TeachOpenCADD goes Deep Learning: Open-source Teaching Platform Exploring Molecular DL Applications

Michael Backenköhler[1†], Paula Linh Kramer[1,2†], Joschka Groß[2†], Gerrit Großmann[2,3†], Roman Joeres[4,5†], Azat Tagirdzhanov[6,7†], Dominique Sydow[8], Hamza Ibrahim[1], Floriane Odje[1], Verena Wolf[2,3], Andrea Volkamer[1,8*]

***For correspondence:**
volkamer@cs.uni-saarland.de (AV)

†These authors contributed equally to this work

[1] Data Driven Drug Design, Center for Bioinformatics, Saarland University; [2] Modeling and Simulation, Saarland University; [3] German Research Center for Artificial Intelligence (DFKI), Saarbrücken; [4] Drug Bioinformatics, Helmholtz Institute for Pharmaceutical Research Saarland; [5] Drug Bioinformatics, Center for Bioinformatics, Saarland University; [6] Human-Microbe Systems Bioinformatics, Helmholtz Institute for Pharmaceutical Research Saarland; [7] Clinical Bioinformatics, Center for Bioinformatics, Saarland University; [8] Structural Bioinformatics and in silico Toxicology, Institute of Physiology, Universitätsmedizin Berlin

Abstract TeachOpenCADD is a free online platform that offers solutions to common computer-aided drug design (CADD) tasks using Python programming and open-source data and packages. The material is presented through interactive Jupyter notebooks, accommodating users from various backgrounds and programming levels.

Due to the tremendous impact of deep learning (DL) methods in drug design, the TeachOpenCADD platform has been expanded to include an introduction to molecular DL tasks. This edition provides an overview of DL and its application in drug design, highlighting the usage of diverse molecular representations in this field. The platform introduces various neural network architectures, including graph neural networks (GNNs), equivariant graph neural networks (EGNNs), and recurrent neural networks (RNNs). It demonstrates how to use these architectures for developing predictive models for molecular property and activity prediction, exemplified by the Quantum Machine 9 (QM9), ChEMBL, and Kinase Inhibitor BioActivity (KiBA) data sets. The DL edition covers methods for evaluating the performance of neural networks using uncertainty estimation. Furthermore, it introduces an application of GNNs for protein-ligand interaction predictions, incorporating protein structure and ligand information. The TeachOpenCADD platform is continuously updated with new content and is open to contributions, bug reports, and questions from the community through its GitHub repository (github.com/volkamerlab/teachopencadd). It can be used for self-study, classroom instruction, and research applications, accommodating users from beginners to advanced levels.

Introduction

CADD in the deep learning era

The process of discovering new drugs remains both expensive and time-consuming. The approval of a single drug typically takes between 10 and 15 years, with average costs exceeding one billion US dollars (*Scannell et al., 2012*). Computer-aided drug design (CADD) has become a crucial component in the drug development process, offering data-driven guidance in the search for or optimization of innovative compounds. Over the last decade, the immense growth of freely available chemical databases such as ChEMBL (*Gaulton et al., 2017*) and Protein Data Bank (PDB) (*Berman et al., 2000*) has further stimulated the development and application of data-driven approaches such as machine and deep learning (DL). The latter has brought about significant advancements in various fields in recent years, as evidenced by innovations like ChatGPT (*Brown et al., 2020*) and AlphaFold (*Jumper et al., 2021*; *Wu et al., 2022*).

In the realm of drug discovery, DL has demonstrated immense potential (*Volkamer et al., 2023*) due to its ability to process and learn from large and complex data sets (*Lavecchia, 2019*). Here, we propose a learning pipeline based on Jupyter notebooks for chemists, biologists, and computer scientists alike. Previous training material is available introducing cheminformatics and DL but with a different scope and setup (*Menke et al., 2023*), or a stronger computer science background (*Ramsundar et al., 2019*). We start from scratch by explaining the theoretical foundations and show practical examples in Python, solving real-world molecular problems using widely known DL methods. The learning pipeline is based on the well-established TeachOpenCADD framework (*Sydow et al., 2019*, *2022*; *Kimber et al., 2021*).

Molecular deep learning in a nutshell

In the field of drug discovery, DL has been applied to many different problem settings, such as molecular activity, and toxicity prediction (*Coley et al., 2017*; *Unke and Meuwly, 2019*; *Wu et al., 2018*; *Mayr et al., 2016*; *Coley et al., 2017*). Moreover, several docking approaches based on DL have been published reporting promising results (*Corso et al., 2022*; *Ganea et al., 2021*; *Stärk et al., 2022*), as well as generative models for *de novo* drug design (*Jin et al., 2020*; *Hoogeboom et al., 2022*).

A DL network typically consists of multiple, connected layers with non-linear, parameterized transformations. The data is provided to the input layer, which then gets processed through a predefined number of hidden layers, and finally, an output layer generating the prediction (see Figure 1 for some drug design examples) (*Goodfellow et al., 2016*). In the process of training a network, the parameters are adjusted to distill large data sets down to relevant features and patterns associated with the prediction task. Neural networks can be trained for a variety of tasks. They can be used for classification tasks, such as determining whether a molecule is toxic or not, or regression tasks, like predicting binding affinity. Depending on the input data, there are many different classes of neural networks suited for handling molecular data, each having different (dis)advantages. For instance, graph neural networks (GNNs) offer a natural architecture for molecular graphs that has several advantages: They capture both atom and bond information, as well as the connectivity between atoms while being invariant to the nodes' input order. They can handle molecules of varying sizes and complexities and learn both local and global features of molecular structures. Convolutional neural networks (CNNs) are often used for image data, while recurrent neural networks (RNNs) and transformers are designed to handle sequential data (such as text). Some of these architectures will be covered in our tutorials.

TeachOpenCADD: Scope and DL extension

As of September 2022, TeachOpenCADD (*Sydow et al., 2022*) contained 28 talktorials covering diverse topics in the broader area of CADD. Most talktorials are exemplified by compound and structural data available for the EGFR kinase (*Herbst, 2004*). The platform contains talktorials introducing the following topics: (i) Cheminformatics basics, e.g. molecular filtering, clustering, and substructure search, as well as similarity search and machine learning models for activity prediction; (ii) chemical database queries, e.g. ChEMBL (*Gaulton et al., 2017*), PDB (*Berman et al., 2000*), PubChem (*Kim et al., 2022*), and KLIFS queries (*Kanev et al., 2020*); (iii) structural bioinformatics, e.g. binding site detection and comparison, docking, protein-ligand interaction profiling, as well as

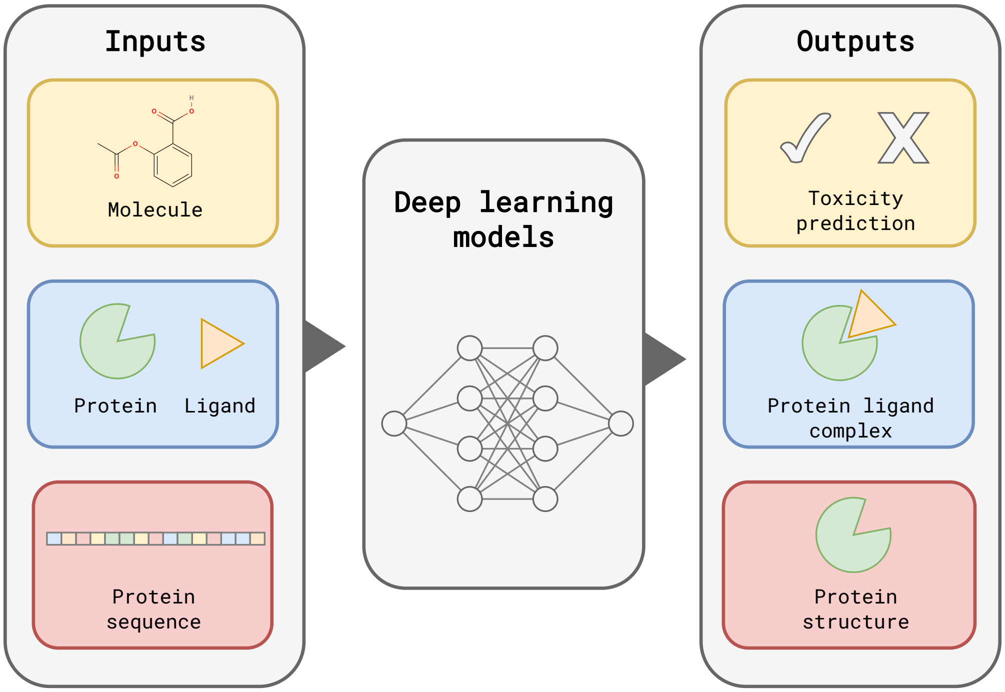

Figure 1. Some applications of DL in the field of drug discovery.

molecular dynamics simulations; and (iv) kinase similarity assessment including different perspectives, e.g. sequence, structure, interaction, and profiling data (***Kimber et al., 2021***).

With the *TeachOpenCADD-DL* edition, we introduce the concepts of DL applied to molecules in six new talktorials. The topics are summarized in Figure 2. As an introduction, we discuss various methods of representing molecules to facilitate their processing by neural networks. For each of the representations, we introduce a class of neural networks: (i) GNNs with molecules represented as a graph, (ii) RNNs where molecules are represented as a SMILES string (***Weininger, 1988***), and (iii) equivariant graph neural networks (EGNNs) which process molecules as point clouds. Each neural network is trained to perform a regression task with the objective of predicting the quantum-mechanical properties of small molecules. In addition to the network architectures, we also cover uncertainty estimations to evaluate the performance of a trained model using molecular fingerprints as input. Finally, we describe an important application of DL for protein-ligand interaction prediction.

Data

In this section, we describe the three molecular data sets used to exemplify the different architectures to solve diverse prediction tasks.

Quantum Machines 9 (QM9) Data Set

QM9 is a public data set that consists of 130k small, organic molecules with up to 9 heavy atoms (***Ramakrishnan et al., 2014***). Each molecule is annotated with various geometric, energetic, electronic, and thermodynamic properties. QM9 is part of MoleculeNet (***Wu et al., 2018***), a widely adopted property prediction benchmark in the molecular machine learning community, e.g., see (***Schütt et al., 2017***; ***Gilmer et al., 2017***; ***Gasteiger et al., 2020***). PyTorch Geometric (***Fey and Lenssen, 2019***) provides pre-implemented classes and methods for working with the QM9 data set in a molecular ML setting.

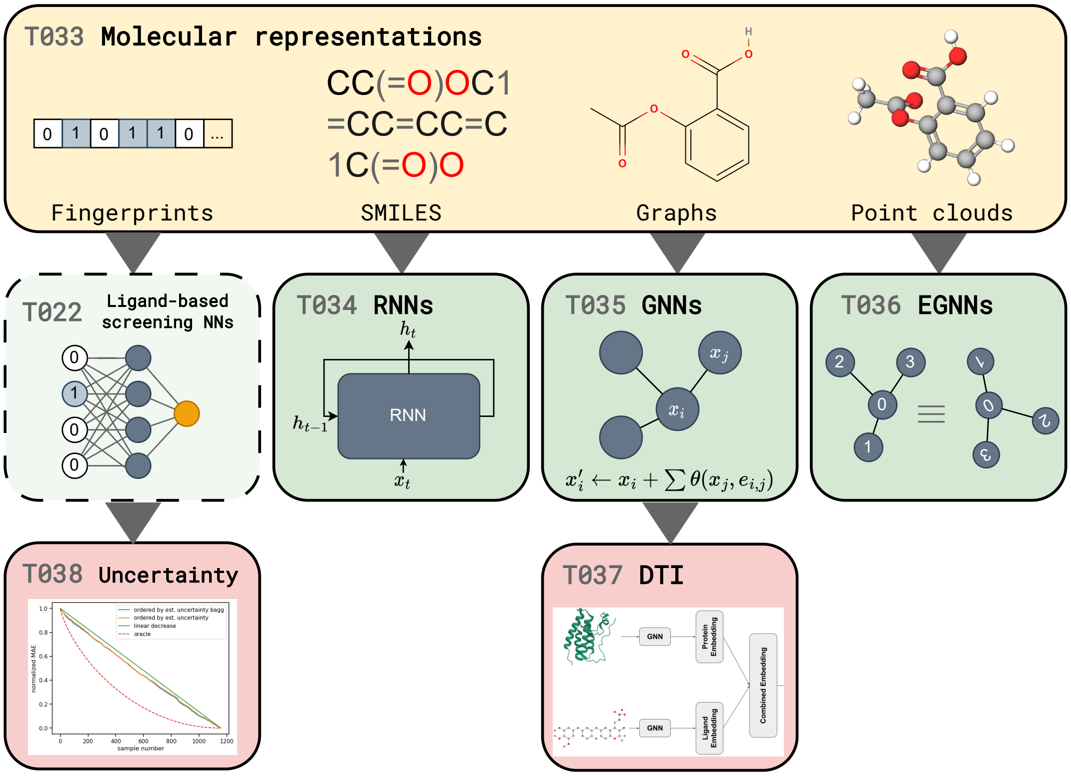

Figure 2. DL-talktorials: The newly contributed talktorials cover molecular representations for machine learning (T033), corresponding deep learning architectures for processing them (T034-36), and more involved topics such as concrete applications (T037) and uncertainty analysis (T038).

ChEMBL EGFR Subset

In the uncertainty estimation talktorial, we make use of activity data available for the EGFR kinase from ChEMBL (*Gaulton et al., 2017*). Protein kinases play a central role in many stages of a cell's life cycle. Dysfunctional signaling of EGFR kinase, e.g., has been associated with cancer progression (*Chen et al., 2016*). The activity data we use is extracted from the public ChEMBL database (*Gaulton et al., 2017*), version 25. Only IC50 data from binding assays (assay_type="B") and exact measurements (standard_type="=") were kept. The data set contains ~3900 compounds with activities from binding assays available as IC50 values.

Kinase Inhibitor BioActivity Data Set

The Kinase Inhibitor BioActivity (KiBA) data set has been assembled from diverse published kinase profiling data sets to provide a large benchmark set for kinase drug-target activity. It is a collection of 467 kinases, 52, 498 ligands, and 246, 088 KiBA scores thereof. The KiBA scores are computed to combine data acquired through different bioactivity experiments and measurements such as IC50, K(i), and K(d) (*Tang et al., 2014*).

For the protein-ligand interaction talktorial (see Section Protein-Ligand Interaction Prediction), we selected a subset of KiBA in order to speed up the training process, reduce memory consumption, and make it trainable on average CPUs in a reasonable time. This is done in two steps: First, all ligands measured against less than 200 kinases are discarded. Second, from the remaining data points, all kinases with data available for less than 10 ligands are removed. Furthermore, a pipeline was provided to scrape the matching PDB structure per kinase starting from UniProt IDs (*Consortium, 2022*) and enforcing some structure quality filters. This resulted in 76 kinases, 275 ligands, and 20, 475 KiBA scores thereof.

Table 1. Summary of the topics covered in the TeachOpenCADD-DL edition.

Topic	Description	Mol. input
Molecular representations	Introduction to molecules and their representations	All below
Recurrent neural networks (RNNs)	RNNs and Gated Recurrent Unit (GRU) for molecular property prediction	SMILES
Graph neural networks (GNNs)	Convolutional and isomorphism GNNs for molecular property prediction	Graph
E(3)-invariant graph neural networks (EGNNs)	EGNNs compared to standard GNNs for molecular property prediction	Point clouds
Uncertainty estimation	Methods for model uncertainty estimation	Fingerprints
Protein-ligand interaction prediction	Applying GNNs to predict protein-ligand interactions	SMILES & PDB

Talktorials

In this section, we describe the six novel topics covered in the TeachOpenCADD-DL edition (see Table 1). Note that all talktorials serve as teaching or starting examples, thus, the architectures were intentionally kept simple and no parameters are tuned to optimize prediction performance.

Molecular Representations

Molecules are intricate, dynamic, three-dimensional (3D) entities composed of atoms, interacting with each forming covalent as well as non-covalent bonds. It is essential to represent molecules in a computer-readable form that corresponds to the information processed through a neural network. In this talktorial, we cover popular molecular representations and discuss their unique implications and (dis-)advantages. This will provide the foundation for subsequent talktorials.

Representing molecules as *graphs* allows for an intuitive and comprehensive representation of their structure. In a graph-based representation, atoms are represented as (labeled) nodes, and bonds are represented as (labeled) edges. However, to represent a graph, we need node ordering. This node ordering, while necessary, is arbitrary, and ideally, a DL predictor should yield the same output regardless of the node order chosen. GNNs address this issue by inherently ensuring this so-called *permutation invariance* by design (***Atz et al., 2021***).

Molecular *fingerprints* are fixed-length, permutation-invariant representations of the molecular graphs. Unlike GNNs, which learn task-specific representations, molecular fingerprints are task-independent. They can be generated based on the occurrences of specific sub-graphs (i.e., molecular fragments or atom environments) (***Rogers and Hahn, 2010***). Generally, it is not feasible to reverse-engineer a fingerprint back to the original molecular graph. Due to their fixed length, fingerprints are compatible with machine learning methods that require a constant input size, such as Multi-Layer Perceptrons (MLPs).

Text-based representations (like SMILES (***Weininger, 1988***), SELFIES(***Krenn et al., 2020***), or InChI (***Grisoni, 2023***)) traverse the molecular graph and convert it into a sequence of characters. However, ambiguity can occur due to the possibility of multiple strings mapping to the same molecule, depending on the order of traversal. To reduce ambiguity, *canonical* SMILES can be used, although what counts as canonicalized SMILES string is not standardized and may differ based on the software package in use. A text-based molecular representation is well-suited for machine learning (ML) models capable of handling sequences with varying lengths. Specifically, they have been successfully used as input to language models (***Wang et al., 2019***; ***Chithrananda et al., 2020***).

Point cloud representations annotate atoms with their 3D coordinates, corresponding to a single conformation. A molecular conformation (conformer) is a specific spatial arrangement of atoms within a molecule, reflecting a single energetically favorable configuration of its 3D structure. Like in GNNs, this necessitates a special type of invariance for DL methods that take point clouds as input. Our specific goal is to attain invariance to Euclidean space transformations (e.g., the output of the neural network model should remain unchanged when the entire molecule is rotated). Point cloud representations are especially advantageous, as they encompass more comprehensive information. In particular, they capture the relative atomic positions, which reflect the collective effect of all forces acting within a molecule, beyond just covalent bonds (***Atz et al., 2021***).

In our talktorial, we discuss the different molecular representations in more detail and demonstrate how to generate and utilize them in Python.

Recurrent Neural Networks

In recent years, DL-based natural language processing (NLP) has made significant progress, with RNNs and transformers among the most successful models. These models proved to be good at capturing text semantics and, when applied to molecular data, can capture the molecular structure in its textual representation. As a result, NLP models have become a powerful tool in numerous drug discovery applications, including *de novo* drug design (***Gupta et al., 2018***), virtual screening (***Karimi et al., 2019***), and molecular property prediction (***Bjerrum, 2017***).

RNNs were originally developed to handle sequential data (***Elman, 1990***). These models can process variable-length sequences of inputs and propagate the information through the sequence using their internal state. In this talktorial, we focus on applying RNNs to SMILES strings. We briefly cover the usual preprocessing steps that transform SMILES into numerical form and discuss two RNN architectures in detail, starting with the Elman network, also known as a simple RNN (***Elman, 1990***). This architecture is suitable for demonstrating the basic principles of RNNs, but in practice, it struggles with learning long-term dependencies in the data. This problem is addressed in the more advanced Gated Recurrent Unit (GRU) (***Cho et al., 2014***) architecture. GRU selectively updates its internal state using gating mechanisms, allowing the model to learn to identify and retain the most important information while discarding irrelevant information.

We implement RNN- and GRU-based regression models and apply them to molecular property prediction using the QM9 data set. As a regression task, we have chosen to predict the dipole moment μ, which is a measure of a molecule's polarity. Our results show that the GRU model learns faster and achieves better performance than the simple RNN model.

Graph Neural Networks

The most natural representation for molecules are graphs spanned by their atoms and bonds. Thus, one intuitive way to apply DL techniques to molecular data is using GNNs. GNNs are widely used in drug discovery, for example for property prediction (***Wu et al., 2018***; ***Wieder et al., 2020***) and *de novo* drug design (***Xia et al., 2019***; ***Tong et al., 2021***).

Instead of the fully connected layers commonly used in standard neural networks, GNNs have message-passing layers, that collect information about the neighboring nodes in the graph (***Kipf and Welling, 2016***). For each node in the graph, all the information from the neighbors is gathered and aggregated using an aggregation function such as the sum. One important property of a GNN is the permutation invariance. This means that changing the arbitrary order of nodes in the graph should not have an effect on the outcome. On the other hand, GNNs should ideally also be able to distinguish between similar graphs.

In our talktorial, we present two commonly used GNN architectures in more detail: one of the simplest GNNs, namely the graph convolutional neural network (GCN (***Kipf and Welling, 2016***)), and a more powerful GNN called the graph isomorphism network (GIN (***Xu et al., 2018***)). GINs are better at distinguishing similar, non-identical graphs compared to GCNs, which often leads to better performance. We demonstrate how to implement GNNs and how to train them using the QM9 data set (see Section Quantum Machines 9 (QM9) Data Set) to predict one quantum-mechanic property of small molecules. We predict the same molecular property as in the previous talktorial (see Section Recurrent Neural Networks).

E(3)-invariant Graph Neural Network

Reasoning about molecular properties is often easier when 3D information (e.g. in the form of conformations) is available. Some tasks may also strictly require the use of molecular representations that include 3D information. Examples of this are binding pose predictions of ligand-protein complexes (***Corso et al., 2022***) or force predictions for molecular dynamics simulations (***Doerr et al., 2021***). It is widely accepted that GNNs which process molecules based on their point cloud representation (see Section Molecular Representations) should satisfy certain invariance or equivariance properties with respect to global *Euclidean transformations* such as translations or rotations.

The Euclidean group that corresponds to these transformations in three dimensions is denoted by $E(3)$. $E(3)$-invariance implies that the output of a GNN is unaffected by rotations or translations of its input point cloud. For example, when predicting binding affinity based on the structure of a ligand-protein complex, this prediction should remain unchanged if the entire complex is translated or rotated. $E(3)$-equivariance implies that rotating or translating the GNN's input should induce an equivalent transformation of its output. For example, when predicting the binding pose of a ligand-based on a given protein structure, rotating the latter should give rise to an equivalently rotated pose prediction.

This talktorial discusses these concepts in more detail in the theory part. It demonstrates how to implement $E(3)$-invariant graph neural networks for property prediction based on the point cloud representation of the molecules included in the QM9 data set. The practical part concludes by training and evaluating such a model in comparison to a plain GNN. The application shows that the theoretical advantages mentioned above also lead to better results in practice.

Uncertainty Estimation

Often researchers pay a lot of attention to the overall accuracy of their predictions. However, when implementing any predictive method in practice, it is equally important to understand the level of confidence in a given estimation. The uncertainty can stem from both the experiments themselves (epistemic) and/or the predictive model (aleatoric). In the former case, the uncertainty of the model arises typically due to a lack of training data while the latter case refers to inherent randomness such as measurement noise (***Der Kiureghian and Ditlevsen, 2009***). Thus, it would be beneficial to obtain not only a point estimate of the prediction but also an indication of how certain we can be about that estimate. The certainty is often modeled by replacing the point estimate with a distributional estimate (***Gawlikowski et al., 2021***). For example, instead of a number as a prediction of an IC50 value, one obtains a distribution of the predicted values.

In this talktorial, we showcase uncertainty estimation on a practical example. We start our demonstration by creating a simple model ensemble. This means we train the same model multiple times with a varying random seed. At test time, we evaluate all models and use the mean as a predictor. The variance across the ensemble serves as a variance estimate for that prediction. We discuss the calibration of this estimator, which – as is typical – under-estimates the actual variance.

In the second step, we improve our ensemble by not only varying the random seed during training but also the data itself. This variation is achieved by bootstrapping the training data. This helps to more accurately estimate uncertainty.

Finally, we showcase test time data augmentation as an alternative to the modification of our predictive model. In this technique, we create variants for each query point in our test set. The variants are created by applying random flips to a fingerprint datum. This way, we get an ensemble of predictions out of a single model, without the need to modify the model itself.

Protein-Ligand Interaction Prediction

Protein-ligand interaction prediction is an important field in drug development, e.g. to screen for novel drug candidates. Classical methods to predict drug-target interactions are based on docking (***de Azevedo Jr et al., 2003***; ***M Bernhardt Levin et al., 2017***), biological networks (***AY et al., 2007***; ***Chen et al., 2012***), and many more (***Zhao et al., 2022***). More recently, models use DL encoders such as MLPs, i.e. CNNs and GNNs, to compute latent space representations, also called embeddings, of biochemical molecules (***Öztürk et al., 2018***; ***Nguyen et al., 2021***). While in classical docking methods, the complex structure is generated and then scored, in these works the two interaction partners are treated separately. The embeddings are combined for each pair of potentially interacting molecules, usually concatenated, and then fed into an MLP to predict the output variable. The variable can either be a proxy value for binding affinity or a classification value separating binding and non-binding pairs of protein and ligand.

The goal of this talktorial is to introduce the reader to the field of protein-ligand interaction prediction using GNNs for proteins and ligands independently. In contrast to previous works in which the protein was encoded as sequence and a CNN was used for the embedding, (***Öztürk et al., 2018***; ***Nguyen et al., 2021***), GNNs are used for both, proteins and ligands. Ligands are represented as graphs constructed from the SMILES string. Representing proteins is more complex and done using Residue Interaction Networks (RINs) (***Doncheva et al., 2011***). These are graphs where nodes represent amino acids and edges represent covalent and non-covalent interactions between amino acids. To compute those, RINminer (***Keller et al., 2020***) can be used or a distance threshold between amino acids in the three-dimensional space as a surrogate of such. The talktorial exemplifies this task of predicting interactions between proteins and ligands using the KiBA subset (see Section Kinase Inhibitor BioActivity Data Set) and shows that predicting interaction on the KiBA dataset is possible with little effort and simple GNNs.

Prerequisites and technical information

Target audience

The talktorials were developed to support researchers who are interested in the topics and are new to the field. The covered scope is intended to further bridge the fields of CADD and DL. The talktorials are recommended for biologists, medicinal chemists as well as computer scientists; and should enable the user to apply the techniques in their own work. Since the talktorials form an extension to the TeachOpenCADD platform, they serve as teaching material in the field of structural bio- and cheminformatics.

Background knowledge

The tutorials are meant to be an introduction to DL and its application to the field of drug discovery. In each talktorial, we first present the theoretical background for the biological and chemical basics as well as the computer science fundamentals. Secondly, we provide thoroughly documented Python code to illustrate the application of DL. However, some proficiency in Python and Jupyter would be helpful.

Software requirements

All talktorials are written in Python and make use of well-known open-source packages such as Pandas (***McKinney, 2011***), NumPy (***Harris et al., 2020***), Matplotlib (***Hunter, 2007***), SciPy (***Virtanen et al., 2020***), RDKit (***Landrum, 2006***). The novel DL talktorials make heavy use of PyTorch (***Paszke et al., 2019***) and PyTorch Geometric (***Fey and Lenssen, 2019***). The user only needs to install the teachopencadd conda-forge package, which will install all relevant packages and save a copy of all TeachOpenCADD notebooks on the user's local machine. A read-only mode of the talktorials is accessible via the TeachOpenCADD website at projects.volkamerlab.org/teachopencadd/.

Structure of the talktorials

The talktorials serve a teaching purpose and are structured as follows: Each Jupyter notebook is split into two parts. We first explain the underlying theory of each topic. We explain the problem setting, give relevant references, and list possible applications. The second part is focusing on the actual implementation in Python. We explain and document each step in the code. We want to make it easy to follow and give the user the chance to extend this to different applications in the field.

Conclusion

This study provides an insightful introduction to DL important for and applied to molecular prediction tasks. We presented six talktorials covering topics such as commonly used representations of molecules and proteins, graph and recurrent neural networks, uncertainty measures, and protein-ligand interaction predictions. Through these talktorials, users can gain a better understanding of DL and its potential applications in drug discovery. We believe that these methods can be used as a starting point and can be adapted for different molecular data sets and more complex questions.

Author Contributions

MB, PK, JG, GG, AT, and RJ implemented the new notebooks. MB, HI, and DS integrated the new material and maintained the repository. All authors reviewed individual talktorials. AV conceptualized the study. VW and AV supervised the project. All authors contributed to writing and reviewing the manuscript.

Acknowledgements

We thank all former and future supporters and users of the TeachOpenCADD platform. And we are grateful for the fruitful discussions around the DL topics with all our colleagues from the NextAID project (see nextaid.cs.uni-saarland.de/) at Saarland University.

Funding Information

We are thankful for funding from Saarland University for the NextAID project through which the positions of MB, PK, JG, GG, and AT are supported. RJ was partially funded by the HelmholtzAI project XAI-Graph.

References

Atz K, Grisoni F, Schneider G. Geometric deep learning on molecular representations. Nature Machine Intelligence. 2021; 3(12):1023–1032. https://doi.org/10.1038/s42256-021-00418-8.

AY M, Goh KI, Cusick ME, Barabasi AL, Vidal M, et al. Drug–target network. Nature biotechnology. 2007; 25(10):1119–1127. https://doi.org/10.1038/nbt1338.

de Azevedo Jr WF, dos Santos GC, dos Santos DM, Olivieri JR, Canduri F, Silva RG, Basso LA, Renard G, da Fonseca IO, Mendes MA, et al. Docking and small angle X-ray scattering studies of purine nucleoside phosphorylase. Biochemical and Biophysical Research Communications. 2003; 309(4):923–928. https://doi.org/10.1016/j.bbrc.2003.08.093.

Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. The protein data bank. Nucleic acids research. 2000; 28(1):235–242. https://doi.org/10.1093/nar/28.1.235.

Bjerrum EJ. SMILES enumeration as data augmentation for neural network modeling of molecules. arXiv preprint arXiv:170307076. 2017; https://doi.org/10.48550/arXiv.1703.07076.

Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, et al. Language models are few-shot learners. Advances in neural information processing systems. 2020; 33:1877–1901. https://doi.org/10.48550/arXiv.2005.14165.

Chen J, Zeng F, Forrester SJ, Eguchi S, Zhang MZ, Harris RC. Expression and Function of the Epidermal Growth Factor Receptor in Physiology and Disease. Physiological Reviews. 2016; 96:1025–1069. doi: 10.1152/physrev.00030.2015.

Chen X, Liu MX, Yan GY. Drug–target interaction prediction by random walk on the heterogeneous network. Molecular BioSystems. 2012; 8(7):1970–1978. https://doi.org/10.1039/C2MB00002D.

Chithrananda S, Grand G, Ramsundar B. Chemberta: Large-scale self-supervised pretraining for molecular property prediction. arXiv preprint arXiv:201009885. 2020; doi: 10.48550/arXiv.2010.09885.

Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:14061078. 2014; https://doi.org/10.48550/arXiv.1406.1078.

Coley CW, Barzilay R, Green WH, Jaakkola TS, Jensen KF. Convolutional embedding of attributed molecular graphs for physical property prediction. Journal of chemical information and modeling. 2017; 57(8):1757–1772. https://doi.org/10.1021/acs.jcim.6b00601.

Consortium TU. UniProt: the Universal Protein Knowledgebase in 2023. Nucleic Acids Research. 2022 11; 51(D1):D523–D531. https://doi.org/10.1093/nar/gkac1052, doi: 10.1093/nar/gkac1052.

Corso G, Stärk H, Jing B, Barzilay R, Jaakkola T. Diffdock: Diffusion steps, twists, and turns for molecular docking. arXiv preprint arXiv:221001776. 2022; https://doi.org/10.48550/arXiv.2210.01776.

Der Kiureghian A, Ditlevsen O. Aleatory or epistemic? Does it matter? Structural safety. 2009; 31(2):105–112.

Doerr S, Majewski M, Pérez A, Kramer A, Clementi C, Noe F, Giorgino T, De Fabritiis G. Torchmd: A deep learning framework for molecular simulations. Journal of chemical theory and computation. 2021; 17(4):2355–2363. https://doi.org/10.1021/acs.jctc.0c01343.

Doncheva NT, Klein K, Domingues FS, Albrecht M. Analyzing and visualizing residue networks of protein structures. Trends in biochemical sciences. 2011; 36(4):179–182. https://doi.org/10.1016/j.tibs.2011.01.002.

Elman JL. Finding structure in time. Cognitive science. 1990; 14(2):179–211. https://doi.org/10.1016/0364-0213(90)90002-E.

Fey M, Lenssen JE. Fast Graph Representation Learning with PyTorch Geometric. CoRR. 2019; abs/1903.02428. https://doi.org/10.48550/arXiv.1903.02428.

Ganea OE, Huang X, Bunne C, Bian Y, Barzilay R, Jaakkola T, Krause A. Independent se (3)-equivariant models for end-to-end rigid protein docking. arXiv preprint arXiv:211107786. 2021; https://doi.org/10.48550/arXiv.2111.07786.

Gasteiger J, Groß J, Günnemann S. Directional message passing for molecular graphs. arXiv preprint arXiv:200303123. 2020; https://doi.org/10.48550/arXiv.2003.03123.

Gaulton A, Hersey A, Nowotka M, Bento AP, Chambers J, Mendez D, Mutowo P, Atkinson F, Bellis LJ, Cibrián-Uhalte E, et al. The ChEMBL database in 2017. Nucleic acids research. 2017; 45(D1):D945–D954. https://doi.org/10.1093/nar/gkw1074.

Gawlikowski J, Tassi CRN, Ali M, Lee J, Humt M, Feng J, Kruspe A, Triebel R, Jung P, Roscher R, et al. A survey of uncertainty in deep neural networks. arXiv preprint arXiv:210703342. 2021; https://doi.org/10.48550/arXiv.2107.03342.

Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE. Neural message passing for quantum chemistry. In: *International conference on machine learning* PMLR; 2017. p. 1263–1272. https://doi.org/10.48550/arXiv.1704.01212.

Goodfellow I, Bengio Y, Courville A. Deep Learning. MIT Press; 2016. http://www.deeplearningbook.org.

Grisoni F. Chemical language models for de novo drug design: Challenges and opportunities. Current Opinion in Structural Biology. 2023; 79:102527. https://doi.org/10.1016/j.sbi.2023.102527.

Gupta A, Müller AT, Huisman BJ, Fuchs JA, Schneider P, Schneider G. Generative recurrent networks for de novo drug design. Molecular informatics. 2018; 37(1-2):1700111. https://doi.org/10.1002/minf.201700111.

Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, Wieser E, Taylor J, Berg S, Smith NJ, Kern R, Picus M, Hoyer S, van Kerkwijk MH, Brett M, Haldane A, del Río JF, Wiebe M, Peterson P, Gérard-Marchant P, et al. Array programming with NumPy. Nature. 2020; 585(7825):357–362. https://doi.org/10.1038/s41586-020-2649-2, doi: 10.1038/s41586-020-2649-2.

Herbst RS. Review of epidermal growth factor receptor biology. International Journal of Radiation Oncology*Biology*Physics. 2004; 59:S21–S26. doi: https://doi.org/10.1016/j.ijrobp.2003.11.041.

Hoogeboom E, Satorras VG, Vignac C, Welling M. Equivariant diffusion for molecule generation in 3D. In: *International Conference on Machine Learning* PMLR; 2022. p. 8867–8887. https://doi.org/10.48550/arXiv.2203.17003.

Hunter JD. Matplotlib: A 2D graphics environment. Computing in Science & Engineering. 2007; 9(3):90–95. doi: 10.1109/MCSE.2007.55.

Jin W, Barzilay R, Jaakkola T. Hierarchical generation of molecular graphs using structural motifs. In: *International conference on machine learning* PMLR; 2020. p. 4839–4848. https://doi.org/10.48550/arXiv.2002.03230.

Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, Tunyasuvunakool K, Bates R, Žídek A, Potapenko A, et al. Highly accurate protein structure prediction with AlphaFold. Nature. 2021; 596(7873):583–589. https://doi.org/10.1038/s41586-021-03819-2.

Kanev GK, de Graaf C, Westerman BA, de Esch IJP, Kooistra AJ. KLIFS: an overhaul after the first 5 years of supporting kinase research. Nucleic Acids Research. 2020 10; 49(D1):D562–D569. https://doi.org/10.1093/nar/gkaa895, doi: 10.1093/nar/gkaa895.

Karimi M, Wu D, Wang Z, Shen Y. DeepAffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks. Bioinformatics. 2019; 35(18):3329–3338. https://doi.org/10.1093/bioinformatics/btz111.

Keller S, Miettinen P, Kalinina OV. Frequent subgraph mining for biologically meaningful structural motifs. BioRxiv. 2020; p. 2020–05. https://doi.org/10.1101/2020.05.14.095695.

Kim S, Chen J, Cheng T, Gindulyte A, He J, He S, Li Q, Shoemaker BA, Thiessen PA, Yu B, Zaslavsky L, Zhang J, Bolton EE. PubChem 2023 update. Nucleic Acids Research. 2022 10; 51(D1):D1373–D1380. https://doi.org/10.1093/nar/gkac956, doi: 10.1093/nar/gkac956.

Kimber TB, Sydow D, Volkamer A. Kinase Similarity Assessment Pipeline for Off-Target Prediction [Article v1.0]. Living Journal of Computational Molecular Science. 2021; 3(1):1599–1599. https://livecomsjournal.org/index.php/livecoms/article/view/v3i1e1599, doi: 10.33011/livecoms.3.1.1599.

Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:160902907. 2016; https://doi.org/10.48550/arXiv.1609.02907.

Krenn M, Häse F, Nigam A, Friederich P, Aspuru-Guzik A. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. Machine Learning: Science and Technology. 2020; 1(4):045024. doi: 10.1088/2632-2153/aba947.

Landrum G, RDKit: Open-source cheminformatics; 2006. https://www.rdkit.org, accessed: 2023-05-22.

Lavecchia A. Deep learning in drug discovery: opportunities, challenges and future prospects. Drug Discovery Today. 2019 Oct; 24(10):2017–2032. https://www.sciencedirect.com/science/article/pii/S135964461930282X, doi: 10.1016/j.drudis.2019.07.006.

M Bernhardt Levin N, Oliveira Pintro V, Boff de Avila M, Boldrini de Mattos B, Filgueira DAJ, et al. Understanding the structural basis for inhibition of cyclin-dependent kinases. New pieces in the molecular puzzle. Current drug targets. 2017; 18(9):1104–1111. doi: 10.2174/1389450118666161116130155.

Mayr A, Klambauer G, Unterthiner T, Hochreiter S. DeepTox: toxicity prediction using deep learning. Frontiers in Environmental Science. 2016; 3:80. https://doi.org/10.3389/fenvs.2015.00080.

McKinney W. pandas: a foundational Python library for data analysis and statistics. Python for High Performance and Scientific Computing. 2011; 14. https://pandas.pydata.org/.

Menke J, Homberg S, Koch O. Introduction to artificial intelligence and deep learning using interactive electronic programming notebooks. Archiv der Pharmazie. 2023; p. e2200628. https://doi.org/10.1002/ardp.202200628.

Nguyen T, Le H, Quinn TP, Nguyen T, Le TD, Venkatesh S. GraphDTA: predicting drug–target binding affinity with graph neural networks. Bioinformatics. 2021; 37(8):1140–1147. https://doi.org/10.1093/bioinformatics/btaa921.

Öztürk H, Özgür A, Ozkirimli E. DeepDTA: deep drug–target binding affinity prediction. Bioinformatics. 2018; 34(17):i821–i829. https://doi.org/10.1093/bioinformatics/bty593.

Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, et al. Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems. 2019; 32. https://doi.org/10.48550/arXiv.1912.01703.

Ramakrishnan R, Dral PO, Rupp M, Von Lilienfeld OA. Quantum chemistry structures and properties of 134 kilo molecules. Scientific data. 2014; 1(1):1–7. https://doi.org/10.1038/sdata.2014.22.

Ramsundar B, Eastman P, Walters P, Pande V. Deep Learning for the Life Sciences. O'Reilly Media; 2019.

Rogers D, Hahn M. Extended-Connectivity Fingerprints. Journal of Chemical Information and Modeling. 2010; 50(5):742–754. https://doi.org/10.1021/ci100050t, doi: 10.1021/ci100050t.

Scannell JW, Blanckley A, Boldon H, Warrington B. Diagnosing the decline in pharmaceutical R&D efficiency. Nature reviews Drug discovery. 2012; 11(3):191–200. https://doi.org/10.1038/nrd3681.

Schütt K, Kindermans PJ, Sauceda Felix HE, Chmiela S, Tkatchenko A, Müller KR. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. Advances in neural information processing systems. 2017; 30. https://doi.org/10.48550/arXiv.1706.08566.

Stärk H, Ganea O, Pattanaik L, Barzilay R, Jaakkola T. Equibind: Geometric deep learning for drug binding structure prediction. In: *International Conference on Machine Learning* PMLR; 2022. p. 20503–20521. https://doi.org/10.48550/arXiv.2202.05146.

Sydow D, Morger A, Driller M, Volkamer A. TeachOpenCADD: a teaching platform for computer-aided drug design using open source packages and data. Journal of Cheminformatics. 2019; 11(1):29. https://doi.org/10.1186/s13321-019-0351-x, doi: 10.1186/s13321-019-0351-x.

Sydow D, Rodríguez-Guerra J, Kimber T, Schaller D, Taylor C, Chen Y, Leja M, Misra S, Wichmann M, Ariamajd A, Volkamer A. TeachOpenCADD 2022: open source and FAIR Python pipelines to assist in structural bioinformatics and cheminformatics research. Nucleic Acids Research. 2022 Jul; 50:W753–W760. https://doi.org/10.1093/nar/gkac267, doi: 10.1093/nar/gkac267.

Tang J, Szwajda A, Shakyawar S, Xu T, Hintsanen P, Wennerberg K, Aittokallio T. Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis. Journal of Chemical Information and Modeling. 2014; 54(3):735–743. https://doi.org/10.1021/ci400709d.

Tong X, Liu X, Tan X, Li X, Jiang J, Xiong Z, Xu T, Jiang H, Qiao N, Zheng M. Generative models for De Novo drug design. Journal of Medicinal Chemistry. 2021; 64(19):14011–14027. https://doi.org/10.1021/acs.jmedchem.1c00927.

Unke OT, Meuwly M. PhysNet: A neural network for predicting energies, forces, dipole moments, and partial charges. Journal of chemical theory and computation. 2019; 15(6):3678–3693. https://doi.org/10.1021/acs.jctc.9b00181.

Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Millman KJ, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey C, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature Methods. 2020; 17(3):261–272. doi: 10.1038/s41592-019-0686-2.

Volkamer A, Riniker S, Nittinger E, Lanini J, Grisoni F, Evertsson E, Rodríguez-Pérez R, Schneider N. Machine Learning for Small Molecule Drug Discovery in Academia and Industry. Artificial Intelligence in the Life Sciences. 2023; p. 100056. https://doi.org/10.1016/j.ailsci.2022.100056.

Wang S, Guo Y, Wang Y, Sun H, Huang J. SMILES-BERT: large scale unsupervised pre-training for molecular property prediction. In: *Proceedings of the 10th ACM international conference on bioinformatics, computational biology and health informatics*; 2019. p. 429–436. doi: 10.1145/3307339.3342186.

Weininger D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. Journal of Chemical Information and Computer Sciences. 1988; 28(1):31–36. https://doi.org/10.1021/ci00057a005, doi: 10.1021/ci00057a005.

Wieder O, Kohlbacher S, Kuenemann M, Garon A, Ducrot P, Seidel T, Langer T. A compact review of molecular property prediction with graph neural networks. Drug Discovery Today: Technologies. 2020; 37:1–12. https://doi.org/10.1016/j.ddtec.2020.11.009.

Wu R, Ding F, Wang R, Shen R, Zhang X, Luo S, Su C, Wu Z, Xie Q, Berger B, et al. High-resolution de novo structure prediction from primary sequence. BioRxiv. 2022; p. 2022–07. https://doi.org/10.1101/2022.07.21.500999.

Wu Z, Ramsundar B, Feinberg EN, Gomes J, Geniesse C, Pappu AS, Leswing K, Pande V. MoleculeNet: a benchmark for molecular machine learning. Chemical science. 2018; 9(2):513–530. https://doi.org/10.1039/C7SC02664A.

Xia X, Hu J, Wang Y, Zhang L, Liu Z. Graph-based generative models for de Novo drug design. Drug Discovery Today: Technologies. 2019; 32:45–53. https://doi.org/10.1016/j.ddtec.2020.11.004.

Xu K, Hu W, Leskovec J, Jegelka S. How powerful are graph neural networks? arXiv preprint arXiv:181000826. 2018; https://doi.org/10.48550/arXiv.1810.00826.

Zhao L, Zhu Y, Wang J, Wen N, Wang C, Cheng L. A brief review of protein-ligand interaction prediction. Computational and Structural Biotechnology Journal. 2022; https://doi.org/10.1016/j.csbj.2022.06.004.