OXFORD

Briefings in Bioinformatics, 2023, **24(1)**, 1–15
https://doi.org/10.1093/bib/bbac533
Problem Solving Protocol

SAM-DTA: a sequence-agnostic model for drug–target binding affinity prediction

Zhiqiang Hu†, Wenfeng Liu†, Chenbin Zhang†, Jiawen Huang†, Shaoting Zhang, Huiqun Yu, Yi Xiong, Hao Liu, Song Ke and Liang Hong

Corresponding authors. Zhiqiang Hu, SenseTime Research, Shanghai 201103, China. E-mail: huzhiqiang@sensetime.com; Song Ke, Shanghai Matwings Technology Co., Ltd, Shanghai 200240, China. E-mail: song.ke@matwings.com

†Zhiqiang Hu, Wenfeng Liu, Chenbin Zhang and Jiawen Huang contributed equally to this work.

Abstract

Drug–target binding affinity prediction is a fundamental task for drug discovery and has been studied for decades. Most methods follow the canonical paradigm that processes the inputs of the protein (target) and the ligand (drug) separately and then combines them together. In this study we demonstrate, surprisingly, that a model is able to achieve even superior performance without access to any protein-sequence-related information. Instead, a protein is characterized completely by the ligands that it interacts. Specifically, we treat different proteins separately, which are jointly trained in a multi-head manner, so as to learn a robust and universal representation of ligands that is generalizable across proteins. Empirical evidences show that the novel paradigm outperforms its competitive sequence-based counterpart, with the Mean Squared Error (MSE) of 0.4261 versus 0.7612 and the R-Square of 0.7984 versus 0.6570 compared with DeepAffinity. We also investigate the transfer learning scenario where unseen proteins are encountered after the initial training, and the cross-dataset evaluation for prospective studies. The results reveals the robustness of the proposed model in generalizing to unseen proteins as well as in predicting future data. Source codes and data are available at https://github.com/huzqatpku/SAM-DTA.

Keywords: drug–target binding affinity, deep learning, sequence-agnostic model

Introduction

Drug discovery is a cost-intensive and time-consuming project [1, 2] that takes billions of dollars and decades of time to find an effective and safe chemical molecule from the laboratory and brings it to the market. Screening molecules with a high affinity toward the target protein is one of the major focus in early-stage drug discovery [3]. Drug target binding affinity (DTA) measures the strengths of the interaction between the drug–target pair and also sees its application in drug repurposing [4, 5] and off-target side effect warning [6, 7]. However, experimental approaches are confronted with the challenge of extremely large search space of both possible ligands and proteins, heavily rely on large-scale ligand/protein libraries and high-throughput instruments and require great efforts and time [8]. As a result, computational

Zhiqiang Hu received his B.S. and M.S. degree from Peking University, China. He is the head of Precision Medicine Division at Sense Time Research, China. His main research interests include artificial intelligence drug design, bioinformatics and medical imaging analysis.
Wenfeng Liu received the B.S. degree in computer science from Nanchang University, China, in 2020. He is a master at the School of Information Science and Engineering, East China University of Science and Technology. His current research interests include drug-target research and deep learning.
Chenbin Zhang received the B.S. degree from Peking University. He is currently pursuing the Ph.D. degree at Peking University. His current research interests include drug discovery and graph neural network.
Jiawen Huang received the B.S. degree in computer science from Yanshan University, China, in 2020. He is a master at the School of Information Science and Engineering, East China University of Science and Technology. His current research interests include software engineering and deep learning.
Shaoting Zhang received the Ph.D. degree in Computer Science from Rutgers in 2012. He is the deputy head of research at SenseTime, and the head of smart health at Shanghai Artificial Intelligence Laboratory. His current research interests are in deep learning algorithms for medical data analytics. He is on the editorial board of Medical Image Analysis and a senior member of the IEEE.
Huiqun Yu received his B.S. degree from Nanjing University, Nanjing, in 1989, M.S. degree from East China University of Science and Technology (ECUST), Shanghai, in 1992, and Ph.D. degree from Shanghai Jiao Tong University, Shanghai, in 1995, all in computer science. He is currently a professor of computer science with the Department of Computer Science and Engineering at ECUST, Shanghai. From 2001 to 2004, he was a visiting researcher in the School of Computer Science at Florida International University, Miami. His research interests include software engineering, high-confidence computing systems, cloud computing, and formal methods. He is a member of ACM, and a senior member of CCF and IEEE.
Yi Xiong is an associate professor at the School of Life Sciences and Biotechnology, Shanghai Jiao Tong University. His main research interests focus on machine learning algorithms and their applications in the protein sequence-structure-function relationship and biomedicine.
Hao Liu received the B.S., M.S., and Ph.D. in Bioinformatics/Computational Chemistry from Shanghai Jiao Tong University (SJTU), Shanghai, China. Now he joined the Institute of Natural Sciences of SJTU as a postdoctoral fellow. His current research interests include computer-aided drug design and artificial intelligence drug design.
Song Ke received the Ph.D. degree in Pharmacy from the Department of pharmacology and toxicology, University of Vienna, in 2015. He was then a postdoc in computational chemistry at the Institute of Natural Sciences, Shanghai Jiao Tong University. He is currently a research & development project director at Matwings Technology. His main interests are computational chemistry, structure-based compound design, as well as drug discovery and screening.
Liang Hong received the Ph. D. degree in polymer science from the University of Akron. He is a full professor at the Institute of Natural Sciences at Shanghai Jiao Tong University, Shanghai, China. His current research interests are in artificial intelligence drug design, artificial intelligence protein design, dynamics of biomacromolecules, and mechanism of cryopreservation of biomacromolecules.
Received: August 19, 2022. **Revised:** October 5, 2022. **Accepted:** November 7, 2022

methods have been attracting attentions that provides alternatives for the efficiency and economy.

This line of research has been studies for decades, consisting of structure-based, sequence-based and similarity-based methods covering both bipartite interaction classification and real-valued affinity regression tasks. Structure-based methods [3, 9–18] act on 3D structures of proteins, ligands or their complexes. These methods, from the point of methodology, can further be categorized as physics-based (such as molecular docking [11, 18], molecular dynamics simulations [14], etc.) and learning-based [10, 12, 13, 16, 17]. Following the dogma 'structure determines function', structure-based methods access the most comprehensive information, backed by strong physicochemical laws and are highly interpretable for their results. However, experimentally validated protein structures are not always available, while those for complexes are even more scarce [19]. As a result, additional computational methods have to be introduced to firstly predict the structures themselves, leading to increased uncertainty as well as heavy time overhead [15]. Alphafold [20] largely alleviates the problem of structural prediction of singleton proteins, but hard cases remain, and the structural prediction of complexes is still a challenge. In general, this extra step essentially prevents large-scale high-throughput virtual screening, just the motivation at the very beginning.

Sequence-based [21–27] and similarity-based [28–31] methods, however, overcome the aforementioned limitations by simplification of the input as residual sequences, Simplified Molecular-Input Line-Entry System (SMILES) sequences, fingerprint sequences, atom-bond graphs or the derived pairwise similarities. Sequence and similarity information can be processed in parallel with well-developed convolutional neural networks (CNNs), recurrent neural networks (RNNs), graph neural networks (GNNs) and multi-layer perceptrons (MLPs) in modern infrastructure with highly optimized efficiency [32]. More importantly, the information is easier to acquire with lower cost, making it possible for training models with large-scale databases [33, 34]. Along with the development of deep learning techniques as well as the accumulation of large databases, sequence-based and similarity-based models are shown to be largely close to structure-based ones in performance while retaining a high efficiency and wide applicability [23].

One may naturally continue to ask whether an even more compact input is feasible. For example, is it possible for a model to work that is agnostic to protein sequences? The question seems incredible at the first glance. After all, residual sequences have long been seen as the most fundamental information that identifies proteins. However, there have already been findings that semantics of an entity can be equivalently represented by its context apart from its intrinsic characteristics. The most well-known example is the semantics of a linguistic word can be well established by the words around it, in the field of natural language processing [35]. Besides, we would also like to emphasize that the question has practical significance as well. Compared with ligands, proteins have a far larger search-space for their longer lengths, which in turn makes sequence-based models prone to overfitting. As a reference, works like Alphafold [20] utilize large-scale databases for auxiliary information, such as multiple sequence alignments involving multiple species. By contrast, a protein-sequence-agnostic model can focus on the ligand representation and facilitate a universal ligand feature extractor that is more readily to generalize across proteins.

In this study we take a step toward answering this question by empirical demonstration, surprisingly, that a model is fully able to achieve strongly competitive and even superior performance without access to any protein-sequence-related information. Furthermore, the sequence-agnostic model has shown its unique advantage in transfer learning for novel proteins, as well as in predicting future data. The key is to treat different proteins separately but are joint trained in a cooperative multi-head manner, which is fundamentally different from the canonical paradigm that processes the inputs of the protein and the ligand separately and then combines them together. We conduct a thorough investigation and cover as comprehensive as possible the common ligand representations, network architectures and the corresponding hyper-parameters. Figure 1 illustrates the overview of the proposed sequence-agnostic framework (SAM-DTA) including singleton models as well as the multi-head training scheme.

Materials and methods

Dataset

The molecular data used in this study are derived from the BindingDB database [34]. Ligands are firstly represented in the SMILES format, and can be converted to the atom-bond graph format using RDKit [36]. For the label, the negative logarithm form of the half maximal inhibitory concentration (IC50), $\mathbf{pIC50} = -\log_{10}\mathbf{IC50}\,(\mathbf{Molar})$, is used as the measure of binding affinity. Prior to taking the logarithm, IC50 is truncated to the range of $[10^{-11}\mathbf{Molar}, 10^{-2}\mathbf{Molar}]$ [23]. Note also that the method can be easily extended to other measures such as $\mathbf{K_i}$, $\mathbf{K_d}$, **EC50**, etc. We build four datasets based on the BindingDB database according to protein families and database timestamps as detailed next.

We start from the dataset curated by Karimi *et al.* [23] from BindingDB at year 2018. Briefly, SMILES strings are retrieved from PubChem [37] using their PubChem CIDs; protein–ligand pairs are filtered to remove samples outside of predefined length ranges, with incomplete information, or with IC50 as a range instead of exact value; multiple IC50 measurements are normalized by geometric mean [23]. On that basis, we group protein–ligand pairs by proteins and remove those groups with <200 samples, since too few samples are unable to give statistically reliable and confident evaluations. Among all the remaining samples, four classes of proteins are withheld for the transfer learning scenario, including nuclear estrogen receptors (ER), ion channels, receptor tyrosine kinases and G-protein-coupled receptors (GPCR), collectively denoted as BindingDB-18ex, whereas the rest denoted as BindingDB-18. Specifically, BindingDB-18 contains 401 proteins/291,504 samples in total and BindingDB-18ex contains another 129 proteins/91,767 samples. For each protein, we random divide the samples into the trainset, valset and testset with the ratio of 7:1:2. More details about the dataset, including the statistics of min, max and quartiles, as well as the histogram, for the distributions of the length of ligand SMILES, the length of proteins, the number of ligands per protein and the number of proteins per ligand, are given in the supplementary materials.

Besides BindingDB-18 and BindingDB-18ex, we build another two datasets from BindingDB at year 2021 and 2022 for prospective studies, denoted as BindingDB-21 and BindingDB-22, respectively. Specifically, we take the officially released files 'BindingDB_All_2D_2021m5.sdf.zip' and 'BindingDB_All_2D_2022m5.sdf.zip', and select samples with the protein in the set of 401 proteins of BindingDB-18. Similar curation steps are also performed as [23] including retrieval of SMILES strings, filtration of ligands with a SMILES string longer than 100 (only 0.67%) and samples with IC50 as a range instead of exact value, and normalization of multiple IC50 measurements by geometric

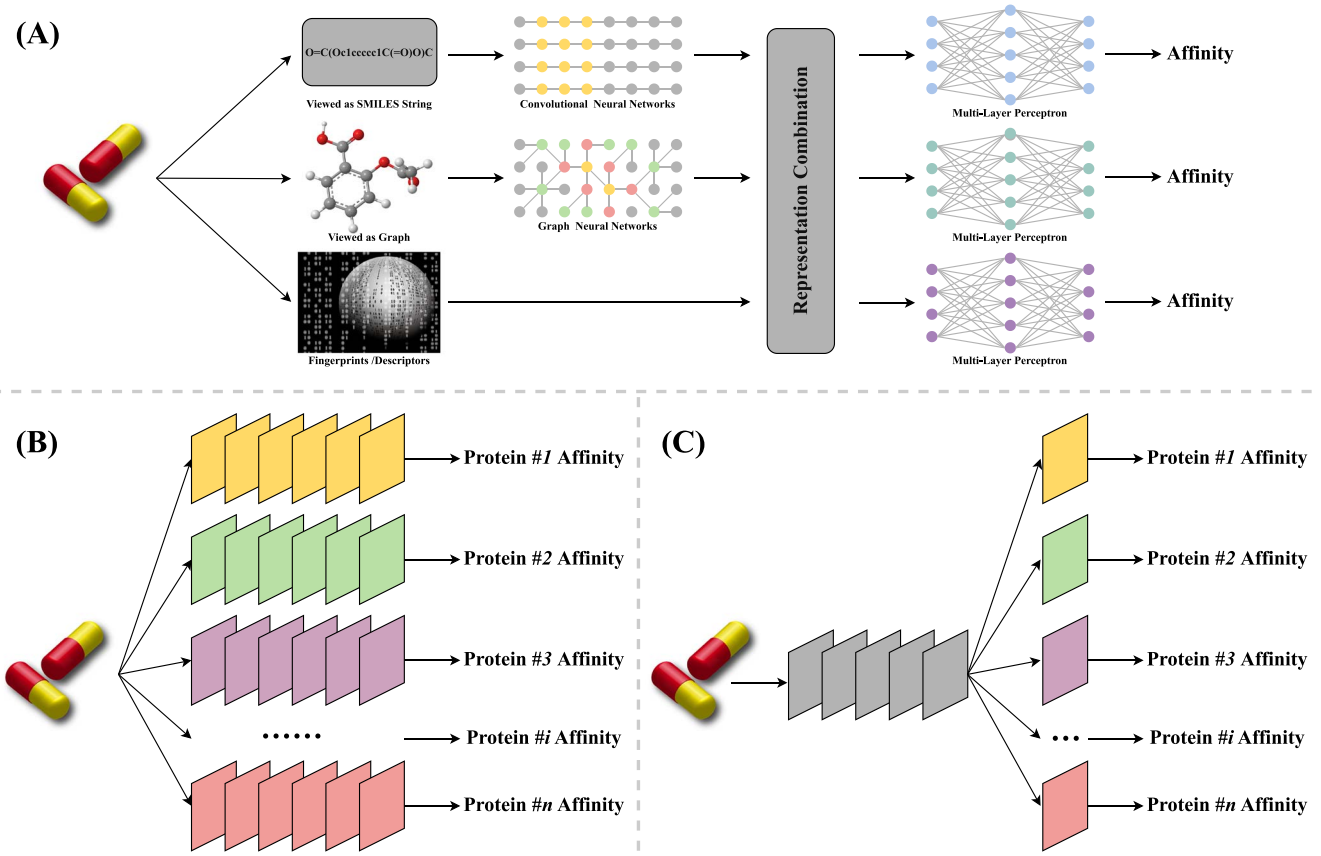

Figure 1. Overview of SAM-DTA, a sequence-agnostic model for drug–target binding affinity prediction. (A) Different kinds of ligand representations and their combinations are evaluated, including SMILES representation, graph-based representation, descriptor representation and fingerprint representation. Different models are designed so as to conform to different input representations. (B) Singleton models that train a standalone model for each protein separately. (C) The multi-head scheme where all proteins are jointly trained by sharing some structures among the heads.

mean. Finally, we remove samples that are already presented in BindingDB-18. In the end, BindingDB-21 and BindingDB-22 contain 180,822 and 232,186 samples in total, respectively. As a result, BindingDB-21 and BindingDB-22 essentially encompass future data for proteins in BindingDB-18. Note that BindingDB-21 is a subset of BindingDB-22.

Model

Input representation

As mentioned above, sequence-related information of proteins is not accessible to our model. The input representations of ligands are elaborated next.

Different kinds of ligand representations and their combinations are compared in this study. We start from the SMILES format of ligands. SMILES are viewed as strings that can be processed by character-based 1D CNNs. The alphabet of 67 SMILES characters is extended with the special START token and STOP token that marks the two ends of the string, and the PADDING token to address the varied lengths. As a result, each character in a SMILES string is represented as a 70-dimensional one-hot vector.

Besides the SMILES format, ligands are also converted to graphs using RDKit [36] with atoms as the nodes and bonds as the edges, so as to be processed by GNNs. The initial atom feature contains the information including the element and degree of the atom, the number of attached hydrogen atoms, the implicit valence, an aromaticity indicator and so on, collectively represented as a 75-dimensional vector, following the work from [38].

Finally, the descriptor and fingerprint representations of ligands are also extracted and evaluated. Specifically, the descriptor representation includes Chi0, Chi1, Chi0n–Chi4n, Chi0v–Chi4v; the number of H-acceptors, H-donors, heteroatoms, rotatable bonds, valence electrons, amide bonds, aromatic/saturated/aliphatic rings or cycles; Molecular Operating Environment (MOE)-type descriptors using either partial charges and surface area contributions, Molar Refractivity (MR) contributions and surface area contributions, LogP contributions and surface area contributions, EState indices and surface area contributions; and so on, in a total of 151 descriptors. The detailed list is given in the supplementary materials. For the fingerprint representation, we investigate four kinds of fingerprints including MACCS Keys, RDKit fingerprints (topological fingerprints), Morgan fingerprints and Avalon fingerprints. The dimensions of resulting vectors are 167, 1024, 1024, 512, respectively, and are concatenated to form a 2727-dimensional vector for each ligand. The descriptor and fingerprint representations are extracted using RDKit [36].

Singleton model

A natural way to the sequence-agnostic paradigm is to train a standalone model for each protein separately. Note that different network architectures should be, respectively, designed so as to conform to different input representations.

For the SMILES-based ligand representation, we build a character-level 1D CNN. The input SMILES string is firstly inserted with the START and STOP token at the two ends, and padded to the maximal length with the PADDING token, to address the issue

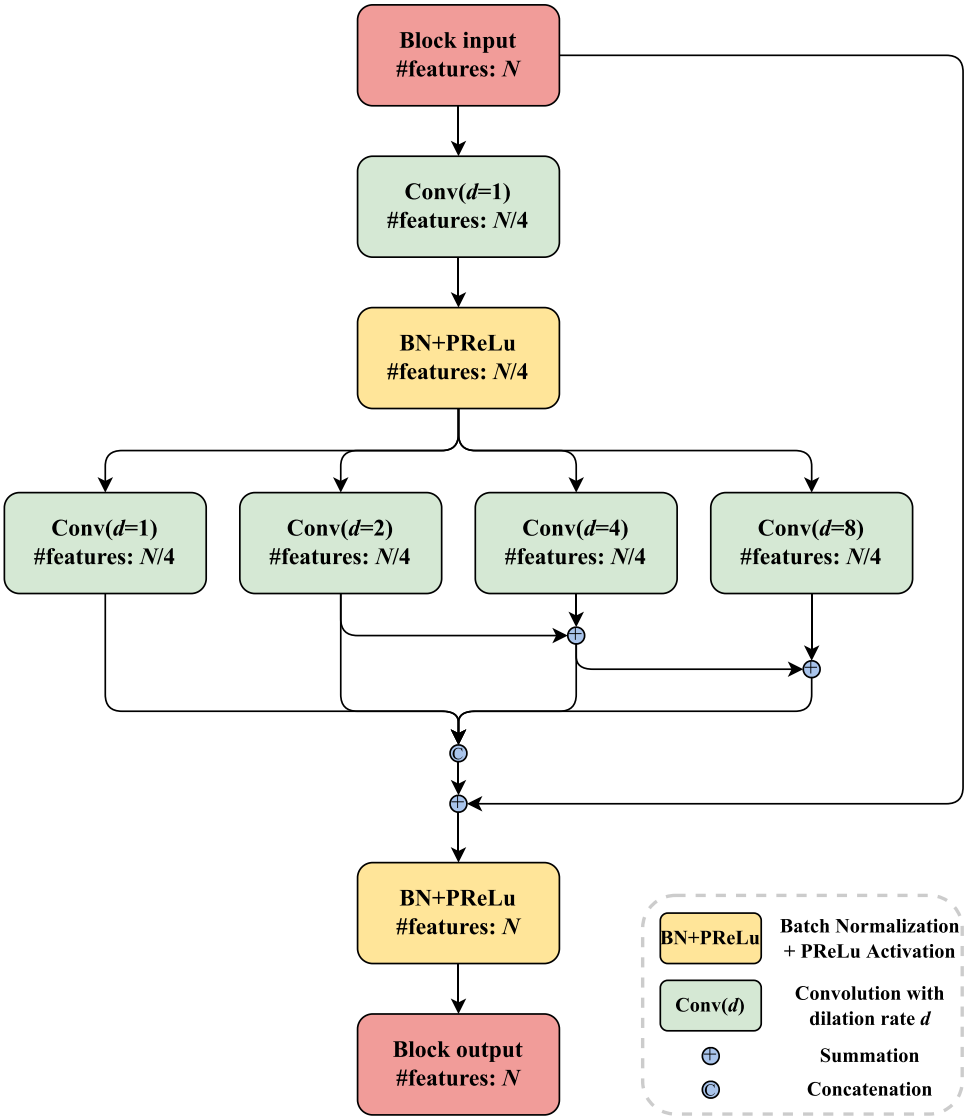

Figure 2. The architecture of the dilated parallel residual block. The block assembles multi-scale features from 1D convolutional layers with different dilation rates, equipped with the residual link.

of varied lengths. After that, each character in the padded string is represented as a 70-dimensional one-hot vector and embedded to a $w/2$-dimensional real-valued vector with the embedding layer. A stack of n dilated parallel residual blocks [39] with w output features is further applied on top of the embedding layer. As illustrated in Figure 2, the block follows the "division-processing-fusion" pipeline that assembles multi-scale features from 1D convolutional layers with different dilation rates, equipped with the residual link [40]. The resulting feature is aggregated in the character dimension by a global average pooling layer, to arrive at an overall SMILES-based representation for the ligand. Note that this representation can be optionally combined with the descriptor or fingerprint representation by concatenation to explore the complementariness of such representations. Finally, the representation, whether SMILES-based or combined with other representation, is processed by a multi-layer perceptron (MLP) module to give the affinity prediction. The MLP is composed of a stack of two fully connected (FC) layers with ReLU and dropout layers in-between. The first FC layer outputs features of dimension h and the second outputs a real-value for affinity prediction for the protein. Different values of w, n and h, as well as other CNN variants including ResNet [40] and DenseNet [41], and RNNs including LSTM [42] and GRU [43], are investigated in experiments.

For the graph-based ligand representation, GNNs are developed to progressively refine the atom features with the information from the neighbors. Given initial atom features described above, GNNs can be generally formulated as

$$x_i^{l+1} = \text{Update}\left(x_i^l, \text{Aggregate}_{j\in\mathcal{N}(i)}\left(\text{Message}(x_i^l, x_j^l)\right)\right) \quad (1)$$

where x_i^l denotes the feature of i-th atom after l GNN layers and x_i^0 the initial atom feature, $\mathcal{N}(i)$ denotes the neighbors of i-th atom, i.e. the atoms that have a chemical bond with i-th atom. 'Update', 'Aggregate' and 'Message' are three differentiable functions that define the particular GNN architectural variant, and the 'Aggregate' function is also required to be invariant to atom permutations. In this study we perform a comprehensive comparison of different GNN architectures including GCN [44], GraphSAGE [45], Set2SetNet [46], GlobalAttentionNet [47], SAGPool [48, 49], TopK [48, 50], SortPool [51], JumpingKnowledge [52], Graclus [53]. After a total of L GNN layers with W output features each, the resulting atom features are also aggregated by average across

atoms to give the overall graph-based ligand representation. Finally, this representation is similarly processed by an MLP module that outputs the affinity prediction. For a fair comparison, the MLP is consistent with the one used in the SMILES-based CNN, with hidden size 3072; values of $L = 3$ and $W = 1024$ are also chosen to keep similar capacity with the SMILES-based CNN.

Multi-head model

Singleton models are unable to take full advantage of data from other proteins. This has two consequences: feature extractor loses the opportunity to learn more generic patterns for better performance; every time a novel protein is met, the model has to be trained totally from scratch. It is thus less optimal either from the perspective of performance or usability. As a result, we further develop a multi-head scheme with the aforementioned singleton models as the building blocks.

The key idea of the multi-head scheme is to enable across-head mutual boosting by *sharing* some structures among the heads. In this study, all but the last FC layer is shared among the heads. Despite a seemingly minor modification, the multi-head scheme has a far different indication from the singleton manner. Unlike singleton models each having its own ligand representation space, the multi-head scheme has a common ligand representation space. Consequently, the learned ligand representation is universal, meaning that it is applicable across proteins and can readily be seen as an independent attribute of the ligand that does not depend on any specific protein.

Training and evaluation

All the aforementioned models are trained with the mean squared error (MSE) loss function,

$$\text{Loss} = \frac{1}{B}\sum_{i=1}^{B}\left(y_i - \hat{y}_i\right)^2 \tag{2}$$

where y_i and $\hat{y}_i$ are observed and predicted pIC50 values, respectively; B is the batchsize and set to 10. For singleton models, each model is trained independently, but the hyper-parameters are kept the same to avoid the *ad hoc* design; in the multi-head scheme, however, one unified model is trained for all the proteins: for each round we iterate through all the proteins in order, group every 64 proteins (except the last containing all the rest) and sum the losses for one optimization step (due to GPU memory constraints, it is not feasible to conduct one optimization step for all the proteins each with batchsize $B = 10$). The optimizer is Adam [54] and weight decay is set to 0.0001. All models are implemented using the PyTorch framework [32] and GNNs are additionally using the PyTorch Geometric library [55].

To evaluate the performance of the models, the predicted pIC50 values are compared with the observed values using both the MSE and the Pearson correlation coefficient (R^2) metrics,

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2 \tag{3}$$

$$R^2 = \frac{\left(\sum_{i=1}^{N}(y_i - m_y)(\hat{y}_i - m_{\hat{y}})\right)^2}{\sum_{i=1}^{N}(y_i - m_y)^2 \sum_{i=1}^{N}(\hat{y}_i - m_{\hat{y}})^2}, \tag{4}$$

where y_i and $\hat{y}_i$ are observed and predicted pIC50 values, respectively; N is the total number of samples; $m_y = \sum_{i=1}^{N} y_i/N$ and $m_{\hat{y}} = \sum_{i=1}^{N} \hat{y}_i/N$ are the mean value of y_i and $\hat{y}_i$, respectively. Note that all samples are aggregated together for evaluation (i.e. micro-average), regardless of whether for singleton models or in the multi-head scheme.

Results and discussion

Comparison with competing methods

We compare the proposed SAM-DTA model with six representative methods from three aspects, including the strong sequence-based counterpart DeepAffinity [23], GraphDTA [56], MDeePred [57], MGraphDTA [58] from literature, the head-to-head variant Sequence-Aware and the singleton version of SAM-DTA. DeepAffinity unifies RNN and CNN with attention mechanism for the ligand and protein feature extraction, and the features are combined by concatenation for affinity prediction, in an end-to-end manner. Notably, the feature extractors of both ligands and proteins are pretrained in large databases. We therefore also take the pretrained parameters and finetune on our dataset with the officially released codes. GraphDTA, on the other hand, utilizes the GNN to extract the feature of ligands, which is also combined with the protein feature extracted by the CNN for affinity prediction. MGraphDTA further extends the architecture of GraphDTA by their proposed multi-scale GNN and CNN, aiming to capture the local and global structure of ligands. Finally, MDeePred focuses their attention on the protein featurization and incorporates multiple types of protein features such as sequence, structural, evolutionary and physicochemical properties, jointly processed by a CNN and concatenated with the ligand feature by a MLP on top of molecular fingerprints for affinity prediction. In summary, the four competing methods from literature are able to cover commonly utilized architectures as well as a variety of featurizations.

Besides the off-the-shelf methods from literature, we also manually equip the proposed SAM-DTA with a dedicated branch for protein feature extraction, the resulting feature of which is concatenated with ligand features for affinity prediction. Note that the structure of the protein branch mimics that of the ligand, with the same number of dilated parallel residual blocks (Figure 2). In this way, we perform a head-to-head comparison between sequence-based and sequence-agnostic methods. This variant is therefore called Sequence-Aware hereinafter.

Finally, we also compare SAM-DTA (multi-head) with its singleton version. Singleton models are trained independently, and the predictions are assembled for evaluation. Based on the ablation studies to be described below, we choose the ligand representation combinations of SMILES and fingerprints, processed by character-level 1D CNN followed by MLP. The architecture is kept the same for both the singleton version and the multi-head version for a fair comparison.

All the models are trained and evaluated on the BindingDB-18 dataset described before, with the same trainset/valset/testset split. Table 1 presents the results, where the standard error is estimated by bootstrapping over 10 times. It can be seen that sequence-agnostic methods, either singleton or multi-head, outperform the sequence-based counterpart, which demonstrates the effectiveness of the proposed sequence-agnostic scheme. Note also that Sequence-Aware performs far worse than DeepAffinity, GraphDTA, MDeePred and MGraphDTA, which rules out the possibility that the performance superiority is derived from the particular network architecture or ligand representation. Within the sequence-agnostic family; however, the multi-head way gives even more accurate predictions than that of singleton. Given that singleton models collectively contain more parameters and thus

Table 1. Performance comparison of the proposed sequence-agnostic model (SAM-DTA) against six representative methods from different aspects, including the strong sequence-based counterpart DeepAffinity, GraphDTA, MDeePred, MGraphDTA from literature, the head-to-head variant Sequence-Aware and the singleton version of SAM-DTA. The standard error is estimated by bootstrapping over 10 times

Model	Trainset MSE	Trainset R^2	Valset MSE	Valset R^2	Testset MSE	Testset R^2
Sequence-Aware	1.0383 ± 0.0020	0.5134 ± 0.0015	1.1225 ± 0.0093	0.4744 ± 0.0033	1.0958 ± 0.0057	0.4805 ± 0.0036
DeepAffinity	0.2945 ± 0.0013	0.8670 ± 0.0005	0.7905 ± 0.0088	0.6490 ± 0.0032	0.7612 ± 0.0064	0.6570 ± 0.0026
GraghDTA	0.1870 ± 0.0005	0.9570 ± 0.0002	0.5404 ± 0.0049	0.8635 ± 0.0028	0.5413 ± 0.0031	0.7417 ± 0.0023
MDeePred	0.1522 ± 0.0006	0.9297 ± 0.0003	0.5378 ± 0.0057	0.7502 ± 0.0023	0.5393 ± 0.0040	0.7471 ± 0.0017
MGraphDTA	0.0288 ± 0.0002	0.9566 ± 0.0001	0.5283 ± 0.0072	0.7229 ± 0.0040	0.5312 ± 0.0038	0.7116 ± 0.0018
Sequence-Agnostic (singleton)	0.0010 ± 0.0000	0.9995 ± 0.0000	0.4633 ± 0.0056	0.7826 ± 0.0022	0.4678 ± 0.0053	0.7781 ± 0.0023
Sequence-Agnostic (multi-head)	0.0055 ± 0.0000	0.9980 ± 0.0000	0.4233 ± 0.0042	0.8019 ± 0.0026	0.4261 ± 0.0032	0.7984 ± 0.0019

have better representation capacity (also shown by lower trainset errors), the result indicates that parameter-sharing in the multi-head training indeed realizes the across-head mutual boosting and learns robust and generalizable ligand features for different proteins.

We also illustrate best-performing and worst-performing predictions as shown in Figure 3(A–B) and (C–D), respectively. Our model demonstrates the ability of handling large value variations (Figure 3A) and non-normal value distributions (Figure 3B) but also maintains reasonable performance in worst cases (Figure 3C and B). One may wonder whether performance has a correlation to the sample capacity of the protein. To this end, we collect the MSE and R-Square value for each individual protein with respect to the number of ligands in the dataset for that protein, as shown in Figure 4A and B. However, the correlations are not significant, as proteins with most ligand samples (thus top sample capacities) are not necessarily the best-performing ones.

Comparison of different ligand representations

We next perform ablation studies to investigate the best settings for the proposed model. Firstly, different ligand representations and their combinations are compared. As shown in Table 2, we compare the SMILES-based and graph-based representations, optionally combined with the fingerprint and descriptor features. Note that different representations are processed by corresponding models, as elaborated in the Model Section.

It can be observed from Table 2 that the SMILES-based representation performs consistently better than the graph-based ones, when processed by RNNs or the dilated parallel residual CNN. We explore multiple GNN variants including GCN [44], GraphSAGE [45], Set2SetNet [46], GlobalAttentionNet [47], SAGPool [48, 49], TopK [48, 50], SortPool [51], JumpingKnowledge [52], Graclus [53]. These architectures differ in the message passing mechanism as defining different 'Update', 'Aggregate' and 'Message' functions. However, performances are comparable, indicating that it is not sensitive to the specific GNN architecture. Considering that GNNs have been shown the effectiveness in plenty of drug-related tasks including ADMET prediction [59–61] and also the sequence-based models of affinity prediction [62, 63], we suspect the performance gap between GNNs and CNNs may result from their different ability in handing large number of multi-heads (e.g. 401 proteins in this study).

With the SMILES-based representation, however, the dilated parallel residual CNN outperforms both other CNN variants as well as RNN architectures. Note also that the two RNN models (LSTM [42] and GRU [43]) perform consistently better than commonly used CNN structures including ResNet [40] and DenseNet [41], showing the advantage of RNNs in modeling long-term correlations. This is also what is explored by the multi-scale information fusion in the dilated parallel residual CNN. Compared with other variants, the dilated parallel residual CNN integrates the long-term modeling ability of RNNs with the optimization advantage of CNNs and demonstrates superior performance with both lower training errors and better testset generalizability.

It is also worthy noting that combining fingerprint or descriptor representations is beneficial to the SMILES-based representation. Moreover, the fingerprint representation offers more improvements than descriptors, indicating that fingerprints contain more information that is complementary to that extracted by deep neural networks. However, the benefits are addictive on top of the other. As a result, practitioners may make their own tradeoffs between performance and efficiency in the selection and combination of ligand representations. In this work, we choose the SMILES-based representation combined with fingerprints only, saving the inference time for computing the descriptors though with the performance slightly lower than the optimum.

Comparison of different architectures

We then focus on the SMILES-based ligand representation and further compare the architectural settings related to both the CNN architecture as well as the following MLP module. These hyper-parameters are searched under the SMILES representation alone and directly applied to the SMILES+fingerprint setting.

For the CNN architecture, different net depth and width are investigated. Specifically, we compare different number of dilated parallel residual blocks (the depth, n) as well as that of output features in each of the blocks (the width, w). Table 3 shows the experimental results. For a fixed net depth, it has been observed clear improvements along with the expanded net width. When the net width is small, the model exhibits a state of underfitting, with relatively high errors even in the trainset; however, training losses are gradually reduced along with the expanded net width, which contributes the most to the improved performance. On the other hand, improvements are not significant when net depth is increased. We speculate that although a deeper network brings more powerful representation capacity, it will also encounter more severe optimization issues (e.g. gradient diminishing). As a result, in terms of the sequence-agnostic affinity prediction task, a wider network should be preferred than a deeper one. In this study, we finally choose a net width of $w = 1024$ with a medium net depth of $d = 3$ blocks.

We also explore the effect of hidden size (h) of the MLP module on the performance. We start from the hidden size of 64 and gradually increase up to 4096 that is four times the output feature size of the preceding CNN module. As shown in Table 4, a larger hidden size immediately leads to a less training error, whereas testset performance is first rising and then reaches the plateau. In

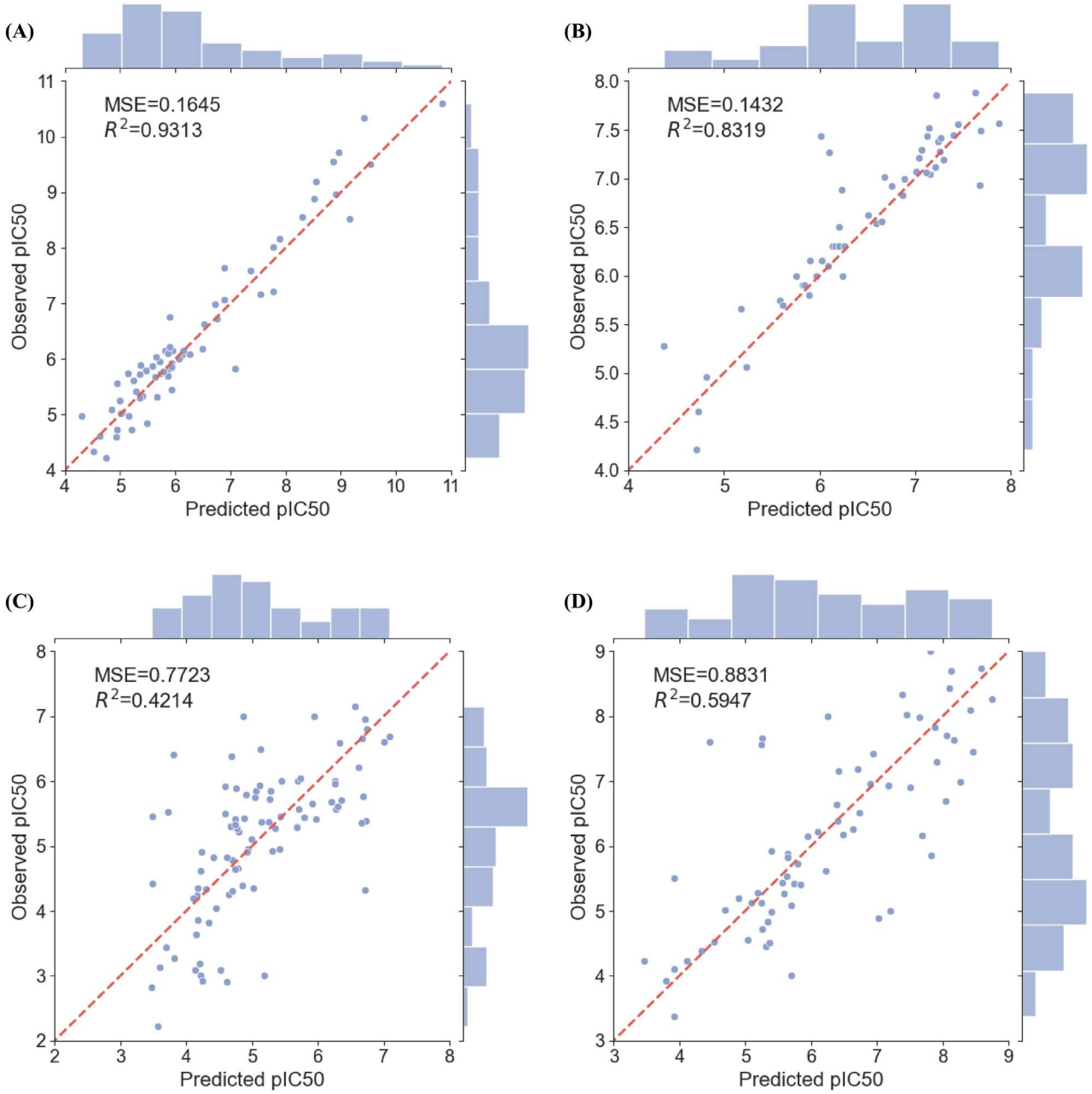

Figure 3. Scatter plots of best-performing [(**A**) THRB, UniProt ID: P10828, #139/401 and (**B**) BRD3, UniProt ID: Q15059, #369/401] and worst-performance [(**C**) IDO1, UniProt ID: P14902, #229/401 and [(**D**) Adora1, UniProt ID: P25099, #278/401] proteins by SAM-DTA. Each dot represents a protein–ligand pair with x-axis the predicted pIC50 and y-axis the observed pIC50; the red dash line represents the ideal $y=x$ relation.

fact, training loss is already low enough that the model has arrived at a state of sufficient fitting, when the hidden size is larger than 512. This, from another side, reveals the representation power of the preceding CNN module that relieve the burden upon the MLP module. The performance has thereby saturated.

Investigation of the transfer learning scenario

Until now we discuss the scenario where the concerned proteins are known beforehand, and models are trained on the corresponding data followed by the immediate testing. However, there are also circumstances that proteins of interest come up sequentially that cannot be perfectly planned at the very beginning. For example, when the model is trained on an initial database and shared in the community, one may take the model and utilize in his/her own projects. This scenario, which we call the transfer learning scenario, is also investigated in this study. Note that the main difference and also the challenge lies at the limitation that the preceding database is not, but only the trained model is accessible to the later phase, which is often the actual conditions in real practice. For the sake of simplification, we consider the case that only one transfer learning phase occur after the initial learning phase. More transfer learning phases can be similarly processed in a one-by-one manner.

To tackle the task, two strategies are investigated and compared. The first one is to view the given model as pretrained parameters, equip it with randomly-initialized last layer, and finetune the whole model on the new data. We also compare to finetune independently for each protein (singleton) against collectively in a multi-head manner. Another way, however, regards

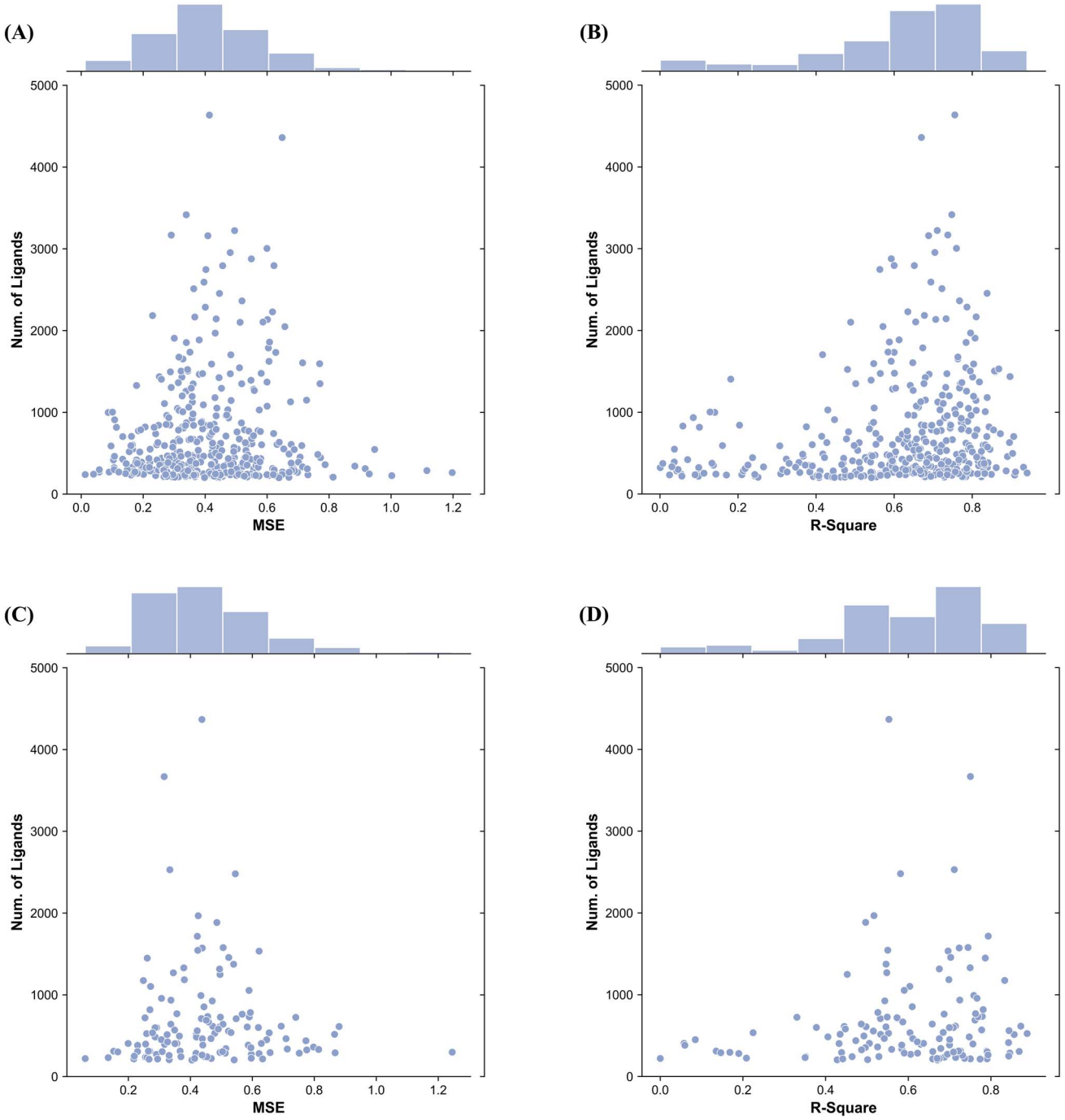

Figure 4. Correlation analysis of the per-protein performance with respect to its sample capacity, in the regular training scenario (A) and (B) and the transfer learning scenario (C) and (D). Each dot represents a protein. Performance is assessed by MSE(A) and (C) and Pearson correlation coefficient [R^2, (B) and (D)] in x-axis, respectively, while sample capacity is measured by the number of ligands for that protein.

the given model as feature extractor, and gets the universal ligand feature from outputs of the last-but-one layer, which is then utilized to predict for the novel proteins, with regressors including MLP, random forest and RBF-kernel support vector machine regressor (RBF-SVR). For comparison, training from scratch is also conducted to serve as the baseline.

Table 5 presents the results. For the same strategy, the proposed sequence-agnostic model, in either singleton or multi-head manner, performs better than Sequence-Aware, DeepAffinity [23], GraphDTA [56], MDeePred [58] and MGraphDTA [57] with a clear margin, again confirming the effectiveness of the sequence-agnostic scheme, while the multi-head way consistently outperforms the singleton one. We also analyze the performance in detail by violin plots across proteins as shown in Figure 5A–C at initial training (401 proteins), training from scratch (129 proteins) and finetuning (129 proteins), respectively. Specifically, we group protein–ligand pairs by their proteins, and then for each protein calculate the MSE for that protein, the results of which are aggregated and finally the violin plot is drawn by the seaborn [64] library. Clear margins are observed for centers of violin plots (the

Table 2. Performance comparison of different ligand representations and their combinations

Representation	Network	+Descriptors?	+Fingerprints?	Trainset MSE	Trainset R^2	Valset MSE	Valset R^2	Testset MSE	Testset R^2
Graph	GCN			0.5097	0.7599	0.6933	0.6730	0.6831	0.6735
	GraphSAGE			0.5094	0.7612	0.6739	0.6825	0.6622	0.6838
	Set2SetNet			0.0754	0.9646	0.6587	0.6954	0.6619	0.6906
	GlobalAttentionNet			0.4781	0.7746	0.6543	0.6914	0.6517	0.6886
	SAGPool			0.4694	0.7795	0.6576	0.6899	0.6414	0.6935
	TopK			0.4708	0.7792	0.6521	0.6926	0.6394	0.6946
	SortPool			0.1792	0.9174	0.649	0.694	0.6357	0.6964
	JumpingKnowledge			0.4746	0.7777	0.6462	0.6955	0.6344	0.6971
	Graclus			0.4416	0.7928	0.6212	0.7072	0.6099	0.7086
SMILES	ResNet			0.0262	0.9878	0.6915	0.6755	0.6809	0.6767
	DenseNet			0.0429	0.9801	0.6675	0.6873	0.6568	0.6881
	GRU			0.0321	0.9911	0.5540	0.7506	0.5533	0.7487
	LSTM			0.0210	0.9917	0.5433	0.7496	0.5454	0.7467
	Dilated Parallel Residual	✓		0.0101	0.9952	0.4974	0.7663	0.4975	0.7636
	CNN		✓	0.0081	0.9962	0.4802	0.7749	0.4880	0.7691
		✓	✓	0.0055	0.9980	0.4233	0.8019	0.4261	0.7984
				0.0054	0.9978	0.4213	0.8027	0.4231	0.7996

Table 3. Performance comparison of different net depth and width for the CNN architecture under the SMILES ligand representation. The depth is measured by the number of dilated parallel residual blocks, whereas the width is measured by the number of output features for the block

Representation	Net depth	Net width	Trainset MSE	Trainset R^2	Valset MSE	Valset R^2	Testset MSE	Testset R^2
SMILES	2 blocks	256	0.0870	0.9609	0.6421	0.7091	0.6413	0.7065
		512	0.0338	0.9841	0.5736	0.7352	0.5770	0.7308
		1024	0.0136	0.9936	0.5144	0.7593	0.5181	0.7551
	3 blocks	256	0.0772	0.9666	0.6459	0.7095	0.6438	0.7077
		512	0.0247	0.9885	0.5544	0.7424	0.5625	0.7362
		1024	0.0101	0.9952	0.4974	0.7663	0.4975	0.7636
	4 blocks	256	0.0660	0.9703	0.6447	0.7066	0.6409	0.7058
		512	0.0201	0.9906	0.5468	0.7448	0.5489	0.7413
		1024	0.0096	0.9955	0.4852	0.7715	0.4845	0.7691

Table 4. Performance comparison of different hidden sizes for the MLP module

Representation	MLP hidden size	Trainset MSE	Trainset R^2	Valset MSE	Valset R^2	Testset MSE	Testset R^2
SMILES	64	0.0477	0.9788	0.5463	0.7425	0.5408	0.7419
	128	0.0331	0.9856	0.5297	0.7507	0.5201	0.7523
	256	0.0226	0.9898	0.5102	0.7598	0.5134	0.7553
	512	0.0176	0.9918	0.5055	0.7617	0.5034	0.7599
	1024	0.0151	0.9930	0.4976	0.7656	0.4957	0.7637
	2048	0.0120	0.9943	0.4955	0.7668	0.4970	0.7634
	3072	0.0101	0.9952	0.4974	0.7663	0.4975	0.7636
	4096	0.0088	0.9959	0.4934	0.7683	0.4984	0.7634

median) across the seven models, indicating that the performance gap is structural but not affected by outliers or corner cases.

Across the strategies, on the other hand, the phenomenon is different for different methods. The proposed SAM-DTA enjoys the benefits from the pretrained parameters while DeepAffinity and Sequence-Aware do not (also by comparison of Figure 5B and C). This essentially reveals that incorporating sequence information does not necessarily help generalize to novel proteins, but a universal ligand feature does.

Similar to the regular training scenario, we also investigate the correlation of the per-protein performance with respect to the number of ligands for that protein, i.e. the sample capacity, in the transfer learning scenario. As shown in Figure 4C and D), also similar to that in the regular training scenario, no significant

Table 5. Performance comparison in the transfer learning scenario of the proposed sequence-agnostic model (SAM-DTA) against six competing methods, with strategies of training from scratch, finetuning or as feature extractor with different following regressors

Strategy	Model	Trainset MSE	Trainset R^2	Valset MSE	Valset R^2	Testset MSE	Testset R^2
Training from scratch	Sequence-Aware	0.4602	0.7306	0.7323	0.5615	0.7403	0.5609
	DeepAffinity	0.3402	0.8110	0.8277	0.5280	0.8170	0.5420
	GraghDTA	0.1358	0.9605	0.5377	0.8237	0.5382	0.6820
	MDeePred	0.1556	0.9155	0.5644	0.6694	0.5573	0.6757
	MGraphDTA	0.0164	0.9477	0.5601	0.6243	0.5661	0.6216
	Sequence-Agnostic (singleton)	0.0016	0.9991	0.4848	0.7107	0.4883	0.7129
	Sequence-Agnostic (multi-head)	0.0063	0.9973	0.4532	0.7298	0.4618	0.7291
Pretrained parameters & finetuning	Sequence-Aware	0.7363	0.5952	0.8842	0.4964	0.8778	0.5023
	DeepAffinity	0.8805	0.4740	0.9402	0.4340	0.9140	0.4570
	GraghDTA	0.1068	0.9707	0.5228	0.8285	0.5285	0.6865
	MDeePred	0.1092	0.9394	0.5241	0.6901	0.5187	0.6965
	MGraphDTA	0.0125	0.9462	0.5236	0.6673	0.5192	0.6740
	Sequence-Agnostic (singleton)	0.0006	0.9997	0.4775	0.7140	0.4812	0.7158
	Sequence-Agnostic (multi-head)	0.0044	0.9979	0.4471	0.7320	0.4486	0.7348
Feature extractor & MLP	Sequence-Aware	1.1814	0.2995	1.2268	0.2722	1.2239	0.2819
	DeepAffinity	1.3157	0.2708	1.3532	0.2515	1.3384	0.2648
	GraghDTA	0.0216	0.9871	1.3731	0.3718	1.3701	0.3739
	MDeePred	0.1065	0.9365	1.6693	0.2307	1.6318	0.2372
	MGraphDTA	0.7684	0.5433	1.2549	0.2972	1.2226	0.3176
	Sequence-Agnostic (singleton)	0.0227	0.9865	0.5715	0.6693	0.5889	0.6653
Feature extractor & random forest	Sequence-Aware	0.1760	0.9158	1.1002	0.3491	1.0753	0.3694
	DeepAffinity	0.1282	0.9473	0.8987	0.4588	0.8928	0.4698
	GraghDTA	0.1108	0.9552	0.7775	0.5348	0.7733	0.5444
	MDeePred	0.1353	0.9483	0.9537	0.4267	0.9470	0.4385
	MGraphDTA	0.1426	0.9438	1.0258	0.3825	1.0218	0.3928
	Sequence-Agnostic (singleton)	0.0803	0.9658	0.5697	0.6646	0.5693	0.6701
Feature extractor & RBF-SVR	Sequence-Aware	1.1340	0.3262	1.1648	0.3049	1.1530	0.3190
	DeepAffinity	1.1006	0.3430	1.1233	0.3244	1.1069	0.3425
	GraghDTA	0.4606	0.7330	0.7519	0.5473	0.7484	0.5557
	MDeePred	0.7243	0.5715	1.0369	0.3788	1.0349	0.3873
	MGraphDTA	1.0114	0.3968	1.1001	0.3417	1.0922	0.3543
	Sequence-Agnostic (singleton)	0.3117	0.8205	0.5187	0.6895	0.5192	0.6932

correlations is observed, again confirming that sample capacity is not the main factor that affects per-protein performance.

For singleton learning, i.e. assuming only one protein in each transfer learning phase, we see that the finetuning strategy performs slightly better than 'feature extractor & regressor' strategies, but the gap is small, especially for the relatively better-performing 'feature extractor & RBF-SVR', also illustrated in the violin plot Figure 5D. Considering that 'feature extractor & regressor' strategies are nearly negligible in additional training costs, this provides an alternative for the usability. In comparison, for cases that multiple proteins show up simultaneously in the transfer learning phase, a multi-head finetuning gives even more accurate predictions, since 'feature extractor & regressor' strategies cannot take advantage of multi-head training due to fixed ligand features.

We further explore how the number of proteins in the initial training phase will affect the performance in the transfer learning phase. To this end, we prepare a series of subsets of BindingDB-18 by randomly drawing 100, 200 and 300 proteins without replacement, and then train models on these subsets and evaluate in the same transfer learning dataset BindingDB-18ex. As shown in Table 6, the 'pretrained parameters & finetuning', the 'feature extractor & random forest' and 'feature extractor & RBF-SVR' strategies demonstrate better performance with more initial proteins, indicating that more proteins in the initial training improve the generalizability of the model. However, for the 'feature extractor & MLP' strategy, the gain is not significant. Although models with whole-set initial training performs best, the gap is small, and performance does not necessarily get better with more initial proteins. Considering the training losses are indeed declining, we speculate overfitting may account for the test error saturation.

Cross-dataset evaluations

Besides the transfer learning for novel proteins, we also assess the generalizability of the proposed SAM-DTA by cross-dataset evaluations. To this end, we take snapshots of the BindingDB dataset at three different time points, namely the year 2018, 2021 and 2022. For the year 2021 and 2022, we take the newly recorded samples compared with year 2018 and constitute the datasets termed BindingDB-21, BindingDB-22. Specifically, BindingDB-21 and BindingDB-22 contain 180,822 and 232,186 samples in total, respectively. The datasets are utilized in two ways. Firstly, we treat the datasets as blind testsets and directly evaluate the model that

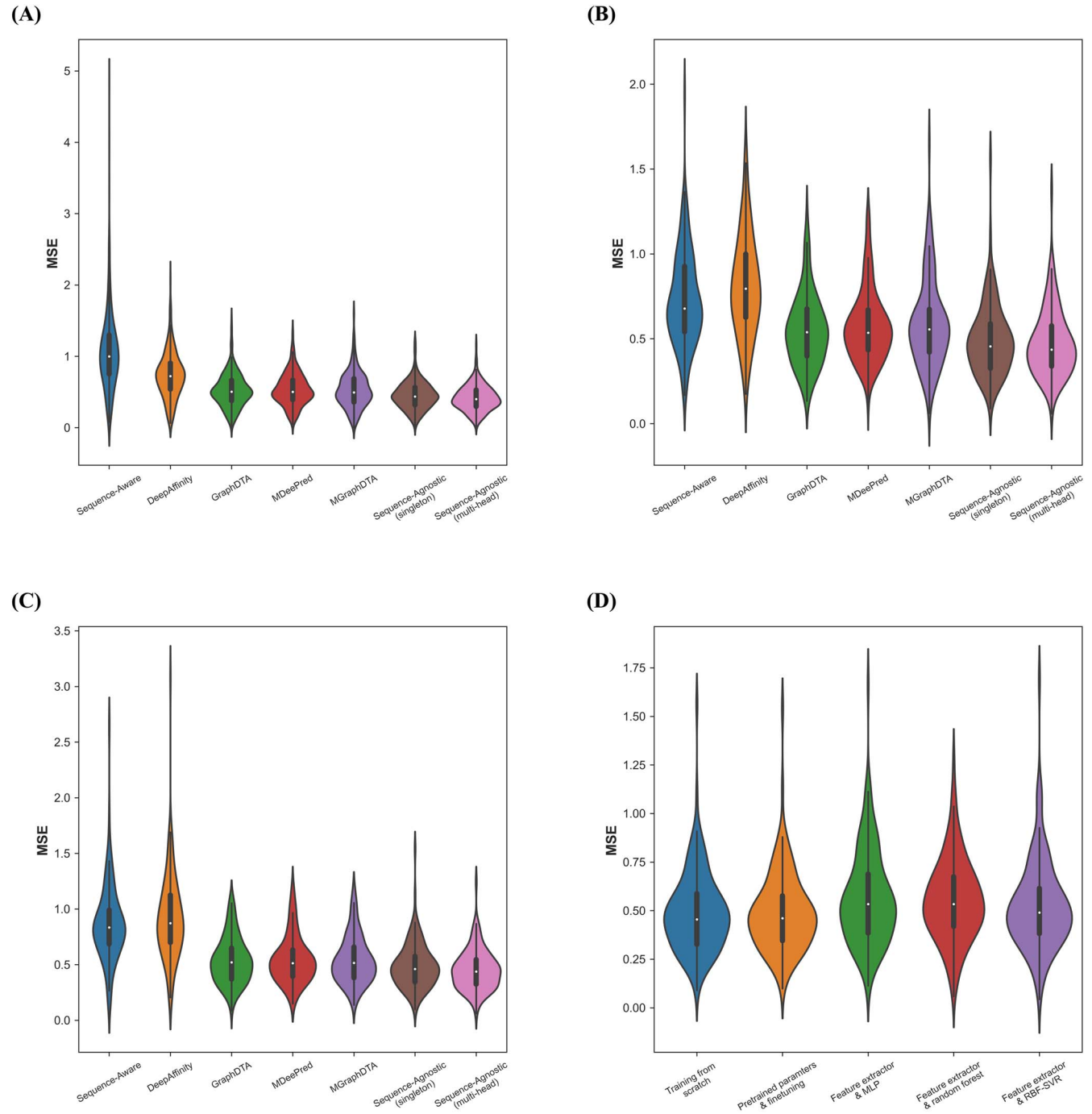

Figure 5. Violin plot analysis across proteins, where protein–ligand pairs are grouped according to proteins and MSE is calculated for each protein. The proposed sequence-agnostic model (SAM-DTA) is compared with six competing models for (A) initial training (401 proteins), (B) training from scratch (129 proteins), (C) finetuning (129 proteins). Different strategies are also compared for SAM-DTA in (D) singleton learning scenario, assuming only one protein in each transfer learning phase.

is trained on the trainset of BindingDB-18. This way essentially performs the prospective studies. Another way, however, is to take the datasets as additional trainsets that is merged with the trainset of BindingDB-18, whereas the valset and testset stay the same. In this way, we can see how the number of training samples will affect the performance given the same testset.

Table 7 presents the results. For the case that BindingDB-21/22 are utilized as blind testsets, the proposed multi-head sequence-agnostic model (SAM-DTA) outperforms DeepAffinity, Sequence-Aware, as well as the singleton version with clear margins. Notably, performance degradations occur for all the four models including SAM-DTA and to find out the reason, we also analyze the violin plots for SAM-DTA between the three testsets. Figure 6 shows the violin plots of different views across (A) proteins and (B) ligands, where protein–ligand pairs are grouped according to proteins/ligands and MSE is calculated for each protein/ligand, respectively (truncated at MSE 5.0). It can be seen that majority of proteins/ligands in BindingDB-21/22 has a performance similar to that in BindingDB-18, but more corner cases have appeared. We think in this case the SAM-DTA model mainly suffers from

Table 6. Studies of how the number of proteins in the initial training affects the performance in the transfer learning phase

Strategy	No. of proteins in initial training	No. of proteins in transfer learning	Trainset MSE	Trainset R^2	Valset MSE	Valset R^2	Testset MSE	Testset R^2
Pretrained parameters & finetuning	100	129	0.0047	0.9977	0.4495	0.7311	0.4558	0.7312
	200	129	0.0046	0.9976	0.4473	0.7319	0.4521	0.7330
	300	129	0.0044	0.9978	0.4488	0.7313	0.4520	0.7329
	400	129	0.0044	0.9979	0.4471	0.7320	0.4486	0.7348
Feature extractor & MLP	100	129	0.0476	0.9717	0.5915	0.6588	0.5998	0.6580
	200	129	0.0286	0.9830	0.5905	0.6585	0.6011	0.6586
	300	129	0.0280	0.9834	0.5987	0.6567	0.6046	0.6589
	401	129	0.0227	0.9865	0.5715	0.6693	0.5889	0.6653
Feature extractor & random forest	100	129	0.0851	0.9646	0.6065	0.6430	0.6012	0.6519
	200	129	0.0823	0.9653	0.5860	0.6544	0.5808	0.6631
	300	129	0.0820	0.9655	0.5864	0.6547	0.5793	0.6648
	401	129	0.0803	0.9658	0.5697	0.6646	0.5693	0.6701
Feature extractor & RBF-SVR	100	129	0.3622	0.7909	0.5547	0.6678	0.5526	0.6737
	200	129	0.3613	0.7911	0.5465	0.6729	0.5462	0.6773
	300	129	0.3029	0.8260	0.5246	0.6857	0.5231	0.6909
	401	129	0.3117	0.8205	0.5187	0.6895	0.5192	0.6932

Table 7. Results of cross-dataset evaluations. BindingDB-21/22 are utilized either as blind testsets, or as additional trainsets

Trainset	Valset	Testset	Model	Trainset MSE	Trainset R^2	Valset MSE	Valset R^2	Testset MSE	Testset R^2
BindingDB-18 trainset	BindingDB-18 valset	BindingDB-21	Sequence-Aware	1.0383	0.5134	1.1225	0.4744	1.8936	0.2021
			DeepAffinity	0.2945	0.8670	0.7905	0.6490	2.0810	0.1720
			Sequence-Agnostic (singleton)	0.0010	0.9995	0.4633	0.7826	1.3905	0.5247
			Sequence-Agnostic (multi-head)	0.0055	0.9980	0.4233	0.8019	1.1548	0.5176
BindingDB-18 trainset	BindingDB-18 valset	BindingDB-22	Sequence-Aware	1.0383	0.5134	1.1225	0.4744	1.8923	0.1870
			DeepAffinity	0.2945	0.8670	0.7905	0.6490	2.0908	0.1620
			Sequence-Agnostic (singleton)	0.0010	0.9995	0.4633	0.7826	1.5289	0.4608
			Sequence-Agnostic (multi-head)	0.0055	0.9980	0.4233	0.8019	1.3151	0.4543
BindingDB-18 trainset	BindingDB-18 valset	BindingDB-18 testset	Sequence-Agnostic (multi-head)	0.0101	0.9952	0.4974	0.7663	0.4975	0.7636
BindingDB-18 trainset & -21				0.0145	0.9942	0.3716	0.8261	0.3680	0.8257
BindingDB-18 trainset & -22				0.0161	0.9926	0.3698	0.8264	0.3672	0.8255

trainset not covering enough corner cases. This conjecture can also be confirmed from results of the case when Binding DB-21/22 are added to the trainset, where SAM-DTA demonstrates comparable trainset errors but clearly better valset and testset performances. It indicates that the performance has not saturated and more data are beneficial for covering more corner cases.

Conclusion

In this paper we propose SAM-DTA, a sequence-agnostic model for drug–target binding affinity prediction. It is fundamentally different from the canonical paradigm that processes the inputs of the protein and the ligand separately and then combines them together. Rather, different proteins are taken separately and are joint trained in a cooperative multi-head manner. We empirically demonstrate that the novel paradigm outperforms its competitive sequence-based counterpart with a clear margin, for example with the MSE of 0.4261 versus 0.7612 and the R-Square of 0.7984 versus 0.6570 against DeepAffinity [23]. Moreover, the sequence-agnostic paradigm further shows its unique advantage in transfer learning of novel proteins (with the MSE of 0.4486 versus 0.9140 and the R-Square of 0.7348 versus 0.4570 against DeepAffinity [23] in the funetuning strategy), as well as in predicting future data (with the MSE of 1.3151 versus 2.0908 and the R-Square of 0.4543 versus 0.1620 against DeepAffinity [23] tested in BindingDB-22). Extensive experiments are also conducted to compare different architectural settings and hyper-parameters, to evaluate different strategies in transfer learning and to assess the effect by the capacity of dataset. Results are analyzed with scatter plots, violin plots and so on.

Note that the proposed sequence-agnostic model also has a connection with sequence-based methods. Recall that in the multi-head scheme, all but the last FC layer is shared among the heads. Equivalently, only a linear layer with a h-dimensional

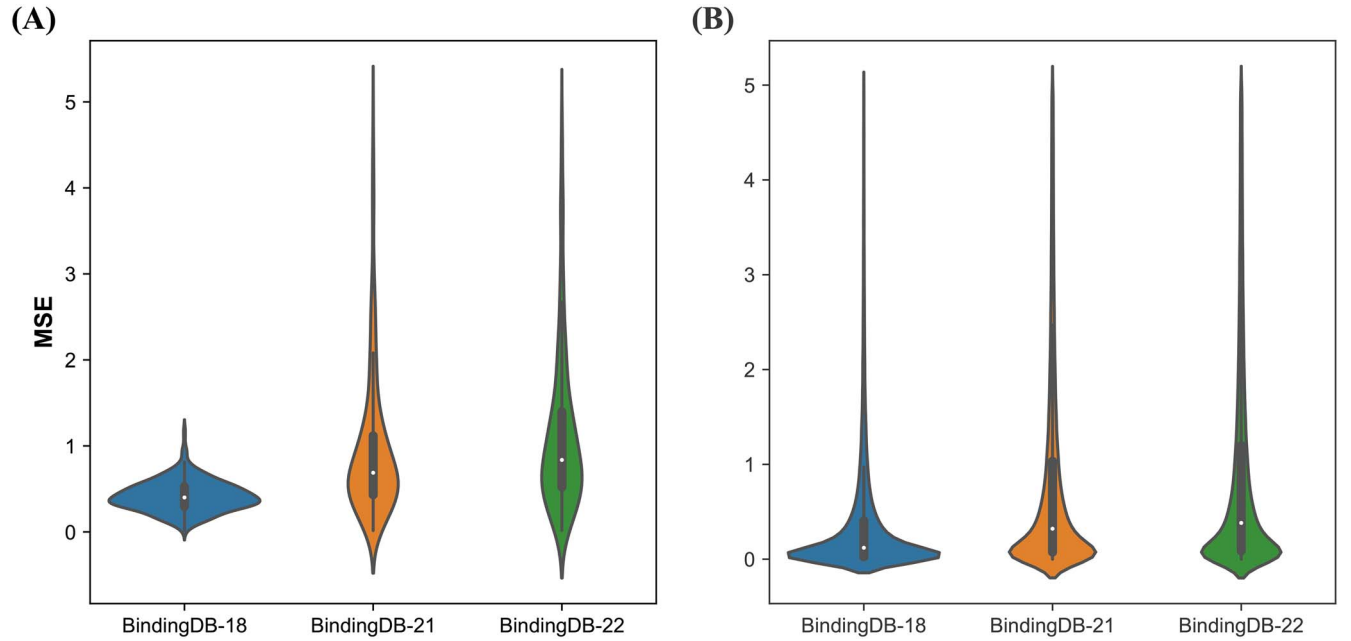

Figure 6. Violin plot analysis between testsets as BindingDB-18 testset, -21 and -22 for SAM-DTA trained on BindingDB trainset. Different views across (A) proteins and (B) ligands are both analyzed, where protein–ligand pairs are grouped according to proteins/ligands and MSE is calculated for each protein/ligand, respectively. Truncated at MSE 5.0.

weight and a scalar bias is peculiar to each protein, whereas others are all shared. As a result, the weight and bias can also be regarded as a 'representation' of the protein. In sequence-based methods, the sequence information of the protein is fed into the model as part of the input, and there is always some module that digests this input and outputs the feature for that protein. For example, in DeepAffinity [23] an embedding layer, an RNN and an attention layer collectively serve to give the protein feature based on the sequence. We say in this case the protein feature is extracted by the model. On the contrary, in the proposed multi-head sequence-agnostic scheme, parameters of the last layer serve as the 'protein feature', but are optimized with gradient descent in the training process. It is thus learned by the model: a protein is characterized completely by the ligands that it interacts. In Figure 7 we visualize the learned 'features' by principal component analysis in the transfer learning scenario, where each dot represents a protein, and the color indicates one of the four classes it is in: nuclear ER, ion channels, receptor tyrosine kinases and GPCR. Note the clustering of these features for proteins in the same class. For example, proteins of receptor tyrosine kinases lie apart from that of other classes. The clustering reveals that proteins in the same class have features that are also with closer proximity in the feature space, which is a sign for the correlation of the learned features to protein relationships. It would also be of interest to investigate the correlation between this learned feature and the sequence itself, and we leave it for future work.

Note also that the proposed sequence-agnostic model can be straightforwardly extended to other measures in affinity prediction, such as K_i, K_d, EC50, etc. Also, the paradigm can be applied and evaluated in other interaction prediction topics, for example cancer drug responses, microbe–drug associations, etc., where cancer cell lines and microbes can be treated separately in the multi-head scheme, like proteins in this study. Considering that cancer cell lines and microbes are in a more macro level and cannot be simply and fully characterized by a sequence, agnostic models have more advantage of tolerability for incomplete or inaccurate quantitative characterization. We also leave it for future work.

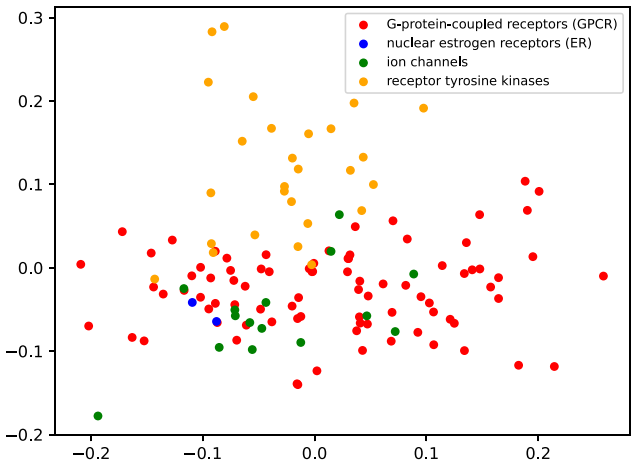

Figure 7. Visualization of the learned parameters of the last layer by principal component analysis in the transfer learning scenario. These parameters are seen as the 'features' for proteins. Each dot represents a protein, the color of which indicates one of the four classes it is in: nuclear ER, ion channels, receptor tyrosine kinases and GPCR.

Key Points

- A novel sequence-agnostic paradigm (SAM-DTA) is proposed for drug–target binding affinity prediction.
- SAM-DTA achieves superior performance than the sequence-based counterpart, without access to any protein-sequence-related information.
- SAM-DTA shows unique advantage in transfer learning of novel proteins, as well as in predicting future data.

Acknowledgments

The authors acknowledge the Center for High Performance Computing at Shanghai Jiao Tong University for computing resources.

Funding

National Science Foundation of China [grant numbers 11504231, 21873101, 31630002, 32030063]; Innovation Program of Shanghai Municipal Education Commission [2019-01-07-00-02-E00076]; Student Innovation Center at Shanghai Jiao Tong University.

References

1. Morgan S, Grootendorst P, Lexchin J, *et al.* The cost of drug development: a systematic review. *Health Policy* 2011;**100**(1):4–17.
2. Schlander M, Hernandez-Villafuerte K, Cheng C-Y, *et al.* How much does it cost to research and develop a new drug? a systematic review and assessment. *Pharmacoeconomics* 2021;**39**(11):1243–69.
3. Gilson MK, Zhou H-X. Calculation of protein-ligand binding affinities. *Annu Rev Biophys Biomol Struct* 2007;**36**(1):21–42.
4. Keiser MJ, Setola V, Irwin JJ, *et al.* Predicting new molecular targets for known drugs. *Nature* 2009;**462**(7270):175–81.
5. Power A, Berger AC, Ginsburg GS. Genomics-enabled drug repositioning and repurposing: insights from an iom roundtable activity. *JAMA* 2014;**311**(20):2063–4.
6. Chang RL, Xie L, Xie L, *et al.* Drug off-target effects predicted using structural analysis in the context of a metabolic network model. *PLoS Comput Biol* 2010;**6**(9):e1000938, 1–18.
7. Mayr A, Klambauer G, Unterthiner T, *et al.* Deeptox: toxicity prediction using deep learning. *Front Environ Sci* 2016;**3**:80.
8. Inglese J, Auld DS. High throughput screening (hts) techniques: applications in chemical biology. *Wiley Encyclopedia of*. *Chem Biol* 2007;**1**:1–15.
9. Ain QU, Aleksandrova A, Roessler FD, *et al.* Machine-learning scoring functions to improve structure-based binding affinity prediction and virtual screening. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, **5**(6):405–24, 2015.
10. Cang Z, Wei G-W. Topologynet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *PLoS Comput Biol* 2017;**13**(7):e1005690, 1–27.
11. Forli S, Huey R, Pique ME, *et al.* Computational protein–ligand docking and virtual drug screening with the autodock suite. *Nat Protoc* 2016;**11**(5):905–19.
12. Gomes J, Ramsundar B, Feinberg EN, *et al.* Atomic convolutional networks for predicting protein-ligand binding affinity. *CoRR*, 2017;**abs/1703.10603**:1–17.
13. Jiménez J, Skalic M, Martinez-Rosell G, *et al.* K deep: protein–ligand absolute binding affinity prediction via 3d-convolutional neural networks. *J Chem Inf Model* 2018;**58**(2):287–96.
14. Karplus M, Andrew J, McCammon. Molecular dynamics simulations of biomolecules. *Nat Struct Biol* 2002;**9**(9):646–52.
15. Leach AR, Shoichet BK, Peishoff CE. Prediction of protein- ligand interactions. docking and scoring: successes and gaps. *J Med Chem* 2006;**49**(20):5851–5.
16. Stepniewska-Dziubinska MM, Zielenkiewicz P, Siedlecki P. Development and evaluation of a deep learning model for protein–ligand binding affinity prediction. *Bioinformatics* 2018;**34**(21):3666–74.
17. Wallach I, Dzamba M, Heifets A. Atomnet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *CoRR*, 2015;**abs/1510.02855**:1–11.
18. Yan Y, Zhang D, Zhou P, *et al.* Hdock: a web server for protein–protein and protein–dna/rna docking based on a hybrid strategy. *Nucleic Acids Res* 2017;**45**(W1):W365–73.
19. Burley SK, Berman HM, Bhikadiya C, *et al.* Rcsb protein data bank: biological macromolecular structures enabling research and education in fundamental biology, biomedicine, biotechnology and energy. *Nucleic Acids Res* 2019;**47**(D1):D464–74.
20. Jumper J, Evans R, Pritzel A, *et al.* Highly accurate protein structure prediction with alphafold. *Nature* 2021;**596**(7873):583–9.
21. Cheng F, Zhou Y, Li J, *et al.* Prediction of chemical–protein interactions: multitarget-qsar versus computational chemogenomic methods. *Mol Biosyst* 2012;**8**(9):2373–84.
22. Cheng Z, Zhou S, Wang Y, *et al.* Effectively identifying compound-protein interactions by learning from positive and unlabeled examples. *IEEE/ACM Trans Comput Biol Bioinform* 2016;**15**(6):1832–43.
23. Karimi M, Di W, Wang Z, *et al.* Deepaffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks. *Bioinformatics* 2019;**35**(18):3329–38.
24. Öztürk H, Özgür A, Ozkirimli E. Deepdta: deep drug–target binding affinity prediction. *Bioinformatics* 2018;**34**(17):i821–9.
25. Shi Y, Zhang X, Liao X, *et al.* Protein-chemical interaction prediction via kernelized sparse learning svm. In: Russ B Altman, A Keith Dunker, Lawrence Hunter, Tiffany A Murray and Teri E Klein (eds.) *Biocomputing 2013*. World Scientific, 2013, 41–52.
26. Tabei Y, Yamanishi Y. Scalable prediction of compound-protein interactions using minwise hashing. *BMC Syst Biol* 2013;**7**(6):1–13.
27. Hua Y, Chen J, Xue X, *et al.* A systematic prediction of multiple drug-target interactions from chemical, genomic, and pharmacological data. *PloS one* 2012;**7**(5):e37608, 1–14.
28. Cichonska A, Ravikumar B, Parri E, *et al.* Computational-experimental approach to drug-target interaction mapping: a case study on kinase inhibitors. *PLoS Comput Biol* 2017;**13**(8):e1005678, 1–28.
29. Cobanoglu MC, Liu C, Feizhuo H, *et al.* Predicting drug–target interactions using probabilistic matrix factorization. *J Chem Inf Model* 2013;**53**(12):3399–409.
30. He T, Heidemeyer M, Ban F, *et al.* Simboost: a read-across approach for predicting drug–target binding affinities using gradient boosting machines. *J Chem* 2017;**9**(1):1–14.
31. Pahikkala T, Airola A, Pietilä S, *et al.* Toward more realistic drug–target interaction predictions. *Brief Bioinform* 2015;**16**(2):325–37.
32. Paszke A, Gross S, Massa F, *et al.* Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 2019;**32**:8026–37.
33. Gaulton A, Bellis LJ, Patricia Bento A, *et al.* Chembl: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res* 2012;**40**(D1):D1100–7.
34. Liu T, Lin Y, Wen X, *et al.* Bindingdb: a web-accessible database of experimentally determined protein–ligand binding affinities. *Nucleic Acids Res* 2007;**35**(suppl_1):D198–201.
35. Devlin J, Chang M-W, Lee K, *et al.* Bert: Pre-training of deep bidirectional transformers for language understanding *CoRR*, 2018;**abs/1810.04805**:1–16.
36. Landrum*Rdkit: Open-source cheminformatics*, 2010. https://www.rdkit.org/docs/Overview.html#citing-the-rdkit.
37. Kim S, Thiessen PA, Bolton EE, *et al.* Pubchem substance and compound databases. *Nucleic Acids Res* 2016;**44**(D1):D1202–13.
38. Duvenaud DK, Maclaurin D, Iparraguirre J, *et al.* Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems* 2015;**28**:2224–32.

39. Wang K, Zhou R, Li Y, *et al.* Deepdtaf: a deep learning method to predict protein–ligand binding affinity. *Brief Bioinform* 2021;**22**(5):bbab072.
40. He K, Zhang X, Ren S, *et al.* Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–8, 2016.
41. Gao H, Liu Z, Van Der Maaten L, *et al.* Densely connected convolutional networks. In: Rama Chellappa, Zhengyou Zhang and Anthony Hoogs (eds.) *Proceedings of the IEEE conference on computer vision and pattern recognition*. Honolulu, HI, USA: IEEE, 2017, 4700–8.
42. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput* 1997;**9**(8):1735–80.
43. Cho K, Van Merriënboer B, Gulcehre C, *et al.* Learning phrase representations using rnn encoder-decoder for statistical machine translation. *CoRR*, 2014;**abs/1406.1078**:1–15.
44. Welling M, Kipf TN. Semi-supervised classification with graph convolutional networks. *CoRR*, 2016;**abs/1609.02907**:1–14.
45. Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs. *Advances in neural information processing systems* 2017;**30**:1025–35.
46. Vinyals O, Bengio S, Kudlur M. Order matters: Sequence to sequence for sets. *CoRR*, 2015;**abs/1511.06391**:1–11.
47. Li Y, Zemel R, Brockschmidt M, *et al.* Gated graph sequence neural networks. *CoRR*, 2016;**abs/1511.05493**:1–20.
48. Knyazev B, Taylor GW, Amer M. Understanding attention and generalization in graph neural networks. *Advances in neural information processing systems* 2019;**32**:4202–12.
49. Lee J, Lee I, Kang J. Self-attention graph pooling. In: Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.) *International conference on machine learning*. Long Beach, California, USA: PMLR, 2019, 3734–43.
50. Gao H, Ji S. Graph u-nets. In: Kamalika Chaudhuri, Ruslan Salakhutdinov (eds.) *In international conference on machine learning*. Long Beach, California, USA: PMLR, 2019, 2083–92.
51. Zhang M, Cui Z, Neumann M, *et al.* An end-to-end deep learning architecture for graph classification. In: *Proceedings of the AAAI conference on artificial intelligence*. Palo Alto, California USA: AAAI Press, Vol. **32**, 2018.
52. Keyulu X, Li C, Tian Y, *et al.* Representation learning on graphs with jumping knowledge networks. In: Jennifer Dy, Andreas Krause (eds.) *International conference on machine learning*. Stockholmsmässan, Stockholm Sweden: PMLR, 2018, 5453–62.
53. Dhillon IS, Guan Y, Kulis B. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE Trans Pattern Anal Mach Intell* 2007;**29**(11):1944–57.
54. Kingma DP, Ba J. Adam: A method for stochastic optimization.*CoRR*, 2014;**abs/1412.6980**:1–15.
55. Fey M, Lenssen JE. Fast graph representation learning with pytorch geometric. *CoRR*, 2019;**abs/1903.02428**:1–9
56. Nguyen T, Le H, Venkatesh S. Graphdta: prediction of drug–target binding affinity using graph convolutional networks. *Bioinformatics*, **37**(8):1140–7.
57. Rifaioglu AS, Atalay RC, Cansen Kahraman D, *et al.* Mdeepred: novel multi-channel protein featurization for deep learning-based binding affinity prediction in drug discovery. *Bioinformatics* 2021;**37**(5):693–704.
58. Yang Z, Weihe Zhong LZ, Chen CY-C. Mgraphdta: deep multiscale graph neural network for explainable drug–target binding affinity prediction. *Chem Sci* 2022;**13**(3):816–33.
59. Zhenqin W, Ramsundar B, Feinberg EN, *et al.* Moleculenet: a benchmark for molecular machine learning. *Chem Sci* 2018;**9**(2):513–30.
60. Xiong G, Zhenxing W, Yi J, *et al.* Admetlab 2.0: an integrated online platform for accurate and comprehensive predictions of admet properties. *Nucleic Acids Res* 2021;**49**(W1):W5–14.
61. Zhang S, Yan Z, Huang Y, *et al.* Helixadmet: a robust and endpoint extensible admet system incorporating self-supervised knowledge transfer. *Bioinformatics*, **38**(13):3444–53.
62. Jiang D, Hsieh C-Y, Zhenxing W, *et al.* Interactiongraphnet: A novel and efficient deep graph representation learning framework for accurate protein–ligand interaction predictions. *J Med Chem* 2021;**64**(24):18209–32.
63. Nguyen T, Le H, Quinn TP, *et al.* Graphdta: Predicting drug–target binding affinity with graph neural networks. *Bioinformatics* 2021;**37**(8):1140–7.
64. Waskom ML. seaborn: statistical data visualization. *Journal of Open Source Software* 2021;**6**(60):3021.