BOOSTING DRUG-TARGET AFFINITY PREDICTION FROM NEAREST NEIGHBORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Precisely predicting Drug-Target binding Affinity (DTA) is essential for drug discovery. Recently, deep learning methods have been popular with DTA prediction. However, the prediction accuracy is still far from satisfaction. In this work, inspired by the recent success of retrieval methods, we propose kNN-DTA, a non-parametric embedding-based retrieval method adopted on a pre-trained DTA prediction model, which can extend the power of the DTA model with no or negligible cost. Compared to traditional chemical similarity retrieval, our embedding-based retrieval shows extremely high efficiency. Different from existing methods, we introduce two neighbor aggregation ways from both embedding space and label space that are integrated in a unified framework. Specifically, we propose a *label aggregation* with *pair-wise retrieval* and a *representation aggregation* with *point-wise retrieval* of the nearest neighbors. This method executes in the inference phase and can efficiently boost the DTA prediction performance with no training cost. In addition, we propose an extension, Ada-kNN-DTA, an instance-wise and adaptive aggregation with lightweight learning. Results on four benchmark datasets show that kNN-DTA brings significant improvements, outperforming previous state-of-the-art (SOTA) results, e.g, on BindingDB IC_{50} and K_i testbeds, kNN-DTA obtains new records of RMSE $\mathbf{0.684}$ and $\mathbf{0.750}$. The extended Ada-kNN-DTA further improves the performance to be $\mathbf{0.675}$ and $\mathbf{0.735}$ RMSE. These results strongly prove the effectiveness of our method. Results on other settings and comprehensive studies/analyses also show the great potential of our kNN-DTA approach. Our code is released at `https://github.com/kNN-DTA/kNN-DTA`.

1 INTRODUCTION

Drug discovery has been more and more important, which is a long and expensive process that usually takes tens of years and billions of dollars. Therefore, Computer-Aided Drug Discovery (CADD) plays an important role to help accelerate the journey, especially in the early stage. Among various CADD applications, Drug-Target binding Affinity (DTA) prediction is an essential one. DTA measures the interaction strength between a drug and a target, and the accurate prediction can greatly benefit Virtual Screening (VS) (Inglese & Auld, 2007) and expedite drug repurposing (Pushpakom et al., 2019), e.g., finding potential drugs for COVID-19 (Zhou et al., 2020). Along the way, various computational methods have been proposed for DTA prediction (Gilson & Zhou, 2007; Trott & Olson, 2010; Salsbury Jr, 2010; Pahikkala et al., 2015).

Recently, Deep Learning (DL) methods have been widely applied for DTA prediction with the increased available affinity data, and huge process has been made (Öztürk et al., 2018; Nguyen et al., 2021; Huang et al., 2020). Though DL-based methods are popular, DTA is still an unsolved problem with unsatisfied accuracy (D'Souza et al., 2020). Besides, the training cost of DL-based DTA methods is still high (e.g., multi-GPUs with tens of hours or days) and there are many different deep models already trained for DTA prediction. Therefore, we are thinking of the following question, *how can we further exploit the potential of these existing DL-based DTA models with no or little effort?*

Luckily, non-parametric methods (e.g., k-nearest neighbors) have shown success in various tasks recently, such as language modeling (Khandelwal et al., 2019), machine translation (Gu et al., 2018; Khandelwal et al., 2020), and question answering Guu et al. (2020a). These methods have demonstrated their effectiveness by making the neural models expressive, adaptable and interpretable.

Therefore, in this paper, we propose kNN-DTA as a solution to answer the above question, which is an embedding-based non-parametric approach for DL-based DTA prediction from nearest neighbors. That is, we utilize the drug and target representations extracted from the trained models to retrieve the nearest samples in a datastore (e.g., the original training data). Compared with traditional chemical similarity retrieval (e.g., Tanimoto similarity (Fligner et al., 2002)), our embedding-based retrieval has much higher efficiency ($100\times$ faster) and also quality guarantee (see Section 5.2). Different from common approaches, kNN-DTA introduces two aggregation ways for the retrieved neighbors from both label and embedding spaces. Specifically, a *label aggregation* is performed on the nearest neighbors of drug-target pairs with a *pair-wise embedding retrieval*. Besides, a *representation aggregation* is also conducted on the nearest drug or target representations with a *point-wise embedding retrieval*. The integrated labels and the model prediction are then combined as the final affinity score. Noting that kNN-DTA only needs to execute in the inference phase for a pre-trained DTA model, hence it boosts affinity predictions without any extra training in an efficient and effective way.

We further introduce an extension of kNN-DTA, Ada-kNN-DTA, with lightweight training cost. In Ada-kNN-DTA, a plug-and-play learning module is designed, where the neighbor distances are taken as input to obtain adaptive and instance-wise weights for aggregation. The intuitive motivation behind is that, since each data sample has different neighbors w.r.t the embedding/label distance closeness, adaptive aggregation can potentially boost more precise prediction from these neighbors. Besides, the light training module can automatically learn how to aggregate the neighbors so to avoid manually hyperparameter tuning cost.

We conduct extensive experiments on four benchmarks for evaluation, including BindingDB IC_{50} and K_i, DAVIS and KIBA datasets. On all datasets, significant performance improvement is achieved by kNN-DTA against pre-trained models, and new state-of-the-art (SOTA) results are obtained. For example, on BindingDB IC_{50} and K_i testbeds, kNN-DTA reaches new best RMSE, $\mathbf{0.684}$ and $\mathbf{0.750}$. With Ada-kNN-DTA, the prediction error is further reduced about $\mathbf{0.01}$ RMSE. We then test on four generalization testsets through zero-shot transfer learning, in which kNN-DTA also demonstrates its potential generalization ability. At last, we also deeply show the effectiveness of kNN-DTA and Ada-kNN-DTA through comprehensive studies.

The contributions of this work are as follows. (1) We propose kNN-DTA, a novel non-parametric embedding-based retrieval method to exploit the great potential of existing DL-based DTA models without extra training, which includes two proposed aggregation ways from both embedding space and label space with different retrieval methods. (2) We further introduce an extension of a lightweight Ada-kNN-DTA framework to learn adaptive aggregations with little cost. (3) We conduct extensive experiments and comprehensive studies to demonstrate the effectiveness and high efficiency of our approaches, and new SOTA results are achieved among various testbeds. (4) Lastly, since affinity prediction is highly crucial for virtual screening so to efficiently select potential drugs, our paper delivers the message to chemists/data scientists that using embedding retrieval method upon deep models is a good way to do DTA prediction. We hope our approach can benefit/inspire more people (especially in AI4Science) to think along this way and do more advanced innovations.

2 Related Work

Drug-Target binding Affinity (DTA) Prediction aims to estimate the strength between drug-target interaction. The experimental assay (Inglese & Auld, 2007) is the most reliable method, but it is labour-intense with high cost. Hence, computational methods have been applied, which can be divided into structure-based and structure-free methods. For structure-based ways, molecular docking (Trott & Olson, 2010; Verdonk et al., 2003) and molecular dynamics simulations (Salsbury Jr, 2010) are typical ones. For structure-free methods, machine learning ways include Random Forest (Shar et al., 2016), kernel-based works (Cichonska et al., 2017), and gradient boosting machines (He et al., 2017). Thanks to the increased available affinity data, deep learning models (Öztürk et al., 2018; Nguyen et al., 2021; Öztürk et al., 2019; Huang et al., 2020; Nguyen et al., 2022) are now dominating the DTA prediction, which take different neural networks (e.g., GNNs, CNNs) for representation learning.

Similarity-based Virtual Screening is commonly adopted in classical binding prediction, which usually generates drug and target similarity matrices (Thafar et al., 2022; Abbasi et al., 2020; Ding et al., 2014; Shim et al., 2021; Shi et al., 2018; Ru et al., 2022; Islam et al., 2021). These similarity

matrices serve as features to be integrated into different methods, such as kernel regression (Yamanishi et al., 2008), matrix factorization (Bolgár & Antal, 2016), gradient boosting machine (Tanoori et al., 2021; Thafar et al., 2021), neural network classifiers (Shim et al., 2021; An & Yu, 2021) and so on. Several works also utilize the drug/target similarity to integrate the affinity labels (Van Laarhoven & Marchiori, 2013; Liu et al., 2022). SVM-KNN (Zhang et al., 2006) is a work that combines the kNN with SVM classifier for prediction, but it differs a lot from ours on motivation and process.

Nearest Neighbor Learning and Memory Networks. Recently, kNN retrieval is popular in Natural Language Processing (NLP) (Kaiser et al., 2017; Gu et al., 2018; Borgeaud et al., 2021; Zheng et al., 2021; Lewis et al., 2020). Khandelwal et al. (2019) is among the first work that successfully combines the language model with kNN retrieval method. Later, Khandelwal et al. (2020) shares the similar idea to apply kNN retrieval to machine translation. After that, the kNN-based methods are widely spread, such as question answering (Lewis et al., 2020), pre-training (Guu et al., 2020a;b), and dialogue conversation (Fan et al., 2021). Our kNN-DTA is inspired by them but differs from the aggregation methods and the regression prediction scenarios. Another related field is Memory Networks (Weston et al., 2015; Sukhbaatar et al., 2015; Zhang et al., 2017) which utilize explicit memory module. However, memory networks are part of the model and must be trained and updated, which are mainly designed for LSTM (Hochreiter & Schmidhuber, 1997) to extend the memory cell.

3 METHOD

In this section, we first mathematically define the DTA prediction task and necessary notations. Next, we introduce our kNN-DTA with two aggregation and retrieval ways. Then, we present the extension Ada-kNN-DTA. Finally, we give some discussions.

Preliminary Let $\mathcal{D} = \{(D, T, y)_i\}_{i=1}^{N}$ denotes a DTA dataset, where (D, T, y) is a triplet sample and N is the dataset size. Here D/T is one drug/target from the dataset, and y is the label measuring the binding affinity strength (e.g., IC_{50}, K_i, K_d) between the drug-target pair (more in Appendix A.1). The DTA prediction is then a regression task that aims to predict the affinity score between the drug-target pair. Mathematically, the goal is to learn a mapping function $\mathcal{F} : D \times T \rightarrow y$. A drug D can be represented by different formats, such as simplified molecular-input line-entry system (SMILES) (Weininger, 1988), or graph with nodes (atoms) and edges (bonds), or a 3D conformation where the coordinates of all atoms are available. Similarly, a target T can be represented by amino acid sequences or a 3D conformation. In this work, we take the SMILES strings for drug and amino acid sequences for target. Due to the superior performance of Transformer (Vaswani et al., 2017), we use two Transformer encoders $\mathcal{M}_\mathcal{D}$ and $\mathcal{M}_\mathcal{T}$ to encode the drug D and target T and obtain R_D and R_T. The R_D and R_T are fed into a prediction module $\mathcal{P}$ to get the predicted affinity $\hat{y}$.

3.1 RETRIEVAL-BASED kNN-DTA

Our kNN-DTA incorporates two retrieval methods and aggreation ways, which are label aggregation with pair-wise retrieval and representation aggregation with point-wise retrieval.

3.1.1 LABEL AGGREGATION WITH PAIR-WISE RETRIEVAL

Intuitively, similar drug-target pairs possibly have similar binding affinity scores. Hence, we propose *label aggregation* with *pair-wise retrieval*, which is to aggregate the ground-truth affinity scores from k nearest neighbors retrieved by the embeddings of drug-target pair. Shortly speaking, we first build a key-value memory datastore that contains the encoded representations of all drug-target pairs and their corresponding labeled affinity values, which can be quickly done through a single forward pass of the pre-trained DTA model. Then, the kNN retrieval is performed when evaluating on test samples.

Datastore The memory datastore is constructed offline with a set of key-value pairs (k_i, v_i). Since the affinity score y corresponds to a specific drug-target pair (D_i, T_i) instead of one drug or target only, the key k_i in our datastore is the concatenated representation of R_{D_i} and R_{T_i}, that is $[R_{D_i}; R_{T_i}]$, and the value v_i is the ground-truth affinity score y_i. This is the why we call *pair-wise* retrieval. The datastore $(\mathcal{K}, \mathcal{V})$ is created by the key-value pairs for all the samples in dataset $\mathcal{D}$,

$$(\mathcal{K}, \mathcal{V}) = \{([R_{D_i}; R_{T_i}], y_i) | ((D_i, T_i), y_i) \in \mathcal{D})\}. \tag{1}$$

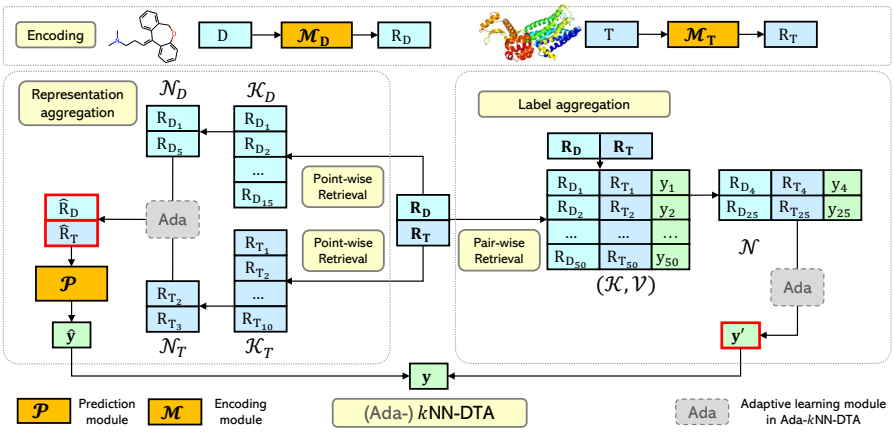


Figure 1: The overall framework of our kNN-DTA and Ada-kNN-DTA. We use two Transformer encoders $\mathcal{M}_D$ and $\mathcal{M}_T$ to encode drug D and target T. The representations R_D and R_T are then separately used for representation aggregation with point-wise retrieval. Meanwhile, the concatenation of R_D and R_T are then used for label aggregation with pair-wise retrieval. The dashed grey 'Ada' parts are the lightweight learning modules in Ada-kNN-DTA. '$\mathcal{P}$' stands for the prediction module, $(\mathcal{K}, \mathcal{V})$, $\mathcal{K}_D$, $\mathcal{K}_T$ are the datastores, and $\mathcal{N}$, $\mathcal{N}_D$, $\mathcal{N}_T$ are retrieved nearest neighbors. The aggregated representation and the affinity are in red outline.

Noting that we only need a single forward pass of the pre-trained DTA model to obtain $(\mathcal{K}, \mathcal{V})$, which can be quickly done.

Pair-wise Retrieval, Label Aggregation, Affinity Prediction Given a test sample (D_t, T_t), we first encode the data through encoder $\mathcal{M}_D$ and $\mathcal{M}_T$ to obtain representations R_{D_t} and R_{T_t}. Then the concatenated $[R_{D_t}; R_{T_t}]$ is used as query to retrieve the k nearest neighbors $\mathcal{N} = \{(k_i, v_i)\} = \{([R_{D_i}; R_{T_i}], y_i)\}$ from the datastore. The retrieval depends on a specific similarity measurement $s(\cdot, \cdot)$ between query and the datastore, such as L_2 distance. With the retrieved nearest neighbor set, we then do *label aggregation* among the labeled affinity from the neighbors in an attentive way. That is, we aggregate the retrieved affinity values y_i by the attention weights α_i to be y'_t. Mathematically,

$$y'_t = \sum_{(k_i, v_i) \in \mathcal{N}} \alpha_i * y_i, \quad \alpha_i = \frac{\exp\left(s([R_{D_t}; R_{T_t}], k_i)/\tau\right)}{\sum_{(k_j, v_j) \in \mathcal{N}} \exp\left(s([R_{D_t}; R_{T_t}], k_j)/\tau\right)}, \tag{2}$$

where τ is the temperature, and y_i equals to v_i in above equations. The integrated affinity score y_t is supposed to produce a good prediction with the help of retrieved neighbors. The aggregated affinity y'_t from neighbors will be used in the unified framework to produce the final prediction (Section 3.1.3).

3.1.2 Representation Aggregation with Point-wise Retrieval

Apart from above label aggregation that directly affects the predicted affinity scores through the label space, we also introduce another *representation aggregation* with *point-wise retrieval* to leverage the nearest neighbors from the embedding space. This is related to the similarity-based VS methods. Different from the above pair-wise retrieval, here we use separate *point-wise* retrieval for k_D nearest drug representations and k_T nearest target representations. Generally speaking, we build separate datastores for drugs and targets, with only the key (drug and target representations) saved in the datastore since the values we need is the same as the keys (also the drug/target representations). Then kNN retrieval is performed on test drug and target to aggregate representations.

Datastore, Point-wise Retrieval, Representation Aggregation, Affinity Prediction We build a datastore $\mathcal{K}_D$ for drugs and a $\mathcal{K}_T$ for targets. Instead of the key-value pairs, these datastores only save keys k_{D_i} and k_{T_i}. That is, the encoded drug/target representation R_{D_i}/R_{T_i} is stored in $\mathcal{K}_D/\mathcal{K}_T$. Noting that these R_{D_i} and R_{T_i} are the same as that in above pair-wise retrieval method. Thus $\mathcal{K}_D = \{R_{D_i} | D_i \in \mathcal{D}\}, \mathcal{K}_T = \{R_{T_i} | T_i \in \mathcal{D}\}$, where D_i and T_i are the unique drugs and targets.

At test time, given the test sample(D_t, T_t), we use R_{D_t}/R_{T_t} as query to retrieve nearest representations from $\mathcal{K}_D/\mathcal{K}_T$ with similarity metric $s(\cdot, \cdot)$. The retrieved sets are $\mathcal{N}_D = \{R_{D_i}\}$ and

$\mathcal{N}_T = \{R_{T_i}\}$. The kNN retrieval is also based on similarity metric $s(\cdot,\cdot)$ between query representation and the ones in datastore. With the retrieved sets $\mathcal{N}_D$ and $\mathcal{N}_T$, attentive representation aggregation is conducted. Same as the label aggregation, the representation aggregation is $R'_{D_t} = \sum_{R_{D_i} \in \mathcal{N}_D} \alpha_i^D * R_{D_i}, \alpha_i^D = \frac{\exp\left(s(R_{D_t}, R_{D_i})/\tau_D\right)}{\sum_{R_{D_j} \in \mathcal{N}_D} \exp\left(s(R_{D_t}, R_{D_j})/\tau_D\right)}$, and R'_{T_t} is calculated in a same way. With R'_{D_t}/R'_{T_t}, we further aggregate them with query R_{D_t}/R_{T_t} to obtain the final drug and target representation, $\hat{R}_{D_t} = \lambda_D * R_{D_t} + (1-\lambda_D) * R'_{D_t}$ and $\hat{R}_{T_t} = \lambda_T * R_{T_t} + (1-\lambda_T) * R'_{T_t}$, which are then inputted to the model $\mathcal{P}$ for affinity prediction $\hat{y}_t = \mathcal{P}(\hat{R}_{D_t}, \hat{R}_{T_t})$.

3.1.3 Unified Framework

Each of the above aggregation methods can be used to enhance DTA prediction. In order to make the best use of above two ways, we systematically combine them in a unified framework, which is shown in Figure 1. Given the test sample (D_t, T_t), the whole test process is as follows. (1) Use encoders $\mathcal{M}_D$ and $\mathcal{M}_T$ to obtain the representations R_{D_t} and R_{T_t}; (2) Concatenate R_{D_t} and R_{T_t} and use it as a query to retrieve the nearest samples from $(\mathcal{K}, \mathcal{V})$. The label aggregation is performed to the retrieved neighbors' affinity values to obtain y'_t; (3) Use R_{D_t}/R_{T_t} as query to separately retrieve the nearest drug/target representations from $\mathcal{K}_D/\mathcal{K}_T$, and aggregate retrieved representations and the query representations to obtain $\hat{R}_{D_t}/\hat{R}_{T_t}$, then get model prediction $\hat{y}_t = \mathcal{P}(\hat{R}_{D_t}, \hat{R}_{T_t})$; (4) The aggregated y'_t are then combined with the predicted $\hat{y}_t$ to produce the final affinity prediction $y_t = \lambda * \hat{y}_t + (1-\lambda) * y'_t$.

3.2 Extension: Adaptive Retrieval-based Ada-kNN-DTA

The above kNN-DTA only requires retrieving nereast neighbors in the inference phase, and the calculation of the aggregation is parameter-free and training-free. Though efficient, the coefficients for aggregation, e.g., λ/λ_D, are manually designed hyper-parameters in current kNN-DTA and shared for all the test data, without considering the aggregation quality for each specific sample. Hence, to further exploit the power of kNN-DTA and reduce the manually tuning cost of these hyperparameters, we propose an adaptive learning extension Ada-kNN-DTA.

In Ada-kNN-DTA, some lightweight modules are introduced to meta-learn the aggregation weights, e.g., α/α_D, and the coefficients, e.g., λ/λ_D. Concretely, the embedding distances between the query and neighbors $s(\cdot,\cdot)$ are fed into a light meta-network to learn the weights/coefficients[1] and then perform the aggregation. Take the label aggregation as an example, these k distances are first put as a vector (denoted as $S = [s_1, ..., s_k]$) and then the calculation is:

$$y_t = \alpha_1 * \hat{y}_t + \sum_{i=2}^{k+1} \alpha_i * y_i, \quad \alpha_i = \frac{\exp(h_i)}{\sum_{h_j \in h} \exp(h_j)}, \quad h = \max(0, SW_1 + b_1)W_2 + b_2, \tag{3}$$

where W and b are the learnable parameters. Specially, the output dimension of h is $k+1$. The first α_1 in α is the coefficient λ. In this way, we now automatically learn the coefficient λ and adaptive weights α for aggregation. Noting that the training is a meta way that is only conducted on the valid set and then the trained meta-network is directly applied on the test data.

3.3 Discussion

We put some clarification and discussion here. (1) For nearest neighbor retrieval, chemists/biologists usually utilize data-specific chemical similarity, such as fingerprint-based (Fligner et al., 2002) Tanimoto similarity for molecule and SW score (Yamanishi et al., 2008) for protein. Though domain specific, we compare them with our embedding similarity retrieval (in Section 5.2). Results show that embedding-based similarity has not only much higher efficiency (100× faster) but also outstanding performances. Hence, this is highly valuable to prove the superiority of the embedding retrieval. (2) Our kNN-DTA builds three datastores, e.g., the drug-target pair datastore $(\mathcal{K}, \mathcal{V})$ and the drug/target datastore $\mathcal{K}_D/\mathcal{K}_T$. Actually, the representations stored in $\mathcal{K}_D/\mathcal{K}_T$ are the same as the ones in paired $(\mathcal{K}, \mathcal{V})$ with only duplicates removing. Thus, only one forward pass is required for constructing these

[1]We also tried a cross-attention way as in Transformer decoder to feed in the query and neighbors' representations (key) for weights learning, but the results do not show much difference in our experiments.

datastores. (3) Different hyperparameters (e.g., λ, k) need to be searched in kNN-DTA. To reduce the cost, we first separately search for the label and representation aggregations, then we slightly search near these best configurations of them for the unified framework. We have compared the time and storage cost with detailed statistics in the Appendix B.2. (4) When aggregating the nearest labels or representations, we use the similarity-based `softmax` for combination. The simplest method is to use average pooling. In our experiments, we find our attentive way is better than the average one. (5) Our current method uses embeddings of drugs and targets for retrieval and aggregation. Since drugs and targets are from two different domains, there remains much possibility for better integration, e.g., interaction-based attentive aggregation, this would be an interesting future point. (6) Finally, the basic assumption of our work depends on the similarity of the drug and target in the datastore (same as similarity-based VS), and we have somehow shown the reason behind the success in Appendix C.1. In reality, the limitation is that when there are extremely different (new) targets, this would be not easy to be effective practically (as shown in Section 5.3)), which is even hard for medicine chemists. This means the future direction towards new targets virtual screening is very important.

4 EXPERIMENTS

To evaluate our kNN-DTA, we first pre-train a DL-based DTA model as test model, and then perform the kNN retrieval. We introduce the experiments with different settings in this section. If not specified, the pre-trained model, the datastore creation, and the testset are all from the same domain.

4.1 DATASETS AND PRE-TRAINED DTA MODELS

We evaluate on four well-known DTA benchmarks, including BindingDB IC_{50} and K_i (Liu et al., 2007), DAVIS (Davis et al., 2011), and KIBA (Tang et al., 2014). Besides, there are four generalization testsets for zero-shot transfer learning. The statistics of these datasets are in the Appendix A.1.

BindingDB (Liu et al., 2007) is a database of different measured binding affinities. Following previous works such as DeepAffinity (Karimi et al., 2019) and MONN (Li et al., 2020), we evaluated on IC_{50} and K_i affinity scores with same data split, which are 60% for training, 10% for validation and 30% for test. The label of affinity scores are transformed to logarithm scales as commonly done. To evaluate the zero-shot transfer ability of kNN-DTA, following (Karimi et al., 2019), we test on four generalization testsets, ion channel/GPCR/tyrosine kinase/estrogen receptor, where the targets are not in the training set. **DAVIS** (Davis et al., 2011) contains selectivity assays of the kinase protein family and the relevant inhibitors with their respective dissociation constant (K_d) values. **KIBA** (Tang et al., 2014) includes kinase inhibitor bioactivities measured in K_i, K_d, and IC_{50}, and the labels were constructed to optimize the consistency between them by using the statistics they embedded in these quantities (Öztürk et al., 2018). Following DeepPurpose (Huang et al., 2020), we split DAVIS and KIBA datasets into $7:1:2$ as train/valid/test sets.

To evaluate kNN-DTA, we first pre-train DL-based DTA models for each dataset, which consist of Transformer (Vaswani et al., 2017) encoders and the upper Transformer layers for affinity prediction. The performance of these DTA models is ensured to be good when comparing to previous works. Then our kNN-DTA/Ada-kNN-DTA are performed upon the pre-trained DTA models. More details of the model architecture and the DTA model training are put in Appendix A.3.

4.2 PARAMETERS OF kNN-DTA AND EVALUATION METRICS

To find the best hyperparameters for kNN-DTA, we do search on each valid set. We tune k, k_D, k_T in $[2^1, 2^2, ...2^7]$, τ, τ_D and τ_T, in $[10^1, 10^2, ..., 10^5]$, λ, λ_D and λ_T in $[0.1, 0.2, ..., 1.0]$. When searching neighbors, we use FAISS (Johnson et al., 2019), which is a library for efficient nearest neighbor search in high-dimensional spaces. The parameters for the best valid performance are applied to the test set. For training Ada-kNN-DTA, the hidden dimension of the meta-network is 32 and we take no more than $5k$ steps training on one GPU on the valid data. The detailed costs are in Appendix B.2.

We follow previous works (Huang et al., 2020; Öztürk et al., 2018; Karimi et al., 2019) to evaluate the performance. Specifically, (a) root-mean-square error (RMSE) and (b) Pearson Correlation coefficient (R) (Abbasi et al., 2020) are used to evaluate on BindingDB datasets, (c) mean-square error (MSE) and (d) Corcondance Index (CI) (Gönen & Heller, 2005) are on DAVIS and KIBA datasets.

Table 1: Performance of different methods on BindingDB IC_{50} and K_i datasets. The ↓ and ↑ indicate the directions of better results. We report the mean (standard deviation) results with three runs.

Dataset	**IC_{50}**		**K_i**	
Method	**RMSE↓**	**R↑**	**RMSE↓**	**R↑**
Random Forest (Karimi et al., 2019)	0.910	0.780	0.970	0.780
DeepAffinity (Karimi et al., 2019)	0.780	0.840	0.840	0.840
DeepDTA (Öztürk et al., 2018)	0.782	0.848	-	-
MONN (Li et al., 2020)	0.764	0.858	-	-
BACPI (Li et al., 2022)	0.740	0.860	0.800	0.860
Pre-trained DTA	0.717 (0.0066)	0.880 (0.0037)	0.785 (0.0016)	0.876 (0.0008)
+ kNN-DTA	0.684 (0.0021)	0.889 (0.0012)	0.750 (0.0016)	0.882 (0.0004)
+ Ada-kNN-DTA	0.675 (0.0004)	0.889 (0.0000)	0.735 (0.0021)	0.884 (0.0008)

Table 2: Performance of different methods on DAVIS and KIBA datasets. We report the mean (standard deviation) results with three runs. The first four baselines are reported in DeepPurpose* (Huang et al., 2020), others are from the original paper. (-) means the standard deviation is not reported.

Dataset	**DAVIS**		**KIBA**	
Method	**MSE↓**	**CI↑**	**MSE↓**	**CI↑**
KronRLS (Pahikkala et al., 2015)	0.329 (0.019)	0.847 (0.006)	0.852 (0.014)	0.688 (0.003)
GraphDTA (Nguyen et al., 2021)	0.263 (0.015)	0.864 (0.007)	0.183 (0.003)	0.862 (0.005)
DeepDTA (Öztürk et al., 2018)	0.262 (0.022)	0.870 (0.003)	0.196 (0.008)	0.864 (0.002)
DeepPurpose* (Huang et al., 2020)	0.242 (0.009)	0.881 (0.005)	0.178 (0.002)	0.872 (0.001)
DeepCDA (Abbasi et al., 2020)	0.248 (-)	0.891 (0.003)	0.176 (-)	0.889 (0.002)
Affinity2Vec (Thafar et al., 2022)	0.240 (-)	0.887 (-)	0.111 (-)	0.923 (-)
Pre-trained DTA	0.205 (0.0008)	0.893 (0.0021)	0.162 (0.0012)	0.866 (0.0004)
+kNN-DTA	0.190 (0.0004)	0.905 (0.0021)	0.146 (0.0004)	0.886 (0.0004)
+Ada-kNN-DTA	0.191 (0.0009)	0.902 (0.0026)	0.147 (0.0000)	0.885 (0.0004)

4.3 Results on BindingDB Benchmark

The RMSE and Pearson Correlation results of BindingDB IC_{50} and K_i are shown in Table 1. For comparison, we take several works and existing best models as baselines (introduced in Appendix A.2), including Random Forest (Karimi et al., 2019), DeepAffinity (Karimi et al., 2019), DeepDTA Öztürk et al. (2018), MONN (Li et al., 2020), and BACPI (Li et al., 2022). These baseline results are reported from original papers (Random Forest is reported in DeepAffinity and DeepDTA is reported in MONN). From Table 1, we can see: (1) Comparing with existing works, our pre-trained DTA models achieve strong performances (e.g., 0.717 RMSE), which outperform the previous best BACPI (Li et al., 2022) on both RMSE and R. (2) After combined with our kNN-DTA, the performances can be largely improved. For instance, RMSE results on IC_{50} and K_i benchmarks are improved to 0.684 and 0.750, which significantly overpass the pre-trained models by 0.033 and 0.035 RMSE. (3) With Ada-kNN-DTA, the performances are further improved. The RMSE is reduced to be 0.675 and 0.735. Therefore, these numbers can clearly demonstrate the effectiveness of our kNN-DTA and also the adaptive learning in Ada-kNN-DTA.

4.4 Results on DAVIS and KIBA Benchmarks

We then evaluate on DAVIS and KIBA datasets, and the results are presented in Table 2. Compared with BindingDB datasets, DAVIS and KIBA are relatively small-scale. The baseline methods are KronRLS (Pahikkala et al., 2015), GraphDTA (Nguyen et al., 2021), DeepDTA (Öztürk et al., 2018), DeepPurpose (Huang et al., 2020), DeepCDA Abbasi et al. (2020), and Affinity2Vec Thafar et al. (2022). Again, we see that our pre-trained Transformer models obtain good performances compared to previous best works, e.g. 0.205 and 0.162 MSE on DAVIS and KIBA respectively. By applying our kNN-DTA, MSE is reduced to be 0.190 and 0.146. However, Ada-kNN-DTA performs similarly to the kNN-DTA. We then study the reason and find the shape of probability density function for DAVIS/KIBA affinity is highly sharp and different from BindingDB (Appendix A.1). We suspect this centralized distribution may hinder the learning effectiveness from the samples that are not

specific and diverse enough. Nevertheless, Ada-kNN-DTA still achieves strong improvements upon the pre-trained DTA model. All of these results demonstrate that kNN-DTA can work well.

4.5 Retrieval from Other Datastore

Apart from above experiments, we further verify whether adding other/external datastore for retrieval is beneficial. In this experiment, we take the pre-trained model on DAVIS. Besides the DAVIS training set as datastore, we also add BindingDB training data in the datastore, hence the datastore is from two different datasets. Note that part of the targets in the DAVIS are also in BindingDB, so this actually enlarge the retrieval datastore. The evaluation is performed on DAVIS testset and the results are presented in Table 3. We compare the kNN-DTA retrieval on DAVIS datastore, and DAVIS+BindingDB, and Ada-kNN-DTA on DAVIS+BindingDB. It can be seen that retrieval method benefits from additional data and improves the DTA performance, e.g., MSE is reduced from 0.189 to 0.168 when comparing the retrieval from DAVIS only with DAVIS+BindingDB. This experiment shows the easy adoption of our method and also the great potential in real applications.

Table 3: Performance of the DAVIS pre-trained model with different datastores on DAVIS testset.

Method	**MSE↓**	**CI↑**
Pre-trained DTA	0.205	0.893
+ kNN-DTA	0.190	0.905
+ kNN-DTA + BindingDB	0.168	0.914
+ Ada-kNN-DTA + BindingDB	0.168	0.916

5 Study

To better understand our work, we conduct extensive studies. Without specific mention, we take BindingDB K_i as testbed. Due to space limitation, many meaningful studies are in Appendix B/C.

5.1 Ablation

We first conduct ablation study to investigate the effect of our two aggregation ways. We remove the label aggregation and representation aggregation from our kNN-DTA separately and check the performance effect. In Table 4, we can see that (1) removing each of the two aggregation methods hurt the prediction performance. (2) Besides, both aggregation methods benefit the DTA prediction (each of the removing experiment still outperforms our pre-trained model). (3) When comparing these two methods, we can conclude that label aggregation contributes more to the success of kNN-DTA, e.g., the performance drop when removing label aggregation (0.748 v.s. 0.762) is more than removing representation aggregation (0.748 v.s. 0.753).

Table 4: Ablation study on BindingDB K_i dataset with RMSE metric.

Method	**Valid**	**Test**
Pre-trained DTA	0.795	0.784
kNN-DTA	0.758	0.748
- Label Aggregation	0.772	0.762
- Representation Aggregation	0.763	0.753

5.2 Retrieval with Chemical/Biological Similarity

In drug discovery, a widely adopted way to retrieval similar molecule/protein is to use chemical/biological similarity measure, e.g., the 2D/3D structure similarity of the molecules/proteins. The most popular one for molecule is Tanimoto similarity (Fligner et al., 2002) based on fingerprint and SW score (Yamanishi et al., 2008) for protein. Hence we make a comparison of our embedding-based retrieval and the chemical/biological similarity retrieval. Specifically, we use RDKit (Landrum et al., 2013) and Biopython (Cock et al., 2009) to do molecule/protein similarity search, and we keep the size of retrieved nearest neighbors to be 32[2]. Then we evaluate kNN-DTA (for simplicity, we only use label aggregation) by these two

Table 5: Retrieval methods comparison on BindingDB K_i with label aggregation.

Method	**Time Cost**	**Valid**
Pre-trained DTA	5.27 (min)	0.795
Chemical Retrieval	655.73 (min)	0.776
Embedding Retrieval	5.30 (min)	0.763

[2]Due to the heavy cost of RDKit and Biopython (Appendix B.1), we first reduce the searching space to be 64 using our embedding search, then retrieve the nearest 32 neighbors by these tools for statistic comparison. For a fair comparison, we first calculate the fingerprints of drugs and save them in advance.

retrieval methods. The retrieval cost and the RMSE performance are shown in Table 5. We can see that our embedding-based retrieval shows strong advantage over the chemical retrieval on both efficiency ($\mathbf{100}\times$) and the prediction performance, which clearly demonstrate its practical value.

5.3 Results on Zero-shot Transfer

Above experiments build datastores from the training set used for pre-trained model, and the testset is from same domain, which intuitively ensure the similarity between the datastore and the testset. To evaluate the generalization ability of our kNN-DTA, we conduct a zero-shot transfer experiment on BindingDB generalization testsets. Specifically, the targets in ER/Ion Channel/GPCR/Tyrosin Kinase are hold out before data splitting, which are unseen and increasingly different from the BindingDB training set. Thus, we take the model pre-trained on BindingDB, build the datastore on BindingDB training set, and then apply kNN-DTA to evaluate on these four testsets. The results of RMSE and Pearson Correlation are reported in Table 6. We can see that though these testsets are much different from the data in datastore, kNN-DTA also improves the results on some specific sets. For instance, on ER K_i, Tyrosin Kinase IC_{50} and K_i, the RMSE reduced about $0.004/0.006$. Noting this zero-shot transfer is extremely hard. Thus, our method has potential towards the generalization ability. From this experiment, we can see this hard setting should be an important direction for future works.

Table 6: RMSE/R performance evaluation of different methods on the BindingDB generalization testsets with IC_{50} and K_i metrics. 'x/y': x is the RMSE score and y is the Pearson Correlation.

Dataset	**ER**		**Ion Channel**		**GPCR**		**Tyrosin Kinase**	
Method	IC_{50}	K_i	IC_{50}	K_i	IC_{50}	K_i	IC_{50}	K_i
Random Forest	1.41/0.26	1.48/0.14	**1.24/0.16**	**1.46/0.21**	1.40/0.25	**1.20/0.19**	1.58/0.11	1.75/0.10
DeepAffinity	1.53/0.16	1.76/0.09	1.34/0.17	1.79/0.23	1.40/0.24	1.50/0.21	1.24/0.39	2.10/0.16
Pre-trained DTA	1.42/0.38	1.40/0.29	1.47/0.13	1.50/0.27	**1.39/0.31**	1.31/0.38	1.26/0.48	1.54/0.40
+kNN-DTA	**1.41/0.40**	**1.34/0.36**	1.46/0.13	1.49/0.27	**1.39/0.31**	1.30/0.38	**1.22/0.49**	**1.51/0.40**

5.4 Case Study of Nearest Neighbors

We finally provide some retrieved cases (more in Appendix C) to better understand the effect of our method. The study is performed on the pair-wise retrieval for simplicity. We randomly choose one sample that improves after applying our kNN-DTA. Then we look into their retrieved nearest pairs for study. For the specific drug-target pair and their retrieved neighbors, we have several findings. (1) The targets in these retrieved pairs are the same (UniProt ID: P13922), which means these drugs can bind to this same target. This meets the reality as we discussed in Section B.5. Hence we can suspect the benefit is from these drugs that bind to the query target. (2) We plot these drugs in Figure 2 and we can see these drugs are indeed similar from their chemical structure space, which further demonstrates that similar drugs can benefit most to our method for DTA prediction.

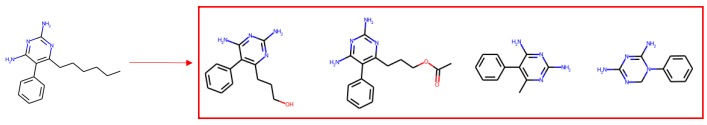


Figure 2: One case sample (left) and its retrieved 4 nearest neighbors (right).

6 Conclusions

In this paper, we propose an embedding-based non-parametric retrieval method, kNN-DTA and its extension Ada-kNN-DTA, for drug-target binding affinity prediction so as to further exploit the potential upon an existing DTA model with no or light cost. Through a label aggregation with pair-wise embedding retrieval and a representation aggregation with point-wise embedding retrieval, kNN-DTA greatly benefits DTA prediction from these retrieved neighbors. We verify the effectiveness of kNN-DTA on four benchmark sets (BindingDB IC_{50} and K_i, DAVIS, KIBA), and obtain significant improvements over previous best models. Comprehensive studies and experiments prove the great potential/practicality of our work. In the future, we will improve our method for better efficiency and also extend it to other applications for drug discovery.

REFERENCES

Karim Abbasi, Parvin Razzaghi, Antti Poso, Massoud Amanlou, Jahan B Ghasemi, and Ali Masoudi-Nejad. Deepcda: deep cross-domain compound–protein affinity prediction through lstm and convolutional neural networks. *Bioinformatics*, 36(17):4633–4642, 2020.

Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pp. 1–6. Ieee, 2017.

Qi An and Liang Yu. A heterogeneous network embedding framework for predicting similarity-based drug-target interactions. *Briefings in Bioinformatics*, 22(6):bbab275, 2021.

Bence Bolgár and Péter Antal. Bayesian matrix factorization with non-random missing data using informative gaussian process priors and soft evidences. In *Conference on Probabilistic Graphical Models*, pp. 25–36. PMLR, 2016.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. *arXiv preprint arXiv:2112.04426*, 2021.

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.

Peter JA Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, et al. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11): 1422–1423, 2009.

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):1046–1051, 2011.

Hao Ding, Ichigaku Takigawa, Hiroshi Mamitsuka, and Shanfeng Zhu. Similarity-based machine learning methods for predicting drug–target interactions: a brief review. *Briefings in bioinformatics*, 15(5):734–747, 2014.

Sofia D'Souza, KV Prema, and Seetharaman Balaji. Machine learning models for drug–target interactions: current knowledge and future directions. *Drug Discovery Today*, 25(4):748–756, 2020.

Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. Augmenting transformers with knn-based composite memory for dialog. *Transactions of the Association for Computational Linguistics*, 9:82–99, 2021.

Robert D Finn, Alex Bateman, Jody Clements, Penelope Coggill, Ruth Y Eberhardt, Sean R Eddy, Andreas Heger, Kirstie Hetherington, Liisa Holm, Jaina Mistry, et al. Pfam: the protein families database. *Nucleic acids research*, 42(D1):D222–D230, 2014.

Michael A Fligner, Joseph S Verducci, and Paul E Blower. A modification of the jaccard–tanimoto similarity index for diverse selection of chemical compounds using binary strings. *Technometrics*, 44(2):110–119, 2002.

Michael K Gilson and Huan-Xiang Zhou. Calculation of protein-ligand binding affinities. *Annu. Rev. Biophys. Biomol. Struct.*, 36:21–42, 2007.

Mithat Gönen and Glenn Heller. Concordance probability and discriminatory power in proportional hazards regression. *Biometrika*, 92(4):965–970, 2005.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor OK Li. Search engine guided neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*, 2020a.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, pp. 3929–3938. PMLR, 2020b.

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):1–14, 2017.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Kexin Huang, Tianfan Fu, Lucas M Glass, Marinka Zitnik, Cao Xiao, and Jimeng Sun. Deeppurpose: a deep learning library for drug–target interaction prediction. *Bioinformatics*, 36(22-23):5545–5547, 2020.

James Inglese and Douglas S Auld. High throughput screening (hts) techniques: applications in chemical biology. *Wiley Encyclopedia of Chemical Biology*, pp. 1–15, 2007.

Sk Mazharul Islam, Sk Md Mosaddek Hossain, and Sumanta Ray. Dti-snnfra: Drug-target interaction prediction by shared nearest neighbors and fuzzy-rough approximation. *Plos one*, 16(2):e0246920, 2021.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

Łukasz Kaiser, Ofir Nachum, Aurko Roy, and Samy Bengio. Learning to remember rare events. *arXiv preprint arXiv:1703.03129*, 2017.

Mostafa Karimi, Di Wu, Zhangyang Wang, and Yang Shen. Deepaffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks. *Bioinformatics*, 35(18):3329–3338, 2019.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*, 2019.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Nearest neighbor machine translation. In *International Conference on Learning Representations*, 2020.

Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, et al. Pubchem in 2021: new data content and improved web interfaces. *Nucleic acids research*, 49(D1):D1388–D1395, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Greg Landrum et al. Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling. *Greg Landrum*, 2013.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33: 9459–9474, 2020.

Min Li, Zhangli Lu, Yifan Wu, and YaoHang Li. Bacpi: a bi-directional attention neural network for compound-protein interaction and binding affinity prediction. *Bioinformatics*, 2022.

Shuya Li, Fangping Wan, Hantao Shu, Tao Jiang, Dan Zhao, and Jianyang Zeng. Monn: a multi-objective neural network for predicting compound-protein interactions and affinities. *Cell Systems*, 10(4):308–322, 2020.

Bin Liu, Konstantinos Pliakos, Celine Vens, and Grigorios Tsoumakas. Drug-target interaction prediction via an ensemble of weighted nearest neighbors with interaction recovery. *Applied Intelligence*, 52(4):3705–3727, 2022.

Tiqing Liu, Yuhmei Lin, Xin Wen, Robert N Jorissen, and Michael K Gilson. Bindingdb: a web-accessible database of experimentally determined protein–ligand binding affinities. *Nucleic acids research*, 35(suppl_1):D198–D201, 2007.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Larry R Medsker and LC Jain. Recurrent neural networks. *Design and Applications*, 5:64–67, 2001.

Thin Nguyen, Hang Le, Thomas P Quinn, Tri Nguyen, Thuc Duy Le, and Svetha Venkatesh. Graphdta: Predicting drug–target binding affinity with graph neural networks. *Bioinformatics*, 37 (8):1140–1147, 2021.

Tri Minh Nguyen, Thin Nguyen, Thao Minh Le, and Truyen Tran. Gefa: Early fusion approach in drug-target affinity prediction. *IEEE/ACM transactions on computational biology and bioinformatics*, 19(2):718–728, 2022.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*, 2019.

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

Hakime Öztürk, Elif Ozkirimli, and Arzucan Özgür. Widedta: prediction of drug-target binding affinity. *arXiv preprint arXiv:1902.04166*, 2019.

Tapio Pahikkala, Antti Airola, Sami Pietilä, Sushil Shakyawar, Agnieszka Szwajda, Jing Tang, and Tero Aittokallio. Toward more realistic drug–target interaction predictions. *Briefings in bioinformatics*, 16(2):325–337, 2015.

Sudeep Pushpakom, Francesco Iorio, Patrick A Eyers, K Jane Escott, Shirley Hopper, Andrew Wells, Andrew Doig, Tim Guilliams, Joanna Latimer, Christine McNamee, et al. Drug repurposing: progress, challenges and recommendations. *Nature reviews Drug discovery*, 18(1):41–58, 2019.

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32, 2019.

Xiaoqing Ru, Xiucai Ye, Tetsuya Sakurai, and Quan Zou. Nerltr-dta: drug–target binding affinity prediction based on neighbor relationship and learning to rank. *Bioinformatics*, 38(7):1964–1971, 2022.

Freddie R Salsbury Jr. Molecular dynamics simulations of protein dynamics and their relevance to drug discovery. *Current opinion in pharmacology*, 10(6):738–744, 2010.

Piar Ali Shar, Weiyang Tao, Shuo Gao, Chao Huang, Bohui Li, Wenjuan Zhang, Mohamed Shahen, Chunli Zheng, Yaofei Bai, and Yonghua Wang. Pred-binding: large-scale protein–ligand binding affinity prediction. *Journal of enzyme inhibition and medicinal chemistry*, 31(6):1443–1450, 2016.

Jian-Yu Shi, Jia-Xin Li, Bo-Lin Chen, and Yong Zhang. Inferring interactions between novel drugs and novel targets via instance-neighborhood-based models. *Current Protein and Peptide Science*, 19(5):488–497, 2018.

Jooyong Shim, Zhen-Yu Hong, Insuk Sohn, and Changha Hwang. Prediction of drug–target binding affinity using similarity-based convolutional neural network. *Scientific Reports*, 11(1):1–9, 2021.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. *Advances in neural information processing systems*, 28, 2015.

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.

Betsabeh Tanoori, Mansoor Zolghadri Jahromi, and Eghbal G Mansoori. Drug-target continuous binding affinity prediction using multiple sources of information. *Expert Systems with Applications*, 186:115810, 2021.

Maha A Thafar, Rawan S Olayan, Somayah Albaradei, Vladimir B Bajic, Takashi Gojobori, Magbubah Essack, and Xin Gao. Dti2vec: Drug–target interaction prediction using network embedding and ensemble learning. *Journal of cheminformatics*, 13(1):1–18, 2021.

Maha A Thafar, Mona Alshahrani, Somayah Albaradei, Takashi Gojobori, Magbubah Essack, and Xin Gao. Affinity2vec: drug-target binding affinity prediction through representation learning, graph mining, and machine learning. *Scientific reports*, 12(1):1–18, 2022.

Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

Twan Van Laarhoven and Elena Marchiori. Predicting drug-target interactions for new drug compounds using a weighted nearest neighbor profile. *PloS one*, 8(6):e66952, 2013.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Marcel L Verdonk, Jason C Cole, Michael J Hartshorn, Christopher W Murray, and Richard D Taylor. Improved protein–ligand docking using gold. *Proteins: Structure, Function, and Bioinformatics*, 52(4):609–623, 2003.

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.

Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2018.

Yoshihiro Yamanishi, Michihiro Araki, Alex Gutteridge, Wataru Honda, and Minoru Kanehisa. Prediction of drug–target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics*, 24(13):i232–i240, 2008.

Hao Zhang, Alexander C Berg, Michael Maire, and Jitendra Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pp. 2126–2136. IEEE, 2006.

Jiani Zhang, Xingjian Shi, Irwin King, and Dit-Yan Yeung. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th international conference on World Wide Web*, pp. 765–774, 2017.

Xin Zheng, Zhirui Zhang, Junliang Guo, Shujian Huang, Boxing Chen, Weihua Luo, and Jiajun Chen. Adaptive nearest neighbor machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 368–374, 2021.

Yadi Zhou, Fei Wang, Jian Tang, Ruth Nussinov, and Feixiong Cheng. Artificial intelligence in covid-19 drug repurposing. *The Lancet Digital Health*, 2(12):e667–e676, 2020.

A EXPERIMENTAL SETTINGS

A.1 DATASET DETAILS

The datasets we used for evaluation are BindingDB IC_{50}, K_i, KIBA, DAVIS, also the BindingDB generalization testsets. Besides, BindingDB K_d dataset is used for out-of-domain datastore creation in Section 4.5. For BindingDB IC_{50} and K_i, we randomly split them into train/valid/test with 6:1:3 as in (Karimi et al., 2019). For KIBA and DAVIS, train/valid/test sets are 7:1:2 as in (Huang et al., 2020). We give the detailed statistics of these datasets in Table 7 and BindingDB generalization testsets in Table 8, including the number of drug-target pairs, the unique molecules and proteins. The affinity label y (floating number) of these datasets are as follows. The affinity values in DAVIS range from 5.0 to 10.8, and the KIBA affinity scores range from 0.0 to 17.2, BinddingDB IC50 and KI range from 2.0 to 11.0. To better show the label information, we further give the label distribution plots of BindingDB IC_{50}, K_i, DAVIS and KIBA datasets in Figure 5. We can see that the affinity distributions of BindingDB are like normal distribution. However, the data distributions of DAVIS and KIBA are different, where the shape is sharp and the values are centered around specific area. This somehow hinders the learning ability of the Ada-kNN-DTA and affects the further performance gain of Ada-kNN-DTA on them.

Table 7: Dataset details.

Information	**Pairs**	**Molecules**	**Proteins**
BindingDB IC_{50}	376,751	255,328	2,782
BindingDB K_i	144,525	87,461	1,620
BindingDB K_d	7,900	63,233	1,504
KIBA	118,254	2,068	229
DAVIS	30,056	68	379

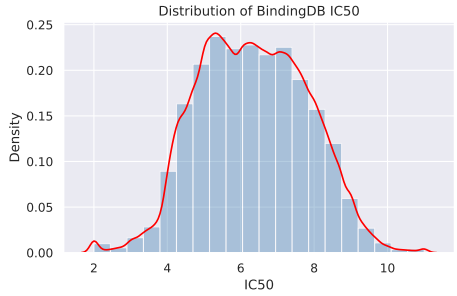
Table 8: BindingDB generalization testsets details.

Dataset		**Pairs**	**Molecules**	**Proteins**
BindingDB IC_{50}	ER	3,374	2,115	6
	Ion Channel	14,599	12,795	125
	GPCR	60,238	48,712	313
	Tyrosin Kinase	34,318	24,608	127
BindingDB K_i	ER	516	287	6
	Ion Channel	8,101	6,838	78
	GPCR	77,994	51,182	323
	Tyrosin Kinase	3,355	2,367	48

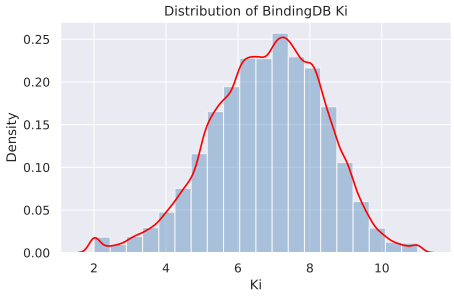
A.2 DETAILS OF COMPARED BASELINE METHODS

The baseline models we take for comparison are described as follows.

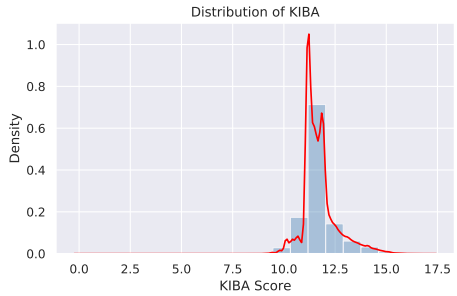
- **DeepDTA** (Öztürk et al., 2018) uses the sequence information for drug and target to extract the features for DTA prediction. The networks are CNN (Albawi et al., 2017)s on both SMILES and protein sequences. DeepDTA was originally evaluated on DAVIS and KIBA datasets, and MONN (Li et al., 2020) evaluated DeepDTA on BindingDB dataset.
- **DeepAffinity** (Karimi et al., 2019) uses RNN (Medsker & Jain, 2001) to encode the SMILES and protein sequences for unsupervised pre-training. Then, CNN layers are appended to make DTA prediction. DeepAffinity was evaluated on BindingDB.
- **MONN** (Li et al., 2020) uses a GCN module to encode molecule and a CNN module for protein. A pairwise interaction module is introduced to link the molecule and protein for drug-target interaction modeling. MONN was evaluated on BindingDB.



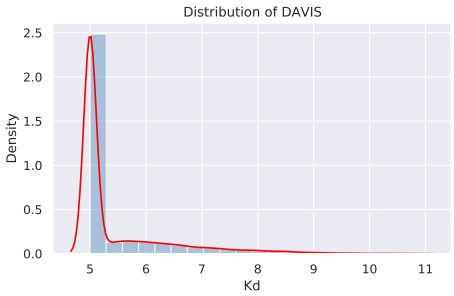
(a) BindingDB IC_{50} label distribution.



(b) BindingDB K_i label distribution.



(c) KIBA label distribution.



(d) DAVIS label distribution.

Figure 3: The label distribution of BindingDB IC_{50} and K_i datasets. The x axis is the affinity value (processed log version), and the y axis is the frequency ratio of the affinity value.

- **KronRLS** (Pahikkala et al., 2015) is a similarity based method. It employs the Kronecker Regularized Least Squares (KronRLS) algorithm that utilizes similarity-based representation of drugs and targets (Pahikkala et al., 2015). KronRLS was evaluated on DAVIS and KIBA.
- **GraphDTA** (Nguyen et al., 2021) uses graph neural networks (Kipf & Welling, 2016; Veličković et al., 2017; Xu et al., 2018) to encode molecule and CNN to encode protein, then a feed-forward layers for prediction. GraphDTA was evaluated on DAVIS and KIBA.
- **DeepPurpose** (Huang et al., 2020) trains customized prediction models by implementing different encoders and various architectures. They were evaluated on DAVIS and KIBA.
- **BACPI** (Li et al., 2022) utilizes a bi-directional attention neural network for integrating the representations from GAT (Veličković et al., 2018) and CNN encoded drugs and targets. BACPI was evaluated on BindingDB.

A.3 Model Configurations

For better understanding the DTA model, we show the architecture of our model in Figure 4. We use two Transformer encoders for molecule encoder $\mathcal{M}_{\mathcal{D}}$ and protein encoder $\mathcal{M}_{\mathcal{T}}$ respectively, and each follows RoBERTa (Liu et al., 2019) architecture and configuration that consists of 16 layers. The first 12 layers of both encoders are initialized from pre-trained molecule model and pre-trained protein model respectively. Specifically, the pre-trained molecule model is from a Transformer-based encoder that trained on molecules from PubChem (Kim et al., 2021) dataset, and the pre-trained protein model is the same as the one in TAPE (Rao et al., 2019) trained on proteins from Pfam (Finn et al., 2014) dataset (but we re-trained using Fairseq (Ott et al., 2019)). As commonly done, both encoders take the masked language modeling objective for pre-training. The remained last 4 Transformer layers are randomly initialized for $\mathcal{M}_{\mathcal{D}}$ and $\mathcal{M}_{\mathcal{T}}$. Then, the total 16 layer encoders and an upper prediction module $\mathcal{P}$ are combined for DTA model training, which is the "Pre-trained DTA" that we used for later kNN retrieval. The embedding/hidden size and the dimension of feed-forward layer are 768 and $3,072$ respectively. The max lengths for molecule and protein are 512 and $1,024$ respectively. The regression prediction head $\mathcal{P}$ is 2-MLP layers with `tanh` activation function and

the hidden dimension is 768. During training, to save the computational cost, the first two pre-trained 12-layer molecule and protein encoders are fixed and used as feature extractors, and only the last 4 Transformer layers and 2-MLP layers are learnable for DTA prediction. The implementation is based on Fairseq toolkit[3]. The model is optimized by Adam (Kingma & Ba, 2014) with learning rate $1e^{-4}$. The dropout and attention dropout of two encoders are 0.1. The learning rate is warmed up in the first 5% update steps and then linearly decayed. The batch size is 32 and we accumulated the gradients 8 times during training.

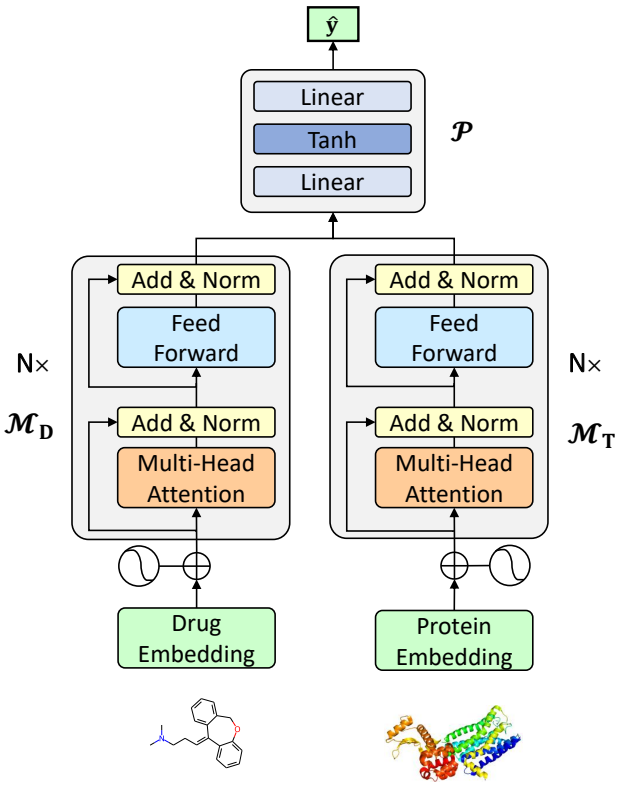


Figure 4: The architecture of our DTA prediction model, which contains one drug encoder and one target encoder ($\mathcal{M}_D$ and $\mathcal{M}_T$), and one upper prediction module ($\mathcal{P}$). Note that the first 12 layers of $N = 16$ layers encoder is pre-trained on unlabeled molecules and proteins and then fixed. Only the last 4 layers are finetuned for DTA prediction.

B More Investigations

B.1 Retrieval Cost Comparison: Tanimoto Similarity/Smith-Waterman Alignment Score v.s. Embedding Similarity

As we discussed before, the commonly adopted retrieval method in drug discovery for molecule is using structure similarity, e.g., Tanimoto similarity (Fligner et al., 2002) based on fingerprint. For protein target, the common method is normalized score of the Smith-Waterman alignment of the amino acid sequence (SW) (Yamanishi et al., 2008), which compares segments of all possible lengths of the protein sequence and optimizes the similarity measure. Hence, we make a separate experiment to compare the retrieval cost by Tanimoto similarity and Smith-Waterman alignment score with our embedding similarity. We take the whole BindingDB K_i training set as the datastore (not like in Section 5.2 that we reducing the search space to 64 at first). For robustness, we use several drugs/targets as the queries, and then count the average retrieval cost on the whole datastore(i.e.

[3] https://github.com/pytorch/fairseq

unique drugs/targets set of the whole set). The result is that, averagely speaking, each drug takes about $\mathbf{0.0156}$ milliseconds for embedding similarity calculation and then ranking, while the RDKit takes about $\mathbf{168.4}$ milliseconds for Tanimoto similarity calculation and ranking. Each target takes about $\mathbf{0.0021}$ milliseconds for embedding similarity search, while calculating Smith-Waterman alignment score by biopython needs about $\mathbf{738k}$ milliseconds. From the comparison, we can see that embedding-based similarity retrieval is extremely faster than traditional chemical similarity search, which greatly shows the high efficiency of the embedding similarity retrieval.

B.2 Model Training and Inference Cost

Table 9: DTA model training time cost on different datasets.

Dataset	**Epochs**	**Cost (GPU hours)**
BindingDB IC_{50}	30	307.2
BindingDB K_i	100	163.2
KIBA	100	149.6
DAVIS	200	104.0

Table 10: Inference cost on different datasets. The time cost is counted by GPU minutes.

Dataset	**BindingDB IC_{50}**		
Method	Pre-trained model	+kNN-DTA	+Ada-kNN-DTA
Training time (mins)	18,424	0	0
Datastore building time (mins)	0	128	128
Meta-network training time (mins)	0	0	76
Inference time (mins per batch)	0.0172	0.0173	0.0177
Inference GPU memory (GB)	4.11	8.56	9.84
Dataset	**DAVIS**		
Method	Pre-trained model	+kNN-DTA	+Ada-kNN-DTA
Training time (mins)	6,240	0	0
Datastore building time (mins)	0	11	11
Meta-network training time (mins)	0	0	20
Inference time (mins per batch)	0.0159	0.0159	0.0175
Inference GPU memory (GB)	4.10	5.30	7.62

To study the training and inference cost of our method, we first count the total cost time for pre-training the DTA models on different datsets (which we used as datastore for retrieval). The training experiments of pre-trained models are conducted on NVIDIA Tesla 8×V100 GPUs. The number of maximal training epochs varies with datasets. We report the total GPU training hours in Table 9. We can see that each training costs more than 100+ GPU hours. In comparison, with the trained models, the inference time cost is low, where the inference is performed on one Tesla V100 GPU. We simply count the inference cost on BindingDB IC_{50} and DAVIS datasets for comparison. During model inference, we set batch size to be 32 for fast inference, and also 32 batch size for building the datastore of our kNN-DTA. The counted results are presented in Table 10, and we can see that our kNN-DTA requires external time for building datastore used for retrieval, but it is less than 1% when compared to the training time cost. For example, it takes $18,424$ minutes for training on BindingDB IC_{50}, but the datastore building only takes 128 minutes, which is only 0.6% of the training cost. As for the inference cost of pre-trained model and our kNN-DTA model, they are comparable (we count the cost of per batch inference data to avoid the effect from testset size). Our kNN-DTA does not increase the inference time. For example, our kNN-DTA costs 0.0173 minutes per batch, and it is similar to pre-trained model (0.0172 minutes per batch). Besides the training time cost, we also put the inference GPU memory cost. Our kNN-DTA indeed needs more GPU memory for retrieved neighbors, but it is affordable since inference will not take too much GPU memory, and it also varies with the size of the datastore. We also separately count for the Ada-kNN-DTA cost. The meta-network is trained on one NVIDIA Tesla V100 GPU. Compared with kNN-DTA, the only extra

requirement is the meta-network training and more GPU memory, and we can see that the cost is not much, which means that the Ada-kNN-DTA method is efficient.

B.3 Effect of The Size of Retrieval Datastore

We further do another study to see the effect of the size of retrieval datastore. We conduct this study on BindingDB K_i dataset, and we vary the datstore size from the full training data to half and quarter of the full set, then we evaluate the performance of the valid and test sets. The results are shown in Table 11. From the table, we can clearly observe that the datastore size indeed impacts the final performance, but they all surpass the original model (without kNN retrieval). Generally, the larger the datastore is, the more possible that we can retrieve for similar drug-target pairs, and the larger performance improvement we can get.

Table 11: Performance effect when varying the size of retrieval datastore on BindingDB K_i.

Retrieval size	**Valid**	**Test**
Pre-trained DTA w/o kNN	0.795	0.784
+kNN-DTA on full set (1)	0.758	0.748
+kNN-DTA on half set (1/2)	0.768	0.759
+kNN-DTA on quarter set (1/4)	0.770	0.760

B.4 Embedding Similarity Measurement

Our kNN-DTA requires specific embedding similarity measurement for retrieval. To see the effect of different similarity measurements, we compare the results of L_2 distance, cosine similarity and dot product on BindingDB K_i dataset. Note that different similarity measurements need different τ because their similarity values are not in the same scale, thus we modify τ to make sure that they are in the same scale. The results are shown in Table 12. We can see that L_2 distance is better than dot product and cosine similarity. Thus we take L_2 distance as similarity metric in our experiments.

Table 12: Effect of embedding similarity measurements on BindingDB K_i dataset w.r.t. RMSE.

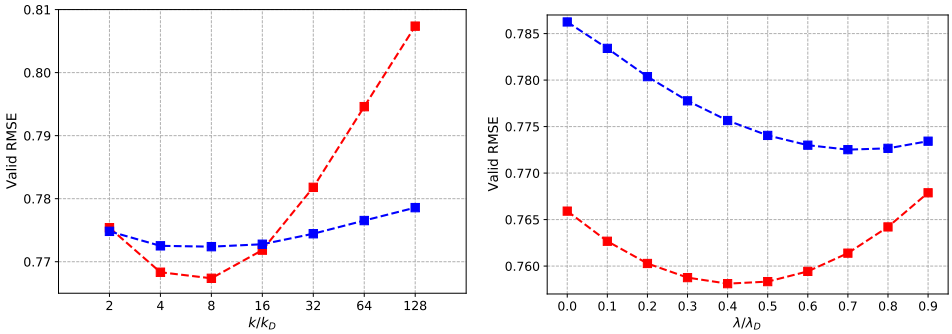
Similarity Measurement	**Valid**	**Test**
L_2 Distance	0.758	0.748
Dot Product	0.777	0.765
Cosine Similarity	0.794	0.782

B.5 Effect of k Nearest Neighbors

It is necessary to study the effect of number k of nearest neighbors, which will affect the retrieval results and also the final prediction performance. We fix all τ as $1,000$ for simplicity and study on BindingDB K_i. Specifically, we vary k and k_D in $[2^1, 2^2, ..., 2^7]$ for label aggregation and representation aggregation respectively, and the performance curves are shown in Figure 5a. We do not study k_T since our preliminary search results show that target neighbors can slightly help the representation aggregation. This is reasonable since usually different drugs can work for one target, but targets are quite different and it is hard for a drug corresponds to multiple targets (this is exactly the difficulty of drug repurposing). Therefore, aggregation of the representations for different targets may not be beneficial a lot. From the figure, we can see that different k and k_D do impact and a relative scale of neighbor size is important.

B.6 Effect of Aggregation Coefficients

We further study the aggregation coefficients λ and λ_D, which affect the integrated labels and representations in kNN-DTA. The study is also on BindingDB K_i. Similarly, we vary the value of λ and λ_D in $[0, 0.1, ..., 0.9]$, and plot the performance changes on BindingDB K_i valid set in Figure 5b. The curves show that different λ and λ_D affects a lot to the prediction performance. This suggests that searching aggregation coefficients is necessary for real-world application, which also somehow indicates the importance of automatic coefficient learning in Ada-kNN-DTA. Hence, we also study the learned $\lambda = \alpha_0$ in Ada-kNN-DTA, and the automatically learned α_0 also ranges in a similar way but is instance different.



(a) Effects of k (red) and k_D (blue).

(b) Effects of λ (red) and λ_D (blue).

Figure 5: The effects of different hyperparameters (k and λ).

B.7 kNN-DTA FOR OTHER BACKBONE MODELS

Generally speaking, our kNN-DTA is model agnostic and it does not depend on specific architecture or what kind of pre-trained DTA model. Hence, in this subsection, we evaluate on different pre-trained DTA models. Besides the Transformer network that used as DTA model in this paper, we also apply our kNN-DTA to graph neural network (GNN)-based DTA model prediction. Specifically, we first take the 12-layer pre-trained molecule/protein encoders and finetune it on DTA. We also take the 4-layer Transformer encoders that trained from scratch for DTA prediction. Above two DTA models are still based on Transformer architecture but with different performances. Besides, we take the recent best GNN work, BACPI Li et al. (2022), as DTA backbone model. Then we apply kNN retrieval on these different pre-trained DTA models to evaluate the performance on BindingDB K_i test set. The results are as shown in Table 13. For the two Transformer-based DTA models, applying our kNN-DTA can consistently improve the model performance as we shown in our main experiments. For our reproduced BACPI, it achieves RMSE score 0.815 and Pearson Correlation 0.856[4] , with kNN-DTA, the results are improved to be 0.797 RMSE and 0.863 Pearson Correlation. These comparisons show the universal effectiveness of our kNN retrieval method. The method can improve performance not only on different model architectures but also on pre-trained DTA models with different performances.

Table 13: Performance of kNN-DTA applied on different pre-trained DTA models on BindingDB K_i.

Pre-trained DTA	**RMSE↓**	**R↑**
12-layer pre-trained DTA	0.854	0.844
+kNN-DTA	0.824	0.853
4-layer DTA train from scratch	0.892	0.827
+kNN-DTA	0.872	0.836
BACPI Li et al. (2022)	0.815	0.856
+kNN-DTA	0.797	0.863

C MORE CASE STUDIES

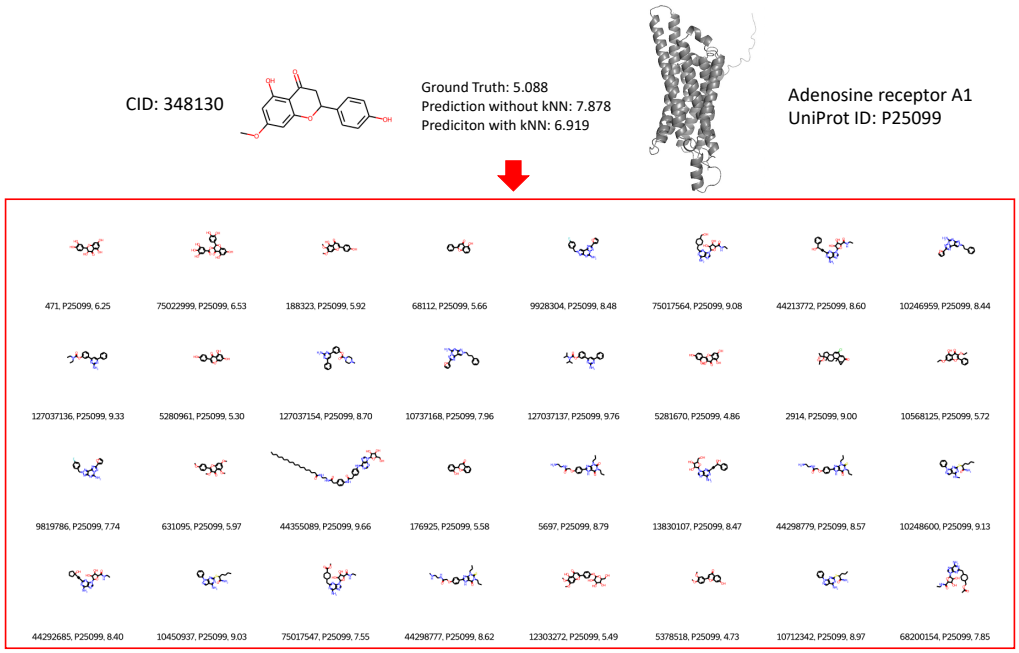
We provide more cases about the retrieved nearest neighbors by the pair-wise retrieval method. We randomly choose some cases that benefit from our kNN-DTA method w.r.t the prediction performance. In Figure 6, we plot the paired cases with their drug (PubChem ID, graph visualization), target (UnitProt ID, 3D visualization), and also their groundtruth binding affinity score (K_i), the pre-trained DTA predicted score and our kNN-DTA predicted score. For the retrieved neighbors of drug-target

[4]We use the officially released code for reproduction, but the result is worse than the reported one in the original paper, e.g., 0.815 RMSE by our reproduction and 0.800 by the original report. But the improvement is still gained by our kNN-DTA.

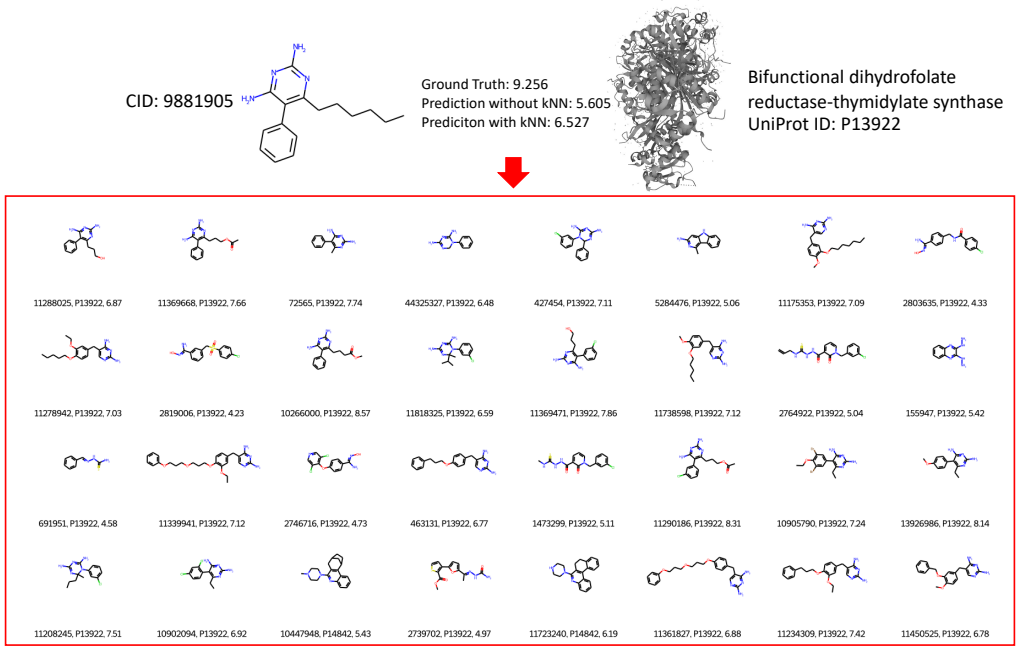
pairs ($k = 32$), we show the graph visualization, PubChem ID of the drugs for clear understanding, and the UnitProt ID of targets, also the affinity scores. From these cases we have several interesting findings. (1) For the retrieved neighbors, almost all of the pairs are with the same target, and the differences are from the drugs. This is reasonable since multiple drugs can be used for one target, and these pairs can help for the test sample. For instance, in case 1, the target is adenosine receptor A1 and it shares for all retrieved neighbors. We can also see from the visualized graphs that the retrieved drugs are in similar structure. (2) Our kNN-DTA model indeed helps the predicted affinity score to be closer to the ground-truth value, specifically for some out-of-distributed pairs. For example, we can see that in case 1 and case 2, the ground-truth values of the test samples are far different from the neighbors. The predictions from our pre-trained model are based on the training data so that the predictions are also far from the ground-truth. With the help of neighbors by our kNN-DTA, the predicted values are pushed to be much closer to the ground-truth. This is interesting and demonstrate the value of kNN-DTA. For case 3, though the prediction of pre-trained model is not far away from ground-truth (in-distribution), our kNN-DTA can make the prediction more accurate.

C.1 Embedding Visualization

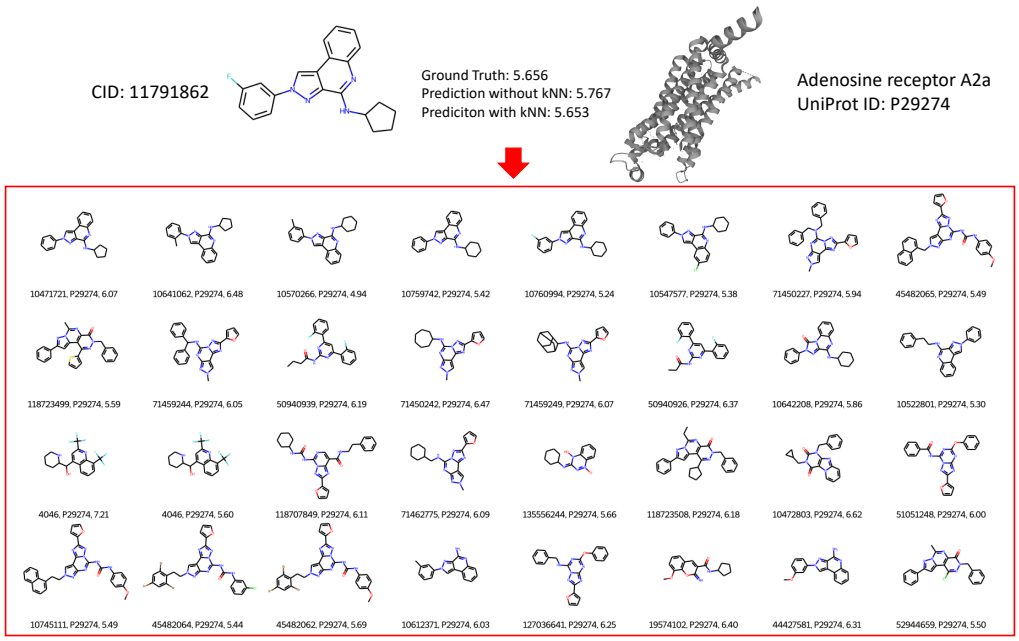
Our retrieval-based method is mainly based on the assumption that for one drug-target pair, other drugs that are similar to the specific query drug may have similar affinity binding scores (e.g., point-wise drug retrieval). In order to better prove this assumption and demonstrate the effect of the retrieval-based method, we provide the drug embedding visualization for case 3 in Figure 6c. Specifically, we plot the embeddings for all drugs that can bind the P29274 (UniProt ID) target, the results are shown in Figure 7. The query drug (CID: 11791862) is in red color, and the nearest 8 drugs are in blue color. The label for each node is its binding affinity score to the P29274 target. From the embedding visualization and the labeled affinity score, we can clearly observe that the nearest neighbors have similar affinity scores, especially when comparing to the right bottom drugs. Hence, this embedding visualization with affinity score can prove the assumption of our kNN retrieval method and support the motivation of our work.



(a) Case 1. Among these 32 neighbors, the target is same for all neighbors.



(b) Case 2. Among these 32 neighbors, 30 are with the same target and 2 are different.



(c) Case 3. Among these 32 neighbors, the target is same for all neighbors.

Figure 6: Three cases of the test samples (top) and retrieved neighbors (bottom). The test sample includes drug (PubChem ID, graph visualization), target (UniProt ID, 3D visualization) and their ground-truth binding affinity in K_i measurement in log space, the pre-trained DTA predicted score and our kNN-DTA predicted score. The retrieved nearest neighbors include drug (PubChem ID, graph visualization), target (UniProt ID) and their binding affinity in K_i measurement in log space.

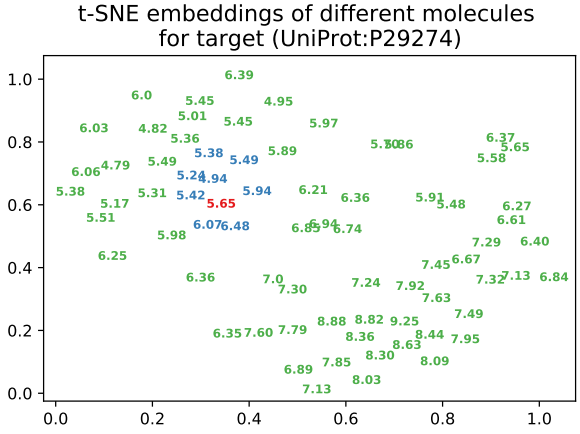


Figure 7: Embedding visualization for all the drugs that can bind to target (UniProt ID: P29274) in Figure 6c. The query drug (CID: 11791862) is in red color, and the nearest 8 drugs are in blue color. The number of each node is the ground-truth binding affinity score.