# Predicting drug–target binding affinity with graph neural networks

Thin Nguyen[1*], Hang Le[2], Thomas P. Quinn[1], Thuc Le[3], Svetha Venkatesh[1]

[1]Deakin University, [2]Nha Trang University, [3]University of South Australia
[*] Correspondence: thin.nguyen@deakin.edu.au

**Abstract**

While the development of new drugs is costly, time consuming, and often accompanied with safety issues, drug repurposing, where old drugs with established safety are used for medical conditions other than originally developed, is an attractive alternative. Then, how the old drugs work on new targets becomes a crucial part of drug repurposing and gains much of interest. Several statistical and machine learning models have been proposed to estimate drug–target binding affinity and deep learning approaches have been shown to be among state-of-the-art methods. However, drugs and targets in these models have been commonly represented in 1D strings, regardless the fact that molecules are by nature formed by the chemical bonding of atoms. In this work, we propose **GraphDTA** to capture the structural information of drugs, possibly enhancing the predictive power of the affinity. In particular, unlike competing methods, drugs are represented as graphs and graph convolutional networks are used to learn drug–target binding affinity. We trial our method on two benchmark datasets of drug–target binding affinity and compare the performance with state-of-the-art models in the research field. The results show that our proposed method can not only predict the affinity better than non-deep learning models, but also outperform competing deep learning approaches. This demonstrates the practical advantages of graph-based representation for molecules in providing accurate prediction of drug–target binding affinity. The application may also include any recommendation systems where either or both of the user- and product-like sides can be represented in graphs.

**Availability and implementation** The proposed models are implemented in Python. Related data, pre-trained models, and source code are publicly available at
https://github.com/thinng/GraphDTA. All scripts and data needed to reproduce the post-hoc analysis are available from https://doi.org/10.5281/zenodo.3603523.

## 1 Background

It costs about 2.6 billion US dollars to develop a new drug [1], and can take up to 10–17 years for FDA approval [2, 3]. Identifying new uses for already approved drugs avoids the expensive and lengthy process of drug development [2, 4]. In order to repurpose drugs effectively, it is useful to know which proteins are targeted by which drugs. High-throughput screening experiments are used to examine the affinity of a drug toward its targets; however, these experiments are costly and time-consuming [5, 6], and an exhaustive search is infeasible because there are millions of drug-like compounds [7] and hundreds of potential targets [8, 9]. As such, there is a strong motivation to build computational models that can estimate the interaction strength of new drug–target pairs based on previous drug–target experiments.

Several computational approaches have been proposed for drug–target affinity (**DTA**) prediction [10, 11, 12]. One approach uses collaborative filtering. For example, the **SimBoost** model uses the affinity similarities among drugs and among targets to build new features. These features are then used as input in a gradient boosting machine to predict the binding affinity for unknown drug–target pairs [13]. Alternatively, the similarities could come from others sources (rather than the training data affinities). For example, kernel-based methods use kernels built from the molecular descriptors of the drugs and targets within a regularized least squares regression (RLS) framework [14, 15]. To speed up model training, the **KronRLS** model computes a pairwise kernel $K$ from the Kronecker product of the drug-by-drug and protein-by-protein kernels [14, 15] (for which any similarity measure can be used).

Another approach uses neural networks trained on 1D representations of the drug and protein sequences. For example, the **DeepDTA** model uses 1D representations and layers of 1D convolutions (with pooling) to capture predictive patterns within the data [16]. The final convolution layers are then concatenated, passed through a number of hidden layers, and regressed with the drug–target affinity scores. The **WideDTA** model is an extension of **DeepDTA** in which the sequences of the drugs and proteins are first summarized as higher-order features [17]. For example, the drugs are represented by the most common sub-structures (the Ligand Maximum Common Substructures (LMCS) [18]), while the proteins are represented by the most conserved sub-sequences (the Protein Domain profiles or Motifs (PDM) from PROSITE [19]). While **WideDTA** [17] and **DeepDTA** [16] learn a latent feature vector for each protein, the **PADME** model [20] uses fixed-rule descriptors to represent proteins, and performs similarly to **DeepDTA** [16].

The deep learning models are among the best performers in DTA prediction [16]. However, these models represent the drugs as strings, which are not a natural way to represent molecules. When using strings, the structural information of the molecule is lost, which could impair the predictive power of a model as well as the functional relevance of the learned latent space. Already, graph convolutional networks have been used in computational drug discovery [21], including interaction prediction, synthesis prediction, *de novo* molecular design, and quantitative structure prediction [22, 23, 24]. However, neither these nor other graph neural networks have been used for DTA prediction.

In this article, we propose **GraphDTA**, a new neural network architecture capable of directly modelling drugs as molecular graphs, and show that this approach outperforms state-of-the-art deep learning models on two hallmark drug–target affinity prediction benchmarks. In order to better understand how the graphical model works, we perform an analysis of the model's latent space and find that our graph neural network automatically assigned importance to well-defined chemical features, such as the number of aliphatic OH groups, without any prior knowledge. We also examine the model's performance and find that a handful of drugs contribute disproportionately to the total prediction error, and that these drugs are inliers (i.e., not outliers) in an ordination of the latent space. Taken together, our results suggest that graph neural networks are highly accurate, abstract meaningful concepts, and yet fail in predictable ways. We conclude with a discussion about how these insights can feedback into the research cycle.

# 2 Methods

## 2.1 Overview of GraphDTA

We propose a novel deep learning model called **GraphDTA** for drug–target affinity (**DTA**) prediction. We frame the DTA prediction problem as a regression task where the input is a drug–target pair and the output is a continuous measurement of binding affinity for that pair. In existing methods, the input drugs and proteins are both represented as 1D sequences. Our approach is different; we represent the drugs as graphs, and so directly capture the bonds among atoms.

## 2.2 Graph representation of drug compounds

## 2.3 Drug representation

**SMILES** (Simplified Molecular Input Line Entry System) was invented to represent molecules to be readable by computers [25], enabling several efficient applications, including fast retrieval and substructure searching. From the SMILES code, drug descriptors like the number of heavy atoms or valence electrons can be inferred and readily used as features for affinity prediction. One could also view the SMILES code as a string. Then, one could featurize the strings with natural language processing (NLP) techniques, or use them directly in a convolutional neural network (CNN).

Instead, we view drug compounds as a graph of the interactions between atoms, and build our model around this conceptualization. To describe a node in the graph, we use a set of atomic features adapted from **DeepChem** [26]. Here, each node is a multi-dimensional binary feature vector expressing five pieces of information: the atom symbol, the number of adjacent atoms, the number of adjacent hydrogens, the implicit value of the atom, and whether the atom is in an aromatic structure. binary feature vector. We convert the SMILES code to its corresponding molecular graph and extract atomic features using the open-source chemical informatics software **RDKit** [27].

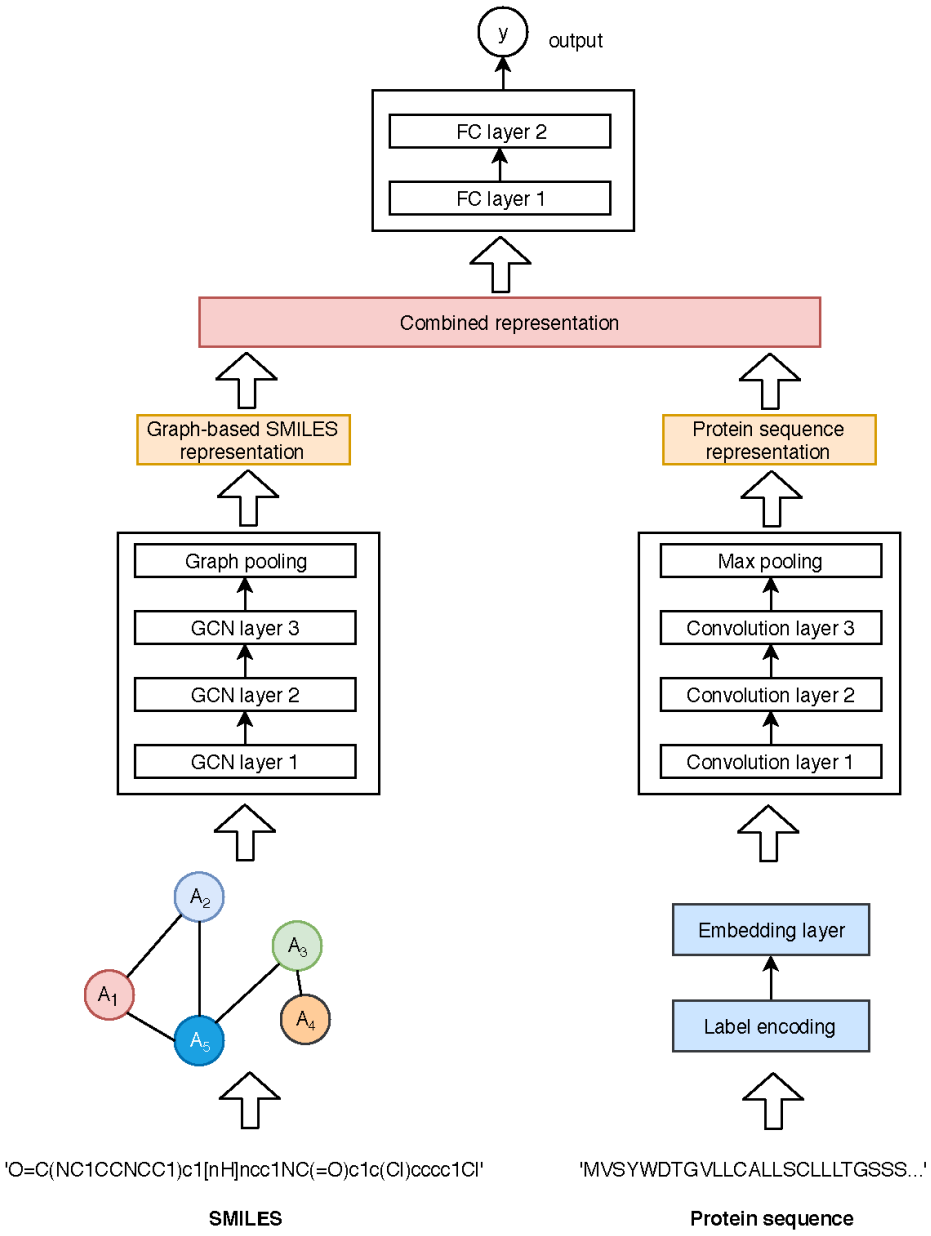


Figure 1: This figure shows the **GraphDTA** architecture. It takes a drug–target pair as the input data, and the pair's affinity as the output data. It works in 3 stages. First, the SMILES code of a drug is converted into a molecular graph, then a deep learning algorithm learns a graph representation. Meanwhile, the protein sequence is encoded and embedded, then several 1D convolutional layers learn a sequence representation. Finally, the two representation vectors are concatenated and passed through several fully connected layers to estimate the output drug–target affinity value.

## 2.4 Deep learning on molecular graphs

Having the drug compounds represented as graphs, the task now is to design an algorithm that learns effectively from graphical data. The recent success of CNN in computer vision, speech recognition, and natural language processing has encouraged research into graph convolution. A number of works have been proposed to handle two main challenges in generalizing CNN to graphs: (1) the formation of receptive fields in graphs whose data points are not arranged as Euclidean grids, and (2) the pooling operation to down-sample a graph. These new models are called graph neural networks.

In this work, we propose a new DTA prediction model based on a combination of graph neural networks and conventional CNN. Figure 1 shows a schematic of the model. For the proteins, we use a string of ASCII characters and apply several 1D CNN layers over the text to learn a sequence representation vector. Specifically, the protein sequence is first categorically encoded, then an embedding layer is added to the sequence where each (encoded) character is represented by a 128-dimensional vector. Next, three 1D convolutional layers are used to learn different levels of abstract features from the input. Finally, a max pooling layer is applied to get a representation vector of the input protein sequence. This approach is similar to the existing baseline models. For the drugs, we use the molecular graphs and trial 4 graph neural network variants, including GCN [28], GAT [29], GIN [30], and a combined GAT-GCN architecture, all of which we describe below.

### 2.4.1 Variant 1: GCN-based graph representation learning

**GCN** model [28] was originally designed for the problem of semi-supervised node classification. The model enables to learn hidden layer representations that capture both local graph structures and features of nodes. This fits well our constructed indirect, node attributed graphs. Formally, denote a graph for a given drug as $G = (V, E)$, where $V$ is the set of $N$ nodes each is represented by a $F$-dimentional vector and $E$ is the set of edges represented as an adjacency matrix $A$. The GCN layer is defined by [28] as

$$\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}X\Theta \tag{1}$$

where $\tilde{A}$ is the graph adjacency matrix with added self loop, $\tilde{D}$ is the graph diagonal degree matrix, $X$ is the node feature matrix, and $\Theta$ is the trainable parameter matrix.

To make the GCN applicable to the task of learning a representation vector of the whole graph, we add a global max pooling layer right after the last GCN layer. In our GCN-based model, we use three consecutive GCN layers each is activated by a ReLU function. Then a global max pooling layer is added to aggregate the whole graph representation.

### 2.4.2 Variant 2: GAT-based graph representation learning

We adopt graph attention network (**GAT**) [29] in our model. Unlike graph convolution techniques, this method proposes an attention-based architecture to learn hidden representations of nodes in a graph by applying a self-attention mechanism. The building block of a GAT architecture is a *graph attention layer.* The GAT layer takes the set of nodes of a graph, the node feature vector $\mathbf{x}$, as input, applies a linear transformation to every node by a weigh matrix $\mathbf{W}$. At the input node $i$ in the graph, the *attention coefficients* between $i$ and its first-order neighbors are computed as

$$a(\mathbf{W}\mathbf{x}_i, \mathbf{W}\mathbf{x}_j) \tag{2}$$

This value indicates the importance of node $j$ to node $i$. These *attention coefficients* are then normalized by applying a soft-max function, then used to compute the output features for nodes as

$$\sigma(\sum_{j\in\mathcal{N}(i)} \alpha_{ij}\mathbf{W}\mathbf{x}_j) \tag{3}$$

where $\sigma(.)$ is a non-linear activation function and $\alpha_{ij}$ are the normalized *attention coefficients*.

In our model, the GAT-based graph learning architecture includes two GAT layers, activated by a ReLU function, then followed a global max pooling layer to obtain graph representation vector. In details, for the first GAT layer, *multi-head-attentions* are applied with the number of heads is set to 10, and the number of output features are set identical with the number of input features. The number of output features of the second GAT are set to 128.

| Method | Protein rep. | Compound rep. | CI | MSE |
|---|---|---|---|---|
| Baseline models | | | | |
| DeepDTA [16] | Smith-Waterman | Pubchem-Sim | 0.790 | 0.608 |
| DeepDTA [16] | Smith-Waterman | 1D | *0.886* | 0.420 |
| DeepDTA [16] | 1D | Pubchem-Sim | 0.835 | 0.419 |
| KronRLS [14, 15] | Smith-Waterman | Pubchem-Sim | 0.871 | 0.379 |
| SimBoost [13] | Smith-Waterman | Pubchem-Sim | 0.872 | 0.282 |
| DeepDTA [16] | 1D | 1D | 0.878 | *0.261* |
| WideDTA [17] | 1D + PDM | 1D + LMCS | *0.886* | 0.262 |
| Proposed method - **GraphDTA** | | | | |
| GCN | 1D | Graph | 0.880 | **0.254** |
| GAT_GCN | 1D | Graph | 0.881 | **0.245** |
| GAT | 1D | Graph | **0.892** | **0.232** |
| GIN | 1D | Graph | **0.893** | **0.229** |

Table 1: Prediction performance on Davis dataset, sorted by MSE. Baseline results are from [16, 17]. Proposed graph-based method is tested with different graph neural networks: GIN [30], GAT [29], GCN [28], and combined GAT-GCN [29, 28]. Italics: best for baseline models, bold: better than baselines.

#### 2.4.3 Variant 3: Graph Isomorphism Network (GIN)

We integrate a recently proposed graph learning method, namely Graph Isomorphism Network (**GIN**) [30]. This model is theoretically proven that it achieves maximum discriminative power among graph neural networks [30]. Specifically, GIN uses a multi-layer perceptron (MLP) model to update the node features as

$$MLP((1+\epsilon)\mathbf{x}_i + \sum_{j \in B(i)} \mathbf{x}_i) \tag{4}$$

where $\epsilon$ is either a learnable parameter or a fixed scalar, $\mathbf{x}$ is the node feature vector, and $B(i)$ is the set of nodes neighbor to *i*.

In our model, the GIN-based graph neural net consists of five GIN layers, each is followed by a batch normalization layer. Finally, a global max pooling layer is added to aggregate a graph representation vector.

#### 2.4.4 Variant 4: GAT-GCN combined graph neural network

We investigate a combination of **GAT-GCN** [29, 28] for learning on graphs in our proposed **GraphDTA** model. In detail, the graph neural network starts by a GAT layer which takes graphs as input and returns the convolved feature matrix to a GCN layer. Each layer is activated by a ReLU function. The graph representation vector is then computed by concatenating the global max pooling and global mean pooling layers from GCN layer output.

### 2.5 Benchmark

To compare our model with the state-of-the-art **DeepDTA** [16] and **WideDTA** [17] models, we use the same 2 datasets from the [16, 17] benchmarks:

- **Davis** contains the binding affinities for all pairs of 72 drugs and 442 targets, measured as Kd constants and ranging from 5.0 to 10.8 [31].
- **Kiba** contains the binding affinities for 2,116 drugs and 229 targets, measured as KIBA scores and ranging from 0.0 to 17.2 [32].

To make the comparison as fair as possible, we use the same train/test data splits from [16, 17] (i.e., 80% used for training and 20% for testing), as well as the same performance metrics: Mean Square Error (**MSE**, the smaller the better) and Concordance Index (**CI**, the larger the better). For all baseline methods, we report the performance metrics originally published in [16, 17].

| Method | Protein rep. | Compound rep. | CI | MSE |
|---|---|---|---|---|
| Baseline models | | | | |
| DeepDTA [16] | 1D | Pubchem-Sim | 0.718 | 0.571 |
| DeepDTA [16] | Smith-Waterman | Pubchem-Sim | 0.710 | 0.502 |
| KronRLS [14, 15] | Smith-Waterman | Pubchem-Sim | 0.782 | 0.411 |
| SimBoost [13] | Smith-Waterman | Pubchem-Sim | 0.836 | 0.222 |
| DeepDTA [16] | Smith-Waterman | 1D | 0.854 | 0.204 |
| DeepDTA [16] | 1D | 1D | 0.863 | 0.194 |
| WideDTA [17] | 1D + PDM | 1D + LMCS | *0.875* | *0.179* |
| Proposed method - **GraphDTA** | | | | |
| GAT | 1D | Graph | 0.866 | 0.179 |
| GIN | 1D | Graph | **0.882** | **0.147** |
| GCN | 1D | Graph | **0.889** | **0.139** |
| GAT_GCN | 1D | Graph | **0.891** | **0.139** |

Table 2: Prediction performance on Kiba dataset, sorted by MSE. Baseline results are from [16, 17]. Proposed graph-based method is tested with different graph neural networks: GIN [30], GAT [29], GCN [28], and combined GAT-GCN [29, 28]. Italics: best for baseline models, bold: better than baselines.

### 2.6 Model interpretation

The activation of nodes within layers of a deep neural network are called latent variables, and can be analyzed directly to understand how a model's performance relates to domain knowledge [33]. We obtained the 128 latent variables from the graph neural network layer, and analyzed them directly through a redundancy analysis. This multivariable statistical method allows us to measure the percent of the total variance within the latent variables that can be explained by an external data source. In our case, the external data source is a matix of 38 molecular descriptors for each drug (available from JoeLib []).

We also compare the value of the principal components from these latent variables with the per-drug test set error. Here, the per-drug (or per-protein) error refers to the median of the absolute error between the predicted DTA and the ground-truth DTA for all test set pairs containing that drug (or that protein). For these analyses, we focus on the GIN model (because of its superior performance) and the Kiba dataset (because of its larger drug catalog).

## 3 Results and Discussion

### 3.1 Graphical models outperform the state-of-the-art

Table 1 compares the performance of our 4 variant **GraphDTA** models with the existing baseline models for the Davis dataset. Here, all 4 variants had the lowest MSE. The best variant had an MSE of 0.229 which is 12.3% lower than the previous best baseline of 0.261. The improvement is less obvious according to the CI metric, where only 2 of the 4 variants had the highest CI. The best CI for a baseline model was 0.886. By comparison, the GAT and GIN methods achieved a CI of 0.892 and 0.893, respectively.

Table 2 compares the performance of our **GraphDTA** models with existing baseline models for the Kiba dataset. Here, 3 of the 4 variants had the lowest MSE and the highest CI, including GIN, GCN, and GAT-GCN. Of note, the best MSE here is 0.139, which is 22.3% lower than the best baseline. Of all variants tested, GIN is the only one that had the best performance for both datasets and for both performance measures. For this reason, we focus on the GIN in all post-hoc analyses.

### 3.2 Graphical models discover known drug properties

A graph neural network works by abstracting the molecular graph of each drug into a new feature vector of *latent variables*. In our model, there are 128 latent variables which together characterise the structural properties of the drug. Since the latent variables are learned during the DTA prediction task, we assume that they represent graphical features that contribute meaningfully to DTA.

Unfortunately, it is not straightforward to determine the molecular sub-structures to which each latent variable corresponds. However, we can regress the learned latent space with a matrix of known molecular descriptors to look for overlap. Figure 2 shows a redundancy analysis of the 128 latent variables regressed

with 38 molecular descriptors (available from JoeLib []). From this, we find that 20.19% of the latent space is explained by the known descriptors, with the "Number of aliphatic OH groups" contributing most to the explained variance. Indeed, two latent variables correlate strongly with this descriptor: hidden nodes V58 and V14 both tend to have high activation when the number of aliphatic OH groups is large. This finding provides some insight into how the graphical model might "see" the drugs through molecular sub-structures, though most of the latent space is orthogonal to the known molecular descriptors.

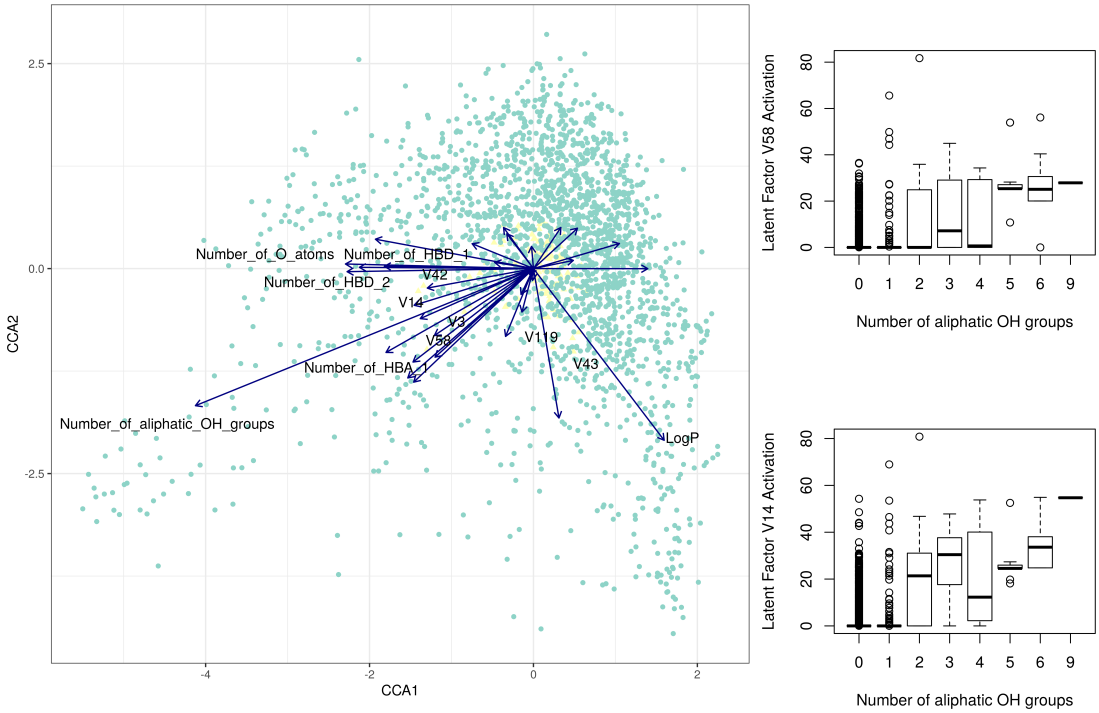


Figure 2: The left panel of the figure shows a redundancy analysis triplot for the 128 Kiba drug latent variables regressed with 38 JoeLib molecular descriptors. The blue dots represent drugs, the yellow dots represent latent variables (with the top 5 labelled explicitly), and the arrows represent molecular descriptors (with the top 5 labelled explicitly). The right panel of the figure shows the activation of two latent variables plotted against the number of aliphatic OH groups in that drug. These results suggest that the graph convolutional network can abstract known molecular descriptors without any prior knowledge.

## 3.3 A few drugs contribute disproportionately to total error

Although the **GraphDTA** model outperforms its competitors, we wanted to know more about why its predictions sometimes failed. For this, we averaged the prediction error for each drug (and each protein), for both the Davis and Kiba test sets. Figures 3 and 4 show the median of the absolute error (MAE) for affinity prediction, sorted from smallest to largest. Interestingly, we see that a handful of drugs (and a handful of proteins) contribute disproportionately to the overall error. Of note, CHEMBL1779202 (an ALK inhibitor), CHEMBL1765740 (a PDK1 inhibitor) and the protein CSNK1E all had an MAE above 2.

We examined the latent space with regard to the prediction error, but could not find any obvious pattern that separated hard-to-predict drugs from easy-to-predict drugs. The only trend we could find is that the easy-to-predict drugs are more likely to appear as outliers in a PCA of the latent space. Supplemental Figure 1 shows the median errors plotted against the first 6 principal components, where we see that the hard-to-predict drugs usually appear close to the origin. We interpret this to mean that drugs with unique molecular sub-structures are always easy to predict. On the other hand, the hard-to-predict drugs may lack unique structures, though this is apparently true for many easy-to-predict drugs too.

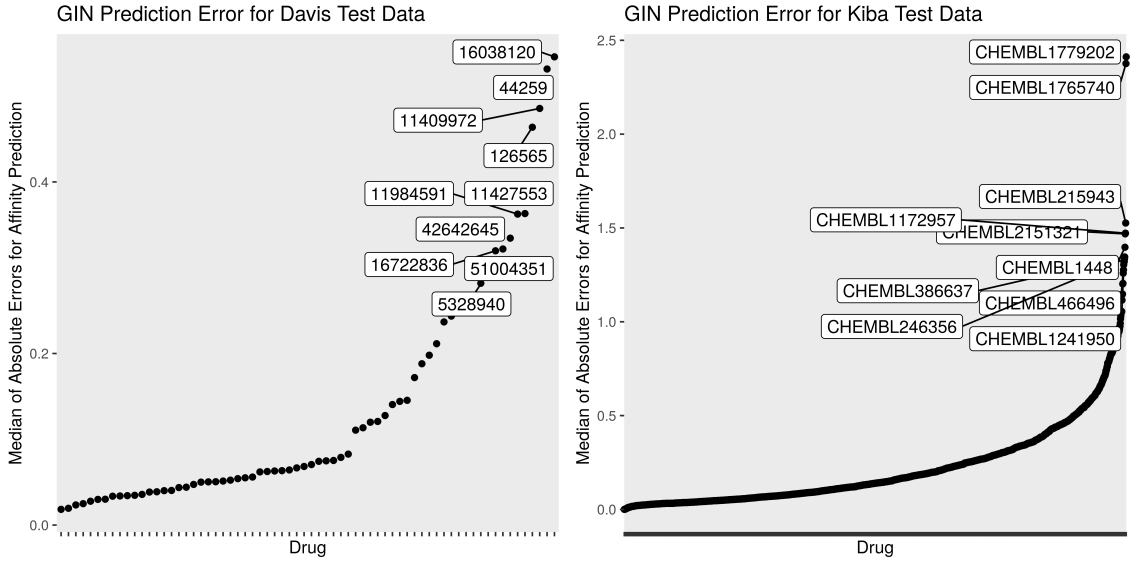


Figure 3: This figure shows the median of the absolute error for each drug, sorted in increasing order, for the Davis and Kiba test sets. Here, we see that the errors are not distributed evenly across the drugs. It is harder to predict the target affinities for some drugs than others.

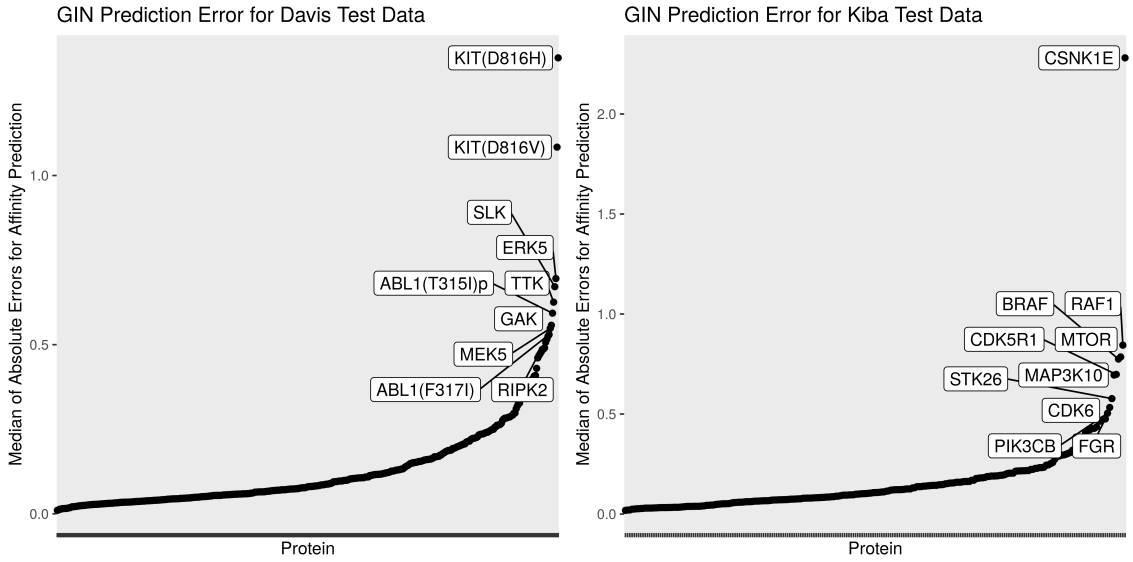


Figure 4: This figure shows the median of the absolute error for each protein, sorted in increasing order, for the Davis and Kiba test sets. Here, we see that the errors are not distributed evenly across the proteins. It is harder to predict the target affinities for some proteins than others.

## 3.4 Model interpretation and the research cycle

Knowing how a model works and when a model fails can feedback into the research cycle. In the post-hoc analysis of our model, we find that a graph neural network can learn the importance of known molecular descriptors without any prior knowledge. However, most of the learned latent variables remain unexplained by the available descriptors. Yet, the model's performance implies that these learned representations are *useful* in affinity prediction. This suggests that there overlaps and differences in how machines "see" chemicals versus how human experts see them. Understanding this distinction may further improve model performance while also helping to elucidate the mechanisms that underly drug–target affinity.

Meanwhile, the distribution of the test set errors suggest that there are "problem drugs" (and "problem proteins") for which prediction is especially difficult. One could action this insight either by collecting more training data for these drugs (or proteins), or by using domain-knowledge to engineer features that

complement the molecular graphs. Indeed, knowing that the graphical outliers are the easiest to predict suggests that some additional feature input may be needed to differentiate between drugs that lack a distinct molecular sub-graph. Although 2D graphs contain more information than 1D strings, our model still neglects the stereochemistry (e.g., the 3D arrangement) of the molecules. Future experiments could test whether representing drugs in 3D (or proteins in 2D) would improve model performance further.

# 4 Summary

In this work, we propose a novel method for estimating drug–target binding affinity, called **GraphDTA**, which represents drugs as graphs. We show that graph neural networks not only predict drugs–target affinity better than non-deep learning models, but also outperform competing deep learning methods. In particular, the GIN method performs consistently well for two separate benchmark datasets and for two key performance metrics. Our results confirm that deep learning models are appropriate for drug–target binding affinity prediction, and that representing drugs as graphs can lead to further improvements. Although we focus on drug–target affinity prediction, our **GraphDTA** model is a generic solution for any collaborating filtering or recommendation problem in which any of the data input can be represented by graphs.

# Declarations

## Ethics approval and consent to participate

Not applicable.

## Consent to publish

Not applicable.

## Availability of data and materials

**GraphDTA** is implemented in Python. Related data, pre-trained models, and source code are publicly available at https://github.com/thinng/GraphDTA.

## Competing interests

The authors declare that they have no competing interests.


## Funding

Not applicable.


## Author's contributions

TN, SV conceived the idea of this work. TN, HL, TL, SV refined the idea. TN, HL designed and performed the experiments. TPQ interpreted the model. TN, HL, TPQ, TL drafted the manuscript. All authors revised the manuscript and approved the final version.


## Acknowledgements

Not applicable.


# References


[1] Asher Mullard. New drugs cost US $2.6 billion to develop. *Nature Reviews Drug Discovery*, 13(12):877, 2014.

[2] Ted T Ashburn and Karl B Thor. Drug repositioning: identifying and developing new uses for existing drugs. *Nature Reviews Drug Discovery*, 3(8):673, 2004.

[3] Allen D Roses. Pharmacogenetics in drug discovery and development: a translational perspective. *Nature Reviews Drug Discovery*, 7(10):807, 2008.

[4] Stephen M Strittmatter. Overcoming drug development bottlenecks with repurposing: old drugs learn new tricks. *Nature Medicine*, 20(6):590, 2014.

[5] Philip Cohen. Protein kinases–the major drug targets of the twenty-first century? *Nature Reviews Drug Discovery*, 1(4):309, 2002.

[6] Martin EM Noble, Jane A Endicott, and Louise N Johnson. Protein kinase inhibitors: Insights into drug design from structure. *Science*, 303(5665):1800–1805, 2004.

[7] Mukund Deshpande, Michihiro Kuramochi, Nikil Wale, and George Karypis. Frequent substructure-based approaches for classifying chemical compounds. *IEEE Transactions on Knowledge and Data Engineering*, 17(8):1036–1050, 2005.

[8] Gerard Manning, David B Whyte, Ricardo Martinez, Tony Hunter, and Sucha Sudarsanam. The protein kinase complement of the human genome. *Science*, 298(5600):1912–1934, 2002.

[9] Shawn J Stachel, John M Sanders, Darrell A Henze, Mike T Rudd, Hua-Poo Su, Yiwei Li, Kausik K Nanda, Melissa S Egbertson, Peter J Manley, Kristen LG Jones, et al. Maximizing diversity from a kinase screen: identification of novel and selective pan-Trk inhibitors for chronic pain. *Journal of Medicinal Chemistry*, 57(13):5800–5816, 2014.

[10] Sarah L Kinnings, Nina Liu, Peter J Tonge, Richard M Jackson, Lei Xie, and Philip E Bourne. A machine learning-based method to improve docking scoring functions and its application to drug repurposing. *Journal of Chemical Information and Modeling*, 51(2):408–419, 2011.

[11] Murat Iskar, Georg Zeller, Xing-Ming Zhao, Vera van Noort, and Peer Bork. Drug discovery in the age of systems biology: the rise of computational approaches for data integration. *Current Opinion in Biotechnology*, 23(4):609–616, 2012.

[12] Steven M Corsello, Joshua A Bittker, Zihan Liu, Joshua Gould, Patrick McCarren, Jodi E Hirschman, Stephen E Johnston, Anita Vrcic, Bang Wong, Mariya Khan, et al. The Drug Repurposing Hub: a next-generation drug library and information resource. *Nature Medicine*, 23(4):405, 2017.

[13] Tong He, Marten Heidemeyer, Fuqiang Ban, Artem Cherkasov, and Martin Ester. SimBoost: a read-across approach for predicting drug–target binding affinities using gradient boosting machines. *Journal of Cheminformatics*, 9(1):24, 2017.

[14] Anna Cichonska, Balaguru Ravikumar, Elina Parri, Sanna Timonen, Tapio Pahikkala, Antti Airola, Krister Wennerberg, Juho Rousu, and Tero Aittokallio. Computational-experimental approach to drug-target interaction mapping: a case study on kinase inhibitors. *PLoS Computational Biology*, 13(8):e1005678, 2017.

[15] Anna Cichonska, Tapio Pahikkala, Sandor Szedmak, Heli Julkunen, Antti Airola, Markus Heinonen, Tero Aittokallio, and Juho Rousu. Learning with multiple pairwise kernels for drug bioactivity prediction. *Bioinformatics*, 34(13):i509–i518, 2018.

[16] Hakime Öztürk, Arzucan Özgür, and Elif Ozkirimli. DeepDTA: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, 2018.

[17] Hakime Öztürk, Elif Ozkirimli, and Arzucan Özgür. WideDTA: prediction of drug-target binding affinity. *arXiv*, (arXiv:1902.04166), 2019.

[18] Michał Woźniak, Agnieszka Wołos, Urszula Modrzyk, Rafał L Górski, Jan Winkowski, Michał Bajczyk, Sara Szymkuć, Bartosz A Grzybowski, and Maciej Eder. Linguistic measures of chemical diversity and the 'keywords' of molecular collections. *Scientific Reports*, 8, 2018.

[19] Christian JA Sigrist, Lorenzo Cerutti, Edouard De Castro, Petra S Langendijk-Genevaux, Virginie Bulliard, Amos Bairoch, and Nicolas Hulo. PROSITE, a protein domain database for functional characterization and annotation. *Nucleic Acids Research*, 38(suppl_1):D161–D166, 2009.

[20] Qingyuan Feng, Evgenia Dueva, Artem Cherkasov, and Martin Ester. PADME: A deep learning-based framework for drug-target interaction prediction. *arXiv*, (arXiv:1807.09741), 2018.

[21] Mengying Sun, Sendong Zhao, Coryandar Gilvary, Olivier Elemento, Jiayu Zhou, and Fei Wang. Graph convolutional networks for computational drug development and discovery. *Briefings in Bioinformatics*, 06 2019.

[22] Ke Liu, Xiangyan Sun, Lei Jia, Jun Ma, Haoming Xing, Junqiu Wu, Hua Gao, Yax Sun, Florian Boulnois, and Jie Fan. Chemi-Net: A molecular graph convolutional network for accurate drug property prediction. *International Journal of Molecular Sciences*, 20(14), 2019.

[23] Maya Hirohara, Yutaka Saito, Yuki Koda, Kengo Sato, and Yasubumi Sakakibara. Convolutional neural network based on SMILES representation of compounds for detecting chemical motif. *BMC Bioinformatics*, 19(19):526, 2018.

[24] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of Computer-Aided Molecular Design*, 30(8):595–608, 2016.

[25] David Weininger. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988.

[26] Bharath Ramsundar, Peter Eastman, Patrick Walters, and Vijay Pande. *Deep Learning for the Life Sciences: Applying Deep Learning to Genomics, Microscopy, Drug Discovery, and More*. O'Reilly Media, 2019.

[27] Greg Landrum. RDKit: Open-source cheminformatics.

[28] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[29] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[30] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[31] Mindy I Davis, Jeremy P Hunt, Sanna Herrgard, Pietro Ciceri, Lisa M Wodicka, Gabriel Pallares, Michael Hocker, Daniel K Treiber, and Patrick P Zarrinkar. Comprehensive analysis of kinase inhibitor selectivity. *Nature Biotechnology*, 29(11):1046, 2011.

[32] Jing Tang, Agnieszka Szwajda, Sushil Shakyawar, Tao Xu, Petteri Hintsanen, Krister Wennerberg, and Tero Aittokallio. Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis. *Journal of Chemical Information and Modeling*, 54(3):735–743, 2014.

[33] Vuong Le, Thomas P. Quinn, Truyen Tran, and Svetha Venkatesh. Deep in the Bowel: Highly Interpretable Neural Encoder-Decoder Networks Predict Gut Metabolites from Gut Microbiome. *bioRxiv*, page 686394, June 2019.