frontiers | Frontiers in Chemistry

ORIGINAL RESEARCH
published: 26 May 2022
doi: 10.3389/fchem.2022.866585

Improving Small Molecule pK_a Prediction Using Transfer Learning With Graph Neural Networks

Fritz Mayr[†‡], *Marcus Wieder*[*†‡], *Oliver Wieder*[†] *and Thierry Langer*[†]

Department of Pharmaceutical Sciences, Pharmaceutical Chemistry Division, University of Vienna, Vienna, Austria

Enumerating protonation states and calculating microstate pK_a values of small molecules is an important yet challenging task for lead optimization and molecular modeling. Commercial and non-commercial solutions have notable limitations such as restrictive and expensive licenses, high CPU/GPU hour requirements, or the need for expert knowledge to set up and use. We present a graph neural network model that is trained on 714,906 calculated microstate pK_a predictions from molecules obtained from the ChEMBL database. The model is fine-tuned on a set of 5,994 experimental pK_a values significantly improving its performance on two challenging test sets. Combining the graph neural network model with Dimorphite-DL, an open-source program for enumerating ionization states, we have developed the open-source Python package pkasolver, which is able to generate and enumerate protonation states and calculate pK_a values with high accuracy.

Keywords: physical properties, PKA, Graph Neural Network (GNN), transfer learning, protonation states

OPEN ACCESS

Edited by:
Marco Tutone,
University of Palermo, Italy

Reviewed by:
Jean-Louis Reymond,
University of Bern, Switzerland
Kun Yao,
Schrodinger, United States

***Correspondence:**
Marcus Wieder
marcus.wieder@gmail.com

†ORCID:
Fritz Mayr
orcid.org/0000-0002-6621-2108
Marcus Wieder
orcid.org/0000-0003-2631-8415
Oliver Wieder
orcid.org/0000-0003-4967-7613
Thierry Langer
orcid.org/0000-0002-5242-1240

‡*These authors have contributed equally to this work*

Specialty section:
This article was submitted to Theoretical and Computational Chemistry, a section of the journal Frontiers in Chemistry

Received: *31 January 2022*
Accepted: *04 April 2022*
Published: *26 May 2022*

Citation:
Mayr F, Wieder M, Wieder O and Langer T (2022) Improving Small Molecule pK_a Prediction Using Transfer Learning With Graph Neural Networks. Front. Chem. 10:866585. doi: 10.3389/fchem.2022.866585

1 INTRODUCTION

The acid dissociation constant (K_a), most often written as its negative logarithm (pK_a), plays a significant role in molecular modeling, as it influences the charge, tautomer configuration, and overall 3D structure of molecules with accessible protonation states in the physiological pH range. All these factors further shape the mobility, permeability, stability, and mode of action of substances in the body (Manallack et al., 2013). In case of insufficient or missing empirical data, the correct determination of pK_a values is thus essential to correctly predict the aforementioned molecular properties.

Authors and studies disagree on the exact percentage of drugs with ionizable groups, but a conservative estimate suggests that at least two-thirds of all drugs contain one or more ionization groups (in a pH range of 2–12) (Manallack, 2007). The importance of pK_a predictions for drug discovery has been widely recognized and has been the topic of multiple blind predictive challenges—most notable the Statistical Assessment of Modeling of Proteins and Ligands (SAMPL) series SAMPL6 (Işık et al., 2021), SAMPL7, (Bergazin et al., 2021) and ongoing SAMPL8 [1] challenge.

Multiple methods have been developed to estimate pK_a values of small molecules, ranging from physical models based on quantum chemistry calculations (Selwa et al., 2018; Tielker et al., 2018) and/or free energy calculations (Prasad et al., 2018; Zeng et al., 2018) to empirical models based on

[1]https://github.com/samplchallenges/SAMPL8

linear free energy relationships using the Hammet-Taft equation or more data driven methods using quantitative structure-property relationship (QSPR) and machine learning (ML) approaches like deep neural network or random forest models (Liao and Nicklaus, 2009a; Rupp et al., 2011; Mansouri et al., 2019; Baltruschat and Czodrowski, 2020a; Bergazin et al., 2021). In general empirical methods require significantly less computational effort than their physics-based counterparts once they are parameterized but require a relatively large number of high-quality data points as training set (Bergazin et al., 2021).

In recent years, machine learning methods have been widely applied to predict different molecular properties including pK_a predictions. Many of these approaches learn pK_a values on fingerprint representations of molecules (Baltruschat and Czodrowski, 2020a; Yang et al., 2020). The pK_a value of an acid and conjugate base pair is determined by the molecular structure and the molecular effects on the reaction center exerted by its neighborhood, including mesomeric, inductive, steric, and entropic effects (Perrin et al., 1981). Ideally, these effects should be included and encoded in a suitable fingerprint or set of descriptors. For many applications, extended-connectivity fingerprints (ECFPs) in combination with molecular features have proven to be a suitable and powerful tool to learn structure-property relationships (Rogers and Hahn, 2010; Jiang et al., 2021).

The emergence of graph neural networks (GNNs) has shifted some focus from descriptors and fingerprints designed by domain experts to these emerging deep learning methods. GNNs are a class of deep learning methods designed to perform inference on data described by graphs and provide straightforward ways to perform node-level, edge-level, and graph-level prediction tasks (Wu et al., 2019; Wieder et al., 2020; Zhou et al., 2020). GNNs are capable of learning representations and features for a specific task in an automated way eliminating the need for excessive feature engineering ((Gilmer et al., 2017)). Another aspect of their attractiveness for molecular property prediction is the ease with which a molecule can be described as an undirected graph, transforming atoms to nodes and bonds to edges encoded both atom and bond properties. GNNs have proven to be useful and powerful tools in the machine learning molecular modeling toolbox (Gilmer et al., 2017; Deng et al., 2021).

Pan et al. (Pan et al., 2021) have shown that GNNs can be successfully applied to pK_a predictions of chemical groups of a molecule, outperforming more traditional machine learning models relying on human-engineered descriptors and fingerprints, developing MolGpka, a web server for predicting pK_a values. MolGpka was trained on molecules extracted from the ChEMBL database (Gaulton et al., 2012) containing predicted pK_a values (predicted with ACD/Labs Physchem software[2]). Only the most acidic and most basic pK_a values were considered for the training of the GNN models.

The goal of this work was to extend the scope of predicting pK_a values for independently ionizable atoms (realized in MolGpka) and develop a workflow that is able to enumerate protonation states and predict the corresponding pK_a values connecting them (sometimes referred to as "sequential pK_a prediction"). To achieve this we implemented and trained a GNN model that is able to predict values for both acidic and basic groups by considering the protonated and deprotonated species involved in the corresponding acid-base reaction. We trained the model in two stages. First, we started by pre-training the model on calculated microstate pK_a values for a large set of molecules obtained from the ChEMBL database (Gaulton et al., 2012). The pre-trained model already performs well on the two independent test sets used to measure the performance of the trained models. To improve its performance we fine-tuned the model on a small training set of molecules for which experimental pK_a values were available. The fine-tuned model shows excellent and improved performance on the two test sets.

We have implemented the training routine and prediction pipeline in an open-source Python package named pkasolver, which is freely available and can be obtained as described in the Code and data availability section. Due to the terms of its licence agreement we are unable to distribute models trained using results generated with Epik. Users with an Epik licence can follow the instructions outlined in the data repository to obtain the fine-tuned models. For users without such a licence we provide models trained without Epik. We also provide a ready-to-use Google Colab Jupyter notebook which includes trained models and can be used to predict pk_a values for molecules without locally installing the package (for further information see the Code and data availability section) (Bisong, 2019).

2 RESULTS AND DISCUSSION

We will start by discussing the performance of the model on the validation set of the ChEMBL data set (which contains pK_a values calculated with Epik on a subset of the ChEMBL database) and the two independent test sets: the Novartis test set (280 molecules) and the Literature test set (123 molecules). This will be followed by a discussion of the fine-tuned model on its validation set (experimental data set), on both test sets, and on the ChEMBL data set. Subsequently, we will discuss the performance of the models trained only on the monoprotic experimental data set (without transfer learning). Finally, we will discuss the developed pkasolver package, its use cases, and limitations.

Performance of the different predictive models is subsequently reported using the mean absolute error (MAE) and root mean squared error (RMSE). For each metric (MAE and RMSE) the median value from 50 repetitions with different training/validation set splits is reported and the 90% confidence interval is shown. To visualize training results a single training run (out of the 50) was randomly selected and the results on the validation set plotted.

In the following sections we will use the term pkasolver to describe the sequential pK_a prediction pipeline using trained GNN models. To distinguish between the transfer learning approach (models trained both on the and experimental data

[2]version 12.01, Advanced Chemistry Development Inc. 2010ACD/Labs

TABLE 1 | Performance of state-of-the-art knowledge-based approaches and commercial software solutions to predict pK_a values on the Novartis and Literature test sets are shown. For each data set, the mean absolute error (MAE) and root mean squared error (RMSE) is calculated. For MolGpKa, Epik, pkasolver-epic, and pkasolver-light the median value and the 90% confidence interval are reported.

Model	Novartis data set		Literature data set	
	MAE	RMSE	MAE	RMSE
Random Forest[1,3]	1.15	1.51	0.53	0.76
ChemAxon Marvin (V20.1.0)[3]	0.86	1.17	0.57	0.87
MolGpKa Pan et al. (2021)	0.87 [0.77;0.97]	1.27 [1.08;1.45]	0.49 [0.40;0.65]	1.00 [0.56;1.53][4]
Epik[2] Pan et al. (2021)	0.83 [0.75;0.91]	1.16 [1.06;1.26]	0.58 [0.48;0.67]	0.92 [0.74;1.12]
pkasolver-epic	0.71 [0.64;0.74]	0.93 [0.85;0.97]	0.52 [0.49;0.56]	0.82 [0.76;0.86]
pkasolver-light	0.86 [0.81;0.94]	1.13 [1.04;1.20]	0.56 [0.51;0.64]	0.82 [0.71;0.93]

[1]*Used a random forest implementation with 1,000 estimators and the FCFP6 fingerprint. Values for the best performing random forest implementation are shown.*

[2]*Epik identified different protonation centers than were reported in the data sets for the Novartis data set for 26 out of 280 molecules. These molecules were excluded from the MAE and RMSE calculation for Epik.*

[3]*values were obtained from Baltruschat and Czodrowski (Baltruschat and Czodrowski, 2020a).*

[4]*the reason for the large confidence interval is the incorrect prediction for a single molecule (Isomeric Smiles: CCNC) by MolGpKa with an error of 8.86 pK_a units*

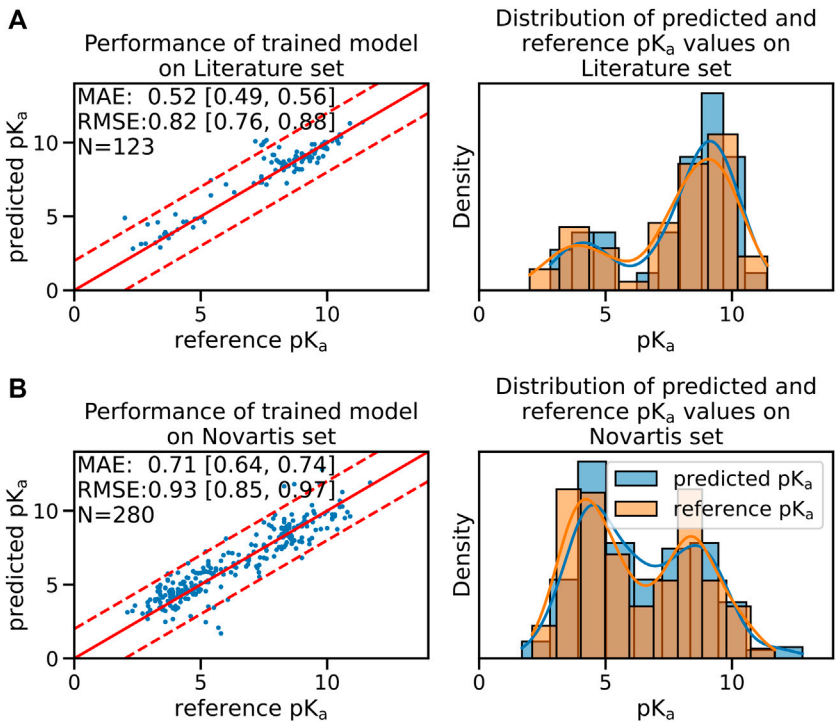

FIGURE 1 | The fine-tuned GNN model is able to predict the pK_a values of the Novartis and Literature test set with high accuracy. Panel **(A)** shows the performance of the fine-tuned model (initially trained with the ChEMBL data set and subsequently fine-tuned on the experimental data set) on the Literature test set. Panel **(B)** shows the performance of the same model on the Novartis test set. The solid red line in the scatter plot indicates the ideal behavior of the reference and predicted pK_a values, the dashed lines mark the ±1 pK_a unit interval. Mean absolute error (MAE) and root mean squared error (RMSE) is shown, the values in bracket indicate the 90% confidence interval calculated from 50 repetitions with random training/validation splits. *N* indicates the number of investigated samples.

set) and the models trained *only* on the experimental data set we will indicate the former with pkasolver-epic and the latter with the keyword pkasolver-light.

2.1 Pre-Training Model Performance

The initial training of the GNN model was performed using the ChEMBL data set (microstate pK_a values calculated with Epik). **Supplementary Figure S3A** shows the results of the best performing model on the hold-out validation set. The MAE and RMSE are 0.29 [90% CI: 0.28; 0.31] and 0.45 [90% CI:0.44; 0.49] pK_a units shows a good fit across the reference pK_a values. The kernel density estimates (KDE) of the distribution of the reference and predicted pK_a values shown in **Supplementary Figure S3A** highlights the ability of the GNN to correctly learn to predict pK_a values throughout the investigated pH range.

The performance of the trained GNN model was assessed on two independent test sets: the Novartis and the Literature test set (both test sets are described in detail in the Methods section) (Baltruschat and Czodrowski, 2020a). The trained model performs well on both test sets with a MAE of 0.62 [90% CI:0.57;0.67] and a RMSE of 0.97 [90% CI:0.89;1.10] pk_a units on the Literature test set and a MAE of 0.82 [90% CI:0.77;0.85] and a RMSE of 1.13 [90% CI:1.05;1.21] pk_a units on the Novartis test set (shown in **Supplementary Figure S2**). The performance is comparable to the performance of Epik and Marvin on both test sets (shown in **Table 1**).

2.2 Fine-Tuned Model Performance

While the performance on the test sets of the pre-trained model was already acceptable we were able to further increase model accuracy by fine-tuning the pre-trained model using a data set of experimentally measured pK_a values. The performance of the fine-tuned model on the validation set of the experimental data set is shown in **Supplementary Figure S3B**. The median performance of the fine-tuned model was improved from a RMSE of 0.97 [90% CI:0.89;1.10] to 0.82 [90% CI:0.76;0.88] pK_a units on the Literature test set and from a RMSE of 1.13 [90% CI:1.05;1.21] to 0.93 [90% CI:0.85;0.97] pK_a units on the Novartis test set (shown in **Figure 1**).

In order to avoid model performance degradation on the ChEMBL data set we randomly added molecules from the ChEMBL data set during the fine-tuning workflow. Adding molecules from the ChEMBL data set to restrict model parameters and avoid overfitting decreased the performance of the fine-tuned model on the ChEMBL data set only slightly (shown in **Supplementary Figure S4**). This was necessary since previous attempts without regularization showed decreased accuracy of the fine-tuned model in regions outside the limited pH range of the experimental data set while improving the performance on the test sets (details to the pH range of both the ChEMBL and experimental data set are shown in **Supplementary Figure S6**). An example of the performance of the fine-tuned model on the ChEMBL data set without regularization is shown in **Supplementary Figure S7**.

To set the performance of the fine-tuned model in context we compare its performance with two recent publications investigating pK_a predictions using machine learning. In **Table 1** the results are summarized for the methods presented in both Baltruschat and Czodrowski (Baltruschat and Czodrowski, 2020a) and Pan et al. (Pan et al., 2021). We extracted data from these publications where appropriate and recalculated values if needed. Pan et al. (Pan et al., 2021) split the reported results into basic and acidic groups making it necessary to recalculate the values reported there for MolGpKa and Epik, the values for Marvin were taken directly from reference (Baltruschat and Czodrowski, 2020a) (reported values were calculated without confidence interval). The fine-tuned GNN model (shown as pkasolver-epic in **Table 1**) performs on a par with the best performing methods reported there.

It is difficult to rationalize MAE/RMSE differences between different methods/models shown in **Table 1**) since training sets and methods are different. The small difference in performance between pkasolver-epic and MolGpka could be attributed to the transfer learning routine which added experimentally measured pK_a values. The random forest model was trained on significantly less data (only on the 5,994 pk_a values present in the experimental data set) than either pkasolver or MolGpka yet performs comparably to both on the Literature data set while significantly worse on the Novartis data set. This might highlight the complexity of the Novartis data set, an observation previously made and investigated in Pan et al. (Pan et al., 2021).

Both Epik and Marvin perform well on both test data sets. It is surprising that pkasolver-epic can slightly outperform Epik, even though its initial training was based on data calculated by Epik. We think this emphasizes the potential of transfer learning as used in this work and data-driven deep learning in general.

2.3 Training on the Experimental Data Set Without Transfer Learning

To provide a ready-to-use pK_a prediction pipeline for which we can distribute the trained models under the MIT licence we trained models exclusively on the experimental data set. The performance on the Novartis and Literature data set of these models is shown in **Supplementary Figure S5** and summarized in **Table 1** (shown as pkasolver-light). While the results are comparable to Epik and MolGpKa on the test sets it is important to stress that both test sets contain only monoprotoic molecules (Baltruschat and Czodrowski, 2020a).

2.4 Sequential pK_a Predictions With Pkasolver

Combining the trained GNN models with Dimorphite-DL, a tool that identifies potential protonation sites and enumerates protonation states, enabled us to perform sequential pK_a predictions. A detailed description of this approach is given in the Detailed methods section. We investigated multiple mono- and polyprotic molecules for qualitative and quantitative agreement between prediction and experimental data. The results for the investigate systems were of excellent consistency using pkasolver-epic and of reasonable accuracy using pkasolver-light. The list of molecules that we tested is included in the pkasolver repository and a subset of molecules of general interest for drug discovery are discussed in detail in the **Supplementary Materials** section.

2.5 Limiations of Pkasolver

The sequential pK_a prediction of pkasolver generates microstates and the calculated pK_a values are microstate pK_a values. One limitation of pkasolver is that only a single microstate per macrostate is generated. Tautomeric and mesomeric states are *never* changed during the sequential de-/protonation (i.e., double bond positions are fixed). For each protonation state the bond patter of the molecule that was proposed by Dimorphite-DL at pH 7.4 is used. This shortcoming has several consequences. First, it leads to unusual protonation states. One example that has been observed throughout the sequential pK_a prediction tests with

pkasolver-epic are amide groups with a negative charge on the nitrogen atom. The more likely position of the charge is the more electronegative oxygen atom. This has little practical consequence since this pattern was also present in the pK_a prediction training set generated with Epik (the mesomeric state was fixed in training too). A far more severe limitation is the fact that it is not possible to model microstates within a singe macrostate, since tautomers can not be changed (Gunner et al., 2020). To overcome this limitation it is necessary to enumerate tautomers for each protonation state and estimate their relative population. Solving this particular problem will be part of future work.

2.5.1 Limitations of Pkasolver-Light

The training set of pkasolver-light contains only monoprotic pK_a data with the majority of pK_a values between 4 and 10 (as shown in **Supplementary Figure S6**) (Baltruschat and Czodrowski, 2020a). The trained models are not necessarily suitable for polyprotic molecules. This limitation becomes apparent in the in depth discussion of some mono- and polyprotic molecules discussed in the **Supplementary Materials** section. For polyprotic molecules it is highly recommended to use pkasolver-epic instead of the pkasolver-light.

2.5.2 Limitations of Pkasolver-Epic

The pre-training data set imposes limitations on the applicability domain of the pK_a predictions with pkasolver-epic. The selection criteria of the pre-training data set are described in the Methods section. In **Supplementary Figure S8** the distribution of several molecular properties (molecular weight, number of heteroatoms, number of hydrogen bond acceptor/donor, frequency of elements) are shown. The transferability of the trained models for molecules outside these distributions has not been tested and the usage of pkasolver-epic for such molecules is not recommended.

3 DETAILED METHODS

3.1 Data Set Generation and Pre-processing

Four different data sets were used in this work: the ChEMBL data set, the experimental data set, the Novartis data set and the Literature data set.

The ChEMBL data set used for pre-training was obtained from the ChEMBL database using the number of Rule-of-Five violations (set to a maximum of one violation) as filter criteria (Gaulton et al., 2012; Davies et al., 2015). For each of the molecules, a pK_a scan for the pH range between zero and 14 was performed using the Schrodinger tool Epik (Shelley et al., 2007; Greenwood et al., 2010) (Version-2121-1). The sequential pK_a scan indicated for 320,800 molecules one or multiple protonation state/s, resulting in a total of 729,375 pK_a values. For each pK_a value, Epik further indicated the protonation center using the atom index of the heavy atom at which either a hydrogen is attached or removed.

To perform transfer learning we obtained a second data set with experimental pK_a values. This data set (subsequently called 'experimental data set') was developed by Baltruschat and Czodrowski (Baltruschat and Czodrowski, 2020b) and can be acquired from their GitHub repository[3]. For a detailed description of the curating steps taken to generate this data set, we point the reader to the Methods section of (Baltruschat and Czodrowski, 2020b). The experimental data set consists of 5,994 unique molecules, each with a single pK_a value and an atom index indicating the reaction center. Some of the molecules had to be corrected to obtain their protonation state at pH 7.4 (examples shown in ??).

To test the performance of the models, two independent data sets were used, which were provided and curated by Baltruschat and Czodrowski (Baltruschat and Czodrowski, 2020b). The Literature data set contains 123 compounds collected by manual curating the literature. The Novartis data set contains 280 molecules provided by Novartis (Liao and Nicklaus, 2009b). For each molecule, a pK_a value and atom index indicating the reaction center was provided. To avoid training the model on molecules present in the Literature or Novartis data set we filtered the ChEMBL data set using the InChIKey and canonical SMILES strings of the neutralized molecules as matching criteria. 50 molecules were identified and removed from the ChEMBL data set. All checks were performed using RDKit (RDKit and Open-Source Chemiformatics, 2022).

3.2 Enumerate Protonation States During Training/Testing

The goal of calculating microstate pK_a values is to find the pH value at which the concentration of two molecular species is equal. To do this efficiently, we provide as input the protonated and deprotonated molecular species of the acid-base pair for which we want to calculate the pK_a value (the Brønsted acid/base definitions are used here and subsequently (McNaught and Wilkinson, 2014)). This approach enables a consistent treatment of acids and bases with a single data structure (the acid-base pair).

This workflow made it necessary that we generate the molecular species at each protonation state starting from the molecule at pH 7.4 by removing or adding hydrogen to the reaction center (which was calculated by Marvin for the experimental, Novartis, and Literature data set and Epik for the ChEMBL data set). We do this by sequentially adding hydrogen atoms from highest to lowest pK_a for acids (i.e., at pH = 0 all possible protonation sites are protonated) and removing hydrogen atoms from lowest to highest pK_a value for bases on the structure present at pH 7.4 (at pH = 14 all possible protonation sites are deprotonated).

This approach presented challenges for the ChEMBL data set for which sequential pK_a values and reaction centers were calculated with Epik. Epik calculates the microstate pK_a value on the most probable tautomeric/mesomeric structure. This leads to potential protonation states that require changes in the double bond pattern and redistribution of hydrogen. Since we

[3]https://github.com/czodrowskilab/Machine-learning-meets-pKa

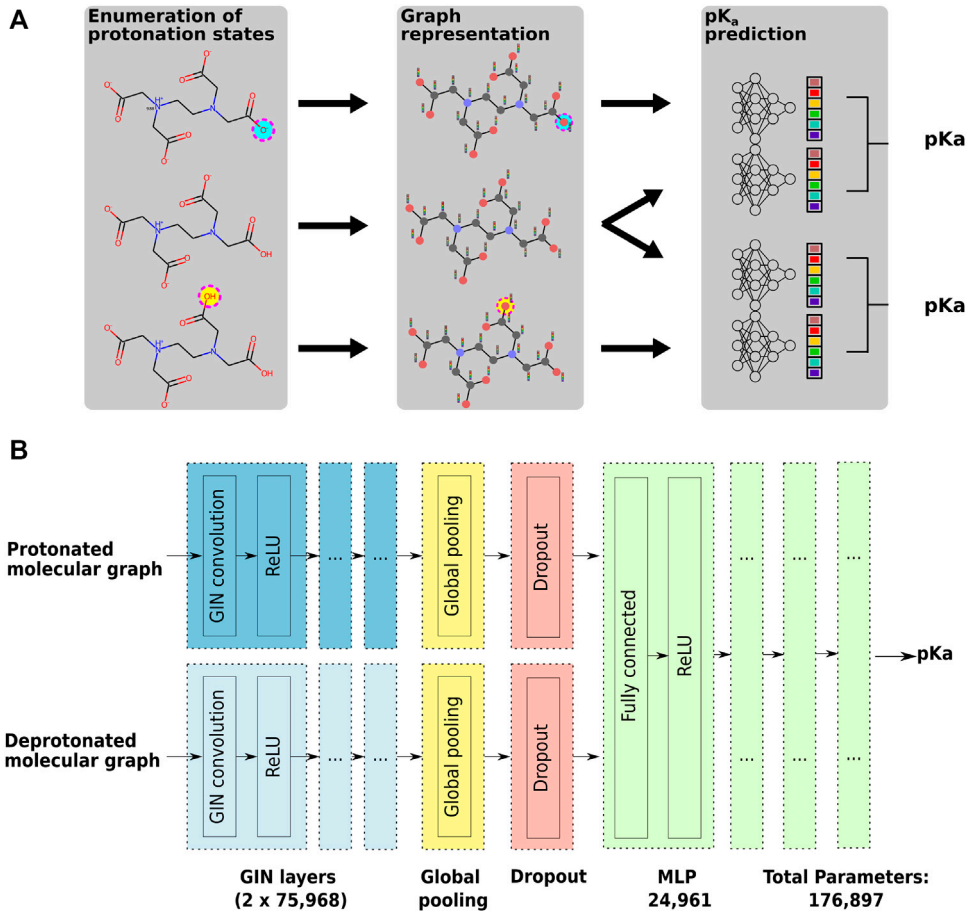

FIGURE 2 | Panel **(A)** shows the general workflow used to train the GNN on pK_a values for a single molecule. During the training and testing phase, each molecule was provided in the structure dominant at pH 7.4 with atom indices indicating the protonation sites and corresponding pK_a values connecting them. In the Enumeration of protonation states phase we generate the protonation state for each pK_a value. The molecular species for each of the protonation states are then translated in their graph representation using nodes for atoms and edges for bonds, with node feature vectors encoding atom properties in the Graph representation phase. In the pK_a prediction phase graphs of two neighboring protonation states are combined and used as input for the GNN model to predict the pK_a value for the acid-base pair [using the Brønsted–Lowry acid/base definition (McNaught and Wilkinson, 2014)]. The architecture of the GNN model is shown in detail in panel **(B)**. For a pair of neighboring protonation states two independent GIN (graph isomorphism network) convolution layers and ReLU activation functions are used for the protonated and the deprotonated molecular graph to pass information of neighboring atoms and achieve the embedding of the chemical environment of each atom (Xu et al., 2019). The output of the convolutional layer is summarized using a global average pooling layer, generating the condensed input for the multilayer perceptron (MLP). To add regularization and to prevent co-adaptation of neurons a dropout layer was added.

do not consider tautomeric changes to the molecular structure in the present implementation, such tautomeric changes can introduce invalid molecules in either the sequential removal or addition of hydrogen atoms. Whenever such molecular structures were encountered we removed these protonation states from further consideration. Additionally, we used RDKit's sanitize function to identify cases for which protonation state changes introduce invalid atom valences. In other cases in which the protonation state change on a mesomeric structure introduces valid yet improbable molecular structures (e.g. protonating the oxygen in an amide instead of the nitrogen) we keep these structures. This reduced the number of molecules and protonation states in the ChEMBL data set to 286,816 molecules and 714,906 protonation states. The distribution of pK_a values for the ChEMBL and experimental data set is shown in **Supplementary Figure S6**.

3.3 Training and Testing With PyTorch Geometric

We use PyTorch and PyTorch geometric (subsequently abbreviated as PyG) for model training, testing, and prediction of pK_a values on the graph data structures (Fey and Lenssen, 2019; Paszke et al., 2019).

3.3.1 Graph Data Structure

A graph G is defined as a set of no the nodes V and edges E connecting the nodes. Each node $v \in V$ has a feature vector x_v, which

encodes atom properties like element, charge, number of hydrogen, as well as the presence of particular SMARTS patterns as a one-hot-encoding bit vector (all atom properties are shown in **Supplementary Table S1**). The adjacency matrix A defines the connectivity of the graph. A is defined as a quadratic matrix with $A_{uv} = 1$ if there is an edge between node u and v and $A_{uv} = 0$ if there is no edge between node u and v.

We used RDKit to generate a graph representation of the molecule with atoms represented as nodes and bonds as edges (in coordinate list format [4] to efficiently represent the spares matrix).

3.3.2 Graph Neural Network Architecture

To predict a single pk_a value the graph neural network (GNN) architecture takes as input two graphs representing the conjugated acid-base pair as shown in **Figure 2**. **Figure 2B** shows the high-level architecture of the used GNN.

There are three phases to predict a pK_a value from a pair of molecular graphs. The first stage involves recurrently updating the node states using GIN (graph isomorphism network) convolution layers and ReLU activation functions (Xu et al., 2019). We used 3 GIN layers with an embedding size of 64 bits each to propagate information throughout the graph and update each node with information about the extended environment. In the second stage, a global average pooling is performed to produce the embedding of the protonated and deprotonated graph, resulting in two 32 bit vectors. Concatenating the two 32 bit vectors produces the input for the third stage, the multilayer perceptron (MLP) with 3 fully connected layers (each with an embedding size of 64). To add regularization and to prevent co-adaptation of neurons a dropout layer randomly zeros out elements of the pooling output vector with $p = 0.5$ during training. Additionally, batch normalization is applied as described in (Ioffe and Szegedy, 2015).

3.3.3 GNN Model Training

Before each training run the ChEMBL and experimental data set were shuffled and randomly split in training (90% of the data) and validation set (10% of the data). To ensure that we can reproduce these splits the seed for each split was recorded.

The mean squared error (MSE) of predicted and reference pK_a values on the training data set was calculated and parameter optimization was performed using the Adam optimizer with decoupled weight decay regularization (Loshchilov and Hutter, 2019) as implemented in PyTorch.

Model performance was evaluated on the validation set and the model with the best performance was selected either for fine-tuning or further evaluation on the test data sets. The performance on the evaluation data set was calculated after every fifth epoch and the corresponding weights were saved. The learning rate for all training runs was dynamically reduced by a factor of 0.5 if the validation set performance did not change within 150 epochs (validation set performance threshold was set to 0.1).

[4]https://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.coo_matrix.html

Pre-training of the GNN was performed on the ChEMBL data set with a learning rate of $1x10^{-3}$ and a batch size of 512 molecules for 1,000 epochs. Fine-tuning was performed using the experimental data set with a learning rate of $1x10^{-3}$ and a batch size of 64 molecules for 1,000 epochs. All parameters of the GNN models were optimized during fine-tuning. To avoid overfitting to the experimental data set we added to each batch of the fine-tuning data set a randomly selected batch (1,024 molecules) of the pre-training data set.

To calculate the confidence intervals of the model performance, pre-training and fine-tuning were repeated 50 times, each with a random training-validation set split resulting in 50 independently fine-tuned models.

3.4 Sequential pK_a Value Prediction With Pkasolver

We use Dimorphite-DL to identify the proposed structure at pH 7.4 and all de-/protonation sites for a given molecule (Ropp et al., 2019).

We iteratively protonate each of the proposed de-/protonation sites generating a molecular pair consisting of the protonated and deprotonated molecular species (in the first iteration the deprotonated molecule is the molecule at pH 7.4). For each of the protonate/deprotonated pairs a pK_a value is calculated. The protonated structure with the highest pK_a value (but below pH 7.4) is kept and the protonation site is removed from the list of possible protonation sites. This is repeated until either (1) all protonation sites are protonated, (2) no more valid molecules can be generated, or (3) the calculated pK_a values are outside the allowed pK_a range.

To enumerate all deprotonated structures we start again with the structure at pH 7.4 and start to iteratively deprotonate each of the proposed de-/protonation sites. Here, we always keep the deprotonated structure with the lowest pK_a value that is above 7.4.

pK_a values are calculated using 25 of the 50 fine-tuned GNN models. For each protonation state, the average pK_a value is calculated and the standard deviation is shown to enable the user to identify molecules or protonation states for which the GNN model estimates are uncertain.

We provide a ready to use implementation of pkasolver to predict sequential pK_a values in our GitHub repository (for further information see the Code and data availability section).

4 CONCLUSION

We have shown that GNNs can be used to predict mono- and polyprotic pK_a values and achieve excellent performance on two external test sets. Training the GNN model in two stages with a pre-training phase using a large set of molecules with calculated pK_a values and a fine-tuning phase on a small set of molecules with experimentally measured pK_a values improves the performance of the GNN model significantly. This performance boost is especially noteworthy on the

challenging Novartis test set (the RMSE was decreased from 1.18 [1.05;1.27] to 0.93 [0.85;0.97] pk_a units). A direct comparison with other software solutions and machine learning models on the two test sets shows that the fine-tuned GNN model performs consistently on a par with the best results of other commercial and non-commercial tools.

We have implemented pkasolver as an open-source and free-to-use Python package under a permissive licence (MIT licence). We provide two versions of the package: pkasolver-epic and pkasolver-light. The former performs best on both test sets and is suitable for sequential pK_a prediction on poloyprotic molecules. It was pretraind on a subset of the ChEMBL data set for which pK_a values were predicted using Epik and fine-tuned on experimental monoprotic pK_a values. Due to the terms of the licence agreement of Epik we are unable to supply the trained models but provide the training pipeline to reproduce the models (which requires an active Epik license). pkasolver-light performs well on both test sets but its application domain is limited to monoprotic molecules. These are the trained models distributed with the pkasolver package.

DATA AVAILABILITY STATEMENT

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found in the article/ **Supplementary Material**. Python package used in this work (release v0.3) and Colabs Jupyter notebook link: https://github.com/mayrf/pkasolver. Data and notebooks to reproduce the plots/figures (release v0.2): https://github.com/wiederm/pkasolver-data.

AUTHOR CONTRIBUTIONS

Conceptualization: FM, OW, TL, and MW; Methodology: FM, OW, and MW; Software: FM and MW; Investigation: FM and MW; Writing–Original Draft: FM, OW, and MW; Writing–Review and Editing: FM, OW, TL, and MW; Funding Acquisition: MW, TL; Resources: TL; Supervision: MW, TL.

FUNDING

MW acknowledges support from an FWF Erwin Schrödinger Postdoctoral Fellowship J 4245-N28. FM and TL gratefully acknowledge funding by the NeuroDeRisk project (https://www.neuroderisk.eu), which has received funding from the Innovative Medicines Initiative 2 Joint Undertaking (IMI2 JU, https://european-union.europa.eu/institutions-law-budget/institutions-and-bodies/institutions-and-bodies-profiles/imi-2-ju_en) under Grant Agreement No. 821528. This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation program and the European Federation of Pharmaceutical Industries and Associations (EFPIA, https://www.efpia.eu).

ACKNOWLEDGMENTS

MW is grateful for discussions with David Bushiri, John Chodera, Josh Fass, Nils Krieger, Magdalena Wiercioch, Steffen Hirte, Thomas Seidel, and the Tautomer Consortium, specifically Paul Czodrowski, Brian Radak, Woody Sherman, David Mobley, Christopher Bayly, and Stefan Kast. MW, OW, FM, and TL are grateful for the help of Gerhard F. Ecker and his group members who performed the reference pK_a calculations with Epik for the ChEMBL data set.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fchem.2022.866585/full#supplementary-material

Supplementary Figure S1 | Protonation state errors in the experimental data set. This exemplary selection shows molecules from the experimental data set provided by Baltruschat and Czodrowski (Baltruschat and Czodrowski, 2020a) for which the protonation state provided does not correspond to the state at pH 7.4. For examples 1, 2, 4 and 5 with experimental pK_a values below 7.4 protonation at the reaction center would result in highly unlikely pentavalent nitrogen. For example 3 and 6 with pK_a values above 7.4 deprotonation at the reaction site can not be performed because of the lack of a suitable hydrogen. These error were corrected during our data preparation.

Supplementary Figure S2 | Performance of the pre-trained GNN model on the Novartis and Literature test set is shown. 50 training runs with different training/validation splits were performed and for each training run the best model was selected based on its performance on the validation set (shown here is a single, randomly selected training run). Panel **(A)** shows the performance of the GNN model on the Literature data set. Panel **(B)** shows the performance of the GNN model on the Novartis data set. The solid red line in the scatter plot indicates the ideal behavior of the reference and predicted pK_a values, the dashed lines mark the ±1 pk_a unit interval. Mean absolute error (MAE) and root mean squared error (RMSE) are shown, the values in bracket indicate the 90% confidence interval calculated from 50 repetitions with random training/validation splits. *N* indicates the number of investigated samples.

Supplementary Figure S3 | Performance of the pre-trained and fine-tuned models are shown on the respective validation sets. 50 training runs with different training/validation splits were performed and for each training run the best model was selected based on its performance on the validation set (shown here is a single, randomly selected training run). Panel **(A)** shows the validation set performance of the best GNN model trained on the ChEMBL data set. Panel **(B)** shows the validation set performance starting from the same pre-trained model after fine-tuning on the experimental training set. The solid red line in the scatter plot indicates the ideal behavior of the reference and predicted pK_a values, the dashed lines mark the ±1 pk_a unit interval. Mean absolute error (MAE) and root mean squared error (RMSE) are shown, the values in bracket indicate the 90% confidence interval calculated from 50 repetitions with random training/validation splits. *N* indicates the number of investigated samples.

Supplementary Figure S4 | The accuracy of the fine-tuned GNN model only decreases slightly when molecules from the ChEMBL data set are used for regularization. 50 fine-tuning runs with different training/validation splits were performed, each initialized using the parameters of 50 pre-training runs, and for each training run the best model was selected based on its performance on the validation set. In order to generate a single plot we selected randomly a single fine-

tuning run and generated the scatter plot with the best performing model on the validation set. The solid red line in the scatter plot indicates the ideal behavior of the reference and predicted pK_a values, the dashed lines mark the ±1 pk_a unit interval. Mean absolute error (MAE) and root mean squared error (RMSE) are shown, the values in bracket indicate the 90% confidence interval calculated from 50 repetitions with random training/validation splits. *N* indicates the number of investigated samples.

Supplementary Figure S5 | The performance of the GNN model trained exclusively on the experimental data set is shown. 50 training runs with different training/validation splits were performed. To generate a single plot a randomly selected training run is shown. The solid red line in the scatter plot indicates the ideal behavior of the reference and predicted pK_a values, the dashed lines mark the ±1 pk_a unit interval. Mean absolute error (MAE) and root mean squared error (RMSE) are shown, the values in bracket indicate the 90% confidence interval calculated from 50 repetitions with random training/validation splits. *N* indicates the number of investigated samples.

Supplementary Figure S6 | The pk_a distribution of ChEMBL and experimental data set are shown.

Supplementary Figure S7 | The performance of the fine-tuned GNN model on the ChEMBL data set is shown. In contrast to the results obtained with the fine-tuned models shown in **Supplementary Figure S4** the models shown here did not use regularization. The performance of the GNN model decreased significantly on the ChEMBL data, shifting pK_a values above 12 and below 2. The solid red line in the scatter plot indicates the ideal behavior of the reference and predicted pK_a values, the dashed lines mark the ±1 pk_a unit interval. Mean absolute error (MAE) and root mean squared error (RMSE) are shown, the values in bracket indicate the 90% confidence interval calculated from 50 repetitions with random training/validation splits. *N* indicates the number of investigated samples.

Supplementary Figure S8 | The distribution of molecular weight, the number of hetereoatoms, hydrogen bond acceptors (HBAs) and hydrogen bond donors (HBDs) and distribution of elements per molecule are shown for the ChEMBL data set.

Supplementary Figure S9 | Results are shown for a sequential pK_a prediction using pkasolver-epic for ethylenediaminetetraacetic acid (EDTA). For each protonation state the base-acid pair is shown and the consensus prediction for the pK_a value with the standard deviation is shown. The protonation site is highlighted for each protonation state.

Supplementary Figure S10 | Results are shown for a sequential pK_a prediction using pkasolver-epic for lisdexamfetamine.

Supplementary Figure S11 | Results are shown for a sequential pK_a prediction using pkasolver-epic for cocaine.

Supplementary Figure S12 | Results are shown for a sequential pK_a prediction using pkasolver-epic for tyrosine.

Supplementary Figure S13 | Results are shown for a sequential pK_a prediction using pkasolver-epic for taurine.

Supplementary Figure S14 | Results are shown for a sequential pK_a prediction using pkasolver-epic for aspergillic acid.

Supplementary Figure S15 | Results are shown for a sequential pK_a prediction using pkasolver-epic for ketamine.

Supplementary Figure S16 | Results are shown for a sequential pK_a prediction using pkasolver-epic for levodopa.

Supplementary Figure S17 | Results are shown for a sequential pK_a prediction using pkasolver-epic for furosemide.

Supplementary Figure S18 | Results are shown for a sequential pK_a prediction using pkasolver-light for furosemide.

Supplementary Figure S19 | Results are shown for a sequential pK_a prediction using pkasolver-epic for an aryl guanidine (SMILES: C1CNC(N1)=NC1=CC=CN=C1).

Supplementary Figure S20 | Results are shown for a sequential pK_a prediction using pkasolver-epic for pyridine.

Supplementary Table S1 | List of one-hot-encoding of atom features used for the node feature vector deposited in the node feature matrix *X*.

Supplementary Table S2 | Experimental and calculated pK_a values for the 24 compounds of the SAMPL6 pK_a challenge (Işık et al., 2018). pK_a values were calculated using pkasolver-epic. pK_a values and standard distribution (shown in parenthesis) are rounded to one significant digit. The pK_a value used to match the experimental pK_a value is shown in red.

REFERENCES

Baltruschat, M., and Czodrowski, P. (2020). *Machine Learning Meets pKa*, 9. [version 2; peer review: 2 approved].

Baltruschat, M., and Czodrowski, P. (2020). *Machine Learning Meets pKa*, 9. [version 2; peer review: 2 approved].

Bergazin, T. D., Tielker, N., Zhang, Y., Mao, J., Gunner, M. R., Francisco, K., et al. (2021). Evaluation of Log P, pKa, and Log D Predictions from the SAMPL7 Blind Challenge. *J. Comput. Aided Mol. Des.* 35, 771–802. doi:10.1007/s10822-021-00397-3

Bisong, E. (2019). In: Building Machine Learning and Deep Learning Models on Google Cloud Platform Berkeley, CA: Apress. *Google Colab.*, 59–64. doi:10.1007/978-1-4842-4470-8_7

CRC Handbook (2007). *CRC Handbook of Chemistry and Physics*. 88th Edition. CRC Press, 88. http://www.amazon.com/CRC-Handbook-Chemistry-Physics-88th/dp/0849304881/ref=sr_1_5?ie=UTF8&qid=1302802093&sr=8-5.

Dardonville, C., Caine, B. A., Navarro De La Fuente, M., Martín Herranz, G., Corrales Mariblanca, B., and Popelier, P. L. A. (2017). Substituent Effects on the Basicity (pKa) of Aryl Guanidines and 2-(arylimino)imidazolidines: Correlations of pH-Metric and UV-Metric Values with Predictions from Gas-phase Ab Initio Bond Lengths. *New J. Chem.* 41 (19), 11016–11028. doi:10.1039/c7nj02497e

Davies, M., Nowotka, M., Papadatos, G., Dedman, N., Gaulton, A., Atkinson, F., et al. (2015). ChEMBL Web Services: Streamlining Access to Drug Discovery Data and Utilities. *Nucleic Acids Res.* 43 (W1), W612–W620. doi:10.1093/nar/gkv352

Deng, D., Chen, X., Zhang, R., Lei, Z., Wang, X., and Zhou, F. (2021). XGraphBoost: Extracting Graph Neural Network-Based Features for a Better Prediction of Molecular Properties. *J. Chem. Inf. Model.* 61 (6), 2697–2705. doi:10.1021/acs.jcim.0c01489

Fey, M., and Lenssen, J. E. (2019). *Fast Graph Representation Learning with PyTorch Geometric*. http://arxiv.org/abs/1903.02428, cite arxiv:1903.02428.

Gaulton, A., Bellis, L. J., Bento, A. P., Chambers, J., Davies, M., Hersey, A., et al. (2012). ChEMBL: a Large-Scale Bioactivity Database for Drug Discovery. *Nucleic Acids Res.* 40 (D1), D1100–D1107. doi:10.1093/nar/gkr777

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). *Neural Message Passing for Quantum Chemistry*. doi:10.1002/nme.2457

Greenwood, J. R., Calkins, D., Sullivan, A. P., and Shelley, J. C. (2010). Towards the Comprehensive, Rapid, and Accurate Prediction of the Favorable Tautomeric States of Drug-like Molecules in Aqueous Solution. *J. Comput. Aided Mol. Des.* 24 (6-7), 591–604. doi:10.1007/s10822-010-9349-1

Gunner, M. R., Murakami, T., Rustenburg, A. S., Işık, M., and Chodera, J. D. (2020). Standard State Free Energies, Not pKas, Are Ideal for Describing Small Molecule Protonation and Tautomeric States. *J. Comput. Aided Mol. Des.* 34 (5), 561–573. doi:10.1007/s10822-020-00280-710.1007/s10822-020-00280-7

Ioffe, S., and Szegedy, C. (2015). *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. CoRR. abs/1502.03167. http://arxiv.org/abs/1502.03167.

Işık, M., Levorse, D., Rustenburg, A. S., Ndukwe, I. E., Wang, H., Wang, X., et al. (2018). pKa Measurements for the SAMPL6 Prediction Challenge for a Set of Kinase Inhibitor-like Fragments. *J. Comput. Aided Mol. Des.* 32 (10), 1117–1138. doi:10.1007/s10822-018-0168-0

Işık, M., Rustenburg, A. S., Rizzi, A., Gunner, M. R., Mobley, D. L., and Chodera, J. D. (2021). Overview of the SAMPL6 pKa Challenge: Evaluating Small Molecule Microscopic and Macroscopic pKa Predictions. *J. Comput. Aided Mol. Des.* 35, 131–166. Springer International Publishing. doi:10.1007/s10822-020-00362-6

Jiang, D., Wu, Z., Hsieh, C.-Y., Chen, G., Liao, B., Wang, Z., et al. (2021). Could Graph Neural Networks Learn Better Molecular Representation for Drug Discovery? A Comparison Study of Descriptor-Based and Graph-Based Models. *J. Cheminform* 13 (1), 1–23. doi:10.1186/s13321-020-00479-8

Latscha, H. P., Klein, H. A., and Linti, G. W. (2004). *Analytische Chemie: Chemie-Basiswissen III. Chemie-Basiswissen*. Springer. https://books.google.pt/books?id=xVJ0WtmKMHQC.

Liao, C., and Nicklaus, M. C. (2009). Comparison of Nine Programs Predicting pKa Values of Pharmaceutical Substances. *J. Chem. Inf. Model.* 49 (12), 2801–2812. doi:10.1021/ci900289x

Liao, C., and Nicklaus, M. C. (2009). Comparison of Nine Programs Predicting pKa Values of Pharmaceutical Substances. *J. Chem. Inf. Model.* 49 (12), 2801–2812. Dec. doi:10.1021/ci900289x

Loshchilov, I., and Hutter, F. (2019). *Decoupled Weight Decay Regularization. 7th International Conference on Learning Representations*. ICLR 2019.

Manallack, D. T., Prankerd, R. J., Yuriev, E., Oprea, T. I., and Chalmers, D. K. (2013). The Significance of Acid/base Properties in Drug Discovery. *Chem. Soc. Rev.* 42 (2), 485–496. doi:10.1039/C2CS35348B

Manallack, D. T. (2007). The pKa Distribution of Drugs: Application to Drug Discovery. *Perspect. Med. Chem.*, 1, 1177391X0700100. doi:10.1177/1177391X0700100003

Mansouri, K., Cariello, N. F., Korotcov, A., Tkachenko, V., Grulke, C. M., Sprankle, C. S., et al. (2019). Open-source QSAR Models for pKa Prediction Using Multiple Machine Learning Approaches. *J. Cheminform* 11 (1), 1–20. doi:10.1186/s13321-019-0384-1

McNaught, A. D., and Wilkinson, A. (2014). *Of Pure IU, Chemistry A, of Chemistry (Great Britain) RS. IUPAC Compendium of Chemical Terminology*. International Union of Pure and Applied Chemistry. https://books.google.at/books?id=l2LojwEACAAJ.

Mech, P., Bogunia, M., Nowacki, A., and Makowski, M. (2020). Calculations of pKa Values of Selected Pyridinium and its N-Oxide Ions in Water and Acetonitrile. *J. Phys. Chem. A* 124 (3), 538–551. doi:10.1021/acs.jpca.9b10319

National Center for Biotechnology Information (2022a). *PubChem Compound Summary for CID 3440*. Lisdexamfetamine. https://pubchem.ncbi.nlm.nih.gov/compound/Lisdexamfetamine.

National Center for Biotechnology Information (2022b). *PubChem Compound Summary for CID 3440*. Cocaine. https://pubchem.ncbi.nlm.nih.gov/compound/Cocaine.

National Center for Biotechnology Information (2022c). *PubChem Compound Summary for CID 3440*. Furosemide. https://pubchem.ncbi.nlm.nih.gov/compound/Furosemide.

Pan, X., Wang, H., Li, C., Zhang, J. Z. H., and Ji, C. (2021). MolGpka: A Web Server for Small Molecule pKa Prediction Using a Graph-Convolutional Neural Network. *J. Chem. Inf. Model.* 61 (7), 3159–3165. doi:10.1021/acs.jcim.1c00075

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32*. Editors H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc.), 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Perrin, D. D., Dempsey, B., and Serjeant, E. P. (1981). *pK a Prediction for Organic Acids and Bases*. Dordrecht: Springer Netherlands. doi:10.1007/978-94-009-5883-8

Prasad, S., Huang, J., Zeng, Q., and Brooks, B. R. (2018). An Explicit-Solvent Hybrid QM and MM Approach for Predicting pKa of Small Molecules in SAMPL6 Challenge. *J. Comput. Aided Mol. Des.* 32 (10), 1191–1201. doi:10.1007/s10822-018-0167-1

RDKit, Open-Source Chemiformatics (2022). *RDKit, Open-Source Chemiformatics*. http://www.rdkit.org.

Rogers, D., and Hahn, M. (2010). Extended-connectivity Fingerprints. *J. Chem. Inf. Model.* 50 (5), 742–754. may. doi:10.1021/ci100050t

Ropp, P. J., Kaminsky, J. C., Yablonski, S., and Durrant, J. D. (2019). Dimorphite-DL: An Open-Source Program for Enumerating the Ionization States of Drug-like Small Molecules. *J. Cheminform* 11 (1), 1–8. doi:10.1186/s13321-019-0336-910.1186/s13321-019-0336-9

Rupp, M., Korner, R., and V. Tetko, I. (2011). Predicting the pKa of Small Molecules. *Cchts* 14 (5), 307–327. doi:10.2174/138620711795508403

Selwa, E., Kenney, I. M., Beckstein, O., and Iorga, B. I. (2018). SAMPL6: Calculation of Macroscopic pKa Values from Ab Initio Quantum Mechanical Free Energies. *J. Comput. Aided Mol. Des.* 32 (10), 1203–1216. doi:10.1007/s10822-018-0138-6

Shelley, J. C., Cholleti, A., Frye, L. L., Greenwood, J. R., Timlin, M. R., and Uchimaya, M. (2007). Epik: a Software Program for pK a Prediction and Protonation State Generation for Drug-like Molecules. *J. Comput. Aided Mol. Des.* 21 (12), 681–691. doi:10.1007/s10822-007-9133-z

Tielker, N., Eberlein, L., Güssregen, S., and Kast, S. M. (2018). The SAMPL6 Challenge on Predicting Aqueous pKa Values from EC-RISM Theory. *J. Comput. Aided Mol. Des.* 32 (10), 1151–1163. doi:10.1007/s10822-018-0140-z

Wieder, O., Kohlbacher, S., Kuenemann, M., Garon, A., Ducrot, P., Seidel, T., et al. (2020). A Compact Review of Molecular Property Prediction with Graph Neural Networks. *Drug Discov. Today Technol.* https://www.sciencedirect.com/science/article/pii/S1740674920300305. doi:10.1016/j.ddtec.2020.11.009

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. (2019). *A Comprehensive Survey on Graph Neural Networks*. jan.

Xu, K., Jegelka, S., Hu, W., and Leskovec, J. (2019). How Powerful Are Graph Neural Networks? 7th International Conference on Learning Representations. *ICLR* 2019, 1–17.

Yang, Q., Li, Y., Yang, J. D., Liu, Y., Zhang, L., Luo, S., et al. (2020). Holistic Prediction of the P K a in Diverse Solvents Based on a Machine-Learning Approach. *Angew. Chem. Int. Ed.* 59 (43), 19282–19291. doi:10.1002/anie.202008528

Zeng, Q., Jones, M. R., and Brooks, B. R. (2018). Absolute and Relative pKa Predictions via a DFT Approach Applied to the SAMPL6 Blind Challenge. *J. Comput. Aided Mol. Des.* 32 (10), 1179–1189. doi:10.1007/s10822-018-0150-x

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., et al. (2020). Graph Neural Networks: A Review of Methods and Applications. *AI Open* 1 (September 2020), 57–81. doi:10.1016/j.aiopen.2021.01.001

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