frontiers | Frontiers in High Performance Computing

TYPE Review
PUBLISHED 13 March 2025
DOI 10.3389/fhpcp.2025.1536501

OPEN ACCESS

EDITED BY
Mohamed Wahib,
RIKEN, Japan

REVIEWED BY
Peng Chen,
National Institute of Advanced Industrial Science and Technology (AIST), Japan
Edgar Martinez-Noriega,
National Institute of Advanced Industrial Science and Technology (AIST), Japan

*CORRESPONDENCE
Adam Weingram
✉ aweingram@ucmerced.edu
Xiaoyi Lu
✉ xiaoyi.lu@ucmerced.edu

RECEIVED 29 November 2024
ACCEPTED 15 January 2025
PUBLISHED 13 March 2025

CITATION
Weingram A, Cui C, Lin S, Munoz S, Jacob T, Viers J and Lu X (2025) A definition and taxonomy of digital twins: case studies with machine learning and scientific applications. *Front. High Perform. Comput.* 3:1536501. doi: 10.3389/fhpcp.2025.1536501

A definition and taxonomy of digital twins: case studies with machine learning and scientific applications

Adam Weingram[1]*, Carolyn Cui[1], Stephanie Lin[1], Samuel Munoz[2], Toby Jacob[1], Joshua Viers[3] and Xiaoyi Lu[1]*

[1]Department of Computer Science and Engineering, University of California, Merced, Merced, CA, United States, [2]Viterbi School of Engineering, University of Southern California, Los Angeles, CA, United States, [3]Department of Civil and Environmental Engineering, University of California, Merced, Merced, CA, United States

As next-generation scientific instruments and simulations generate ever larger datasets, there is a growing need for high-performance computing (HPC) techniques that can provide timely and accurate analysis. With artificial intelligence (AI) and hardware breakthroughs at the forefront in recent years, interest in using this technology to perform decision-making tasks with continuously evolving real-world datasets has increased. Digital twinning is one method in which virtual replicas of real-world objects are modeled, updated, and interpreted to perform such tasks. However, the interface between AI techniques, digital twins (DT), and HPC technologies has yet to be thoroughly investigated despite the natural synergies between them. This paper explores the interface between digital twins, scientific computing, and machine learning (ML) by presenting a consistent definition for the digital twin, performing a systematic analysis of the literature to build a taxonomy of ML-enhanced digital twins, and discussing case studies from various scientific domains. We identify several promising future research directions, including hybrid assimilation frameworks and physics-informed techniques for improved accuracy. Through this comprehensive analysis, we aim to highlight both the current state-of-the-art and critical paths forward in this rapidly evolving field.

KEYWORDS

digital twin, high-performance computing, machine learning, artificial intelligence, world models

1 Introduction

The concept of digital twin (DT) is gaining traction in both academic and industry contexts, as shown in Figure 1 (Haße et al., 2022). DT-based approaches aim to offer enhanced forecasting abilities (i.e., "what if" scenarios) that more closely represent reality (Thelen et al., 2022a). Through the use of data from the physical system with which they are associated, they improve over time. There are countless examples of computer models being used to great effect in real scientific and industrial applications, ranging from simulations of molecular dynamics to quantum computer models, to global weather (Stocks et al., 2024; Liu et al., 2021; Taylor et al., 2023; Schmude et al., 2024; Dunbar et al., 2024). A DT goes further and allows for even more in-depth analysis by focusing on a *single instance* of a physical system and using finer-grained and more accurate modeling techniques (Grieves, 2014).

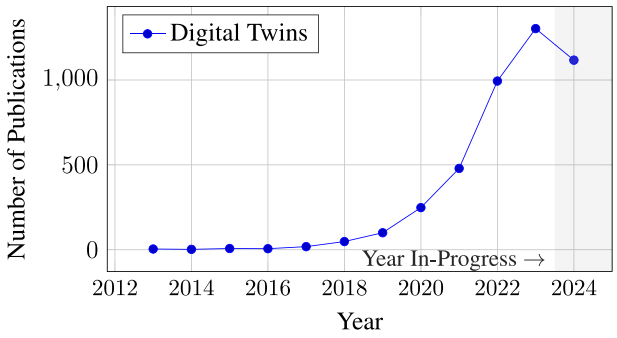

FIGURE 1
Number of publications related to digital twins. Note that the numbers from 2024 are incomplete and thus an underestimate as only published works are shown.

In recent years, various deep neural network-based machine learning techniques have shown promise in a wide array of prediction, analysis, and even general-purpose reasoning tasks (Touvron et al., 2023a,b; Radford et al., 2019). Examples include language models tuned to follow instructions (Brown et al., 2020), models capable of generating visual and auditory art (Yang et al., 2023), AI systems capable of scoring highly on difficult mathematics competitions (Trinh et al., 2024), programs that use a variety of techniques to give a model the ability to think step-by-step (Wei et al., 2022; Yao et al., 2023; Besta et al., 2024), and models developed to predict the structure of proteins (Jumper et al., 2021; Abramson et al., 2024). Other recent work has conducted early investigations on how generative AI models can be used for simulated prediction tasks (Yang et al., 2024). Although it is conceivable that the current level of growth is unsustainable for general techniques due to compute requirements or lack of data availability, there are already many areas where such techniques can be applied in the context of HPC and scientific computing. Furthermore, digital twins offer opportunities to take advantage of previously untapped data streams for learning tasks.

A digital twin is, at the fundamental level, a learning system. It must react to changes in the behavior of the physical system by updating its internal knowledge of the physical system's dynamics so that it can more accurately model the physical system's behavior. A digital twin without this learning characteristic is more accurately described as a simulator (we discuss our definition in depth in Section 2.4). For these reasons, digital twins and AI/ML techniques are inherently synergistic. With few exceptions, AI/ML models improve with exposure to and adequate processing of increasing amounts of *high-quality* data (Halevy et al., 2009; Sun et al., 2017; Sutton, 2019). Due to the assumption that digital twins are associated with a single physical system, one can assume that the data collected will be more specific and more high quality as a result.

However, the promise of digital twins cannot be fully realized without adequate computing power. Recent data-driven "frontier" models were reported to have taken millions of GPU-hours to train, effectively requiring high-performance computing (HPC) resources (Touvron et al., 2023a). Luckily, fine-tuning of pre-trained models can be performed with fewer resources (Hu et al., 2021; Dettmers et al., 2023). However, we identify HPC as a key component of digital twin development moving forward.

1.1 Key contributions

Through this work, we aim to achieve the following:

1. **Definition:** present a simple and generalizable language for formulating digital twins and their associated procedures that enables building on a rich body of existing literature from multiple disciplines.
2. **Fundamentals:** survey the foundational concepts upon which digital twins are built supported by a review of a carefully curated set of digital twin case studies.
3. **Taxonomy:** introduce a taxonomy for machine learning-enabled digital twins rooted in an analysis of the literature.
4. **Review and future directions:** present case studies demonstrating the state of the art in digital twins and explore promising future research directions and enabling technologies with a specific focus on scalable digital twins.

1.2 Related studies

There are other digital twin survey studies that are worth specifically mentioning. There are a set of two surveys (Thelen et al., 2022a,b) that provide a comprehensive view of digital twins as they exist today. These surveys focus on discussing the many existing techniques that can be combined to design digital twins, which are covered at a high level. Jones et al. (2020) attempts to characterize digital twins from a manufacturing perspective but does not discuss internal details of the digital twin. Haße et al. (2022) suggests certain digital twin design principles that may be useful in business contexts. VanDerHorn and Mahadevan (2021) uses a case study of a container ship to motivate the use of digital twins.

Other surveys choose to address the area of digital twins comprehensively, but at a higher level of abstraction, or in the context of specific applications. In this survey, we highlight the distinctive learning component of digital twins. Our analysis specifically addresses the interaction between machine learning methodologies and digital twins.

2 Preliminaries and digital twin definition

The term "digital twin" has been used to describe a wide variety of different approaches to modeling, making the design of a single comprehensive abstraction difficult. However, there are certain processes and rules that apply to many examples of digital twins that we find particularly interesting or useful. We also notice that there are numerous common elements across the world modeling, digital twin, and dynamic modeling domains. To provide a clearer understanding of digital twins (DTs), we present a *focused definition* and a *high-level schematic* that give us insight into their core functionalities. In addition, we break down the DT into a set of

abstract tasks, illustrating the specific operations that occur within digital twins. We then connect each of these tasks to a set of digital twin axioms. Furthermore, we aim to unify the key underlying concepts for digital twins by grounding them in the context and language of established fields such as control systems, decision processes and reinforcement learning, world modeling, machine learning, as well as existing works on digital twins specifically.

There is a substantial amount of prior work concerning sequence models and decision-making processes. These include Markov decision processes (POMDP) (Åström, 1965), the probabilistic graphical models (PGMs) (Pearl, 1985) that inspired (Kapteyn et al., 2021), and recent work on learned interactive environments (Valevski et al., 2024). We draw on many of these ideas to build a definition suited to the task of digital twinning.

2.1 Definitions

2.1.1 Physical system

The physical twin or physical system is the object or system that the digital twin attempts to mirror through modeling and updates based on observations from the real world. Some examples include a nuclear power generation plant, a manufacturing facility, and a spacecraft operating far from human reach on the moon (Digital Twins, 2023; Jones et al., 2020; Allen, 2021). A key feature shared by all of these examples is the availability of data from the physical system; each can be instrumented to provide data for the DT using sensors, internet of things, or even human observers. It is also possible to create the digital twin before or alongside the physical twin, as accurate models can be used for tasks such as finding optimal sensor placements (Wang et al., 2024). In other cases, the digital twin may be created after the physical twin (Grieves, 2014).

2.1.2 Digital twin

A digital twin is a virtual representation of a single instance of a physical system. Critically, a digital twin possesses a mechanism to integrate or "assimilate" data collected from the physical system to better estimate the state or predict the behavior of the physical system. Also at the core of a digital twin is a model that attempts to mirror the behavior or characteristics of the physical system for a set of goals defined by those designing the DT (Grieves, 2014; Zhang et al., 2023). While DTs themselves are not *only* simulations, the core of every DT contains a simulation, capability, or model that is constructed to depict the physical system. Simulation is a well-studied field, and there are numerous lessons that can be adapted for use within DTs. Models or simulations can be used for a variety of purposes, from testing performance characteristics to making predictions about future behavior and potential failures in the physical system (Grieves, 2014; Kapteyn et al., 2021). Outside of DT, models need not be overly complex; there are many situations where highly abstract models are sufficient, or even preferable. However, digital twins are most powerful when used in non-trivial cases where the models are detailed, highly-annotated, and have outputs that closely resemble the behavior of the physical system (Grieves, 2014). It is important to note that a given DT may employ multiple simulations or models, in which different approaches are selected for different contexts. Light-weight versions of more detailed models that eschew certain details that are not required for the specific task under consideration may be essential for applications where latency, power, or available computational capabilities are a concern (Grieves, 2014).

2.2 Digital twin phases

We outline the fundamental stages that a digital twin undergoes as the **construction phase**, **tuning phase**, **control and insight phase**, and **decommissioning phase**, all of which are visualized in Figure 2 (Kapteyn et al., 2021). The DT is designed and implemented in the construction phase according to the design resources of the physical system or by studying the physical system if it already exists. The DT is then tuned and calibrated using data collected from the physical system during the tuning phase. At this point, the physical system exists and is operational, but the digital twin is not yet providing insights or making control decisions. The insight and control phase is the longest-running portion in which the digital twin provides insights, controls, or both. Simultaneously, the DT updates its internal model based on data gathered from the physical system as they becomes available. Decommissioning takes place when the links between the physical system and the digital twin is severed. Data from the physical system are no longer available to the digital twin, and the digital twin no longer provides insights or controls the physical system.

2.3 Axioms

Here, we state a set of axioms that we use to guide our definition and problem formulation.

1. States are influenced by past states and any events that occur. Therefore, we say that states are *causally* linked. We explicitly do not consider the possibility that future states or actions impact past states.
2. A digital twin receives data from the physical system in the form of **observations**, which may or may not represent the true underlying state of the physical system.
3. The digital twin is a learning system; data from the physical system are assimilated into the internal model of the digital twin, providing the internal model with additional *information* about the behavior of the physical system.
4. A digital twin possesses what we refer to here as the "prediction anytime" property. Once constructed, a digital twin can start generating predictions immediately, although predictions might initially lack accuracy or precision. As more data are assimilated, the predictive capabilities of the digital twin improve. This is similar to the "anytime" property (Zilberstein, 1996) for computation where the algorithm's solution improves by some metric when given increasing amounts of processing time.

FIGURE 2
A high-level overview of digital twin phases.

2.4 Digital twinning is sequence modeling

After reviewing the existing digital twin literature, we observe that previous work does not adequately capture the centrality of *learning* in the digital twin context. Therefore, we start by formulating the problem of digital twinning as one of sequence modeling, which has strong support in the literature (Yang et al., 2024; Micheli et al., 2023; Lin et al., 2024; Valevski et al., 2024). States $s_t \in \mathcal{S}$ at discrete points in time t causally influence future states $s_{>t}$, as demonstrated in Equation 1. Here, $\mathcal{S}$ is the set of all possible states. In the episodic case, an episode τ consists of a set of T states $\tau := \{s_i\}_0^T$ where $\tau \in \Upsilon$ and Υ is the set of episodes. The central goal of a digital twin is to predict the sequence of states that most closely match the actual states of the physical system at each discrete point in time.

In the simplest case, a simulator must be able to predict future states given some initial state or trajectory "stub" (i.e., an unfinished trajectory). The simulator's job is to complete these trajectories, which are sequences of states and events. We identify each component with a specific time $t \in \mathcal{T}$ where $\mathcal{T}$ is the index set for a given trajectory. Note that we do not require the wall time between time steps to be consistent.

Our formalization assumes that the particular physical system can be mirrored using a digital twin that operates on discrete moments in time. It assumes that individual events to not occur at *exactly* the same time. It does not assume that any particular assimilation or update method is used, and does not assume that the actual amount of time that passes between different discrete time steps is uniform. This is beneficial because it means that the practitioner can choose the assimilation method and simulation strategy that fit their particular needs. The relationship between the physical state, the observation, and the belief state is shown in Figure 6.

2.4.1 System function modeling

If, for a moment, we assume that both the physical system and the digital twin are deterministic and reduce the physical system to a simple function $f : C_{\text{full}} \rightarrow \mathcal{S}$ where $C_{\text{full}} := \{s_i\}_0^t \subseteq \mathcal{S}$ is the set of states that have already occurred, then the goal of twinning is to learn some function $\widetilde{f} : C_{\text{full}} \rightarrow \mathcal{S}$ that approximates f. Although the optimal digital twin function f^* would produce states that *exactly* match those of the physical system function[1], $f^* = f$, this outcome is unlikely to be achieved in practice.

$$\widetilde{f}(\{s_i\}_0^t) = s_{t+1} \tag{1}$$

This formulation abides by Axiom 1. However, using the entire history of a trajectory may be prohibitively expensive for two reasons. The first is in long-running or high-frequency situations, *storing* entire trajectories may occupy a large amount of space. The second is that even when a large amount of space is available, the cost of determining the next state by *processing* all of the past states grows, at a minimum, linearly with T. Even worse, if we cannot assume that the trajectories are episodic, then the space and processing cost are unbounded.

As this is clearly unacceptable, one must find a way to reduce the size of the recorded or relevant history. We represent this subset of the history as C_{sub}. If one can assume that the sequence is Markovian, then one only needs to know the state s_t to determine s_{t+1} and can ignore the rest of the history (Weisstein, 2024). However, many applications do not have the luxury of assuming Markovianness, or cannot assume Markovianness without storing the entire history within each state (Sutton and Barto, 2020; Tennenholtz et al., 2023). To illustrate the limitations of the Markov property, consider a natural language processing system trying to understand the meaning of pronouns[2] to help predict future sentences. A purely Markovian model that only looks at the current sentence would struggle to determine what "**it**" refers to in a passage such as: "The *airplane* flew two loops over his head. He looked up. *It* was red with yellow stripes." In a Markovian model, only the most recent sentence is used, so the system will lack the information that "it" refers to the airplane. We show this example visually in Figure 3. In practice, many language tasks need to consider contextual information from multiple previous sentences or paragraphs. Hence, these tasks violate the strict Markov assumption because crucial information can be buried further back in the text than just the immediately preceding sentence.

Naturally, then, the question becomes "if we must keep history, how do we determine which parts of the history to throw away and which parts to keep?", or more importantly, "**which data should we use during computation** to determine the next state?" A naive approach to generate C_{sub} would be to treat it as a sliding window of size W, as shown in Equation 2.

$$\widetilde{f}(\{s_i\}_{t-W}^t) = s_{t+1} \tag{2}$$

While easy to implement, bounded in cost, and simple to understand, the windowed approach clearly fails when there are long temporal dependencies between states. A simple but unfortunate example of this might be a medical model that fails to take into account an accident involving radiation exposure early in

1 Remember that we are discussing the deterministic case here.

2 Pronoun resolution is a well-studied but difficult problem in NLP (Denis and Baldridge, 2007).

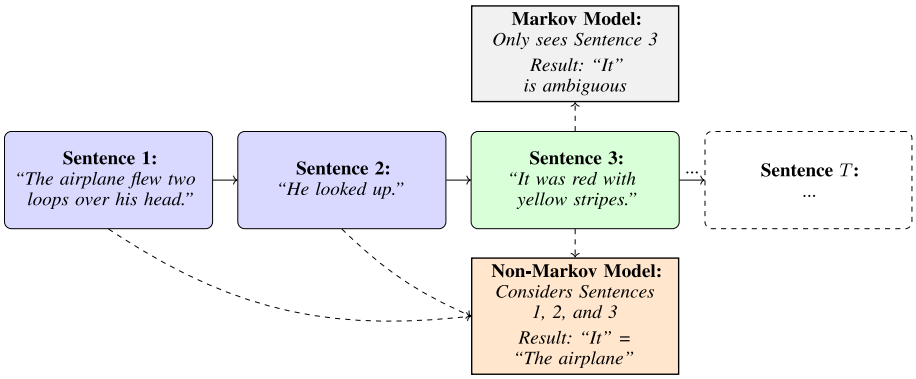

FIGURE 3
Markovian and Non-Markovian models compared through a simple example.

a person's life that causes him to become observably ill decades later, despite intervening years of good health. One potential solution is to use some heuristic function $k : \mathcal{S} \rightarrow \mathbb{R}$ that at each time step assigns a value to the state, and therefore whether it is worth keeping or whether it should be thrown away. This approach is in many ways analogous to human record keeping. $C_{\text{sub}}^{\text{prio}}$ is represented by a priority queue of bounded size l, where $|C_{\text{sub}}^{\text{prio}}| \leq l$. States are inserted into $C_{\text{sub}}^{\text{prio}}$ until it becomes full. When $|C_{\text{sub}}^{\text{prio}}| = l$, either the current state or lowest-priority entry in $C_{\text{sub}}^{\text{prio}}$ is dropped depending on which is less valuable. Then, only the current state and the most important parts of the history are taken into account when determining the next state, as shown in Equation 3.

$$\widetilde{f}(s_t, C_{\text{sub}}^{\text{prio}}) = s_{t+1} \tag{3}$$

Yet, this approach requires the creator of the digital twin to have advance insight so that the heuristic function k can be concretely defined, making it unattractive for most applications. Another solution is to use a fixed-size "hidden" state h similar to how a recursive neural network (RNN) operates, which we show in Equation 4. This is attractive if computation on the history is infeasible and the modeling approach is capable of *automatically* learning how to generate these hidden states.

$$\widetilde{f}(s_t, h_t) = s_{t+1}, h_{t+1} \tag{4}$$

However, empirical results with RNNs show that this approach may have significant limitations compared to history-preserving attention-based approaches, demonstrating how important history can be (Peng et al., 2023). In addition, the method of creating these hidden states is not clear for modeling approaches that are not based on neural networks, making the technique less generalizable.

The problem of history is not unique to digital twins nor to world modeling, and has been extensively studied in the machine learning community (Huang et al., 2024; Tennenholtz et al., 2023; Lin, 1992; Eysenbach et al., 2019; Fedus et al., 2020). Overall, the best approach to history keeping is context- and application-dependent; currently, there is no single solution. Different modeling approaches require different amounts of history, and may require keeping the history in different forms. This pattern is clearly evident across many previous works (Das et al., 2024; Zeng et al., 2023; Bodnar et al., 2024; Yin et al., 2024; Liu H. et al., 2024).

Furthermore, if we assume that the dynamics of the physical system are instead *stochastic* rather than deterministic, then our formulation must change to account for this. Instead of the physical system function returning a single "next state" s_{t+1}, it instead returns a probability distribution over all possible states, $f(s_{t+1}|s_{\leq t})$.

Stochasticity also introduces additional complexity for predictions, especially those that extend far into the future. Luckily, there are numerous methods for computing predictions with stochastic models, many of which are highly scalable and fit for HPC applications, such as the various Monte Carlo methods, which are often described as "embarrassingly parallel" due to their usual lack of inter-simulation dependencies (Metropolis and Ulam, 1949; LeBeau, 1999; Meeds and Welling, 2015). Another benefit of this approach is that ensemble forecasting often allows one to quantify uncertainty in the predictions of a stochastic model (Leutbecher and Palmer, 2008; Parker, 2013).

2.4.2 Event-based sequence modeling

Intuitively, changes to the state of the physical world we live in do not "just happen." Instead, changes are caused by *events*. The system function-based modeling strategy is conceptually simple, however, it lacks the capacity to explain the underlying causes of state transitions. Therefore, *interventions* are also ignored, despite the vital importance of these components in "what-if" analyses.

Here, we formulate the digital twin as an *event-based sequence model* that explicitly takes stochasticity into account. This formulation, shown visually in Figure 4, is heavily inspired by Markov Decision Processes (MDP), but does not require the Markov Property.

- **Event:** $a_t \in A(s_t) \subseteq \mathcal{A}$ where the shorthand $A(s_t)$ gives the set of events that can take place given a state s_t. The set $\mathcal{A}$ represents all possible events across all states. Critically, events can be related to specific interventions, such as a choice of how much fertilizer to use in an agricultural application, or related to the natural behavior of the physical system,

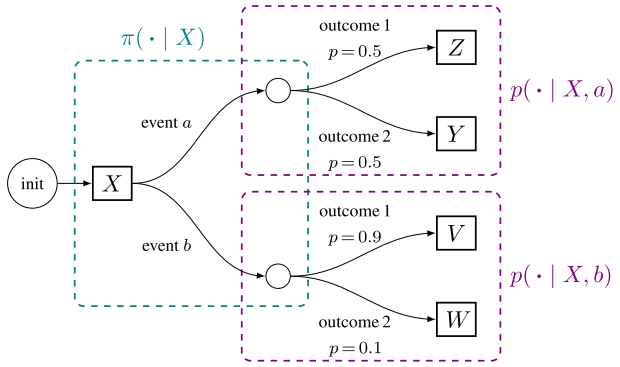

FIGURE 4
The sequence of event decisions and simulated dynamics can be visualized as a tree or directed acyclic graph.

such as whether or not it rains. The ability to "force" certain events to occur (i.e., interventions) is not easily representable in the simpler system function approach, nor in standard probabilistic graphical models, where causal relationships are defined only between states. Interventions may not have deterministic effects on the state, so the ability to model stochastic event outcomes is critical. Put simply, interventions let us perform "what if" predictions. MDPs fail to capture the influence of history. Though one can theoretically include history *in* the state to get close, this is often very difficult to *learn* effectively due to the state space explosion (Sutton and Barto, 2020).

The use of events between as the fundamental driver of state changes also more closely resembles reality. In the real world, state does not change on its own. Instead, events cause changes that are reflected in the state.

- **Event selection policy:** $a_t \sim \pi(a_t|s_{\leq t})$ is responsible for selecting the next event to simulate in the sequence. The event selection policy can heavily influence the dynamics of the digital twin, but can be "overridden" to test interventions.
- **Physical transition probability function:** $s_{t+1} \sim p(s_{t+1}|s_{\leq t}, a_t)$, which represents the dynamics of physical system when events occur.

This framework, while more descriptive, is equivalent to the system function modeling approach described previously when a dummy event is used and always occurs with complete certainty. The outcome of the dummy event, given the current state, leads to the next state.

2.4.3 Partial observability

Physical systems cannot be perfectly observed by digital means for a variety of reasons, including measurement error, limitations of sensors, the scope of what they can reliably observe, and even the discretization that is required in order to represent data in a digital format. A digital twin must rely on observations $o \in \Omega$ to improve its internal knowledge of the physical system. Here, Ω is the set of all possible observations. The observation function $O : \mathcal{S} \to \Omega$ represents the process of actually observing some property of the physical system, including any noise or error that such an action may entail. The handling of partial observability is required by Axiom 2.

- **Belief state:** $b : \mathcal{S} \to [0, 1]$. Every belief state is a probability distribution over $\mathcal{S}$. Consequently, we can say that in the discrete case $\sum_{s \in \mathcal{S}} b(s) = 1$, and in the continuous case $\int_{s \in \mathcal{S}} b(s)\, ds = 1$. This naturally gives rise to the question of how a belief state can be updated, given that one of the most prominent goals of the digital twin is to assign higher probabilities to states that actually occur at their respective times. The answer to this question lies in the **data assimilation** approach (see Section 3.3) chosen by the practitioner.

  Predictions can be conditioned on ground-truth observations from the physical system as they become available, similar to the teacher forcing technique (Williams and Zipser, 1989) used in recurrent neural network training. This approach helps prevent divergence between the physical system and digital twin models.
- **Observation:** $o_t \in \Omega$ where Ω is the set of all observations. An observation o_t is obtained through the observation function $O(s_t)$. An observation represents a partial or potentially noisy view of the physical system, often collected by some sensor as the actual state of the physical system cannot be directly examined. The observation function is an abstract representation of the sensing process.

2.4.4 The digital twin improvement loop

Previous work has conceptualized the prediction of future states as a sequence modeling task and proposed formulations to describe this behavior (Yang et al., 2024). In fact, the formulation we present above has much in common with these works. However, previous work misses the most critical aspect of the dynamic world model: updates from the physical system. In past work, samples from the physical system were used only to condition future predictions, and not to update the model's underlying representation of the physical system. Note that both state estimation and system dynamics prediction must be updated. We explicitly include the update step in our formulation.

A digital twin can operate in either an **online**, such as in Figure 5, or an **offline** capacity. In the conceptually simpler online case, the digital twin internal state evolves in lockstep with the physical system's state. The digital twin's predictions are compared to physical system observations immediately upon availability. In the offline case, the digital twin can evolve entirely independently of the digital twin while still assimilating information in the form of observations from the physical system as they become available. In our formulation, updates may come from the specific physical system instance associated with the digital twin or possibly from other instances in a manner similar to off-policy reinforcement learning (Sutton and Barto, 2020).

2.5 Controlled-environment agriculture example

Agricultural AI involves a spectrum of attributes that include interacting biophysical processes (both known and unknown), high

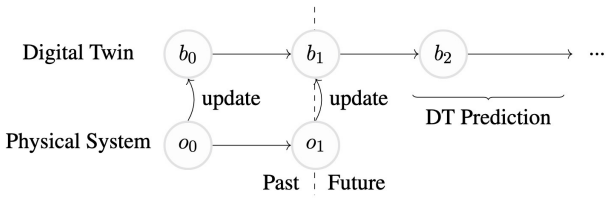

FIGURE 5
The physical system and its associated digital twin are shown undergoing discrete transitions as time progresses during the insight phase. This **online** digital twin is coupled to the physical twin through regular updates up until the current moment. The digital twin is then shown making predictions of the future trajectory of the physical system, annotated by "DT prediction." In this example, no control component is displayed.

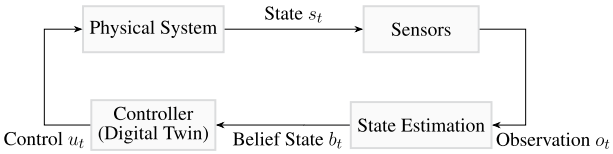

FIGURE 6
A focused view of the relationship between actual states, observations, and the belief state in a control context. For simplicity, the details of the digital twin are not shown.

uncertainty, multiscale spatio-temporal data, a range of decision timescales, and multiple decision point workflows (Kalyanaraman et al., 2022; Ng et al., 2023). Thus, it makes an ideal testbed for DTs. To illustrate our formulation, we present Controlled Environment Agriculture (CEA), or greenhouse-based food production systems, as an example that uses our event-based sequence model. CEA has several advantages for sustainable food production as they are optimized for resource efficiency and continuous high-quality production. Such integrated physical and simulated environments are the basis for autonomous food production systems (Avigal et al., 2021).

This example CEA system is equipped with sensors that measure the time of day, temperature, relative humidity, average soil moisture, and the CO_2 concentration once per time step (Axiom 2). For simplicity, we assume that these measurements perfectly describe the true state of the physical system, so $o_t = s_t$[3]. A greenhouse environment responds to both natural phenomena *and* controlled events, such as fluctuations in outdoor temperature and the activation or deactivation of climate control systems such as irrigation and heating. These events are represented by a_t. We know that s_{t+1} depends on $s_{\leq t}$ due to physical processes such as evaporation and heat exchange with the outside world (Axiom 1). It is also clear that events should have some impact on the state of the system. The model $p(s_{t+1}|s_{\leq t}, a_t)$ *predicts* the time of day, temperature, relative humidity, soil moisture, and CO_2 concentration for the next time step based on the history and an event. During data assimilation, p learns these dynamics. For example, the model may learn that the event of turning on the irrigation system for the current time step often results in an increase in soil moisture for the next time step, or more rarely a slight decrease if, perhaps, the water tank is empty (Axioms 3 and 4). However, some exogenous events are not controllable but are predictable, such as the outside temperature becoming colder at night (event), impacting the internal temperature of the system (state). The event selection policy π learns to predict such events. Together, π and p form the model component of the digital twin. This example demonstrates how an event-based sequence model provides a structured, learnable representation of real-world dynamics, forming the core of a digital twin that continuously refines its predictions over time.

3 We highlight that this is a simplifying assumption—partial observability is a more general case.

3 Fundamental tasks

In order to disambiguate digital twins from conceptually similar techniques and to clarify the means through which digital twins operate, we present a set of fundamental tasks that digital twins perform. We start by explaining each task at an abstract level and then discuss concrete methods for completing each task along with examples from the literature.

The digital twin problem can be thought of as a system containing two co-evolving components: the physical system and the digital twin tasked with mirroring the physical system. A visual representation of this pattern is shown in Figure 6.

3.1 Data acquisition

Data acquisition is the process of sensing, collecting, and transmitting data from their origin and other monitoring equipment in a physical system to a digital representation that can be stored, retrieved, and used by the digital twin (Correia et al., 2023). The scope of the digital representation of the physical system can vary widely ranging from individual machines or a group of machines to entire cities, farms, or industries (Friederich et al., 2022; Uhlemann et al., 2017). Data acquisition can be manual, automated, or some combination of the two. The manual acquisition entails documenting changes that happen in the system through direct human action. However, manual collection is slow, tedious, expensive, and often low-frequency. In the past decades, automated sensors have become widespread and low cost (Mao et al., 2019). As a result, many digital twins implement automated data acquisition (van der Valk et al., 2020).

3.1.1 Pre-deployment

As outlined in Figure 2, the a digital twin must first be constructed and tuned before it can be used (during "control and insight"). Data from the physical system play an important role in both of these phases. The nature of the data depends on the digital twin being constructed, what knowledge is already available, and what sensing systems can be deployed.

3.1.1.1 Design artifact re-use

In general, the manufacturing process for complex physical items involves creating annotated three-dimensional models, along with bills of processes (BOPs) and bills of materials (BOMs) (Grieves, 2014). When these resources are available, crafting digital twins can become simpler and less expensive as these design artifacts can be re-used or modified to support the creation of the DT. In many cases, DTs may also be able to exploit the relationships between data stored in existing comprehensive lifecycle management systems.

3.1.1.2 Manual measurement and 3D design

One of the more common types of modeling in the literature is the manual creation of three-dimensional computer-aided design (CAD) models of the physical system (Kapteyn et al., 2021; Matulis and Harvey, 2021). A human's "highest-bandwidth" method of absorbing information is through visual sight, and a three-dimensional (3D) model is conducive to this as it can be directly viewed (Grieves, 2014). While helpful for visualizing data, the same models can also be used within simulations. There are a variety of approaches to creating these models, but a common approach is to have a human worker take measurements of the physical system, then use the measurements as well their own knowledge of the system to create a three-dimensional representation using 3D modeling software. This technique can produce extremely detailed and accurate 3D models, but is extremely costly and the results entirely depend on the skills of the human worker.

3.1.1.3 Active and passive sensing

As the name implies, laser scanning uses an array of lasers to find precise distances between the scanner and the object being studied. These distance measurements are combined with known information about the orientation of the scanner relative to the object to create a three-dimensional point cloud. The point cloud is then transformed into a 3D mesh that under ideal conditions, precisely reproduces the object's physical representation in virtual space (Scott et al., 2003).

Another technique is photogrammetry, where 3D models are recreated from many 2D images. Unlike laser scanning, photogrammetry generally does not require specialized equipment in the field other than a camera and some measurement tools. Photogrammetry-derived 3D models have been used in popular software tools such as Google Earth (Google Maps 101, 2019). The relatively low cost of this technique makes it attractive for creating DT assets, however, the accuracy and precision of the resulting 3D models are often lower than both laser scanned and manually created assets.

3.1.1.4 Generative AI

High quality 3D meshes are necessary for many digital twin applications. Recent advancements in generative artificial intelligence (GenAI) have led to massive improvements in the capabilities of publicly available GenAI technologies for creating meshes. Siddiqui et al. (2023) explored how transformer models can be used to generate triangle meshes, thereby creating 3D models without manual design effort. Their MeshGPT approach demonstrated significantly better performance than previous mesh generation solutions in multiple benchmarks while retaining its ability to generate novel shapes, that is, not directly output training data. This is vital for DT applications, as DTs must represent single instances of physical systems, no two of which are exactly alike.

3.1.2 Post-deployment

Once the digital twin is constructed, tuned, and ready for the control and insight phase, continuous data streams must be created, establishing the link between the physical system and the digital twin as required by Axiom 2. Some examples of the tools used to collect these data are sensors, unmanned aerial vehicles (UAVs) and satellite imagery (van der Valk et al., 2020; Huang et al., 2021). In many cases, these data are collected autonomously, often by systems that can be described as "internet-of-things" (IoT). IoT frequently serves as a way to get an efficient, reliable, and continuous flow of data for the digital twin. This is because the combination of frequent data acquisition (high frequency), remote access, and automatic collection are highly desirable for DT applications. An example of this is from Guo et al. (2023) where the authors employed IoT to assist in acquiring data for their performance on Array Antennas Segovia and Garcia-Alfaro (2022).

This information is recorded and stored in large databases located on cloud servers or data application servers, which we discuss in detail in Section 3.2 (Correia et al., 2023). The collected data are divided into categories such as **static information**, which includes specifications or performance information that do not appreciably or regularly change, and **dynamic information**, which includes parameters and measurements that change or are subject to change over time (Friederich et al., 2022; Uhlemann et al., 2017).

Ideally, these data are collected and stored seamlessly; however, several factors prevent this from being the case. Because digital twins often ingest data from multiple disparate sources, these data are not uniformly structured, meaning that the raw inputs vary in format between structured data like tables and unstructured data like videos. Raw data can also be recorded in different serialization formats, adding an extra level of complexity when organizing for data analysis (Correia et al., 2023). Additionally, digital twins demand high precision, low latency, and continuously flowing data to provide real-time updates.

Optimally, the post-collection data would be updated and processed quickly enough for real-time changes to reflect almost immediately in the digital twin; however, slow computing times as well as problems with data storage space filling up from the large data pool prevent this from happening.

3.2 Data storage

Data frequency and reliability are crucially important to maintaining the link between the physical system and the digital twin, thus preserving accurate modeling, assimilation, and prediction. Reliability requires secure, fault-resistant, and well-documented access to the data. The high update frequency of physical twin sensor and data collection systems requires designs capable of accepting, storing, and organizing incoming information with minimal delay. This section tracks the flow of data from post-acquisition to storage and through organization, outlining

the methods used in data management and data storage, and highlighting notable trends in storage literature that pertain to digital twins.

We break the storage process into five subtasks, namely: (1) communication of data from the acquisition system to the DT storage system, (2) validation and annotation of the received data by the storage system, (3) organization of the data and write to storage media within the storage system, (4) retrieval of data by the digital twin, and (5) transmission of data from the storage system to the digital twin.

Data may originate from IoT sensors as outlined in Section 3.1, or alternatively, it might be synthetic in nature (Zheng et al., 2019). Before it is sent to long-term storage, the data must be pre-processed. The original analog electric signals of varying formats are often first converted to digital representations and preprocessed by edge devices to filter out noise, de-duplicate, detect errors and apply compression for more efficient communication (Li et al., 2023). The data, metadata, and information about the origin of the data are combined to form a package which is then sent to the storage system, generally over some kind of network.

Once the data arrive in the storage system, additional transformations may be performed. These include normalizing data to create uniformity among fields and records, allowing for easier retrieval and standardization (Gorelick et al., 2017). Other important pre-processing steps can involve indexing that will later allow more efficient access (Lü et al., 2011).

In general, data receipt, handling, management, and querying take place within a central storage system where all data are stored and maintained at one location (Lü et al., 2011). However, a single logical data storage system may actually consist of multiple specialist systems working together in a distributed fashion. Organizing the data by group ("silo") is one sharding method. Creating silos is a simple way to organize information systematically across the network. This method is formalized by decentralized policies such as outsourcing or splitting the organization into separate entities and providing these groups with individual budgets for infrastructure, thereby encouraging independent data management (Cromity and De Stricker, 2011). However, this configuration is frequently undesirable for building DTs as these silos restrict data sharing and access, and can lead to duplicate or incomplete data. Although sharding was common in the past, it has now become a problem when building DTs that require access to organization-wide data (Sun et al., 2020). Some works have called for a shift in the traditional approach to more accessible data through policies such as software-defined infrastructure or data ecosystems (Sun et al., 2020). Such a system can be created with the advice of experts of respective data areas.

Distributed data storage uses data that are physically scattered with limited access across clouds, data centers, edges, networks, etc. To connect disparate sources, Uhlemann et al. (2017) envisions a storage fabric infrastructure that can aggregate data into a more interconnected system. This idea is based on software-defined infrastructure, which offers a continuously adaptable infrastructure that is hardware-agnostic and scalable. This configuration shares similarities with the federated infrastructure resource pooling approach, in which each system remains autonomous while collaborating to share resources.

3.2.1 HPC storage challenges for digital twins

Digital twins create unique demands on HPC storage systems that extend beyond traditional scientific computing workloads. HPC centers are designed to support large-scale parallel computations; however, digital twin applications introduce additional complexities that require careful consideration. Whereas many large-scale scientific applications exhibit relatively regular access patterns, digital twins can generate data requests at varied frequencies and granularities, spanning multiple time scales.

As discussed in previous sections, this can require the continuous ingestion of sensor and observational data in real-time while simultaneously providing fast access to large volumes of historical data for model updates, analytics, and prediction tasks. This dual requirement has the potential to strain existing systems that typically assume more predictable I/O patterns.

Existing HPC deployments sometimes leverage advanced caching, burst buffers, or object-based storage to mitigate these issues, but the demands of continuous and unpredictable data queries can still result in performance bottlenecks (Liu et al., 2018; Romanus et al., 2015; Khetawat et al., 2019). Overall, the impact of existing parallel storage designs on digital twin applications has not been adequately studied.

3.3 Data assimilation

Data assimilation may be the most important and fundamental task that a digital twin performs, as the process serves as the bridge between information obtained from the physical system and the digital twin's internal understanding of the physical system. Assimilation is required as part of our definition of a digital twin (Axiom 3). The data assimilation process uses information from the physical instance to update and improve properties of or representations in the digital twin. Such updates may be useful for a variety of purposes, ranging from improving context awareness to tuning a model to more accurately predicting physical twin behavior, as shown in Figure 7 (Rodríguez et al., 2023). Data types, update frequency, veracity, and other properties can be determined by considering the requirements for the specific digital twin system being implemented (VanDerHorn and Mahadevan, 2021).

The specifics of the update strategy are also important to consider. Although the strategy chosen is often application-specific—there is no "one-size-fits-all solution"—there are methods that serve as excellent starting points once certain properties of the desired system are known.

We outline two distinct areas where data assimilation plays a role. The first is to improve **state estimation**, where observations are used to infer the state of the physical system. The second is to improve **prediction**, where the digital twin predicts the future states of the physical system or events that they may be influenced by. There are numerous ways in which one can approach the problem of data assimilation, though many are based on Bayes' rule, such as Kalman Filtering and Particle Filtering (Bertsekas, 2020). These smoothing algorithms are often implemented offline to refine state estimates by incorporating past and future data points. The accuracy of state estimation is strongly influenced by the chosen state representation, the fidelity of the dynamics model,

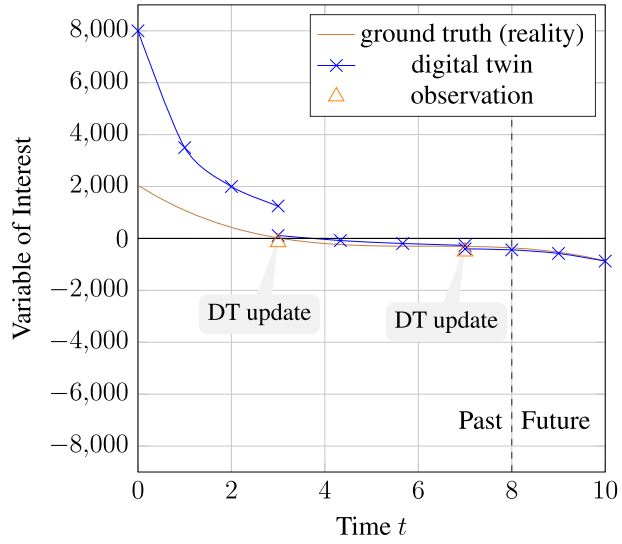

FIGURE 7
An abstract example of a DT model's predictions becoming closer to the ground truth, and thus more accurate over time as additional observations are taken into account. Blue "X"s represent the DT's prediction for a given time step.

the quality of sensor data, and the chosen estimation technique. Other more recent techniques involve pre-training or fine-tuning deep neural networks.

However, there is a key gap in the research. Although the crucial role of data assimilation in digital twin technology is widely acknowledged, its implementation remains a significant challenge and an area that requires further investigation. Current research on data assimilation within the context of digital twins is still in its early stages.

The reasons for this gap are multifaceted. The complex and heterogeneous nature of physical systems, coupled with the diverse range of modeling and simulation techniques used in digital twins, pose significant challenges for developing universal data assimilation methods (Grieves, 2014). Furthermore, the computational demands of real-time data assimilation, especially for high-fidelity digital twin models, can be substantial, requiring efficient algorithms and potentially specialized hardware (Thelen et al., 2022a,b). Addressing these challenges through rigorous research and development of robust, scalable, and computationally efficient data assimilation techniques will be crucial for unlocking the full potential of digital twin technology across various domains.

3.3.1 Deep neural networks

DNNs excel at learning complex patterns and relationships from vast datasets, enabling them to model intricate system dynamics and improve prediction accuracy (LeCun et al., 2015). In fact, the universal approximation theorem states that when the number of neurons is not bounded, a neural network is theoretically capable of representing any function and is therefore a "universal approximator," though *learnability* is not guaranteed (Hornik et al., 1989). Notably, transformers, recurrent neural networks (RNNs), especially long short-term memory (LSTM) networks, have shown promise in handling temporal dependencies in data, which is crucial for capturing the evolving behavior of physical systems. Furthermore, techniques like transfer learning and pre-training on large, generic datasets can accelerate the development of effective data assimilation models by leveraging existing knowledge.

3.3.2 Physics-informed neural networks

Traditionally, our understanding of physical systems has been built upon physics-based models. These models are rooted in fundamental physical laws and first principles. We translate these laws into mathematical equations and use constitutive models to describe how materials interact, their properties, and how forces affect them. The beauty of physics-based models is their interpretability. Each parameter has a clear physical meaning. Beyond prediction, physics-based models allow for extrapolation, meaning we can often predict behavior in situations *beyond* the data we have observed. However, these complex simulations can be computationally expensive, demanding significant time and resources. Also, in our efforts to simplify reality enough to represent its behavior with analytical or numerical models, we make assumptions that can sometimes lead to discrepancies between our model and the actual physical system.

Machine learning has been proposed as an alternative to these simulations and models, built on the idea that if we have enough data, we can train these models to have essentially the same behavior as a physics model. However, regular data-driven machine learning has drawbacks, including extremely poor out-of-distribution behavior. Furthermore, the results of an ML-based model can only ever be as good as the data it has seen so far. Critically, machine learning model architectures are not based on physical laws, and may predict impossible outcomes (Ritto and Rochinha, 2021).

Physics-informed neural networks (PINNs), thus, aim to take advantage of both approaches. PINNs integrate the rigor of physical laws with the adaptability of neural networks, and they often work by introducing known physics as a regularization term (Karniadakis et al., 2021). In simple terms, the neural networks are being taught to respect the fundamental rules of physics while still learning from data; that is, the models should fit the data while ensuring the output is consistent with the physics equations we know about. This leads first to improved accuracy. By incorporating physical knowledge, we can guide the learning process and enhance predictive capabilities, especially in data-limited scenarios. Second, PINNs promote enhanced interpretability. The physical constraints act as metaphorical guardrails, helping observers understand the model's predictions and identify potential inconsistencies. Lastly, physics-informed models often have less stringent data requirements. Physical knowledge acts as a powerful supplement when data quantities are limited, making PINNs more versatile and sample-efficient. This type of model holds immense potential for building more robust digital twins, as DTs are modeling the real world to begin with.

3.3.3 Attention and transformer architectures

Transformer models have been shown to be extremely successful for language modeling tasks, and have became popular for their strong ability to generate coherent text from prompts and even follow user instructions (Radford et al., 2019; Brown et al., 2020; Touvron et al., 2023a,b; Ouyang et al., 2022). In a natural language context, tokens represent words, or more frequently, sub-word chunks arranged in order (Sennrich et al., 2016). Tokenizing with sub-word units has demonstrated benefits compared with entire-word dictionary encoding and is intuitively well-founded (Sennrich et al., 2016). A positional encoding is used to capture the temporal dependencies between tokens as the attention mechanism is not by itself order-aware (Vaswani et al., 2023).

By the definition presented in Section 2.4, digital twins are a form of time series modeling because they operate over the temporal domain (i.e., time steps). Unfortunately, for such time series modeling applications, traditional applications of the transformer architecture often struggle to perform well and are, in fact, frequently outperformed by extremely simple alternatives that do not use the attention mechanism (Zeng et al., 2023; Das et al., 2024). To address these shortcomings, recent work has proposed the use of an "inverted" transformer design, frequently referred to as an iTransformer (Liu Y. et al., 2024). The key to this idea is using the attention mechanism across variates (i.e., variables) rather than across time steps, treating the entire input time series for each variate as a single token. In the process, the model is able to learn multivariate correlations. One benefit of this approach is its greater interpretability compared to traditional transformers, as the score maps from iTransformers can be easily inspected. The original iTransformer uses an encoder-only architecture and does not modify any of the encoders' internal components. Generation is handled by linear layers.

3.3.4 Transfer learning

Traditionally, transfer learning is used to create models with limited domain-specific data. Appropriately applied, transfer learning can help one take knowledge learned from one domain or task (the "source") and apply it to improve learning in another domain or task (the "target") (Weiss et al., 2016). Transfer learning has contributed to recent developments in deep learning and assimilation. Specifically, foundation models have shown promise in general tasks, meaning they are often able to learn about target tasks from source datasets more easily. We discuss this in the next section.

3.3.5 Foundation models

In line with existing literature, we define foundation models as general-purpose deep learning models that have been pre-trained on diverse and massive datasets, which can then be fine-tuned to perform new tasks or those that the model was not adequately trained on (Bommasani et al., 2022). Essentially, they provide a "foundation" upon which more specialized models can be built.

Presently, the most well-known foundation models are large language models (LLMs), products of the field of natural language processing (NLP). These include models such as GPTs, LLaMAs, and BERTs (Radford et al., 2018; Touvron et al., 2023a,b; Devlin et al., 2019). Multimodality in LLMs is also becoming increasingly popular (Radford et al., 2021). As a whole, foundation models' use cases extend far beyond language and into domains like climate, biology, and computer vision. The most popular current foundation models are based on the Transformer architecture (Bommasani et al., 2023; Bommasani and Liang, 2021; Vaswani et al., 2023).

Two defining characteristics of foundation models are transfer learning and scale (Bommasani et al., 2022; Thrun, 1998). We have discussed transfer learning and what makes it a powerful technique. Specifically in the context of foundation models, we look at *pre-training* and *fine-tuning* as the primary means of transfer learning, where models are initially trained on broad datasets through "surrogate tasks" to capture relationships in the data. Then, these models are further trained for specific downstream tasks through fine-tuning, all while retaining the overall context. Scale, then, encompasses the general idea of growing to expand model capabilities, which was enabled by three components: improved computer hardware, the scalability transformer model architecture, and the increased availability of training data. Everything hinges on the availability of data and the ability to extract information from data.

This cycle of pre-training followed by fine-tuning for specific tasks is adjacent to assimilating data into an existing digital twin, and advancements in this technology should be monitored as foundation models continue to be investigated in the digital twin space. One example of a geospatial model with digital twin potential is Prithvi (Jakubik et al., 2023), a transformer-based foundation model specializing in analyzing multispectral satellite imagery of the Earth's surface. Pre-trained on over 1TB of data from the Harmonized Landsat Sentinel-2 (HLS) dataset (Claverie et al., 2018), which provides global surface reflectance data, Prithvi excels in capturing spatial patterns and relationships relevant to land surface applications. This model has been successfully fine-tuned to achieve state-of-the-art performance on downstream segmentation tasks such as flood mapping, wildfire scar identification, and crop type classification. Prithvi's ability to learn from large-scale, unlabeled satellite imagery showcases the potential of foundation models for data-efficient and generalizable geospatial artificial intelligence.

Ultimately, they promote a sort of centralization, in which one single model can be the backbone for countless smaller tasks and various implementations, paving the way for the future of AI systems and other areas like multi-task learning. This is not unlike the way digital twins can also be regarded as a set of models. The applications of foundation models outside natural language processing is an active area of research, especially in areas where highly complex simulations are traditionally used, such as weather prediction. Interest in applying foundation models to future digital twins in these domains has been published (Roy and Schmude, 2024). We conclude that the investigation of foundation models is a future research direction with strong prospects.

3.3.6 Low-rank adapter tuning

Low-rank adaptation (LoRA) has gained popularity in recent years because it requires vastly fewer resources than full model

pre-training, or even full model fine-tuning (Hu et al., 2021). The resulting trained adapters have zero runtime cost and can be easily shared. Many variations of LoRA have been created, including QLoRA (Dettmers et al., 2023) and DoRA (Liu S.-Y. et al., 2024). These techniques are often applied in combination with the foundation models discussed above, and can serve as a much more efficient way to perform data assimilation (Bodnar et al., 2024).

3.3.7 Surrogate models

Surrogate modeling, also known as metamodeling, is a powerful technique employed to approximate the behavior of complex and computationally expensive systems. Surrogate models can be simplified mathematical representations or function approximators that capture the input-output relationships of the original system, often computationally demanding, high-fidelity model. This simplification is achieved by training the surrogate model on a dataset of input-output pairs generated by the high-fidelity model. The surrogate, once trained, can then be used to predict the system's response to new inputs, significantly reducing the computational burden associated with evaluating the original model. This efficiency gain makes surrogate models particularly attractive in scenarios where numerous model evaluations are required, such as optimization, uncertainty quantification, and design exploration (Willard et al., 2022; Chakraborty et al., 2021).

The process of constructing a surrogate model typically involves several steps. First, a design of experiments is employed to strategically sample the input space of the high-fidelity model. This sampling strategy aims to maximize the information gained about the system's behavior with a limited number of simulations or experiments. Next, the high-fidelity model is evaluated at the selected input points, generating a dataset of the corresponding outputs. This dataset is then used to train the surrogate model, which can be chosen from a variety of mathematical forms, including polynomial regression models, Gaussian processes, radial basis functions, and neural networks (Willard et al., 2022). The choice of surrogate model depends on the characteristics of the system being approximated, the desired level of accuracy, and the computational resources available. Finally, the trained surrogate model is validated against additional data points to assess its predictive accuracy and ensure its suitability for the intended application.

Surrogate models can be particularly valuable in the context of digital twins, where computationally efficient models are often used for real-time decision support and analysis. By approximating the behavior of complex physical models, surrogate models can act as simplified digital twins, enabling rapid analysis and prediction without the computational burden of the original model. Surrogate models can also be used to explore a wider range of scenarios and sensitivities within a digital twin framework, as they can be evaluated quickly and efficiently; that is, surrogate models, in conjunction with digital twins, can be used for simulation tasks.

One generally wants to ensure that a digital twin in use adheres as closely as possible to the physical system it represents. Previously, we described physics-informed neural networks and their advantages. The digital twin, being based on real systems and intended to mimic the behavior of its physical counterpart, could benefit from introducing first principles and other "guardrails" to optimize performance in machine learning-based digital twins. As long as the physical twin is already constrained by some established physical or mathematical law, that law can be translated into some regularization term. Neural networks especially tend to suffer from problems such as overfitting and hallucination. Another benefit is less dependency on large datasets, which PINNs can alleviate in cases where data acquisition is expensive or difficult due to the aforementioned "guardrails" (Karpatne et al., 2017). PINNs, then, can serve as both a surrogate model for computationally expensive physical models and as a machine learning framework that is guided by fundamental physical principles. The main challenge lies in determining the appropriate laws, meaning domain expertise is required and highly pertinent.

3.3.8 Future research directions for data assimilation

To address the crucial gaps in data assimilation methods for digital twins identified at the start of this section, we highlight several promising research directions that warrant investigation:

First, hybrid assimilation frameworks that combine traditional techniques like Kalman filtering with modern machine learning approaches may offer a path forward. These frameworks could leverage the theoretical guarantees of classical methods while benefiting from the flexibility and scalability of deep learning.

Second, specialized architectures for handling multi-modal, multi-scale data streams need development. Digital twins often need to assimilate heterogeneous data types (images, time series, text) at different temporal and spatial scales. Transformer-based architectures with hierarchical attention mechanisms could potentially address this challenge by learning to appropriately weight and combine different data sources.

Third, physics-informed assimilation techniques that explicitly incorporate domain knowledge and physical constraints could improve both accuracy and computational efficiency. These methods could build on recent advances in physics-informed neural networks (PINNs) while adding mechanisms for sequential updating and uncertainty quantification.

Finally, distributed and federated assimilation algorithms that can operate across computing resources deserve exploration. Such approaches could enable digital twins to scale to larger systems while maintaining real-time performance requirements. Recent work in federated learning provides promising building blocks for this direction.

These research directions should be pursued while considering the specific challenges of digital twin applications, including real-time performance requirements, the handling of streaming data, and the need for uncertainty quantification.

3.3.9 Connection to world modeling

In the literature, world modeling and digital twins are treated as separate topics. Here, we will draw connections between these two concepts by highlighting similarities and overlapping ideas. Both involve creating virtual representations of systems—world models focusing on learning generalizable models of environment dynamics and digital twins aiming to mirror specific physical assets

or systems (Ha and Schmidhuber, 2018). But physical twins do not exist in a vacuum; the environment as a whole should be accounted for. Thus, we identify a first key intersection point in digital twin development, where the world model is used to provide that critical context. This notably parallels the methods discussed above.

However, integrating world models into digital twin frameworks presents challenges, particularly in aligning the general nature of world models with the specificity of digital twins. World models are typically designed to learn broadly applicable representations of environments, while digital twins focus on representing a particular physical asset. At the same time, the digital twin should be flexible with the capacity to generalize. Bridging this gap requires carefully tailoring the world model to capture the specific features and constraints of the target system while retaining its capacity for learning and prediction. Once more, this seems to point at potential solutions like foundation models, lending to a seemingly inevitable convergence between digital twins and emerging AI solutions.

3.4 Simulation and prediction

The goal of simulation and prediction is to generate predicted future states of the physical system. These predictions can then be used to make control decisions. Currently, two of the most prevalent methods of modeling a Digital Twin that exist are computational models and data-driven models. Either can be used as the primary model or the surrogate model, which can reduce the cost of running predictions. Computational models use numerical processes and simulations to reflect the characteristics and events that occur within the physical twin system while surrogate models employ the use of simplified or data-driven AI/ML models to approximate the state of the physical twin, reducing computational costs and enhancing efficiency (Bauer et al., 2015; Chakraborty et al., 2021). For the purposes of digital twins, the model needs to be capable of generating predictions even when it has not had the opportunity to assimilate new data from the physical system, as outlined in Axiom 4.

3.4.1 Simulation fundamentals

For various reasons, many problems cannot be solved with analytical methods. It is also possible that closed-form representations are possible but have not yet been found. Additionally, physical systems in active use can be extremely complicated, requiring more abstract renderings that leave out important information. In these situations, numerical methods or simulation techniques are often the best option.

Unlike other applications of simulation, the DT paradigm requires updates to be made to the simulation component based on data and observations from the real world. There is no hard limitation on the form that such simulations can take when applied within DTs; only that they must be able to accept updates based on observations collected from the physical system (see Section 3.3 where data assimilation is discussed in detail). Here, we outline multiple types of simulation employed in digital twin contexts.

3.4.2 Discrete time simulation

In contrast to continuous simulation where time itself is continuous and differential equations are used to calculate outputs, discrete time simulations (DTS) break time into a set of discrete time steps, $t_0 ... t_N$, where N is the number of time steps in an episode. Each time step is associated with a state $s_k \in \mathcal{S}$ where $\mathcal{S}$ is the set of all possible states. A time step t_k is associated with a state s_k. A simplified version of a discrete time simulation is given in Algorithm 1.

```
N := Total number of time steps to simulate
t ← 0
s_0 ← s ∈ I
while t < N do
    s_t ← Simulate for time step t, s_t ∈ S
    t ← t + 1                ▷ Move to next time step
end while
```

Algorithm 1. Simplified discrete time simulation loop.

The specific method used to discretize a real-world process can have large impacts on the results of the simulation. It is important to note that the discretization process itself can introduce accuracy issues.

3.4.3 Discrete event simulation

While similar in name to discrete time simulation, discrete event simulations (DES) do not have a time loop that iterates over time slices. Instead, the core of DES is an *event loop*. Events happen at specific moments in time and are processed in order. By avoiding explicit time steps and dealing only with events, DES avoids many of the accuracy issues inherent to DTS. An example of DES is shown in Algorithm 2.

```
N := Total number of events to simulate
E := Sorted list of events
k ← 0
while k < N do
    e_k ← Next event from E
    s_k ← Simulate event e_k
    k ← k + 1
end while
```

Algorithm 2. Simplified discrete event simulation loop.

3.4.4 Data-driven approaches

In some cases, data-driven approaches can yield excellent results, especially when paired with more traditional massive mathematical modeling. In Frnda et al. (2022), a small neural network was used to augment and correct ECMWF short-time weather forecasts. The accuracy of the neural network closely resembled that of the much more computationally expensive numerical weather prediction models at much finer geographical granularity. There are many other prominent and effective

applications of AI techniques for weather modeling, and there is currently a significant amount of interest and work conducted in this area (Lang et al., 2024; Nipen et al., 2024; Nguyen et al., 2023; Bi et al., 2023; Bodnar et al., 2024; Chen K. et al., 2023; Keisler, 2022; Pathak et al., 2022; Chen L. et al., 2023).

DT simulations can be deterministic or stochastic in nature. In one example of a stochastic model deployment for DT, Li et al. (2017) developed a technique that uses a Bayesian network to predict aircraft wing fatigue crack growth over time.

In Section 3.3, we discussed inverted transformers (Liu Y. et al., 2024). In addition to the benefits to assimilation, iTransformers have strong benefits for the generation and simulation phase as well. During generation, iTransformers use only the linear layers and not the attention mechanism. This makes generation much less computationally intensive (Liu Y. et al., 2024).

It can be significantly more difficult to apply purely data-driven techniques like machine learning in applications where available data are insufficient to fully describe the system, such as in the geosciences (Carrassi et al., 2018). In these situations, other techniques may be considered instead of, or in addition to, data driven techniques.

3.5 Analysis

After simulation and prediction is complete, the resulting outputs must analyzed and either presented to human decision-makers, or used by automated controllers to make decisions that impact the physical system. We label this fundamental task "analysis" as it encompasses both of these sub-tasks. The analysis task is extremely important because it serves as part of the interface between the digital twin model and the physical system.

After algorithms are run on the data to make inferences, these results need to be displayed in a coherent format for human consumers. Therefore, visualization is a key component of human-in-the-loop configurations. The results of predictions are typically presented through visual interfaces that display data analytics, predictive models, and trends. Although mentioned less frequently than visualization, *interaction* must also be carefully considered. Visualization as a topic in computer science has been studied for decades. However, many digital twin prototypes use simple visualization techniques such as dashboards and two-dimensional projections of three-dimensional models (VanDerHorn and Mahadevan, 2021). Interaction, where discussed at all, often requires expert-level knowledge as the user must know how to configure, set the initial conditions of, and run the digital twin. More recent digital twin works have begun to adopt more ambitious uses of visualization and interaction technology, such as augmented reality (AR) and virtual reality (VR) (Brewer et al., 2024; Vysocký and Riha, 2024). AR and VR mediums have a significantly higher potential information "bandwidth" than 2D dashboards and text output as they have more dimensions to work with and provide fundamentally new ways to *interact* with data. Although there are still many research avenues to explore and engineering challenges to overcome, VR and AR interfaces are exciting new directions for truly interactive digital twins (Cruz-Neira, 2024).

Most papers do not mention where the data are stored after being worked on, showing that this field is up to each niche case. These interfaces often feature dashboards, graphs, and charts to allow users to interpret and interact with the predictions easily. Using a long history of previous output data also allows for predictions from machine learning-based digital twins, to have higher accuracy compared to general methods. This also appears in a more digestible fashion when being displayed in graphs (Fahim et al., 2022).

3.6 Feedback and control

The feedback loop or control system, a cornerstone principle within control theory (Åström and Hägglund, 2006; Dorf Bishop, 2017), recontextualized within DTs, depicts interactions between the physical and digital twins. Data from the physical twin are processed by the digital twin; then, insights gleaned from the digital twin are applied back to the physical twin, which will generate new data for the digital twin. The digital twin should update accordingly. This process continuously repeats. The essence of a feedback loop lies in the continuous monitoring of a system's output, comparing it to a desired setpoint, and then using the discrepancy to adjust the system's input to drive it toward the desired state (Åström and Hägglund, 2006).

A similar pattern is applicable to digital twins. While data assimilation forms the "right" side of the feedback loop shown in Figure 6, the feedback and control system forms the "left" connection from the digital twin back to the physical system. We "unroll" this process in Figure 8. We outline two main approaches for the implementation of the control link, though the boundary between these two is somewhat fuzzy.

3.6.1 Human-in-the-loop

The first is "human-in-the-loop" approaches, where real-time feedback from the digital twin provides actionable insights to a human operator (Nunes et al., 2015). The human operator then performs operations that impact the physical twin. The objectives can include performance optimization and the prediction and advance correcting of potential problems (Errandonea et al., 2020). Currently, human-in-the-loop models are popular (Pylianidis et al., 2022; Ritto and Rochinha, 2021; Verdouw et al., 2021) and there is a growing emphasis on real-time feedback (Dang et al., 2022; Cao et al., 2023; Bordukova et al., 2024); digital twins are increasingly expected to provide immediate insights and enable prompt responses to dynamic changes in the physical twin.

3.6.2 Full autonomy

The second approach is fully autonomous control, or the typical closed-loop control system, where control systems are integrated into the digital twin to automate responses such as optional parameter adjustment and maintenance actions (Glaessgen and Stargel, 2012). Fully autonomous control can be difficult to achieve for non-trivial real-world systems, often lacks generalizability, and may be dangerous if applied in the wrong contexts (Huang et al., 2023). Control theory is a long-studied discipline with

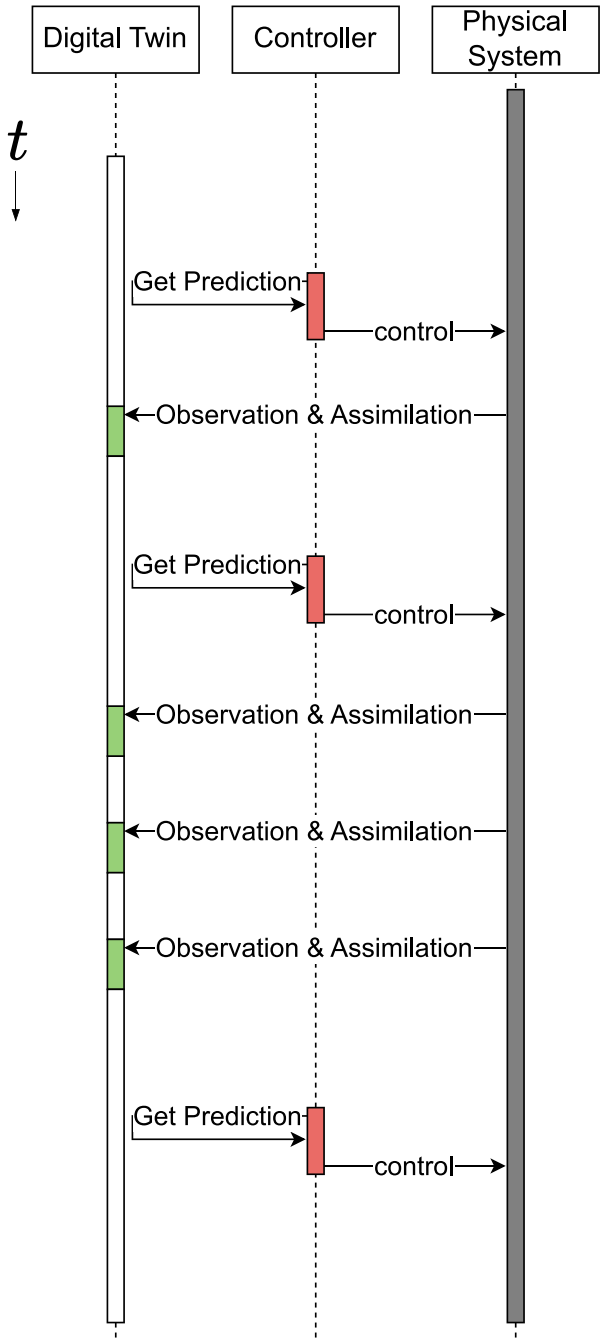

FIGURE 8
A physical twin and its associated digital twin evolve alongside and influence one another. In this diagram, we unroll the interaction loop from Figure 6 and expand the digital twin and controller into discrete parts to highlight the nature of these co-evolving components over time. Note that in our definition, we do not prescribe a strict order for the predictions, updates, and controls.

many applicable techniques. Modern control theory, building upon classical foundations, has expanded the scope of feedback control to encompass non-linear systems, stochastic processes, and adaptive control strategies. The choice of technique is often specific to the application and practitioners must take into account the complexity of the system and the desired control objectives. One prominent example is proportional-integral-derivative (PID) control, which is a widely used technique that utilizes a combination of proportional, integral, and derivative terms to calculate control actions, providing flexibility in tuning the system's response (Åström and Hägglund, 2006). Others include robust control, which addresses systems with uncertainties or disturbances by designing controllers that maintain stability and performance despite variations in the system's parameters or external factors (Zhou et al., 1996), model predictive control (MPC) which uses a dynamic model of the system to predict future behavior and optimize control actions over a finite time horizon, enabling more sophisticated control strategies (Camacho and Bordons, 2007), and various forms of optimal control, which focus on minimizing or maximizing some defined cost function (e.g., energy consumption, time to reach the setpoint) by employing mathematical optimization techniques (Kirk, 2004). These techniques can be used in combination with digital twins (Abro and Abdallah, 2024).

3.6.3 Partial autonomy

As a result of their close relationship with infrastructure, digital twins must be designed to preserve public trust. Safety is paramount, especially in critical applications such as power plant management or transportation systems, where a single faulty automated control could have severe consequences. Developers and stakeholders must prioritize ethical frameworks and establish safeguards to ensure fairness, safety, and the protection of personal privacy in emerging DT applications.

Partially autonomous approaches aim to combine the advantages of full autonomy with the presumed relative safety of human-in-the-loop designs. Despite their attractiveness, there remain significant ethical concerns related to responsibility, accountability, and human agency in partially-autonomous systems. One key consideration for the implementation of the feedback loop lies in determining the optimal level of automation; that is, how can human oversight be balanced with autonomous control? Questions relating to responsibility and accountability become particularly complex: If the recommendation of a digital twin leads to harm when combined with a human operator's judgment, it is often unclear whether the fault lies with the technology or the user.

This conversation is at the intersection of ethics and human safety. On the one hand, autonomous control, where the digital twin makes decisions and implements actions without human intervention, offers significant benefits. It enables real-time responses to dynamic changes, optimization of complex processes, and potentially even the prevention of accidents or failures that humans might miss. However, relinquishing complete control to an automated system raises clear ethical concerns, particularly if the system's decision-making process is opaque or poorly understood.

Partial autonomy can be desirable because it can blend benefits from both approaches, but also requires careful design with human considerations taken into account, such as **automation complacency**, or worse, **automation bias**, where operators become hesitant to override system recommendations even when their experience suggests otherwise (Kim and Yang, 2017; Lyell and

Coiera, 2017). The optimal level of automation in a digital twin's feedback loop depends on various factors, including the specific application domain, the complexity of the physical system, the maturity of the AI control algorithms, and the societal acceptance of autonomous decision-making.

4 A taxonomy for digital twins

Here, we present a taxonomy for digital twins that explores the interface between DT technology, high-performance computing (HPC), AI, and ML techniques. We observe that this interface is not static and that there are a wide variety of approaches currently under exploration.

4.1 Machine learning and HPC for digital twins

High-performance computing resources provide new opportunities to create highly accurate and precise digital twins. High-performance networking technologies, such as Infiniband, lower communication latency and increase bandwidth (Panda et al., 2022). Graphics processing units (GPUs) and specialized accelerators, such as Google's Tensor processing units (TPUs) and Amazon's Trainium, make highly parallel operations much faster (NVIDIA, 2023; Google, Inc., 2024; Amazon Web Services, 2024). Advanced I/O technologies such as NVMe bring some of the designs from high-performance networking technologies to I/O, along with increases in speed (Ng et al., 2024). Many universities have HPC systems that can be accessed by faculty and students for research tasks. In the United States, the Department of Energy and National Science Foundation make supercomputers available to researchers (High Performance Computing, 2024; National Science Foundation, 2022). Cloud providers such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud offer cloud-based HPC systems as well (AWS High Performance Computing, 2024; Microsoft Azure, 2024; Google Cloud, 2024), making HPC technology much more accessible than it once was. Recent advancements in low-power hardware capabilities have enabled the execution of certain machine learning tasks at the edge (Hui et al., 2020).

One core component of digital twins is the model, which is designed to map to the real-world physical system with which the digital twin is associated. Modeling non-trivial real-world systems accurately and precisely often requires intensive manual effort. In order to make effective use of DTs without spending large amounts of time in the modeling stage, automatic updates must be made to the digital twin as information about the physical system comes in. It is also probable that the physical system itself will experience changes over time that are not included in the original digital twin model. By updating the digital twin from the observations of the physical twin via assimilation, the digital twin can adapt to unforeseen events that would otherwise render the digital twin useless for decision making (Kapteyn et al., 2021). Recent trends in machine learning and optimization are applicable to such tasks.

Overall, we see HPC and machine learning technologies as critical components of current and future digital twins. Next, we will explore case studies that show the diversity of data-driven digital twins that are being created for scientific applications.

4.1.1 AI weather models

Aurora (Bodnar et al., 2024) is a foundation model for atmospheric simulation and prediction, pioneering a large-scale model trained on an unprecedented volume of diverse weather and climate data that can handle both regular forecasting and extreme weather at an impressive resolution. It uses an architecture specifically suited for large heterogeneous datasets: a multi-scale 3D Swin Transformer U-Net backbone with a 3D Perceiver encoder and decoder. The authors report that it is efficient and versatile; Aurora can quickly predict various atmospheric phenomena, including air pollution, on par with or better than existing state-of-the-art technology. At the time of writing, this paper was only recently submitted and thus has not yet been published in any journal; however, as Aurora was spearheaded by researchers at Microsoft's AI4Science, the company is currently promoting the model. Aurora's position as a foundation model and the paper's methods lend themselves to further research in modeling other Earth subsystems; ultimately, this paper introduced a new paradigm in atmospheric modeling with numerous practical implications, from climate studies to weather forecasting. Aurora is a foundation model for atmospheric simulation and prediction, pioneering a large-scale model trained on an unprecedented volume of diverse weather and climate data that can handle both regular forecasting and extreme weather at an impressive resolution. Aurora makes use of ECMWF reanalysis products in its training, which are continuously updated. It uses an architecture specifically suited for large heterogeneous datasets: a multi-scale 3D Swin Transformer U-Net backbone with a 3D Perceiver encoder and decoder. The authors report that it is efficient and versatile; Aurora can quickly predict various atmospheric phenomena, including air pollution, on par with or better than existing state-of-the-art technology. Aurora's position as a foundation model and the paper's methods lend themselves to further research in modeling other Earth subsystems; ultimately, this paper introduced a new paradigm in atmospheric modeling with numerous practical implications, from climate studies to weather forecasting. However, Aurora is not the only AI-based model for predicting weather and climate. This topic has generated much interest and research in recent years, which is visible in the literature (Lang et al., 2024; Nipen et al., 2024; Nguyen et al., 2023; Bi et al., 2023; Bodnar et al., 2024; Chen K. et al., 2023; Keisler, 2022; Pathak et al., 2022; Chen L. et al., 2023).

4.1.2 Data-driven calibration and evolution of a UAV structural twin

One instance of a machine learning-based digital twin in the context of robotics is presented in the 2021 paper "A Probabilistic Graphical Model Foundation for Enabling Predictive Digital Twins at Scale" where the authors utilized a Bayesian statistical approach to model an unmanned aerial vehicle (UAV) (Kapteyn et al., 2021). The primary goal was to create a predictive structural simulation of the UAV's airframe.

The authors broke the DT update process into two distinct phases. The first was the initial calibration phase where geometry, material properties, mass, and damping aspects were tested, post-processed, and used to calibrate the digital twin. The second, matching the standard DT approach, was the dynamic phase, where the digital twin performed automatic updates based on sensed data.

An important takeaway from this work is that the authors specifically designed the state in such a way that real-world variance can be represented. This is important when formulating the DT. It is also clear that as a result of the limited computing power available in "edge" deployments, such as use with UAVs, one must very carefully design their DT approach. Deep neural networks and detailed simulations can be extremely computationally expensive (Touvron et al., 2023a) and are therefore often infeasible to deploy at the edge. Instead, one can use effective low-cost methods, such as simpler Bayesian data assimilation or LoRA.

4.1.3 Metamodeling and digital twins in simulation-assisted machine learning

As digital twins want to mirror their physical counterparts with as much accuracy as possible, and knowing the existing success of machine learning as an alternative to typically expensive computational models, there have been attempts to combine the surrogate models and machine learning to enable greater accessibility.

Pylianidis et al. (2022) took a metamodeling approach to construct a machine learning-based digital twin for predicting pasture nitrogen response rates (NRR). Driven by the need for accessibility and operability in real-world agricultural settings, their research addressed the challenge of limited data availability and the computational expense of traditional process-based models (PBMs). To overcome data scarcity, they used the Agricultural Production Systems Simulator (APSIM), a well-established process-based model, to generate a large synthetic dataset encompassing various environmental conditions and management practices. This dataset served as a virtual laboratory for exploring pasture nitrogen dynamics.

This study showcases the potential of simulation-assisted ML for operational digital twins in data-constrained domains like agriculture; we see that their machine learning-based digital twin can provide accurate NRR predictions in both sampled and unsampled locations. Overall, the major shortcoming lies in how the study did not explicitly implement a complete feedback loop with dynamic data assimilation.

4.1.4 Cloud-based digital twins for structural health monitoring

Dang et al. (2022) outlines a feedback loop that uses the digital twin's analysis to provide real-time alerts and guide maintenance decisions for the physical structure. When the digital twin detects anomalies or potential damage based on the assimilated data, it triggers alerts to notify engineers, who can then take action to improve the physical system.

4.2 Digital twins for HPC

HPC systems also serve as interesting physical targets for digital twins. Recent work has demonstrated that digital twins can be highly effective for modeling supercomputers. ExaDigiT (Brewer et al., 2024) is an example of such a digital twin that is capable of simulating power, cooling, and other important parameters of the current second most powerful supercomputer in the world: Frontier.[4] These components are brought together through an advanced visualization layer that leverages augmented reality to provide intuitive system interaction. This architecture allows operators and engineers to study complex behaviors that may occur in the physical system and explore "what-if" scenarios.

The ExaMon framework represents another example of an HPC digital twin, providing an integrated approach to monitoring and maintaining supercomputing systems (Borghesi et al., 2023). ExaMon combines lightweight data collection infrastructure with specialized databases suited for heterogeneous data sources with the purpose of enabling real-time analysis and prediction. The authors of the framework report that it has been successfully deployed across several production supercomputers, including CINECA's Marconi system, demonstrating its ability to handle the scale and complexity of modern HPC environments.

These examples demonstrate how digital twins can transform HPC system management from reactive to predictive, enabling more efficient resource utilization and reduced downtime. They also validate our DT definition's emphasis on learning and adaptation capabilities, as these systems continuously improve their predictions through the assimilation of new data from their physical counterparts.

4.3 Digital twins for machine learning

Another way of approaching the intersection of AI and digital twins is by using digital twins to generate data for machine learning models. Many machine learning methods may derive benefit from inexpensive or parallelizable virtual training environments, as access to training data in the appropriate quantities is frequently a challenge (Sun et al., 2017). Typically, synthetic data generation is considered in the context of machine learning from the other way around; that is, machine learning models are the ones doing the data generation and annotating when real-world data are scarce or unavailable. Digital twins, as representations of a physical system, can serve a similar purpose and further augment datasets for training machine learning models.

Digital twins can be used as training environments for reinforcement learning algorithms. One case of this approach being applied is in Matulis and Harvey (2021). The authors first designed a robotic arm using CAD software and then used 3D printers to create a physical manifestation of the design. Stepper motors were inserted into the printed components to give the arm motion capabilities. However, as with many examples of digital twin in literature, this study does not attempt to address how the particular

4 See the full Top500 list: https://top500.org/lists/top500/2024/11/.

application of DT explored could scale beyond the laboratory, especially with regards to compute.

The robotic arm work is another example of how digital twins can be powerful tools when used to *train* reinforcement learning algorithms. The relatively low cost of training—a result of the lower risk to expensive physical systems—greater parallelization, and the lower required human intervention, suggest that digital twins should be strongly considered for training tasks that will eventually be applied in physical environments. Additionally, users of the digital twin may realize many of the other benefits described in previous sections in addition to simply providing an updated training environment once the digital twin is implemented and the model trained.

5 Conclusion

Digital twins (DTs) are a class of virtual replicas of real-world objects that are modeled, updated, and interpreted to perform decision-making tasks with continuously evolving datasets. This paper has explored the interface between DT technology, scientific computing, and machine learning (ML) by presenting a consistent definition for the digital twin, performing an analysis of the literature to build a taxonomy of digital twins and AI/ML, and discussed case studies from various scientific domains. Looking ahead, we highlighted several promising research directions and outlined the ways in which digital twins, AI/ML, and HPC techniques are synergistic. As the field continues to evolve, these research directions will be crucial for realizing the full potential of digital twins across scientific and industrial applications.

Author contributions

AW: Conceptualization, Funding acquisition, Methodology, Visualization, Writing – original draft, Writing – review & editing. CC: Writing – original draft, Writing – review & editing. SL: Writing – original draft, Writing – review & editing. SM: Writing – original draft, Writing – review & editing. TJ: Writing – original draft, Writing – review & editing. JV: Data curation, Conceptualization, Funding acquisition, Resources, Writing – original draft, Writing – review & editing. XL: Conceptualization, Funding acquisition, Supervision, Writing – original draft, Writing – review & editing.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This publication was prepared with the financial support provided by the Farms Food Future (F3) Initiative funded by the Economic Development Administration Build Back Better Regional Challenge Award #07-79-07913. This work was partly supported by NSF research grants OAC #2321123 and #2340982 and a DOE research grant DE-SC0024207. Additional partial support was provided by the United States Department of Agriculture–National Institute of Food and Agriculture (NIFA) award #2021-67021-35344.

Conflict of interest

The authors declare that the research was conducted without any commercial or financial relationships that could potentially create a conflict of interest.

Generative AI statement

The author(s) declare that no Gen AI was used in the creation of this manuscript.

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.

References

Åström, K. J. (1965). Optimal control of markov processes with incomplete state information I. *J. Math. Anal. Appl.* 10, 174–205. doi: 10.1016/0022-247X(65)90154-X

Åström, K. J., and Hägglund, T. (2006). "Advanced PID control," in *ISA - The Instrumentation, Systems, and Automation Society*. Research Triangle Park, NC.

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., et al. (2024). Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature* 630, 493–500. doi: 10.1038/s41586-024-07487-w

Abro, G. E. M., and Abdallah, A. M. (2024). Digital twins and control theory: a critical review on revolutionizing quadrotor UAVs. *IEEE Access* 12, 43291–43307. doi: 10.1109/ACCESS.2024.3376589

Allen, B. D. (2021). *Digital Twins and Living Models at NASA*. Hampton, VA: Langley Research Center.

Amazon Web Services (2024). *Trainium Architecture*. Available at: https://awsdocs-neuron.readthedocs-hosted.com/en/latest/general/arch/neuron-hardware/trainium.html (accessed November 1, 2024).

Avigal, Y., Deza, A., Wong, W., Oehme, S., Presten, M., Theis, M., et al. (2021). "Learning seed placements and automation policies for polyculture farming with companion plants," in *2021 IEEE International Conference on Robotics and Automation (ICRA)* (Xi'an: IEEE), 902–908. doi: 10.1109/ICRA48506.2021.9561431

AWS High Performance Computing (2024). Amazon Web Services, Inc. Available at: https://aws.amazon.com/hpc/ (accessed January 27, 2024).

Bauer, P., Thorpe, A., and Brunet, G. (2015). The quiet revolution of numerical weather prediction. *Nature* 525, 47–55. doi: 10.1038/nature14956

Bertsekas, D. P. (2020). *Rollout, Policy Iteration, and Distributed Reinforcement Learning*. Belmont, MA: Athena Scientific.

Besta, M., Blach, N., Kubicek, A., Gerstenberger, R., Podstawski, M., Gianinazzi, L., et al. (2024). Graph of thoughts: solving elaborate problems with large language models. *Proc. AAAI Conf. Artif. Intell.* 38, 17682–17690. doi: 10.1609/aaai.v38i16.29720

Bi, K., Xie, L., Zhang, H., Chen, X., Gu, X., Tian, Q., et al. (2023). Accurate medium-range global weather forecasting with 3D neural networks. *Nature* 619, 533–538. doi: 10.1038/s41586-023-06185-3

Bodnar, C., Bruinsma, W. P., Lucic, A., Stanley, M., Brandstetter, J., Garvan, P., et al. (2024). *Aurora: A Foundation Model of the Atmosphere*.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., et al. (2022). On the opportunities and risks of foundation models. *arXiv*. arXiv:2108.07258. doi: 10.48550/arXiv.2108.07258

Bommasani, R., Klyman, K., Longpre, S., Kapoor, S., Maslej, N., Xiong, B., et al. (2023). The foundation model transparency index. *arXiv* [Preprint]. arXiv:2310.12941. doi: 10.48550/arXiv.2310.12941

Bommasani, R., and Liang, P. (2021). *Reflections on Foundation Models*. Stanford HAI. Available at: https://hai.stanford.edu/news/reflections-foundation-models (accessed July 2, 2024).

Bordukova, M., Makarov, N., Rodriguez-Esteban, R., Schmich, F., and Menden, M. P. (2024). Generative artificial intelligence empowers digital twins in drug discovery and clinical trials. *Expert Opin. Drug Discov* 19, 33–42. doi: 10.1080/17460441.2023.2273839

Borghesi, A., Burrello, A., and Bartolini, A. (2023). ExaMon-X: a predictive maintenance framework for automatic monitoring in industrial IoT systems. *IEEE Internet Things J*. 10, 2995–3005. doi: 10.1109/JIOT.2021.3125885

Brewer, W., Maiterth, M., Kumar, V., Wojda, R., Bouknight, S., Hines, J., et al. (2024). "A digital twin framework for liquid-cooled supercomputers as demonstrated at exascale," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis, SC '24* (Atlanta, GA: IEEE Press), 1–18. doi: 10.1109/SC41406.2024.00029

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., et al. (2020). Language models are few-shot learners. *arXiv*. arXiv:2005.14165. doi: 10.48550/arXiv.2005.14165

Camacho, E. F., and Bordons, C. (2007). *Model Predictive Control. Advanced Textbooks in Control and Signal Processing*. London: Springer. doi: 10.1007/978-0-85729-398-5

Cao, H., Zhang, D., and Yi, S. (2023). Real-timemachine learning-based fault detection, classification, and locating in large scale solar energy-based systems: digital twin simulation. *Solar Energy* 251, 77–85. doi: 10.1016/j.solener.2022.12.042

Carrassi, A., Bocquet, M., Bertino, L., and Evensen, G. (2018). Data assimilation in the geosciences: An overview of methods, issues, and perspectives. *WIREs Clim. Change* 9:e535. doi: 10.1002/wcc.535

Chakraborty, S., Adhikari, S., and Ganguli, R. (2021). The role of surrogate models in the development of digital twins of dynamic systems. *Appl. Math. Model.* 90, 662–681. doi: 10.1016/j.apm.2020.09.037

Chen, K., Han, T., Gong, J., Bai, L., Ling, F., Luo, J.-J., et al. (2023). FengWu: pushing the skillful global medium-range weather forecast beyond 10 days lead. *arXiv*. arXiv:2304.02948. doi: 10.48550/arXiv.2304.02948

Chen, L., Zhong, X., Zhang, F., Cheng, Y., Xu, Y., Qi, Y., et al. (2023). FuXi: a cascade machine learning forecasting system for 15-day global weather forecast. *NPJ Clim. Atmos. Sci*. 6, 1–11. doi: 10.1038/s41612-023-00512-1

Claverie, M., Ju, J., Masek, J. G., Dungan, J. L., Vermote, E. F., Roger, J.-C., et al. (2018). The harmonized Landsat and sentinel-2 surface reflectance data set. *Remote Sens. Environ*. 219, 145–161. doi: 10.1016/j.rse.2018.09.002

Correia, J. B., Abel, M., and Becker, K. (2023). Data management in digital twins: a systematic literature review. *Knowl. Inf. Syst.* 65, 3165–3196. doi: 10.1007/s10115-023-01870-1

Cromity, J., and De Stricker, U. (2011). Silo persistence: it's not the technology, it's the culture! *New Rev. Inf. Netw* 16, 167–184. doi: 10.1080/13614576.2011.619924

Cruz-Neira, C. (2024). "National academies: foundational reseach gaps and future directions for digital twins," in *The 2nd Digital Twins Workshop for High-Performance Computing at SC'24* (Atlanta, GA).

Dang, H. V., Tatipamula, M., and Nguyen, H. X. (2022). Cloud-based digital twinning for structuralhealth monitoring using deep learning. *IEEE Trans. Ind. Inf.* 18, 3820–3830. doi: 10.1109/TII.2021.3115119

Das, A., Kong, W., Leach, A., Mathur, S., Sen, R., Yu, R., et al. (2024). long-term forecasting with TiDE: time-series dense encoder. *arXiv*. arXiv:2304.08424. doi: 10.48550/arXiv.2304.08424

Denis, P., and Baldridge, J. (2007). "A ranking approach to pronoun resolution," in *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence, Volume 158821593* (Hyderabad).

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. (2023). QLoRA: efficient finetuning of quantized LLMs. *arXiv*. arXiv:2305.14314. doi: 10.48550/arXiv.2305.14314

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (Minneapolis, MN: Association for Computational Linguistics), 4171–4186. doi: 10.18653/v1/N19-1423

Digital Twins (2023). *NRC Web*. Available at: https://www.nrc.gov/reactors/power/digital-twins.html (accessed January 27, 2024).

Dorf, R. C., and Bishop, R. H. (2017). *Modern Control Systems, Thirteenth Edition, Global Edition ed*. London: Pearson.

Dunbar, J., Moyer, M., and Pitta, K. (2024). "Six ways supercomputing advances our understanding of the universe," *NASA News and Events*. NASA.

Errandonea, I., Beltrán, S., and Arrizabalaga, S. (2020). Digital Twin for maintenance: a literature review. *Comput. Ind.* 123:103316. doi: 10.1016/j.compind.2020.103316

Eysenbach, B., Salakhutdinov, R. R., and Levine, S. (2019). "Search on the replay buffer: bridging planning and reinforcement learning," in *Advances in Neural Information Processing Systems, Volume 32* (Red Hook, NY: Curran Associates, Inc).

Fahim, M., Sharma, V., Cao, T-. V., Canberk, B., and Duong, T. Q. (2022). Machine learning-based digital twin for predictive modeling in wind turbines. *IEEE Access* 10, 14184–14194. doi: 10.1109/ACCESS.2022.3147602

Fedus, W., Ramachandran, P., Agarwal, R., Bengio, Y., Larochelle, H., Rowland, M., et al. (2020). "Revisiting fundamentals of experience replay," in *Proceedings of the 37th International Conference on Machine Learning* (PMLR), 3061–3071.

Friederich, J., Francis, D. P., Lazarova-Molnar, S., and Mohamed, N. (2022). A framework for data-driven digital twins of smart manufacturing systems. *Comput. Ind.* 136:103586. doi: 10.1016/j.compind.2021.103586

Frnda, J., Durica, M., Rozhon, J., Vojtekova, M., Nedoma, J., Martinek, R., et al. (2022). ECMWF short-term prediction accuracy improvement by deep learning. *Sci. Rep*. 12:7898. doi: 10.1038/s41598-022-11936-9

Glaessgen, E., and Stargel, D. (2012). "The digital twin paradigm for future NASA and U.S. air force vehicles," in *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference
 20th AIAA/ASME/AHS Adaptive Structures Conference
 14th AIAA* (Honolulu, Hawaii: American Institute of Aeronautics and Astronautics). doi: 10.2514/6.2012-1818

Google Cloud (2024). *Google Cloud HPC Solutions*. Available at: https://cloud.google.com/solutions/hpc (accessed January 27, 2024).

Google Maps 101 (2019). *Google Maps 101: How imagery powers our map*. Available at: https://blog.google/products/maps/google-maps-101-how-imagery-powers-our-map/ (accessed December 21, 2024).

Google, Inc. (2024). *TPU v6e*. Available at: https://cloud.google.com/tpu/docs/v6e (accessed October 31, 2024).

Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., Moore, R., et al. (2017). Google earth engine: planetary-scale geospatial analysis for everyone. *Remote Sens. Environ*. 202, 18–27. doi: 10.1016/j.rse.2017.06.031

Grieves, M. (2014). Digital twin: manufacturing excellence through virtual factory replication. *White Paper* 1, 1–7.

Guo, X., Liu, L., Wang, Z., Wang, H., Du, X., Shi, J., et al. (2023). Research on data collection methods for assembly performance of array antennas in digital twin workshops. *Processes*11:2711. doi: 10.3390/pr11092711

Ha, D., and Schmidhuber, J. (2018). "Recurrent world models facilitate policy evolution," in *Advances in Neural Information Processing Systems, Vol. 31* (Red Hook, NY: Curran Associates, Inc).

Halevy, A., Norvig, P., and Pereira, F. (2009). The unreasonable effectiveness of data. *IEEE Intell. Syst*. 24, 8–12. doi: 10.1109/MIS.2009.36

Haße, H., van der Valk, H., Möller, F., and Otto, B. (2022). Design principles for shared digital twins in distributed systems. *Bus. Inf. Syst. Eng.* 64, 751–772. doi: 10.1007/s12599-022-00751-1

High Performance Computing (2024). Energy.gov. Available at: https://www.energy.gov/science/high-performance-computing (accessed January 21, 2024).

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

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., et al. (2021). *LoRA: Low-Rank Adaptation of Large Language Models*.

Huang, C., Zhang, Z., Mao, B., and Yao, X. (2023). An overview of artificial intelligence ethics. *IEEE Trans. Artif. Intell*. 4, 799–819. doi: 10.1109/TAI.2022.3194503

Huang, Y., Xu, J., Lai, J., Jiang, Z., Chen, T., Li, Z., et al. (2024). Advancing transformer architecture in long-context large language models: a comprehensive survey. *arXiv*. arXiv:2311.12351. doi: 10.48550/arXiv.2311.12351

Huang, Z., Shen, Y., Li, J., Fey, M., and Brecher, C. (2021). A survey on AI-driven digital twins in industry 4.0: smart manufacturing and advanced robotics. *Sensors* 19:6340. doi: 10.3390/s21196340

Hui, Y., Lien, J., and Lu, X. (2020). "Early experience in benchmarking edge AI processors with object detection workloads," in *Benchmarking, Measuring, and*

Optimizing, eds. W. Gao, J. Zhan, G. Fox, X. Lu, and D. Stanzione (Cham: Springer International Publishing), 32–48. doi: 10.1007/978-3-030-49556-5_3

Jakubik, J., Roy, S., Phillips, C. E., Fraccaro, P., Godwin, D., Zadrozny, B., et al. (2023). Foundation models for generalist geospatial artificial intelligence. *arXiv* [Preprint]. arXiv:2310.18660. doi: 10.48550/arXiv.2310.18660

Jones, D., Snider, C., Nassehi, A., Yon, J., and Hicks, B. (2020). Characterising the Digital Twin: a systematic literature review. *CIRP J. Manuf. Sci. Technol.* 29, 36–52. doi: 10.1016/j.cirpj.2020.02.002

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., et al. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature* 596, 583–589. doi: 10.1038/s41586-021-03819-2

Kalyanaraman, A., Burnett, M., Fern, A., Khot, L., and Viers, J. (2022). Special report: the AgAID AI institute for transforming workforce and decision support in agriculture. *Comput. Electron. Agric.* 197:106944. doi: 10.1016/j.compag.2022.106944

Kapteyn, M. G., Pretorius, J. V. R., and Willcox, K. E. (2021). A probabilistic graphical model foundation for enabling predictive digital twins at scale. *Nat. Comput. Sci.* 1, 337–347. doi: 10.1038/s43588-021-00069-0

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., Yang, L., et al. (2021). Physics-informed machine learning. *Nat. Rev. Phys.* 3, 422–440. doi: 10.1038/s42254-021-00314-5

Karpatne, A., Atluri, G., Faghmous, J. H., Steinbach, M., Banerjee, A., Ganguly, A., et al. (2017). Theory-guided data science: a new paradigm for scientific discovery from data. *IEEE Trans. Knowl. Data Eng.*29, 2318–2331. doi: 10.1109/TKDE.2017.2720168

Keisler, R. (2022). Forecasting global weather with graph neural networks. *arXiv*. arXiv:2202.07575. doi: 10.48550/arXiv.2202.07575.

Khetawat, H., Zimmer, C., Mueller, F., Atchley, S., Vazhkudai, S. S., Mubarak, M., et al. (2019). "Evaluating burst buffer placement in HPC systems," in *2019 IEEE International Conference on Cluster Computing (CLUSTER)* (Albuquerque, NM: IEEE), 1–11. doi: 10.1109/CLUSTER.2019.8891051

Kim, H. J., and Yang, J. H. (2017). Takeover requests in simulated partially autonomous vehicles considering human factors. *IEEE Trans. Hum.-Mach. Syst.* 47, 735–740. doi: 10.1109/THMS.2017.2674998

Kirk, D. E. (2004). *Optimal Control Theory: An Introduction (Dover Books on Electrical Engineering)*. Mineola, NY: Dover Publications.

Lang, S., Alexe, M., Chantry, M., Dramsch, J., Pinault, F., Raoult, B., et al. (2024). AIFS-ECMWF's data-driven forecasting system. *arXiv*. arXiv:2406.01465. doi: 10.48550/arXiv.2406.01465

LeBeau, G. J. (1999). A parallel implementation of the direct simulation Monte Carlo method. *Comput. Methods Appl. Mech. Eng.* 174, 319–337. doi: 10.1016/S0045-7825(98)00302-8

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

Leutbecher, M., and Palmer, T. N. (2008). Ensemble forecasting. *J. Comput. Phys.* 227, 3515–3539. doi: 10.1016/j.jcp.2007.02.014

Li, C., Mahadevan, S., Ling, Y., Choze, S., and Wang, L. (2017). Dynamic Bayesian network for aircraft wing health monitoring digital twin. *AIAA J.* 55, 930–941. doi: 10.2514/1.J055201

Li, Y., Kashyap, A., Guo, Y., and Lu, X. (2023). "Characterizing lossy and lossless compression on emerging bluefield DPU architectures," in *2023 IEEE Symposium on High-Performance Interconnects (HOTI)* (Hot Interconnects), 33–40. doi: 10.1109/HOTI59126.2023.00019

Lin, J., Du, Y., Watkins, O., Hafner, D., Abbeel, P., Klein, D., et al. (2024). Learning to model the world with language. *arXiv*. arXiv:2308.01399. doi: 10.48550/arXiv.2308.01399

Lin, L.-J. (1992). Self-improving reactive agents based on reinforcement learning, planning and teaching. *Mach. Learn.* 8, 293–321. doi: 10.1007/BF00992699

Liu, H., Yan, W., Zaharia, M., and Abbeel, P. (2024). World model on million-length video and language with blockwise ring attention. *arXiv*. arXiv:2402.08268. doi: 10.48550/arXiv.2402.08268

Liu, J., Koziol, Q., Butler, G. F., Fortner, N., Chaarawi, M., Tang, H., et al. (2018). "Evaluation of HPC application I/O on object storage systems," in *2018 IEEE/ACM 3rd International Workshop on Parallel Data Storage Data Intensive Scalable Computing Systems (PDSW-DISCS)* (New York, NY: ACM), 24–34. doi: 10.1109/PDSW-DISCS.2018.00005

Liu, S.-Y., Wang, C.-Y., Yin, H., Molchanov, P., Wang, Y.-C. F., Cheng, K.-T., et al. (2024). DoRA: weight-decomposed low-rank adaptation. *arXiv*. arXiv:2402.09353. doi: 10.48550/arXiv.2402.09353

Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., et al. (2024). iTransformer: inverted transformers are effective for time series forecasting. *arXiv*. arXiv:2310.06625. doi: 10.48550/arXiv.2310.06625

Liu, Y. A., Liu, X. L., Li, F. N., Fu, H., Yang, Y., Song, J., et al. (2021). "Closing the "quantum supremacy" gap: achieving real-time simulation of a random quantum circuit using a new Sunway supercomputer," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '21* (New York, NY: Association for Computing Machinery), 1–12. doi: 10.1145/3458817.3487399

Lü, X., Cheng, C., Gong, J., and Guan, L. (2011). Review of data storage and management technologies for massive remote sensing data. *Sci. China Technol. Sci.* 54, 3220–3232. doi: 10.1007/s11431-011-4549-z

Lyell, D., and Coiera, E. (2017). Automation bias and verification complexity: a systematic review. *J. Am. Med. Inform. Assoc.* 24, 423–431. doi: 10.1093/jamia/ocw105

Mao, F., Khamis, K., Krause, S., Clark, J., and Hannah, D. M. (2019). Low-cost environmental sensor networks: recent advances and future directions. *Front. Earth Sci.* 7:221. doi: 10.3389/feart.2019.00221

Matulis, M., and Harvey, C. (2021). A robot arm digital twin utilising reinforcement learning. *Comput. Graph.* 95, 106–114. doi: 10.1016/j.cag.2021.01.011

Meeds, T., and Welling, M. (2015). "Optimization monte carlo: efficient and embarrassingly parallel likelihood-free inference," in *Advances in Neural Information Processing Systems, Vol. 28* (Red Hook, NY: Curran Associates, Inc).

Metropolis, N., and Ulam, S. (1949). The Monte Carlo Method. *J. Am. Stat. Assoc.*, 44, 335–341. doi: 10.1080/01621459.1949.10483310

Micheli, V., Alonso, E., and Fleuret, F. (2023). Transformers are sample-efficient world models. *arXiv*. arXiv:2209.00588. doi: 10.48550/arXiv.2209.00588

Microsoft Azure (2024). *Microsoft Azure High Performance Computing*. Available at: https://azure.microsoft.com/en-us/solutions/high-performance-computing (accessed January 27, 2024).

National Science Foundation (2022). *CISE Computing Fact Sheet*. Available at: https://web.archive.org/web/20241004115000/https://www.nsf.gov/news/factsheets/ComputingFactsSheet_EditsSep22_JMv02_Final.pdf (accessed October 4, 2024).

Ng, D., Lin, A., Kashyap, A., Li, G., and Lu, X. (2024). "NVMe-oPF: designing efficient priority schemes for NVMe-over-fabrics with multi-tenancy support," in *2024 IEEE International Parallel and Distributed Processing Symposium (IPDPS)* (San Francisco, CA: IEEE), 519–531. doi: 10.1109/IPDPS57955.2024.00052

Ng, D., Schmierer, C., Lin, A., Liu, Z., Yu, F., Newsam, S., et al. (2023). "Benchmarking object detection models with mummy nuts datasets," in *Benchmarking, Measuring, and Optimizing, Lecture Notes in Computer Science*, eds. A. Gainaru, C. Zhang, and C. Luo (Cham: Springer International Publishing), 102–119. doi: 10.1007/978-3-031-31180-2_7

Nguyen, T., Brandstetter, J., Kapoor, A., Gupta, J. K., and Grover, A. (2023). ClimaX: a foundation model for weather and climate. *arXiv*. arXiv:2301.10343. doi: 10.48550/arXiv.2301.10343

Nipen, T. N., Haugen, H. H., Ingstad, M. S., Nordhagen, E. M., Salihi, A. F. S., Tedesco, P., et al. (2024). Regional data-driven weather modeling with a global stretched-grid. *arXiv*. arXiv:2409.02891. doi: 10.48550/arXiv.2409.02891

Nunes, D. S., Zhang, P., and Sá Silva, J. (2015). A survey on human-in-the-loop applications towards an internet of all. *IEEE Commun. Surv. Tutor.* 17, 944–965. doi: 10.1109/COMST.2015.2398816

NVIDIA (2023). *NVIDIA H100 Tensor Core GPU Architecture v1.04*. Whitepaper, NVIDIA.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., et al. (2022). Training language models to follow instructions with human feedback. *arXiv*. arXiv:2203.02155. doi: 10.48550/arXiv.2203.02155

Panda, D. K., Lu, X., and Shankar, D. (2022). *High-Performance Big Data Computing*. Cambridge, MA: The MIT Press. doi: 10.7551/mitpress/11451.001.0001

Parker, W. S. (2013). Ensemble modeling, uncertainty and robust predictions. *WIREs Clim. Change* 4, 213–223. doi: 10.1002/wcc.220

Pathak, J., Subramanian, S., Harrington, P., Raja, S., Chattopadhyay, A., Mardani, M., et al. (2022). FourCastNet: a global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv*. arXiv:2202.11214. doi: 10.48550/arXiv.2202.11214

Pearl, J. (1985). "Bayesian networks: a model of self-activated memory for evidential reasoning," in *Conference of the Cognitive Science Society* (Irvine, CA: University of California, Irvine), 14.

Peng, B., Alcaide, E., Anthony, Q., Albalak, A., Arcadinho, S., Biderman, S., et al. (2023). "RWKV: reinventing RNNs for the transformer era," in *Findings of the Association for Computational Linguistics: EMNLP 2023* (Singapore: ACL), 14048–14077. doi: 10.18653/v1/2023.findings-emnlp.936

Pylianidis, C., Snow, V., Overweg, H., Osinga, S., Kean, J., Athanasiadis, I. N., et al. (2022). Simulation-assisted machine learning for operational digital twins. *Environ. Modell. Softw* 148:105274. doi: 10.1016/j.envsoft.2021.105274

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., et al. (2021). "Learning transferable visual models from natural language supervision," in *Proceedings of the 38th International Conference on Machine Learning* (PMLR), 8748–8763.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). *Improving Language Understanding by Generative Pre-Training*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). *Language Models are Unsupervised Multitask Learners*. OpenAI Blog.

Avilable at: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

Ritto, T. G., and Rochinha, F. A. (2021). Digital twin, physics-based model, and machine learning applied to damage detection in structures. *Mech. Syst. Signal Process* 155:107614. doi: 10.1016/j.ymssp.2021.107614

Rodríguez, F., Chicaiza, W. D., Sánchez, A., and Escaño, J. M. (2023). Updating digital twins: methodology for data accuracy quality control using machine learning techniques. *Comput. Ind*. 151:103958. doi: 10.1016/j.compind.2023.103958

Romanus, M., Ross, R. B., and Parashar, M. (2015). Challenges and considerations for utilizing burst buffers in high-performance computing. *arXiv*. arXiv:1509.05492. doi: 10.48550/arXiv.1509.05492

Roy, S., and Schmude, J. (2024). *Towards Digital Twin: Introduction to Foundation Models for Geoscience*. IEEE GRSS. Available at: https://www.grss-ieee.org/events/towards-digital-twin-introduction-to-foundation-models-for-geoscience/

Schmude, J., Roy, S., Trojak, W., Jakubik, J., Civitarese, D. S., Singh, S., et al. (2024). Prithvi WxC: foundation model for weather and climate. *arXiv*. arXiv:2409.13598 doi: 10.48550/arXiv.2409.13598

Scott, W. R., Roth, G., and Rivest, J.-F. (2003). View planning for automated three-dimensional object reconstruction and inspection. *ACM Comput. Surv*. 35, 64–96. doi: 10.1145/641865.641868

Segovia, M., and Garcia-Alfaro, J. (2022). Design, modeling and implementation of digital twins. *Sensors* 22:5396. doi: 10.3390/s22145396

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. *arXiv*. arXiv:1508.07909. doi: 10.48550/arXiv.1508.07909

Siddiqui, Y., Alliegro, A., Artemov, A., Tommasi, T., Sirigatti, D., Rosov, V., et al. (2023). MeshGPT: generating triangle meshes with decoder-only transformers. *arXiv*. arXiv:2311.15475. doi: 10.48550/arXiv.2311.15475

Stocks, R., Vallejo, J. L. G., Yu, F. C. Y., Snowdon, C., Palethorpe, E., Kurzak, J., et al. (2024). "Breaking the million-electron and 1 EFLOP/s barriers: biomolecular-scale Ab initio molecular dynamics using MP2 potentials," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis, SC '24* (Atlanta, GA: IEEE Press), 1–12. doi: 10.1109/SC41406.2024.00015

Sun, C., Shrivastava, A., Singh, S., and Gupta, A. (2017). "Revisiting unreasonable effectiveness of data in deep learning Era," in *2017 IEEE International Conference on Computer Vision (ICCV)* (Venice: IEEE), 843–852. doi: 10.1109/ICCV.2017.97

Sun, S., Zheng, X., Villalba-Díez, J., and Ordieres-Meré, J. (2020). Data handling in industry 4.0: interoperability based on distributed ledger technology. *Sensors* 20:3046. doi: 10.3390/s20113046

Sutton, R. (2019). *The Bitter Lesson*. Incomplete Ideas. Available at: http://www.incompleteideas.net/IncIdeas/BitterLesson.html (accessed February 18, 2024).

Sutton, R. S., and Barto, A. (2020). *Reinforcement Learning: An Introduction. Adaptive Computation and Machine Learning*, 2nd Edn. Cambridge, MA: The MIT Press.

Taylor, M., Caldwell, P. M., Bertagna, L., Clevenger, C., Donahue, A., Foucar, J., et al. (2023). "The simple cloud-resolving E3SM atmosphere model running on the frontier exascale system," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '23* (New York, NY: Association for Computing Machinery), 1–11.

Tennenholtz, G., Merlis, N., Shani, L., Mladenov, M., and Boutilier, C. (2023). Reinforcement Learning with History Dependent Dynamic Contexts. In Proceedings of the 40th International Conference on Machine Learning, pages 34011-34053. PMLR.

Thelen, A., Zhang, X., Fink, O., Lu, Y., Ghosh, S., Youn, B. D., et al. (2022a). A comprehensive review of digital twin – part 1: Modeling and twinning enabling technologies. *Struct. Multidiscip. Optim*. 65:354. doi: 10.1007/s00158-022-03425-4

Thelen, A., Zhang, X., Fink, O., Lu, Y., Ghosh, S., Youn, B. D., et al. (2022b). A comprehensive review of digital twin–part 2: roles of uncertainty quantification and optimization, a battery digital twin, and perspectives. *Struct. Multidiscip. Optim*. 66:1. doi: 10.1007/s00158-022-03410-x

Thrun, S. (1998). "Lifelong learning algorithms," in *Learning to learn*, eds. S. Thrun and L. Pratt (Boston, MA: Springer US), 181–209.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., et al. (2023a). LLaMA: open and efficient foundation language models. *arXiv*. arXiv:2302.13971. doi: 10.48550/arXiv.2302.13971

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., et al. (2023b). Llama 2: open foundation and fine-tuned chat models. *arXiv*. arXiv:2307.09288. doi: 10.48550/arXiv.2307.09288

Trinh, T. H., Wu, Y., Le, Q. V., He, H., and Luong, T. (2024). Solving olympiad geometry without human demonstrations. *Nature* 625, 476–482. doi: 10.1038/s41586-023-06747-5

Uhlemann, T. H. J., Schock, C., Lehmann, C., Freiberger, S., and Steinhilper, R. (2017). The digital twin: demonstrating the potential of real time data acquisition in production systems. *Procedia Manuf*. 9, 113–120.doi: 10.1016/j.promfg.2017.04.043

Valevski, D., Leviathan, Y., Arar, M., and Fruchter, S. (2024). Diffusion models are real-time game engines. *arXiv*. arXiv:2408.14837. doi: 10.48550/arXiv.2408.14837

van der Valk, H., Hasse, H., Möller, F., Arbter, M., Henning, J.-L., and Otto, B. (2020). "A taxonomy of digital twins," in *AMCIS 2020 Proceedings, Organizational Transformation & Information Systems (SIGORSA)*. AIS Electronic Library.

VanDerHorn, E., Mahadevan, S. (2021). Digital twin: generalization, characterization and implementation. *Decis. Support Syst*. 145:113524. doi: 10.1016/j.dss.2021.113524

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2023). Attention is all you need. *arXiv*. arXiv:1706.03762. doi: 10.48550/arXiv.1706.03762

Verdouw, C., Tekinerdogan, B., Beulens, A., and Wolfert, S. (2021). Digital twins in smart farming. *Agric. Syst*. 189:103046. doi: 10.1016/j.agsy.2020.103046

Vysocký, O., and Riha, L. (2024). "VR dashboard for data center monitoring," *The 2nd Digital Twins Workshop for High-Performance Computing at SC'24* (Atlanta, GA).

Wang, S., Lai, X., He, X., Li, K., Lv, L., Song, X., et al. (2024). Optimal sensor placement for digital twin based on mutual information and correlation with multi-fidelity data. *Eng. Comput*. 40, 1289–1308. doi: 10.1007/s00366-023-01858-z

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Adv. Neural Inf. Process. Syst*. 35, 24824–24837.

Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *J. Big Data* 3:9. doi: 10.1186/s40537-016-0043-6

Weisstein, E. W. (2024). "Markov sequence," *MathWorld*. Wolfram Research, Inc. Available at: https://mathworld.wolfram.com/MarkovSequence.html (accessed November 27, 2024).

Willard, J., Jia, X., Xu, S., Steinbach, M., and Kumar, V. (2022). Integrating scientific knowledge with machine learning for engineering and environmental systems. *ACM Comput. Surv*. 55:66. doi: 10.1145/3514228

Williams, R. J., and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Comput*. 1, 270–280. doi: 10.1162/neco.1989.1.2.270

Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., et al. (2023). A diffusion models: a comprehensive survey of methods and applications. *ACM Comput. Surv*. 56, 105:1–105:39. doi: 10.1145/3626235

Yang, M., Du, Y., Ghasemipour, K., Tompson, J., Kaelbling, L., Schuurmans, D., et al. (2024). Learning interactive real-world simulators. *arXiv*. arXiv:2310.06114. doi: 10.48550/arXiv.2310.06114

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., et al. (2023). Tree of thoughts: deliberate problem solving with large language models. *Adv. Neural Inf. Process. Syst*. 36, 11809–11822.

Yin, J., Liang, S., Liu, S., Bao, F., Chipilski, H. G., Lu, D., et al. (2024). A scalable real-time data assimilation framework for predicting turbulent atmosphere dynamics. *arXiv*. arXiv:2407.12168. doi: 10.48550/arXiv.2407.12168

Zeng, A., Chen, M., Zhang, L., and Xu, Q. (2023). Are transformers effective for time series forecasting? *Proc. AAAI Conf. Artif. Intell*. 37, 11121–11128. doi: 10.1609/aaai.v37i9.26317

Zhang, H., Qi, Q., Ji, W., and Tao, F. (2023). An update method for digital twin multi-dimension models. *Robot. Comput. Integr. Manuf*. 80:102481. doi: 10.1016/j.rcim.2022.102481

Zheng, Y., Yang, S., and Cheng, H. (2019). An application framework of digital twin and its case study. *J. Ambient Intell. Humaniz. Comput*. 10, 1141–1153. doi: 10.1007/s12652-018-0911-3

Zhou, K., Doyle, J. C., and Glover, K. (1996). *Robust and Optimal Control*. Upper Saddle River, NJ: Prentice Hall.

Zilberstein, S. (1996). Using anytime algorithms in intelligent systems. *AI Mag*. 17, 73–73.