possible to determine the mechanical chain behavior from non-homogeneous experiments at the continuum scale by simply solving a linear system of equations (i.e., in an even more efficient manner than the subsequent simulations of the polymer behavior). Another key aspect is that Penalized B-splines (P-splines) preserve the general form of the energy function while retaining sufficient tools for enforcing specific desired conditions on the sought functions. The methodology at hand recovers the analytical free energies used as starting point in the virtual tests. Therefore, any finite element simulation performed with the reversed-engineered energy function will provide identical results to the original material model. From the authors perspective, this new approach opens new possibilities for data-driven characterization of the micro-structure from non-homogeneous tests at the macro-scale. Further research has to be conducted to evaluate the generalization of the approach in Amores et al. (2020) to more complex material behaviors from which the applicability of this same methodology has to be assessed.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

VA: Conceptualization, methodology, software, and writing original draft. FM: Conceptualization, methodology, funding acquisition, and supervision. EC: Methodology and supervision. FC: Conceptualization, methodology, funding acquisition, and supervision. All authors contributed to manuscript revision, read, and approved the submitted version.

FUNDING

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 101007815.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fmats.2022.879614/full#supplementary-material

REFERENCES

Alastrué, V., Martínez, M. A., Doblaré, M., and Menzel, A. (2009). Anisotropic Micro-sphere-based Finite Elasticity Applied to Blood Vessel Modelling. *J. Mech. Phys. Sol.* 57, 178–203. doi:10.1016/J.JMPS.2008.09.005

Amores, V. J., Benítez, J. M., and Montáns, F. J. (2019). Average-chain Behavior of Isotropic Incompressible Polymers Obtained from Macroscopic Experimental Data. A Simple Structure-Based WYPiWYG Model in Julia Language. *Adv. Eng. Softw.* 130, 41–57. doi:10.1016/j.advengsoft.2019.01.004

Amores, V. J., Benítez, J. M., and Montáns, F. J. (2020). Data-driven, Structure-Based Hyperelastic Manifolds: A Macro-Micro-Macro Approach to Reverse-Engineer the Chain Behavior and Perform Efficient Simulations of Polymers. *Comput. Structures* 231, 106209. doi:10.1016/J.COMPSTRUC.2020.106209

Amores, V. J., Nguyen, K., and Montáns, F. J. (2021). On the Network Orientational Affinity assumption in Polymers and the Micro-macro Connection through the Chain Stretch. *J. Mech. Phys. Sol.* 148, 104279. doi:10.1016/J.JMPS.2020.104279

Amores, V. (2022). PSPlines. doi:10.5281/zenodo.6084053

Arruda, E. M., and Boyce, M. C. (1993). A Three-Dimensional Constitutive Model for the Large Stretch Behavior of Rubber Elastic Materials. *J. Mech. Phys. Sol.* 41, 389–412. doi:10.1016/0022-5096(93)90013-6

Bergström, J. S., and Hayman, D. (2016). An Overview of Mechanical Properties and Material Modeling of Polylactide (PLA) for Medical Applications. *Ann. Biomed. Eng.* 44, 330–340. doi:10.1007/S10439-015-1455-8/TABLES/2

Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. (2017). Julia: A Fresh Approach to Numerical Computing. *SIAM Rev.* 59, 65–98. doi:10.1137/141000671

Carlsson, K., and Ekre, F. (2021). Ferrite.jl.

Chagnon, G., Ohayon, J., Martiel, J.-L., Favier, D., Modeling, D. F. H., Chagnon, G., et al. (2017). "Hyperelasticity Modeling for Incompressible Passive Biological Tissues," in *Biomechanics of Living Organs*. Academic Press (Elsevier), 3–30. doi:10.1016/B978-0-12-804009-6.00001-8

Chagnon, G., Rebouah, M., and Favier, D. (2015). Hyperelastic Energy Densities for Soft Biological Tissues: A Review. *J. Elast* 120, 129–160. doi:10.1007/S10659-014-9508-Z

Crespo, J., and Montáns, F. J. (2018). A Continuum Approach for the Large Strain Finite Element Analysis of Auxetic Materials. *Int. J. Mech. Sci.* 135, 441–457. doi:10.1016/J.IJMECSCI.2017.11.038

Eggersmann, R., Kirchdoerfer, T., Reese, S., Stainier, L., and Ortiz, M. (2019). Model-Free Data-Driven Inelasticity. *Comput. Methods Appl. Mech. Eng.* 350, 81–99. doi:10.1016/J.CMA.2019.02.016

Eilers, P. H. C., and Marx, B. D. (2021). *Practical Smoothing: The Joys of P-Splines*. Cambridge University Press. doi:10.1017/9781108610247

Flaschel, M., Kumar, S., and De Lorenzis, L. (2021). Unsupervised Discovery of Interpretable Hyperelastic Constitutive Laws. *Comput. Methods Appl. Mech. Eng.* 381, 113852. doi:10.1016/J.CMA.2021.113852

Flory, P. J., and Rehner, J. (1943). Statistical Mechanics of Cross-Linked Polymer Networks II. Swelling. *J. Chem. Phys.* 11, 521–526. doi:10.1063/1.1723792

Gent, A. N. (1989). Rubberlike Elasticity-A Molecular Primer, by James E. Mark and Burak Erman, John Wiley & Sons, New York, 1988, 196 Pp. *J. Polym. Sci. C Polym. Lett.* 27, 405–406. doi:10.1002/POL.1989.140271012

González, D., Chinesta, F., and Cueto, E. (2018). Thermodynamically Consistent Data-Driven Computational Mechanics. *Continuum Mech. Thermodyn.* 31 (1), 239–253. doi:10.1007/S00161-018-0677-Z

Holzapfel, G. A. (2002). Nonlinear Solid Mechanics: a Continuum Approach for Engineering Science. *Meccanica* 37, 489–490. doi:10.1023/a:1020843529530

Ibañez, R., Abisset-Chavanne, E., Aguado, J. V., Gonzalez, D., Cueto, E., and Chinesta, F. (2018). A Manifold Learning Approach to Data-Driven Computational Elasticity and Inelasticity. *Arch. Computat Methods Eng.* 25, 47–57. doi:10.1007/S11831-016-9197-9/FIGURES/11

Ibañez, R., Borzacchiello, D., Aguado, J. V., Abisset-Chavanne, E., Cueto, E., Ladevèze, P., et al. (2017). Data-driven Non-linear Elasticity: Constitutive Manifold Construction and Problem Discretization. *Comput. Mech.* 60, 813–826. doi:10.1007/S00466-017-1440-1

Khiêm, V. N., and Itskov, M. (2016). Analytical Network-Averaging of the Tube Model:. *J. Mech. Phys. Sol.* 95, 254–269. doi:10.1016/J.JMPS.2016.05.030

Kirchdoerfer, T., and Ortiz, M. (2016). Data-driven Computational Mechanics. *Comput. Methods Appl. Mech. Eng.* 304, 81–101. doi:10.1016/J.CMA.2016.02.001

Kirchdoerfer, T., and Ortiz, M. (2018). Data-driven Computing in Dynamics. *Int. J. Numer. Meth. Engng* 113, 1697–1710. doi:10.1002/NME.5716

Korzeniowski, T. F., and Weinberg, K. (2021). A Multi-Level Method for Data-Driven Finite Element Computations. *Comput. Methods Appl. Mech. Eng.* 379, 113740. doi:10.1016/J.CMA.2021.113740

Latorre, M., and Montáns, F. J. (2013). Extension of the Sussman-Bathe Spline-Based Hyperelastic Model to Incompressible Transversely Isotropic Materials. *Comput. Structures* 122, 13–26. doi:10.1016/J.COMPSTRUC.2013.01.018

Latorre, M., and Montáns, F. J. (2014). What-You-Prescribe-Is-What-You-Get Orthotropic Hyperelasticity. *Comput. Mech.* 53, 1279–1298. doi:10.1007/S00466-013-0971-3

Liu, X., Tao, F., Du, H., Yu, W., and Xu, K. (2020). Learning Nonlinear Constitutive Laws Using Neural Network Models Based on Indirectly Measurable Data. *J. Appl. Mech. Trans. ASME* 87. doi:10.1115/1.4047036/1083320

Marckmann, G., and Verron, E. (2006). Comparison of Hyperelastic Models for Rubber-like Materials. *Rubber Chem. Techn.* 79, 835–858. doi:10.5254/1.3547969

Mooney, M. (1940). A Theory of Large Elastic Deformation. *J. Appl. Phys.* 11, 582–592. doi:10.1063/1.1712836

Moreno, S., Amores, V. J., Benítez, J. M., and Montáns, F. J. (2020). Reverse-engineering and Modeling the 3D Passive and Active Responses of Skeletal Muscle Using a Data-Driven, Non-parametric, Spline-Based Procedure. *J. Mech. Behav. Biomed. Mater.* 110, 103877. doi:10.1016/J.JMBBM.2020.103877

Nguyen-Thanh, V. M., Zhuang, X., and Rabczuk, T. (2020). A Deep Energy Method for Finite Deformation Hyperelasticity. *Eur. J. Mech. - A/Solids* 80, 103874. doi:10.1016/J.EUROMECHSOL.2019.103874

Ogden, R. W. (1972). Large Deformation Isotropic Elasticity - on the Correlation of Theory and experiment for Incompressible Rubberlike Solids. *Proc. R. Soc. Lond. A.* 326, 565–584. doi:10.1098/RSPA.1972.0026

Sáez, P., Alastrué, V., Peña, E., Doblaré, M., and Martínez, M. A. (2011). Anisotropic Microsphere-Based Approach to Damage in Soft Fibered Tissue. *Biomech. Model. Mechanobiol* 11, 595–608. doi:10.1007/S10237-011-0336-9

Sussman, T., and Bathe, K.-J. (2009). A Model of Incompressible Isotropic Hyperelastic Material Behavior Using Spline Interpolations of Tension-Compression Test Data. *Commun. Numer. Meth. Engng.* 25, 53–63. doi:10.1002/CNM.1105

Treloar, L. R. G. (1944). Stress-Strain Data for Vulcanized Rubber under Various Types of Deformation. *Rubber Chem. Techn.* 17, 813–825. doi:10.5254/1.3546701

Treloar, L. R. G. (1975). *The Physics of Rubber Elasicity*. OUP Oxford, 322.

Volokh, K. (2016). *Mechanics of Soft Materials*. Singapore: Springer, 337.

Zheng, Z., Ye, H., Zhang, H., Zheng, Y., and Chen, Z. (2020). Multi-Level K-D Tree-Based Data-Driven Computational Method for the Dynamic Analysis of Multi-Material Structures. *Int. J. Mult. Comp. Eng.* 18, 421–438. doi:10.1615/INTJMULTCOMPENG.2020035167

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

GLOSSARY

$\boldsymbol{\Omega_1}$ Weight matrix for the first differences

$\boldsymbol{\Omega_2}$ Weight matrix for the second differences

$\boldsymbol{A}$ Green Lagrange strain tensor

$\boldsymbol{b}$ Volumetric forces

$\boldsymbol{C}$ Right Cauchy Green tensor

$\boldsymbol{D_1}$ First differences matrix

$\boldsymbol{D_2}$ Second differences matrix

$\boldsymbol{f}^e_{\text{ext}}$ Element external nodal forces

$\boldsymbol{f}^e_{\text{int}}$ Element internal nodal forces

$\boldsymbol{f}_{ext}$ External nodal forces

$\boldsymbol{f}_{int}$ Internal nodal forces

$\boldsymbol{N_i}$ Principal referential direction of deformation number i

$\boldsymbol{P}$ PK1 (First Piola-Kirchhoff) stress tensor

$\boldsymbol{r}$ Arbitrary direction of the unit sphere in the reference configuration

$\boldsymbol{S}$ PK2 (Second Piola-Kirchhoff) stress tensor

$\boldsymbol{t}$ Surface forces

$\boldsymbol{u}$ Displacement field

$\boldsymbol{u^a}$ Displacement vector of node a

$\boldsymbol{W}$ Weight matrix for the system of equation

$\boldsymbol{X}$ Gradient of deformation tensor, $\boldsymbol{X} = \partial^t \boldsymbol{x}/\partial^0 \boldsymbol{x}$

∇ Nabla operator, $\nabla = \partial_i \boldsymbol{e}_i$

$\boldsymbol{\delta u}$ Virtual displacement field

$\boldsymbol{\delta W}$ Virtual work

$\boldsymbol{\delta W}_{ext}$ External virtual work

$\boldsymbol{\delta W}_{int}$ Internal virtual work

$\boldsymbol{\Gamma}$ Area in the reference configuraion

$\boldsymbol{\kappa}$ Bulk modulus

$\mathbb{P}$ Fourth order projector tensor $\mathbb{P} = \mathbb{I}^S - 1/3\boldsymbol{C}^{-1} \otimes \boldsymbol{C}$

$^0\boldsymbol{\xi}$ Local coordinates for the unit reference element

$\boldsymbol{\mu}$ Shear modulus

Ω volume in the reference configuraion

Ω volume in the reference configuraion

Ψ Continuum free energy function

$\boldsymbol{\psi_{ch}}$ Chain free energy function

$\square$ Volume of the unit reference element

$B_i(x)$ B-Spline basis function corresponding to the vertex $\boldsymbol{i}$

$\boldsymbol{h^a}$ Shape function of node a in the reference unit element

I_1^C First principal invariant of $\boldsymbol{C}$, $I_1^C = \lambda_1^2 + \lambda_2^2 + \lambda_3^2\lambda$

$\boldsymbol{J}$ Volume ratio $J = \det(X)$

J^e_j Jacobian of the element $\boldsymbol{e}$ at integration point $\boldsymbol{j}$

$\boldsymbol{MSE}$ Mean square error

$\boldsymbol{n}_{\textbf{dofs}}$ Number of DOFs per element

$\boldsymbol{n}_{\textbf{vert}}$ Number of vertices of the BSpline

$\boldsymbol{n_{el}}$ Number of elements

$\boldsymbol{n_n}$ Number of nodes

$\boldsymbol{n_{qp}}$ Number of quadrature points per element

$\boldsymbol{n_{qS}}$ Number of quadrature points on the microspheres

$\boldsymbol{P_{ch}}$ Chain mechanical behavior function, $\boldsymbol{P_{ch}} = d\psi_{ch}/d\lambda_{ch}$

$\boldsymbol{r_i}$ Director cosine of the vector $\boldsymbol{r}$ with respect to $\boldsymbol{N}_i$

$\boldsymbol{S_i}$ Principal value i of the PK2 stress tensor

$\boldsymbol{w_j}$ Quadrature weight for the integration point j of the element

$\boldsymbol{w_j^S}$ Quadrature weight for the integration point j on the sphere

$\boldsymbol{\lambda_{ch}^C}$ Affine chain stretch, $\lambda_{ch}^C = \sqrt{\boldsymbol{r} \cdot \boldsymbol{C} \cdot r}$

$\boldsymbol{\lambda_i^d}$ Deviatoric principal stretch i

$\boldsymbol{\lambda_{ch}}$ Non-affine chain stretch, $\lambda_{ch} = \boldsymbol{r} \cdot \boldsymbol{U} \cdot r$

$\boldsymbol{\lambda_i}$ Principal stretch i

U Volumetric contribution to the free energy function

W Deviatoric contribution to the free energy function

Advantages of publishing in Frontiers

OPEN ACCESS
Articles are free to read for greatest visibility and readership

FAST PUBLICATION
Around 90 days from submission to decision

HIGH QUALITY PEER-REVIEW
Rigorous, collaborative, and constructive peer-review

TRANSPARENT PEER-REVIEW
Editors and reviewers acknowledged by name on published articles

Frontiers
Avenue du Tribunal-Fédéral 34
1005 Lausanne | Switzerland

Visit us: www.frontiersin.org
Contact us: frontiersin.org/about/contact

REPRODUCIBILITY OF RESEARCH
Support open data and methods to enhance research reproducibility

DIGITAL PUBLISHING
Articles designed for optimal readership across devices

FOLLOW US
@frontiersin

IMPACT METRICS
Advanced article metrics track visibility across digital media

EXTENSIVE PROMOTION
Marketing and promotion of impactful research

LOOP RESEARCH NETWORK
Our network increases your article's readership