HybridDTA: Hybrid Data Fusion through Pairwise Training for Drug-Target Affinity Prediction

Hongyu Luo[1,*], Yingfei Xiang[1,*], Xiaomin Fang[1,†], Wei Lin[2], Fan Wang[1,†], Hua Wu[3], Haifeng Wang[3]

1 Baidu Inc., Shenzhen, China.
2 Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong.
3 Baidu Inc., Beijing, China.

***Equal contributions. †Corresponding authors.**

Abstract

Estimating drug-target binding affinity (DTA) is crucial for various tasks, including drug design, drug repurposing, and lead optimization. Advanced works adopt machine learning techniques, especially deep learning, to DTA estimation by utilizing the existing assay data. These powerful techniques make it possible to screen a massive amount of potential drugs with limited computation cost. However, a typical DNN-based training paradigm directly minimizes the distances between the estimated scores and the ground truths, suffering from the issue of data inconsistency. The data inconsistency caused by various measurements, e.g., K_D, K_I, and IC_{50}, as well as experimental conditions, e.g., reactant concentration and temperature, severely hinders the effective utilization of existing data, thus deteriorating the performance of DTA prediction. We propose a novel paradigm for effective training on hybrid DTA data to alleviate the data inconsistency issue. Since the ranking orders of the affinity scores with respect to measurements and experimental batches are more consistent, we adopt a pairwise paradigm to enable the DNNs to learn from ranking orders instead. We expect this paradigm can effectively blend datasets with various measurements and experimental batches to achieve better performances. For the sake of verifying the proposed paradigm, we compare it with the previous paradigm for various model backbones on multiple DTA datasets. The experimental results demonstrate the superior performance of our proposed paradigm. The ablation studies also show the effectiveness of the design of the proposed training paradigm.

Introduction

Drug-target binding affinity prediction is a fundamental task in the drug discovery industry, which refers to estimating the binding affinity scores between numerous candidate compounds and a designated target protein. The compounds with the highest affinity scores are chosen to be the promising compounds for further validation, e.g., in vitro and in vivo experiments. Traditional approaches measure the binding affinity scores through simulation experiments or biological experiments [25], which are laborious, high-cost, and time-consuming. Recently, advanced studies apply machine learning methods [3–5, 12, 24], especially deep neural networks (DNNs) [14, 15, 20, 26–30, 34, 35] that have achieved great success in various domains, to drug-target affinity (DTA) prediction. DNNs attract increasing attention due to their low cost and high time efficiency. These powerful techniques make it possible to screen a huge amount of potential drugs with limited computation cost, significantly accelerating the drug discovery process.

A typical DNN-based training paradigm [14, 15, 20, 26, 29, 30] utilizes the pointwise training method, which minimizes the distance between the estimated score and the value of the actual affinity score for each protein-compound pair. The pointwise training method suffers from data inconsistency. First, the affinity assays may adopt different measurements, such as dissociation constant (K_D), inhibition constant (K_I), half maximal inhibitory concentration (IC_{50}), and half maximal effective concentration (EC_{50}), to evaluate the binding affinities. However, conversion among those measurements is not straightforward. The data inconsistency caused by various measurements is an obstacle to utilizing the existing available data to train a DTA model. The existing works tend to take each measurement independently to address this issue, and each DTA model is trained on the dataset that only concerns a specific measurement. Nevertheless, the data for a specific measurement may not be sufficient to train a DTA model, limiting the model's power. Second, even considering a specific measurement, the assay results of multiple experimental batches for the same protein-compound pair usually differ, as the assay results are affected by a variety of intractable experimental conditions, e.g., reactant concentration and temperature. However, the DTA model trained by the existing training paradigm can not distinguish the assay results from different experimental batches, making it hard to provide a consistent estimation.

Aiming to alleviate the issue of data inconsistency, we propose a novel training paradigm for DTA, called HybridDTA. HybridDTA takes advantage of hybrid data, including data with multiple measurements and experimental batches, aiming to utilize the available data fully. Rather than directly approximate the values of the affinity scores, we adopt a pairwise training paradigm to learn from ranking orders of the affinity scores. For a given target protein, the ranking orders of the candidate compounds are usually more consistent among different measurements and experimental batches. More concretely, we construct numerous fused affinity pairs regarding measurements and experimental batches, and a DNN model is trained on these pairs by a pairwise ranking method. This way can alleviate the negative impact caused by data inconsistency and

fully utilize the existing available data. Thus, our approach can achieve better performance.

To demonstrate the superiority of HybridDTA, we compare it with the previous pointwise training paradigm. We apply HybridDTA and the previous paradigm to various widely used DTA backbone models, including DeepDTA, GraphDTA, and MolTrans, and evaluate their performances on three DTA datasets. HybirdDTA achieves better performance concerning different backbone models on all the datasets. The experimental results also verify the positive impact of blending hybrid data and the design of HybridDTA.

Our contributions can be summarized as follows:

- We propose a novel training paradigm for DTA by hybrid data fusion to address the issue of data inconsistency.
- We adopt a pairwise training method to learn the ranking orders of numerous fused affinity pairs with respect to various measurements and experimental batches.
- Extensive experiments demonstrate the superiority of the proposed paradigm with various backbone models on multiple DTA datasets.

Related Work

Drug-Target Affinity Prediction.

The task of DTA prediction estimates the binding affinity scores between target proteins and candidate compounds. Advanced studies have made significant progress in DTA prediction by applying machine learning methods to predict the affinity scores. These studies can be categorized into traditional machine learning methods and DNN-based methods.

Traditional machine learning methods can be further classified into feature-based methods [8, 28] and similarity-based methods [12, 31]. The feature-based methods first extract the features from the proteins and compounds by feature engineering. Then, the extracted features are used as inputs of the traditional machine learning models, such as Random Forest (RF) and Support Vector Machine (SVM), to estimate the affinity scores. On the other hand, the similarity-based methods exploit the similarity matrix to estimate the affinity score of a protein-compound pair in accordance with the similar proteins and compounds. However, the traditional methods rely on feature engineering, requiring massive expert knowledge to extract valuable features.

Since Deep Neural Networks (DNNs) have achieved considerable success in numerous fields, many researchers employ DNNs for DTA prediction. A typical DNN-based approach usually obtains the descriptors of the proteins and compounds and then estimates their interactions. Existing studies exploit either fingerprints, sequence-based representation, or graph-based representation to encode a compound and utilize sequence-based representation to encode a protein

described by an acid amino sequence. For example, DeepDTA [29] adopts the sequence-based representation method for the SMILES strings of the compounds, while GraphDTA [26] regards a compound as a graph and leverages GNNs [39] to encode the compounds. Instead of using whole sequences of compounds and proteins, MolTrans [14] leverages the Frequent Consecutive Sub-sequence (FCS) mining method to obtain sub-structural information of the compounds and proteins. However, these works utilize the pointwise training paradigm that directly optimizes the distance between the estimated scores and the ground truths, trapping in data inconsistency.

Learning to Rank.

The learning to rank (LTR) technique is widely used in information retrieval and recommender systems [1, 23], which learns the ranking orders of the candidates. The LTR methods can be divided into three categories: pointwise, pairwise, and listwise.

The pointwise techniques optimize the models by directly approximating the ground truths. Then, they rank the candidates according to the estimated scores of the candidates. These approaches regard each candidate independently without considering the relative order between any two candidates, which is easily interrupted by the outliers. In contrast, the pairwise approaches [1, 9, 16] and the listwise approaches [2] consider the order of the candidates. The pairwise approaches optimize the orders of the candidate pairs, while the listwise approaches [2] consider the order of all the candidates. The pairwise approaches are more prevalent than the listwise approaches in the industry due to their computational efficiency.

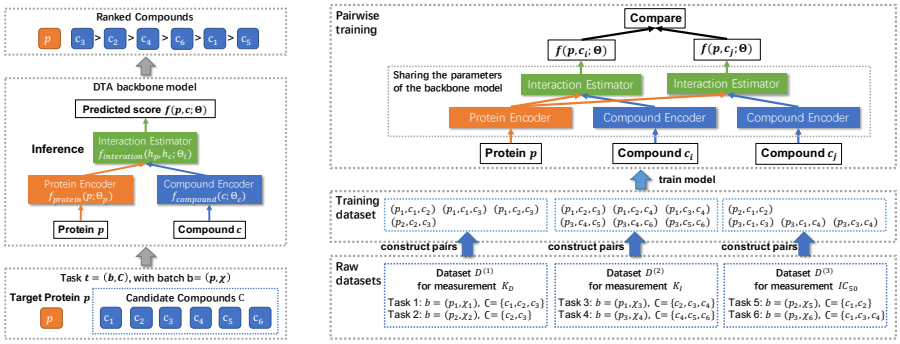
Recently, the LTR techniques have been applied to bioinformatics, such as disease name normalization [18], protein remote homology detection [21]. In particular, the work [38] directly adopts a listwise ranking approach to traditional machine learning methods for DTA. However, simply using LTR techniques without designs fails to address data inconsistency and can not fully utilize the available data.

Materials and Methods

Problem Formulation.

DTA prediction is regarded as a ranking problem, as shown in Fig 1(a). For a ranking task, we apply a DTA backbone model to predict the affinity scores and rank the candidate compounds within the ranking task.

First, a ranking task is defined as $t = (b, \mathcal{C})$, containing an experimental batch b and a set of the candidate compounds χ. An experimental batch b consists of a pair (p, χ), where p denotes the protein and χ denotes the corresponding experimental conditions, including pH degree, temperature, substrate concentration, and so on. The set of candidate compounds is defined as $\mathcal{C} = \{c_i\}|_{i=1}^{n}$,



(a) Problem Formulation.

(b) Pairwise training concerning measurements and experimental batches.

Figure 1: The framework of HybridDTA.

where n represents the number of candidate compounds. We expect to screen out the most promising compounds in $\mathcal{C}$ as the potential drugs for target protein p within experimental batch b.

Then, a DTA backbone model is adopted to estimate the affinity score between protein p and each compound $c \in \mathcal{C}$. A typical DNN-based backbone model for DTA prediction consists of three components: Protein Encoder, Compound Encoder, and Interaction Estimator, as shown in Fig 1(a).

- **Protein Encoder.** The primary structure of a protein is described by an amino acid sequence. A Protein Encoder applies various kinds of sequence-based neural networks, e.g., CNNs [17,19] and Transformers [37], to produce the representation vectors of the proteins. We formalize the Protein Encoder as a function $f_{protein}(p;\Theta_p)$, taking a protein p as input, where Θ_p represents the corresponding model parameters. The representation vector of a protein p is defined as $h_p = f_{protein}(p;\Theta_p)$.

- **Compound Encoder.** A Compound Encoder typically generates the compounds' representation vectors through molecular fingerprints (e.g., ECFP [33] and MACCS [7]), sequence-based representation methods (e.g., LSTMs [13] and Transformers), or graph-based representation methods (e.g., GNNs). We describe the Compound Encoder as a function $f_{compound}(c;\Theta_c)$, taking a compound c as input, where Θ_c represents the model parameters needed to be optimized. Then, the representation vector of a compound c can be written as $h_c = f_{compound}(c;\Theta_c)$.

- **Interaction Estimator.** The Interaction Estimator estimates the interaction score between a protein p and a compound c in accordance to a function $f_{interation}(h_p,h_c;\Theta_i)$, where Θ_i denotes the function's learnable parameters. Function $f_{interation}(h_p,h_c;\Theta_i)$ takes the representation vectors h_c and h_p as inputs and outputs an estimated affinity score.

The pipeline of backbone model $f(p, c; \Theta)$ can be formulated as Equation 1, where $\Theta = \{\Theta_c, \Theta_p, \Theta_i\}$,

$$f(p, c; \Theta) = f_{interaction}(f_{protein}(p; \Theta_p), f_{compound}(c; \Theta_c); \Theta_i). \quad (1)$$

With the affinity scores estimated by the backbone model, we can rank the candidate compounds. The compounds with the highest estimated affinity scores are selected as promising drugs for further validation.

The traditional pointwise training paradigm takes each measurement independently. It optimizes the model parameters Θ by minimizing the distance between the ground-truth $y_{p,c}$ and the estimated score $f(p, c; \Theta)$. Rather than utilizing the traditional pointwise training paradigm, we design a novel training paradigm that adopts pairwise training, which will be elaborated in the following section.

HybridDTA.

Since the previous pointwise training paradigm falls into data inconsistency, we propose a well-designed pairwise training paradigm called HybridDTA to address this issue, taking advantage of hybrid DTA data. The framework of the HybridDTA paradigm is shown in Fig 1(b), learning the parameters of the DTA backbone model from compounds pairs within an experimental batch to select the most promising candidates. The HybridDTA paradigm comprises two main steps: hybrid data fusion and pairwise training. Firstly, numerous hybrid affinity pairs are constructed concerning measurements and experimental batches. The hybrid pairs are fused and taken as the input of the DTA backbone model. Secondly, a pairwise training method is applied to optimize the model parameters of the DTA backbond by learning the orders of pairs.

Hybrid Data Fusion.

Multiple datasets are publicly available for DTA prediction. We attempt to fully explore the potential values of the available data, and hybrid data from various sources (datasets) is used to train the DTA backbone model. Usually, a DTA dataset may contain assay results of one or more measurements, e.g., K_D, K_I, and IC_{50}. As shown in Fig 1(b), rather than independently regarding each measurement in each dataset, we fuse the data for multiple measurements from various raw DTA datasets by generating numerous hybrid affinity pairs.

We denote a raw DTA dataset by $\mathcal{D}^{(k)}$ with k as the index. A typical raw dataset usually contains hundreds to thousands of ranking tasks. For a ranking task $t = (b, \mathcal{C})$ with $b = (p, \mathcal{X})$ in the raw dataset, we can generate many affinity pairs from the candidates $\mathcal{C}$. An affinity pair is defined as the (p, c_i, c_j), which contains a protein and two compounds. Since it is valueless to identify the order between any two compounds for different experimental batches, we concentrate on learning the order of two candidate compounds c_i and c_j with respect to a given batch $b = (p, \mathcal{X})$. The idea is inspired by the previous work Bayesian Personalized Ranking (BPR) [32] for recommender systems. More concretely,

the affinity pairs are generated based on the order of the candidate compounds concerning a given experimental batch.

To formalize the order of the candidate compounds, we introduce a new operator $>_b$ for comparison. If $c_i >_b c_j$, the ground-truth affinity score of compound c_i is significantly larger than that of compound c_j for experimental batch b. Since the ground-truth affinity scores collected from the assays are usually noisy, a hyper-parameter, i.e., deviation ϵ, is introduced to determine whether the two affinity scores differ significantly. That means $c_i >_b c_j$ indicates that $y_{p,c_i} > y_{p,c_j} + \epsilon$, where y_{p,c_i} and y_{p,c_j} denotes the ground-truth affinity scores of compound c_i and compound c_j for target protein p in batch $b = (p, \mathcal{X})$. Then, for a ranking task $t = (p, \mathcal{C})$, a generated binding pair (p, c_i, c_j) should satisfy $c_i >_b c_j$ and $c_i, c_j \in \mathcal{C}$.

Theoretically, we can produce $O(|C|^2)$ pairs for a ranking task $t = (b, \mathcal{C})$. In order to balance the effects of different ranking tasks for training and reduce the computational cost for each iteration, we randomly sample N pairs for each compound within a ranking task t for each iteration, where N is the sampling times. By sampling, for each ranking task, we produce $O(N|C|)$ pairs in each iteration. The number of the samples, i.e., affinity pairs, used to train the DTA model is linear to the number of protein-compound iterations in the raw datasets. All the affinity pairs generated from the available raw datasets $\{\mathcal{D}^{(k)}\}$ are fused to train the DTA backbone model. The set of the fused affinity pairs is defined as the training dataset $\mathcal{D}_{train} = \{(p, c_i, c_j)\}$, used for pairwise training.

Pairwise Training.

Compared with the previous pointwise training paradigm focusing on evaluating the values of the affinity scores, we exploit the pairwise training methods to consider the ranking orders instead. The pairwise training approach takes an affinity pair (p, c_i, c_j) as input. It optimizes the model parameters through a weight sharing of a DTA backbone model to learn the order of compounds c_i and c_j with respect to protein p. Following the classical pairwise ranking method RankNet [1], we regard the pairwise order learning problem as a binary classification task, where the cross-entropy is used as the loss function for an affinity pair (p, c_i, c_j):

$$
\begin{aligned}
L(p, c_i, c_j; \Theta) =& - \hat{P}_{p,c_i,c_j} ln P_{p,c_i,c_j} \\
& - (1 - \hat{P}_{p,c_i,c_j}) ln(1 - P_{p,c_i,c_j}), \\
\hat{P}_{p,c_i,c_j} =& \delta(y_{p,c_i}, y_{p,c_j}), \\
P_{p,c_i,c_j} =& \frac{exp(o_{p,c_i,c_j})}{1 + exp(o_{p,c_i,c_j})}, \\
o_{p,c_i,c_j} =& f(p, c_i; \Theta) - f(p, c_j; \Theta).
\end{aligned}
\tag{2}
$$

$\hat{P}_{p,c_i,c_j}$ indicates whether the ground-truth affinity score of compound c_i, i.e., y_{p,c_i} is larger than that of compound c_j, i.e., y_{p,c_j}. $\delta(x, y)$ is an indicator

function that $\delta(x, y) = 1$ if $x > y$, and $\delta(x, y) = 0$ if $x \leq y$. Besides, P_{p,c_i,c_j} represents the estimated probability whether the affinity score of compound c_i is larger than that of compound c_j. P_{p,c_i,c_j} applies Sigmoid function [11] on o_{p,c_i,c_j}, i.e., the difference between the estimated affinity scores $f(p, c_i; \Theta)$ and $f(p, c_j; \Theta)$. The cross-entropy between the ground-truth probability $\hat{P}_{p,c_i,c_j}$ and the estimated probability P_{p,c_i,c_j} is minimized to learn the model parameters of the DTA backbone model.

As an affinity pair $(p, c_i, c_j) \in \mathcal{D}_{train}$ satisfies $c_i >_b c_j$ and $y_{p,c_i} > y_{p,c_j}$, $\hat{P}_{p,c_i,c_j} = 1$, the loss function for an affinity pair (p, c_i, c_j) in Equation 2 can be simplified as

$$\begin{aligned} L(p, c_i, c_j; \Theta) &= -log\frac{exp(o_{p,c_i,c_j})}{1 + exp(o_{p,c_i,c_j})}, \\ o_{p,c_i,c_j} &= f(p, c_i; \Theta) - f(p, c_j; \Theta). \end{aligned} \tag{3}$$

Then, to utilize the fused data in $\mathcal{D}_{train}$, we define the loss function of dataset $\mathcal{D}_{train}$ as

$$L(\mathcal{D}_{train}; \Theta) = \frac{1}{|\mathcal{D}_{train}|} \sum_{(p,c_i,c_j) \in \mathcal{D}_{train}} L(p, c_i, c_j; \Theta). \tag{4}$$

We minimize the loss function $L(\mathcal{D}_{train}; \Theta)$ to optimize the model parameters of the DTA backbone model $f(p, c; \Theta)$.

Experimental Results

Datasets.

Three DTA datasets are used to evaluate the various paradigms' performance, including two high-quality datasets Davis [6] and KIBA [36], and a web-accessible dataset BindingDB [22]. Datasets Davis and KIBA are widely used in previous studies, as their records are relatively clean. The Davis dataset contains binding affinities measured by K_D, while the KIBA dataset contains binding affinities measured by KIBA scores, integrating multiple measurements (K_D, K_I, and IC_{50}). In this paper, these two datasets are utilized to verify that hybrid data fusion from multiple datasets contributes to improving the accuracy of DTA. On the other hand, the BindingDB dataset collected binding affinities with various measurements (K_D, K_I, IC_{50}, and EC_{50}) as well as the corresponding experimental conditions from multi-sources, containing more noise than datasets Daivs and KIBA. Dataset BindingDB is used to demonstrate that hybrid data fusion from multiple measurements and experimental batches also improves the estimations' accuracy. We pre-processed all these three datasets, the details of which are described in the supplementary information.

Splitting Method.

Most of the previous works [15, 26, 30] randomly split the protein-compound pairs in a dataset into a training set and a test set. However, by using random splitting, a target protein in the test set could have already been observed in the training set. Besides, the assay results of the same protein-compound pair may appear in the training set and test set simultaneously due to multiple experimental batches. This paper splits the data based on the ranking tasks to ensure that the proteins observed in the training set will not appear in the test set, which is more in line with the real-world applications. We generate multiple ranking tasks for each dataset. For the Davis and KIBA datasets, the ranking tasks are generated according to the target proteins. For the BindingDB dataset, the ranking tasks are generated in accordance with the proteins, experimental conditions, e.g., pH degrees, temperatures, and data sources. The details of the datasets are shown in Table 1.

Table 1: Details of the datasets.

Dataset	Measurement	Compounds	Proteins	Interactions	Ranking tasks
Davis	K_D	68	442	30,056	442
KIBA	KIBA score	2,111	229	118,254	229
BindingDB	K_D	9,039	595	44,794	1,106
	K_I	164,730	1,244	323,972	11,730
	IC_{50}	537,019	2,635	851,140	21,000
	EC_{50}	97,346	707	134,889	3,255

Training and Evaluation Settings.

To compare the performance of *Pointiwse* training paradigm and the proposed *HybridDTA* training paradigm, we train multiple DTA backbone models by these two paradigms.

Training Settings.

For the *Pointwise* paradigm, we adopt the hyper-parameters reported in the previous works to train the backbone models. The mean squared error (MSE) between the models' predicted values and the ground-truth affinity scores are taken as the loss function to optimize the model parameters. For the *HybridDTA* paradigm, we apply grid search to search appropriate hyper-parameters, including sample times, learning rate, and batch size. The details of hyper-parameter settings are described in the supplementary information. The loss function for *HybridDTA* has been introduced in section Pairwise Training.

Evaluation Settings.

Concordance Index (CI) [10] is used to evaluate the performance of various paradigms in terms of ranking tasks. CI for a ranking task $t = (b, \mathcal{C})$ with $b = (p, \mathcal{X})$ is defined as:

$$CI(t) = \frac{1}{|\mathcal{C}|^2} \sum_{c_i, c_j \in \mathcal{C}} I(y_{p,c_i} - y_{p,c_j}) \cdot I(f(p, c_i; \Theta) - f(p, c_j; \Theta)), \tag{5}$$

where $I(.)$ is an indicator function. $I(x) = 1$ if $x > 0$, and $I(x) = 0$ otherwise.

Furthermore, we define the overall CI for all the test ranking tasks $\mathcal{T} = \{t_1, t_2, \cdots, t_T\}$ by summarizing $CI(t)$, where T is the number of test ranking tasks. The overall CI is formalized as

$$CI = \frac{1}{\sum_{t \in \mathcal{T}} w(t)} \sum_{t \in \mathcal{T}} w(t) CI(t), \tag{6}$$

where $w(t) = |\mathcal{C}|$ for task $t = (b, \mathcal{C})$ is introduced to balance the impact of different ranking tasks.

DTA Backbone Models.

We compare *Pointwise* paradigm and *HybridDTA* paradigm on three DTA backbone models:

- **DeepDTA** [29] employs three-layers Convolutional Neural Networks (CNNs) as Protein Encoder and Compound Encoder to encode the protein sequences and the compound SMILES strings. Then, for the Interaction Estimator, the encoded protein and compound are concatenated to predict the affinity score.
- **GraphDTA** [26] regards each compound as a graph and attempts several GNNs, such as GIN, GAT, GCN, and GAT-GCN, as the Compound Encoders to represent the compounds. In the meantime, GraphDTA regards each protein as a sequence and adopts CNNs as the Protein Encoder to encode the proteins. The Interaction Estimator is the same as DeepDTA's.
- **MolTrans** [14] decomposes the compounds' SMILES strings and the proteins' acid amino sequences into high-frequency sub-sequences. Then, it applies Transformers as the Compound Encoder and Protein Encoder to obtain the augmented representation with the chemical semantics. MolTrans uses the outer-product operator and CNN blocks as Interaction Estimator to capture the high-order interaction between the compounds and proteins.

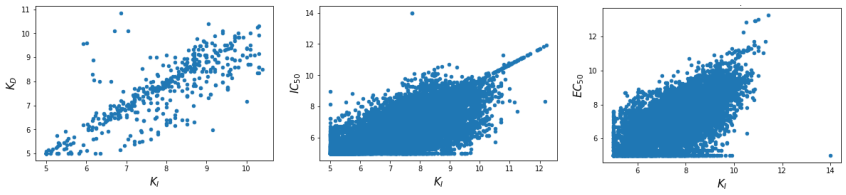
For all the backbone models, we use the hyper-parameters suggested by the corresponding papers.

Table 2: Results of hybrid data fusion of multiple datasets for four DTA backbond models.

Testing dataset	Davis			KIBA		
Training paradigm	Pointwise	HybridDTA		Pointwise	HybridDTA	
Training dataset	Davis	Davis	Davis + KIBA	KIBA	KIBA	Davis + KIBA
GraphDTA (GCN)	0.7906 (0.0057)	0.7995 (0.0044)	**0.8037 (0.0016)**	0.7210 (0.0343)	0.7519 (0.0052)	**0.7529 (0.0054)**
GraphDTA (GATGCN)	0.7917 (0.0034)	0.8023 (0.0009)	**0.8049 (0.0018)**	0.7140 (0.0210)	0.7520 (0.0059)	**0.7575 (0.0025)**
DeepDTA	0.7966 (0.0074)	0.8028 (0.0111)	**0.8029 (0.0060)**	0.7317 (0.0130)	0.7516 (0.0138)	**0.7664 (0.0131)**
MolTrans	0.8047 (0.0083)	0.8079 (0.0037)	**0.8091 (0.0046)**	0.7392 (0.0066)	0.7588 (0.0115)	**0.7644 (0.0183)**

Note: The standard deviations (std) are given in parenthesis.

Figure 2: Scatter plot and Pearson correlation between K_I and other measurements.



(a) K_I & K_D (Pearson correlation: 0.83) (b) K_I & IC_{50} (Pearson correlation: 0.74) (c) K_I & EC_{50} (Pearson correlation: 0.64)

Results.

Hybrid Data Blending Multiple Datasets.

We compare the *Pointwise* paradigm and *HybridDTA* paradigm on the Davis and KIBA datasets, using hybrid data fusion from multiple datasets. We first reserve 1/6 of all the ranking tasks for testing. As Davis or KIBA contain only hundreds of ranking tasks, we conduct 5-fold cross-validation (CV). Table 2 shows the average CI over 5-fold CV on datasets Davis and KIBA for four DTA backbones, including *GraphDTA (GCN)*, *GraphDTA(GATGCN)*, *DeepDTA*, and *MolTrans*. We can draw the following conclusions: First, *HybridDTA* always achieves better performance than *Pointwise* for all the DTA backbone models when training on only a single dataset, because the pairwise training paradigm focuses on learning the ranking orders of the candidate compounds, and it is less likely to be affected by the outliers. Second, *HybridDTA* that utilizes multiple datasets for training works better than that only utilizes a single dataset. It demonstrates that taking advantage of the available external data indeed boosts the models' performance.

Hybrid Data Blending Multiple Measurements.

Dataset BindingDB contains the assay results with various measurements and is used to investigate the effect of data fusion regarding multiple measurements. Besides, as the records in BindingDB are collected from multiple sources with

diverse experimental conditions, the issue of data inconsistency is more prominent, and we can better observe the impact of data inconsistency. Concerning the quantity and quality of assay results for different measurements, we choose to evaluate the paradigms' performance on the assays for measurement K_I. We split the BindingDB dataset into a training set, a validation set, and a test set with the ratio 8:1:1. We train each backbone model on the assays for multiple measurements, e.g., K_I, K_I+K_D, and $K_I+K_D+IC_{50}$, by *HybridDTA* paradigm respectively. Then, we select the model based on the validation set that only contains the assays for K_I and evaluate the performance of the trained models on the test set that only contains the assays for K_I. The settings of the used training measurements are designed according to the Pearson correlations between K_I and other measurements K_D, IC_{50}, and EC_{50}, as shown in Figure 2. We expect the measurement with a higher correlation to K_I will play a more important role when incorporating the corresponding assays for training. Since the assay results for measurement EC_{50} are noisy and the Pearson correlation between K_I and EC_{50} is low, we do not consider utilizing the assays for EC_{50} in this work.

Table 3: Results of hybrid data fusion of multiple measurements on K_I of BindingDB for four DTA backbone models.

Training paradigm	Pointwise	HybridDTA		
Training measurements	K_I	K_I	$K_I + K_D$	$K_I + K_D + IC_{50}$
GraphDTA (GCN)	0.6197 (0.0029)	0.6247 (0.0106)	0.6283 (0.0076)	**0.6297 (0.0101)**
GraphDTA (GATGCN)	0.6289 (0.0046)	0.6305 (0.0063)	0.6341 (0.0090)	**0.6390 (0.0065)**
DeepDTA	0.6176 (0.0036)	0.6244 (0.0008)	0.6322 (0.0057)	**0.6347 (0.0060)**
MolTrans	0.6210 (0.0048)	0.6573 (0.0021)	**0.6595 (0.0004)**	0.6564 (0.0026)

Note: The standard deviations (std) are given in parenthesis.

Table 3 shows the average CI over 5 runs of *Pointwise* and *HybridDTA* for four DTA backbone models. The experimental results indicate that *HybridDTA* surpasses *Pointwise* for all the backbone models. Moreover, blending the assays of multiple measurements could enhance the performance of *HybridDTA*. When adding K_D to the training dataset, all the backbone models work better. If we further add measurement IC_{50}, most of the backbone models achieve further small improvement. This phenomenon reveals that the data with higher correlation contributes more to improving the performance, since the Pearson correlation between K_I and K_D is much higher than that between K_I and IC_{50}.

Table 4: Impact of various designs for ranking tasks.

Design	Batch	Protein	Random
GraphDTA (GCN)	**0.6247 (0.0106)**	0.6097 (0.0145)	0.6062 (0.0114)
GraphDTA (GATGCN)	**0.6305 (0.0063)**	0.6075 (0.0076)	0.5974 (0.0058)
DeepDTA	**0.6244 (0.0008)**	0.6015 (0.0067)	0.6166 (0.0049)
MolTrans	**0.6573 (0.0021)**	0.6389 (0.0094)	0.6420 (0.0020)

Impact of the Design for the Ranking Tasks.

We also conducted ablation studies to analyze the impact of design of the ranking tasks. We compare several ways to generate the ranking tasks for training a DTA model on dataset BindingDB since it recorded the experimental conditions and data sources. Three generation ways of the ranking tasks are designed:

- *Batch* considers both the proteins and the experimental conditions. It is the baseline design of ranking tasks used in the previous experiments.
- *Protein* only considers proteins when constructing ranking tasks. The protein-compound pairs with the same proteins but different experimental conditions are gathered in the same ranking task.
- *Random* randomly generates the ranking tasks. That means a ranking task could contain the protein-compound pairs with different proteins.

Table 4 shows the average CI over 5 runs, comparing the impact of various designs for ranking tasks. We could observe the performance declines for all backbone models under the experimental setting of *Protein* and *Random*, compared with *Batch*. The results suggest that a well-designed method to generate the ranking tasks is one of the keys to accurately learn the order of the candidates.

Conclusions

The task of DTA prediction estimates the affinity scores between the target proteins and candidate compounds. However, the previous pointwise training paradigm suffers from data inconsistency caused by various measurements and experimental batches. In this work, we address the DTA prediction as a ranking problem and construct lots of ranking tasks regarding the measurements as well as the experimental batches. We leverage a pairwise ranking method that blends different kinds of data to take advantage of all the existing data to achieve better performance. We evaluate our proposed paradigm on three DTA datasets. The results from extensive experiments demonstrate that the paradigm based on the data fusion could boost the performance of four backbone models. We also analyze the impact of the design of the ranking tasks to verify the effectiveness of our proposed training framework.

Data availability

The model scripts are available at `https://github.com/PaddlePaddle/PaddleHelix/tree/dev/apps/drug_target_interaction/hybriddta`. All the datasets used in the paper are public resources. Davis is available at `http://staff.cs.utu.fi/~aatapa/data/DrugTarget`, KIBA is available at `https://pubs.acs.org/doi/suppl/10.1021/ci400709d/suppl_file/ci400709d_si_002.xlsx` and BindingDB is available at `https://www.bindingdb.org/bind/index.jsp`. The pre-

processed data can be downloaded from `https://baidu-nlp.bj.bcebos.com/PaddleHelix/datasets/dti_datasets/HybridDTA_data.zip`.

Supporting Information

S1 Data pre-processing

Davis and KIBA.

We follow the previous work [29] to pre-process Davis and KIBA datasets. The details can be found at `https://github.com/hkmztrk/DeepDTA/tree/master/data`.

BindingDB.

The BindingDB dataset with the version of May 2021 contains 2,221,487 compound-protein pairs, including 7,965 protein and 963,425 compounds.

We pre-process the raw BindingDB data following steps below: (1) We keep compound-protein pairs with at least one of the measurements (K_D, K_I, IC_{50} and EC_{50}). (2) We remove the affinity values with '>' or '<'. (3) We modify the extreme affinity values by replacing the values more than 10,000 with 10,000. (4) We drop the duplicates. (5) We define a ranking task by considering the proteins, pH degrees ('pH'), temperatures ('Temp (C)'), and data sources ('Curation/DataSource', 'Article DOI', 'PMID', 'PubChem AID', 'Patent Number', 'Authors', 'Institution). (6) For a protein-compound pair in a ranking task, we keep the median affinity value of that pair. (7) We remove the ranking tasks with no less than ten candidate compounds.

For the backbone model GraphDTA, we conduct additional data cleaning. We remove the illegal SMILES sequence which can not be converted by the Cheminformatics software RDKit (https://rdkit.org) and the single atom sequence (['F', '[SH-]', '[I-]', 'S', 'I', '[F-]']) that can not be converted to a molecular graph. We also remove the protein sequences which do not conform to FASTA format (https://zhanggroup.org/FASTA/).

Besides, instead of using original K_D score as the binding affinity value to make the prediction, we normalize and transform it into pK_D (shown in Equation 7), which is similar to the previous works [12, 15, 26, 29]:

$$pK_D = -log10(\frac{K_D}{10^9}). \tag{7}$$

Note that, K_I, IC_{50} and EC_{50} are normalized by the same way.

S2 Hyper-parameters of HybridDTA

For Davis and KIBA datasets, we use grid search to search the best hyper-parameters for each DTA backbone model on each dataset. The candidate settings of the hyper-parameters for searching are shown in Table 5. We use

Adam optimizer and train 100, 30, 50 epochs for GraphDTA, DeepDTA, and MolTrans, respectively.

For BindingDB, the candidate settings of the hyper-parameters for searching are shown in Table 6. We use Adam optimizer and train 200, 200, 50 epochs for GraphDTA, DeepDTA, and MolTrans, respectively.

Table 5: Candidate settings of hyper-parameters for Davis and KIBA.

Hyper-parameter	Candidate settings
deviation ϵ	0.2
sample times (original dataset)	10
sample times (fused dataset)	{0.5,1,3,5}
learning rate	$\{1 \times 10^{-3}, 5 \times 10^{-4}, 1 \times 10^{-5}\}$
batch size	{32, 256, 512}

Table 6: Candidate settings of hyper-parameters for BindingDB.

Hyper-parameter	Candidate settings
deviation ϵ	0.2
sample times (K_I)	10
sample times (K_D)	{1,3,5}
sample times (IC_{50})	{0.2, 0.5, 1}
learning rate	$\{1 \times 10^{-3}, 5 \times 10^{-4}, 1 \times 10^{-5}\}$
batch size	{32, 256, 512}

References

1. C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96, 2005.

2. Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136, 2007.

3. R. Chen, X. Liu, S. Jin, J. Lin, and J. Liu. Machine learning for drug-target interaction prediction. *Molecules*, 23(9), 2018.

4. A. Cichonska, T. Pahikkala, S. Szedmak, H. Julkunen, A. Airola, M. Heinonen, T. Aittokallio, and J. Rousu. Learning with multiple pairwise kernels for drug bioactivity prediction. *Bioinformatics*, 34(13):i509–i518, 2018.

5. A. Cichonska, B. Ravikumar, E. Parri, S. Timonen, T. Pahikkala, A. Airola, K. Wennerberg, J. Rousu, and T. Aittokallio. Computational-experimental approach to drug-target interaction mapping: a case study on kinase inhibitors. *PLoS computational biology*, 13(8):e1005678, 2017.

6. M. I. Davis, J. P. Hunt, S. Herrgard, P. Ciceri, L. M. Wodicka, G. Pallares, M. Hocker, D. K. Treiber, and P. P. Zarrinkar. Comprehensive analysis of kinase inhibitor selectivity. *Nature biotechnology*, 29(11):1046–1051, 2011.

7. J. L. Durant, B. A. Leland, D. R. Henry, and J. G. Nourse. Reoptimization of MDL keys for use in drug discovery. *J. Chem. Inf. Comput. Sci.*, 42(5):1273–1280, 2002.

8. L. Folkman, B. Stantic, and A. Sattar. Towards sequence-based prediction of mutation-induced stability changes in unseen non-homologous proteins. In *BMC genomics*, volume 15, pages 1–12. Springer, 2014.

9. Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of machine learning research*, 4(Nov):933–969, 2003.

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

11. J. Han and C. Moraga. The influence of the sigmoid function parameters on the speed of backpropagation learning. In *International workshop on artificial neural networks*, pages 195–201. Springer, 1995.

12. T. He, M. Heidemeyer, F. Ban, A. Cherkasov, and M. Ester. Simboost: a read-across approach for predicting drug–target binding affinities using gradient boosting machines. *Journal of cheminformatics*, 9(1):1–14, 2017.

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

14. K. Huang, C. Xiao, L. M. Glass, and J. Sun. Moltrans: Molecular interaction transformer for drug–target interaction prediction. *Bioinformatics*, 37(6):830–836, 2021.

15. M. Jiang, Z. Li, S. Zhang, S. Wang, X. Wang, Q. Yuan, and Z. Wei. Drug–target affinity prediction using graph neural network and contact maps. *RSC Advances*, 10(35):20701–20712, 2020.

16. T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, 2002.

17. A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

18. R. Leaman, R. Islamaj Doğan, and Z. Lu. Dnorm: disease name normalization with pairwise learning to rank. *Bioinformatics*, 29(22):2909–2917, 2013.

19. Y. LeCun, Y. Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

20. I. Lee, J. Keum, and H. Nam. Deepconv-dti: Prediction of drug-target interactions via deep learning with convolution on protein sequences. *PLoS computational biology*, 15(6):e1007129, 2019.

21. B. Liu, J. Chen, and X. Wang. Application of learning to rank to protein remote homology detection. *Bioinformatics*, 31(21):3492–3498, 2015.

22. T. Liu, Y. Lin, X. Wen, R. N. Jorissen, and M. K. Gilson. Bindingdb: a web-accessible database of experimentally determined protein–ligand binding affinities. *Nucleic acids research*, 35(suppl_1):D198–D201, 2007.

23. T.-Y. Liu. Learning to rank for information retrieval. 2011.

24. Y. Luo, X. Zhao, J. Zhou, J. Yang, Y. Zhang, W. Kuang, J. Peng, L. Chen, and J. Zeng. A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information. *Nature Communications*, 8(1):573, 2017.

25. R. Macarron, M. N. Banks, D. Bojanic, D. J. Burns, D. A. Cirovic, T. Garyantes, D. V. S. Green, R. P. Hertzberg, W. P. Janzen, J. W. Paslay, U. Schopfer, and G. S. Sittampalam. Impact of high-throughput screening in biomedical research. *Nature Reviews Drug Discovery*, 10(3):188–195, 2011.

26. T. Nguyen, H. Le, T. P. Quinn, T. Nguyen, T. D. Le, and S. Venkatesh. GraphDTA: predicting drug–target binding affinity with graph neural networks. *Bioinformatics*, 37(8):1140–1147, 10 2020.

27. T. M. Nguyen, T. Nguyen, T. M. Le, and T. Tran. Gefa: early fusion approach in drug-target affinity prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2021.

28. R. Özçelik, H. Öztürk, A. Özgür, and E. Ozkirimli. Chemboost: A chemical language based approach for protein–ligand binding affinity prediction. *Molecular Informatics*, 2020.

29. H. Öztürk, A. Özgür, and E. Ozkirimli. DeepDTA: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, 09 2018.

30. H. Öztürk, E. Ozkirimli, and A. Özgür. Widedta: prediction of drug-target binding affinity. *arXiv preprint arXiv:1902.04166*, 2019.

31. T. Pahikkala, A. Airola, S. Pietilä, S. Shakyawar, A. Szwajda, J. Tang, and T. Aittokallio. Toward more realistic drug–target interaction predictions. *Briefings in bioinformatics*, 16(2):325–337, 2015.

32. S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

33. D. Rogers and M. Hahn. Extended-connectivity fingerprints. *J. Chem. Inf. Model.*, 50(5):742–754, 2010.

34. B. Shin, S. Park, K. Kang, and J. C. Ho. Self-attention based molecule representation for predicting drug-target interaction. In *Machine Learning for Healthcare Conference*, pages 230–248. PMLR, 2019.

35. M. M. Stepniewska-Dziubinska, P. Zielenkiewicz, and P. Siedlecki. Development and evaluation of a deep learning model for protein–ligand binding affinity prediction. *Bioinformatics*, 34(21):3666–3674, 2018.

36. J. Tang, A. Szwajda, S. Shakyawar, T. Xu, P. Hintsanen, K. Wennerberg, and T. 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.

37. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

38. Q. Yuan, J. Gao, D. Wu, S. Zhang, H. Mamitsuka, and S. Zhu. Druge-rank: improving drug–target interaction prediction of new candidate drugs or targets by ensemble learning to rank. *Bioinformatics*, 32(12):i18–i27, 2016.

39. J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.