Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

KNU-DTI: KNowledge United Drug-Target Interaction prediction

Ryong Heo [a,c], Dahyeon Lee [b], Byung Ju Kim [c], Sangmin Seo [d], Sanghyun Park [d], Chihyun Park [a,b,c,e,*]

[a] Interdisciplinary Graduate Program in Medical Bigdata Convergence, Kangwon National University, Chuncheon-si, 24341, Gangwon-do, Republic of Korea
[b] Department of Data Science, Kangwon National University, Republic of Korea
[c] UBLBio Corporation, Yeongtong-ro 237, Suwon, 16679, Gyeonggi-do, Republic of Korea
[d] Department of Computer Science, Yonsei University, Yonsei-ro 50, Seodaemun-gu, 03722, Seoul, Republic of Korea
[e] Department of Computer Science and Engineering, Kangwon National University, Republic of Korea

ARTICLE INFO

Keywords:
Drug discovery
Sequence based drug-target protein interactions
Multimodal representation learning
Feature fusion

ABSTRACT

Motivation: Accurately predicting drug-target protein interactions (DTI) is a cornerstone of drug discovery, enabling the identification of potential therapeutic compounds. Sequence-based prediction models, despite their simplicity, hold great promise in extracting essential information directly from raw sequences. However, the focus in recent DTI studies has increasingly shifted toward enhancing algorithmic complexity, often at the expense of fully leveraging robust sequence representation learning methods. This shift has led to the underestimation and gradual neglect of methodologies aimed at effectively capturing discriminative features from sequences. Our work seeks to address this oversight by emphasizing the value of well-constructed sequence representation algorithms, demonstrating that even with simple interaction mapping algorithm techniques, accurate DTI models can be achieved. By prioritizing meaningful information extraction over excessive model complexity, we aim to advance the development of practical and generalizable DTI prediction frameworks.
Results: We developed the KNowledge Uniting DTI model (KNU-DTI), which retrieves structural information and unites them. Protein structural properties were obtained using structural property sequence (SPS). Extended-connectivity fingerprint (ECFP) was used to estimate the structure-activity relationship in molecules. Including these two features, a total of five latent vectors were derived from protein and molecule via various neural networks and integrated by elemental-wise addition to predict binding interactions or affinity. Using four test concepts to evaluate the model, we show that the model outperforms recently published competitors. Finally, a case study indicated that our model has a competitive edge over existing docking simulations in some cases.

1. Introduction

Drug discovery or repurposing is an active study area in the field of systemic biology. Here, drug-target interaction (DTI) prediction is crucial to predict DTI relationships. Despite the substantial amount of invested funding and time, traditional wet lab-based drug discovery research has recorded a success rate of only 10 % during the clinical trial phase. For drugs that have not yet entered clinical trials, the success rate tends to be <10 % [1]. Furthermore, the past few decades have produced a considerable amount of high-throughput biological and chemical data, which, in its sheer volume, cannot be evaluated using traditional wet lab-based studies.

These limitations led to the development of computational machine learning (ML) that dramatically decreased the cost and time associated with the early steps of drug discovery [2]. From the perspective of computer science, DTIs prediction can be defined into two classifications: i) only identifying the presence or absence of an interaction between a drug and target and ii) regression, which predicts real-valued binding affinity as a sophisticated problem. Many ML-based DTI prediction models have been previously developed based on binary classification [2]. Models such as KronRLS [3] or Simboost [4] are ML-based regression models. Virtual screening using well-refined 3D structural data, sequence similarity-based models, and matrix factorization-based models have achieved high scores in ML-based DTI prediction studies. Although virtual screening has achieved a high success rate in DTI prediction, it presents high computational cost and time complexity.

[*] Corresponding author. Interdisciplinary Graduate Program in Medical Bigdata Convergence, Kangwon National University, Chuncheon-si, 24341, Gangwon-do, Republic of Korea
E-mail address: chihyun@kangwon.ac.kr (C. Park).

https://doi.org/10.1016/j.compbiomed.2025.109927
Received 1 October 2024; Received in revised form 17 January 2025; Accepted 24 February 2025
Available online 1 March 2025

Deep learning (DL)-based models have achieved better performance than ML-based models in various fields, and consequently, research has rapidly shifted to using DL-based models.

DeepAffinity [5] used an structural property sequence (SPS) that was designed to solve the limitations of a recurrent neural network (RNN). This model has an input-sequence length-dependent gradient vanishing and exploding problem owing to the repeated multiplication of the recurrent weight matrix [6]. The SPS is a four-letter code sequence where different fractions are algorithmically determined for the raw protein sequence. Although SPS preserves protein structure information as much as possible, its length is $>$ 10 times less than that of the raw protein sequences. This allows the RNN to circumvent the sequence-length-dependent gradient-vanishing issue.

GraphDTA [7] was the first model to incorporate a graph neural network (GNN) [8] into binding affinity prediction. Previous GraphDTA models used one-dimensional (1D) compound sequences such as the simplified molecular-input line-entry system (SMILES) [9]. However, Nguyen et al. suggested that 1D compound sequences would lose their topological information. Rather, they used graph-structured compound data to preserve topological features. Four GNN variants, namely, the Graph Convolution Network (GCN), Graph Attention Network (GAT), Graph Isomorphism Network (GIN), and GAT-GCN combined models, were used to learn the representation vectors.

MolTrans [10] uses a transformer encoder [11] to obtain the representation vectors for proteins and compounds. These representation vectors were matrix multiplied to create an interaction map. Subsequently, the 2D CNN block learns local patterns from the interaction map. Moltrans uses the FCS mining algorithm to generate subword tokens for protein and compound sequences [12]. This approach is useful in that some protein domain, active site residues are highly conserved [13] and compound functional groups also appear repeatedly.

TransformerCPI [14] maintains the encoder-decoder structure of the transformer but changes the internal representation vector learning block. For protein input, 3-g word-embedding is performed using a pretrained Word2Vec [15] model. Gated Convolution Neural Net (GCNN) [16] learns protein representation vector. The compound is then converted into 2D graph-structured data and processed using a GCN and a self-attention block. Finally, a cross-attention block learns the interaction map between the protein and compound representation vectors.

In contrast to the general attention mechanism, which uses only the attention score between two sequences, HyperAttentionDTI [17] proposes a hyperattention mechanism that calculates the attention score from deep feature spatial vectors. The CNN block learns protein and compound representation vectors. The hyperattention module is then operated for each representation vector. The representation vectors from the CNN block and the hyperattention vectors are concatenated into the decision vector. A fully connected (FC) layer is then used for interaction predictions. The hyper-attention mechanism expands the existing low-resolution sequence spatial attention mechanism to a high-resolution feature spatial attention mechanism.

PerceiverCPI [18] uses a directed message-passing neural network (D-MPNN) [19] and the Morgan circular fingerprint [20] (i.e., ECFP) to learn topological features from chemical compounds. The D-MPNN and GCNN block learn the compound and protein representation vectors, respectively. The second FC layer learns an ECFP representation vector. Two compound-derived vectors are converted into one vector using a cross-attention block. Another cross-attention block creates an interaction map from the compound and protein representation vectors. As large language models have recently overwhelmed other representation learning methods, the formation of an advanced interaction map is a promising step in DTI predictions. This model demonstrated an advanced cross-attention method to construct an interaction map. Additionally, recent advancements in drug-target interaction prediction, such as the MT-DTA [21] and TDGraphDTA [22] models, highlight a growing trend toward designing sophisticated and complex neural network architectures. The MT-DTA model excels in capturing both local and global molecular features through self-attention mechanisms and variational autoencoders, enhancing the interpretability and accuracy of interaction predictions. Similarly, the TDGraphDTA model leverages multi-scale information interaction and graph optimization to refine molecular graph representations and extract fine-grained contextual features. These approaches underscore the potential of advanced neural networks to not only improve the learning of interaction maps between drugs and targets but also offer enhanced scalability, interpretability, and predictive performance.

Although these studies have provided relevant insights, the drawbacks of the existing approaches can be summarized as follows.

- Although SPS still provides structural information, using only SPS without a raw protein sequence leads to a loss of non-structural information.
- In Moltrans, the 2D CNN is an optimal module for learning local patterns but not global ones. When it comes to drug-target pairs, important sequences for interaction are usually distant, which means that global patterns are critical in DTI prediction.
- The main idea of the transformer encoder-decoder block used in transformed CPI is machine translation. In other words, it is optimal to generate tasks that are not suitable for classification or regression. Instead, using a transformer encoder to learn the representation vectors and create an advanced interaction map would be appropriate for a DTI prediction task.
- HyperAttentionDTI uses a 1D CNN block to learn representation vectors. In Moltrans, the CNN specializes in learning local patterns. In DTIs prediction, protein sequences often exhibit an average length of $>$1000 residues, posing a challenge in effectively capturing global patterns using small-sized CNN kernels.

Inspired by Sun et al. [23], who demonstrated that combining orthogonal features improves prediction performance by capturing useful information remaining in other subspaces, extracting only one feature is insufficient as it may fail to capture complementary information. We recognize that many types of intrinsic information, such as secondary or tertiary structural features, chirality, chemical information and molecular structural information are present in raw sequences. Deriving as much information as possible using learning modules that aim to capture different features is an important contributor to DTI prediction. However, obtaining all types of information from the sequence data alone is impossible. Thus, we focused on two main properties expected to have the greatest influence on predicting DTIs. The first is structural features, which capture the geometric and topological characteristics of proteins and compounds. The second is attention score-based contextual features, which retrieve the contextual relationships within the entire sequence. We propose a knowledge-uniting model that focuses on obtaining those features using three types of representation-learning modules. The GCNN was used to learn the protein structural representations with convolution and gating functions. A GNN was used on graph-structured compounds to learn molecular topological features from compound sequences and we utilized a Multi-Layer Perceptron (MLP) to learn the molecular structural features of compound sequences. Attention-based transformer models are used by both proteins and compounds to retrieve contextual relationships in an entire sequence. Considering aforementioned drawbacks, our contributions are as follows:

Enhanced Representation Learning with Suitable Modules: We address limitations in existing DTA prediction models by employing more appropriate learning modules for representation learning. Unlike prior works that rely on local patterns or suboptimal translation-inspired architectures, our model effectively captures both local and global patterns. Specifically, we utilize GCNN for protein structural representations, GNN for compound topological features, and attention mechanisms to learn global contextual relationships within sequences.

This design ensures a more comprehensive and effective integration of DTI-related information.

Integration of Diverse Feature Spaces: The model unifies independent representations of proteins and compounds through element-wise addition, capturing complementary information from orthogonal latent spaces. This knowledge-unifying approach enables the integration of diverse properties (e.g., structural, topological, and contextual) into a single, enriched latent representation, minimizing information loss and enhancing prediction performance.

Balancing Model Accessibility and Predictive Performance: Our approach achieves an optimal balance between accessibility and performance by combining straightforward neural network modules with algorithmically extracted features. This design provides researchers with an intuitive understanding while maintaining strong predictive accuracy, making the model both accessible and scalable for practical DTI prediction tasks to researchers.

2. System and methods

In this section, we provide an overview of KNU-DTI in terms of the data used and the proposed representation learning method (Fig. 1). All representation vectors described in this section are obtained from end-to-end learning modules trained directly on the data, without the use of pretrained models

2.1. Protein representation

SPS representation($\mathbf{O_{SPS}}$): we used SSpro and ACCpro to generate the SPS [24]. Both tools extract secondary structure sequences and solvent accessibility sequences from raw protein sequences. Using the extracted and raw protein sequences, a corresponding four-letter SPS code was obtained for each fraction. The first code represents the secondary structures: alpha (A), beta (B), and coil (C). The second represents the solvent exposure feature, which is either not exposed (N) or exposed (E). The third category represents the property features: nonpolar (G), polar (T), acidic (D), or basic (K). The last letter represents the fraction length: short (S), medium (M), or long (L). The SPS generation method is described in detail in the supplementary information. SPS was embedded and passed through the GCNN layer to learn detailed structural features. This model possesses a CNN layer as well as an output gate (i.e. Gated Linear Unit, GLU), which allows the network to control which information should be passed through the next layer. Intuitively, in language modeling, a gate mechanism allows for the selection of words or features that are important for predicting the next word. The CNN layer flows as follows:

$$\mathrm{h_l(X) = X * W + s} \tag{1}$$

Where $\mathrm{X} \in \mathrm{R}^{\mathrm{n}\times\mathrm{m_1}}$ is the input feature of layer $\mathrm{h_l}$, $W \in R^{k\times m_1\times m_2}$, $s \in R^{m_2}$. n is the maximum length of SPS. $\mathrm{m_2}$ size is two times of $\mathrm{m_1}$ and here, the * symbol indicates the 1D CNN operator, not matrix multiplication. Zero padding is used to keep the size of the feature map passing through the GLU. After the CNN operation, $\mathrm{h_l(X)}$ is divided into two halves.

$$\mathrm{A = h_l[:\ half]} \tag{2}$$

$$\mathrm{B = h_l[half:]} \tag{3}$$

where $\mathrm{A, B} \in \mathrm{R}^{\mathrm{n}\times\mathrm{m_1}}$ followed by gating operation.

$$\mathrm{h_l = A \otimes \delta(B)} \tag{4}$$

Where $\mathrm{h_l} \in \mathrm{R}^{\mathrm{n}\times\mathrm{m_1}}$. δ is a non-linear activation function. The default sigmoid function is used. Finally, adding the skip connection layer, followed by multiplying by the scaling factor, returns the hidden state output.

$$\mathrm{h_{l+1} = (h_l \oplus X) \times scale} \tag{5}$$

Protein representation ($\mathbf{O_{Tp}}$): Huang et al. proposed the Explainable Substructure Partition Fingerprint (ESPF) [12], an algorithm inspired by Byte Pair Encoding (BPE) [25] from natural language

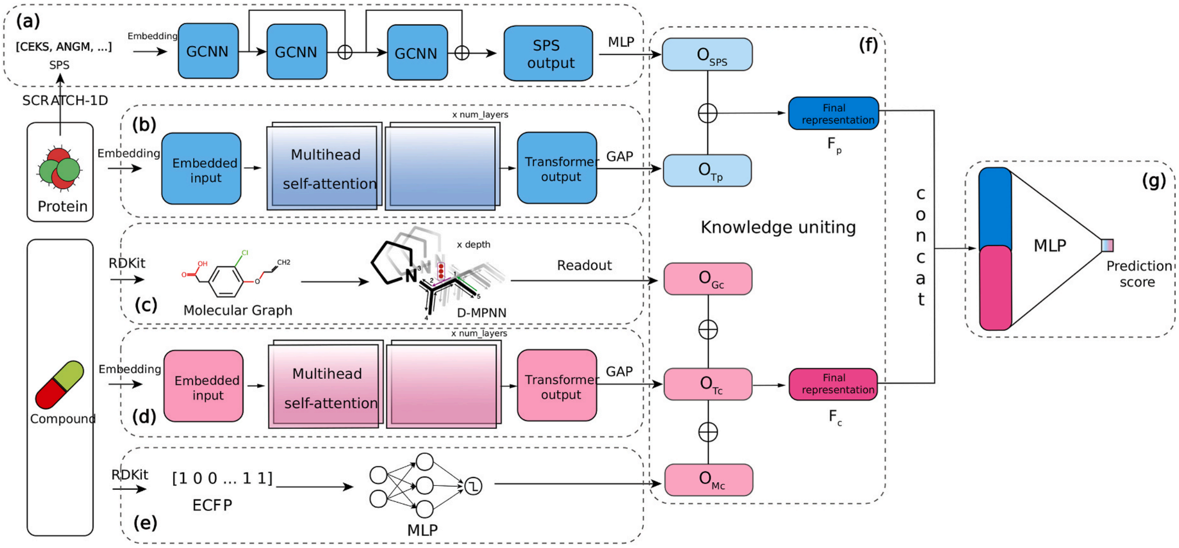

Fig. 1. Workflow of proposed model (a) Structural property sequence (SPS) derived from protein sequence is embedded. SPS representation vector ($\mathbf{O_{SPS}}$) learned structural information is obtained by passing through a GCNN layer. (b) Raw protein sequences are embedded, and Multihead self-attention encoder block learns protein representation vector ($\mathbf{O_{Tp}}$). (c) Raw compound sequences are converted into 2D matrix molecular graph structures using RDKit. By passing t-step message passing and last readout phase, graph representation vector ($\mathbf{O_{Gc}}$), which has topological information, is obtained. (d) Similar to step (b), raw compound sequences are converted into compound representation vector ($\mathbf{O_{Tc}}$) (e) Raw compound sequences are converted into 2048-dimensional binary vectors, also known as Morgan Circular Fingerprints, by RDKit. The MLP layer returns learned ECFP representation vector ($\mathbf{O_{Mc}}$). (f) All five representation vectors have the same shape. Elemental-wise adding, which makes final knowledge united representation vectors ($\mathbf{F_p}$ and $\mathbf{F_c}$), is applied. Notably, $\mathbf{F_p}$ and $\mathbf{F_c}$ have a deeper color than each representation vector before knowledge uniting. We described that knowledge united vectors have more information than each representation vectors before knowledge uniting. (g) Vectors concatenated into decision vector and the final MLP layer returns the final prediction score. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

processing. ESPF identifies frequently occurring substructures in sequences, such as SMILES or amino acid chains, and generates interpretable fingerprint vocabularies for drugs and proteins. The algorithm iteratively merges the most frequent token pairs to create a substructure vocabulary, allowing sequences to be decomposed into meaningful substructures represented as bit vectors. Furthermore, its flexibility enables applications to other biomedical sequence data, such as DNA. Using ESPF, raw protein sequences are processed into sub-word tokens, which are then converted into embedding vectors and passed through a transformer encoder to generate O_{Tp}. The transformer architecture [11], originally designed for machine translation tasks, has demonstrated exceptional performance across various fields, including natural language processing (NLP) and bioinformatics [26]. Protein and compound sequences, such as amino acids, SMILES, and SELFIES [27], have been successfully modeled using transformers for DTI prediction, producing representation vectors that encapsulate both contextual and general features [28]. The transformer encoder employs multi-head self-attention blocks, where queries (Q), keys (K), and values (V) are derived from the same source. The attention mechanism is mathematically represented as follows:

$$\mathrm{attention}(Q, K, V) = \mathrm{softmax}\left(QK^{T} \Big/ \sqrt{d_k}\right)V \tag{6}$$

2.2. Compound representation

Graph representation ($\mathbf{O_{Gc}}$): A compound is converted into a graph $G(\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ is the atom (vertex) feature matrix, and each atom feature is denoted as $\mathbf{x}_v$. Similarly, $\mathbf{E}$ represents the bond (edge) feature matrix, with each bond feature denoted as $\mathbf{e}_{vw}$. We adopted the same graph initialization method as the standard D-MPNN. The features composing atoms and bonds are summarized in Table 1. Although D-MPNN closely follows the structure of the existing MPNN [29], it accounts for the directionality of bonds, resulting in bond hidden states ($\mathbf{h}_{vw}$ and $\mathbf{h}_{wv}$) that have distinct values depending on their orientation. The D-MPNN operates in two distinct phases: the message passing phase and the readout phase. During the message passing phase, the following equation is used:

$$m_{vw}^{t+1} = \sum_{k \in N(v)} M_t\left(x_v, x_k, h_{kv}^{t}\right) \tag{7}$$

$$h_{vw}^{t+1} = U_t\left(h_{vw}^{t}, m_{vw}^{t+1}\right) \tag{8}$$

where $N(\mathbf{v})$ means Neighbors of each vertex $\mathbf{v}$. Σ is an aggregation function for aggregating message values($\mathbf{m}_{vw}$) flowed from neighbors. M_t and U_t are message function and update function which you can use any functions, respectively. At the end of the message passing step T, final hidden state of each atom is calculated as follows:

$$m_v = \sum_{k \in N(v)} h_{kv}^{T} \tag{9}$$

$$h_v = \tau(W_a \bullet \mathrm{cat}(x_v, m_v)) \tag{10}$$

Where $\mathrm{cat}(\mathbf{x}_v, \mathbf{m}_v)$ is concatenated vector of initialized atom feature vector and message value of each vertex $\mathbf{v}$, $\mathbf{W}_a$ is learnable matrix and τ is an activation function. The subsequent readout phase is straightforward: the hidden states of all atoms in the graph are summed to produce the molecular graph feature vector:

$$h = \sum_{v \in G} h_v \tag{11}$$

Compound representation ($\mathbf{O_{Tc}}$): Similar to protein representation ($\mathbf{O_{Tp}}$), the ESPF algorithm was applied to SMILES embeddings. Approximately 2 million molecular sequences from the ChEMBL [30] dataset were processed into ESPF sub-word vocabularies. SMILES were converted into ESPF sub-word tokens using these ChEMBL vocabularies and passed through a transformer encoder block to generate rich compound representation vectors.

Morgan Representation ($\mathbf{O_{Mc}}$): The Extended Connectivity Fingerprint (ECFP) [20] is a hashing-based algorithm that represents molecular structures as fixed-length bit vectors. ECFP generates fingerprints by iteratively exploring circular neighborhoods around each atom in a molecule, capturing both atomic and bond features. During each iteration, the features of neighboring atoms are combined and hashed to create unique identifiers for substructures. The earlier version of ECFP, known as Morgan fingerprints [31], was originally designed to address the isomorphism problem. However, in the 2000s, the algorithm was enhanced to better capture molecular activity-related features and reintroduced as Extended Connectivity Fingerprints (ECFPs). The ECFP algorithm is known for its strong capability to capture chemo-structural features. This iterative process continues for a predefined number of iterations (radius) to include increasingly larger molecular substructures. In our study, SMILES strings were converted into ECFPs using RDKit [32] with parameters set to a radius of 2 and 2048 nBits. The resulting 2048-dimensional binary vectors were then processed through an MLP layer to produce the final Morgan representation vector ($\mathbf{O_{Mc}}$).

2.3. Knowledge uniting

To construct the final knowledge-unified representation vectors, we build on the idea that embedding spaces reflect meaningful relationships between entities. As demonstrated by Mikolov et al. [33], entities with similar properties are typically mapped into vectors with similar directions, while entities with independent or unrelated characteristics are likely mapped into orthogonal directions due to their independence. Extending this concept, we propose that learning modules focusing on similar features—such as structural and topological properties—are expected to map their outputs to representation vectors with aligned directions. Conversely, learning modules designed to capture distinct and independent features—such as structural and sequence properties—are expected to produce representation vectors that exhibit orthogonality. Building on this foundation, we refer to the study by Sun et al. [23], which demonstrated the utility of extracting multiple independent features through orthogonal projection vectors in ordinal regression tasks. This study showed that combining orthogonal vectors creates more informative decision boundaries, leading to improved predictive performance. These findings highlight the importance of leveraging independent features not only to enhance accuracy in ordinal regression tasks but also to improve performance in broader machine learning contexts, such as regression and classification. However, effectively utilizing orthogonal representation vectors requires methods to combine them optimally while preserving their independent features. In this regard, Word2vec [15] demonstrated that simple element-wise operations within a well-trained embedding space effectively capture

Table 1
Atom and bond initial featurization

Atom features	description	size
atom type	type of atom (ex. C, N, O), by atomic number	100
# bonds	number of bonds the atom is involved in	6
formal charge	integer electronic charge assigned to atom	5
chirality	unspecified, tetrahedral CW/CCW, or other	4
# Hs	number of bonded hydrogen atoms	5
hybridization	sp, sp2, sp3, sp3d, or sp3d2	5
aromaticity	whether this atom is part of an aromatic system	1
atomic mass	mass of the atom, divided by 100	1
Bond features	**description**	**size**
bond type	single, double, triple, or aromatic	4
conjugated	whether the bond is conjugated	1
in ring	whether the bond is part of a ring	1
stereo	none, any, E/Z or cis/trans	6

relationships between words, enabling concise representations of both similar and contrasting concepts. Inspired by this paradigm, we assume that the element-wise addition of orthogonal representation vectors preserves all original information while synthesizing a more comprehensive and informative representation. Based on these considerations, we formulated the following assumptions.

- Since each learning module targets distinct properties, the representation vectors learned by these modules will exhibit orthogonality.
- If the representation vectors are orthogonal, performing an elemental-wise addition operation can effectively integrate the independent information into a unified latent space.
- If this knowledge united representation vector enhances the performance of the DTI prediction model, it validates the effectiveness and reasonability of the knowledge uniting operation.

These assumptions imply that the element-wise addition of each representation vector results in an enriched knowledge-united representation vector that captures complementary properties from multiple independent directions. Intuitively, protein and compound representation vectors containing more information make it easier to differentiate positive and negative samples in an interaction map. We then concatenate the two final knowledge-united vectors to extract interaction patterns, followed by an MLP layer to predict interactions.

$$\mathrm{F_p} = \mathrm{O_{SPS}} + \mathrm{O_{Tp}}, \mathrm{F_c} = \mathrm{O_{Gc}} + \mathrm{O_{Tc}} + \mathrm{O_{Mc}} \tag{12}$$

$$\widehat{\mathrm{y}} = \mathrm{MLP}\left(\mathrm{concat}\left(\mathrm{F_p}, \mathrm{F_c}\right)\right) \tag{13}$$

We set the last MLP layer output size such that the model could be used for both classification and regression tasks and BCE and MSE losses are used, respectively. For the regression task, PerceiverCPI, adopts a weighted update method to deal with skewed datasets, such as Davis and KIBA [34]. We also adopted this weighted update method to focus on the label range that the model expects to learn. For the detailed weighting method refer to supplementary materials section 3.

2.4. Benchmark datasets

BindingDB [35] is one of the most extensive public databases of experimentally determined protein-ligand binding affinities [2]. ELECTRA-DTA [36] confirmed that the BindingDB data used in DeepAffinity contained overlapped pairs. The dataset was refined to 129,109 samples in the removal process and merged with the Human protein-SPS pair table provided by DeepAffinity. For unmatched proteins, we constructed SPSs with SCRATCH-1D. Two protein sequences were dropped during SPS generation, and we obtained the BindingDB dataset with 129,107 samples. For the classification task, we split the positive and negative pairs using a threshold $\mathrm{pK_i}$ value of 6.5.

The BindingDB protein class dataset contains data from DeepAffinity, which provides protein compound pair sequences with four classes of protein labels obtained using gene ontology (GO) terms. G-protein coupled receptors (GPCR) (n = 73,042), estrogen receptors (ER, n = 495), kinases (n = 3211), and ion channels (n = 7548) were obtained after excluding duplicates from the original dataset. This dataset had a distribution different from that of BindingDB as it was not randomly curated.

Davis [37] studied 442 proteins and 68 compounds, forming 30,056 DT pairs. Similarly, we used Davis without duplicates provided by ELECTRA-DTA. In total, 24,548 samples were used, including the protein-SPS pair. Davis had a severely skewed $\mathrm{pK_i}$ value distribution of approximately 5.0. Using the same threshold value as in BindingDB creates a class imbalance. Thus, we used a threshold value of 6.0 for the Davis model to relieve the class imbalance.

PDBbind [38], was used to assess generalizability of proposed model. As our goal was to predict DTIs using only 1D sequences, we did not use any 3D structured information. We used the curated PDBbind 2016v data from DeepDTAF [39]. The original dataset from DeepDTAF is separated into three parts. The general, refined, and core sets included 9,221, 3,685, and 290 complexes, respectively. We selected protein-ligand complexes that satisfied the following three conditions.

1) The protein sequence length was >500 amino acids.
2) The Smith–Waterman similarity ratio was <40 % compared to all unique BindingDB proteins (n = 1614), which were used as training data.
3) *Homo sapiens* origin

We obtained 218 protein-ligand pairs without duplicates. The binding-affinity distributions of the datasets and SPS lengths are illustrated in Figs. S1 and S2, respectively. Table 2 describes the details of the datasets used in this study.

3. Implementation and discussion

3.1. Experimental setup

We conducted four experiments to assess whether the proposed model is applicable to both classification and regression tasks and whether it is generalizable. Because we designed two tests to account for model generalizability, we did not conduct the novel-pair test, which assesses model performance using only the unseen test pair set. All DL-based models mentioned in the introduction were used as competitive models. Because some of these have been used in either classification or regression tasks, we modified the last output layer shape and predicted the scoring function to perform classification and regression tasks.

Evaluation metrics: We evaluated both binding affinity prediction and interaction prediction scores. To evaluate the regression task, we used the root mean squared error (RMSE), mean squared error (MSE), and concordance index (CI), which can determine the proportion of two random pairs with correctly ordered predicted labels per total number of pairs, R2 score, and Pearson correlation coefficient (PCC). For the classification task, we used the AUROC, AUPRC, accuracy, recall, and precision metrics.

Conducted environments and Hyperparameters: We followed the same hyperparameter settings as described in the papers on the baseline models. We empirically searched for the proposed model hyperparameters using a grid search (Table S1).

3.2. Model performance test

Using the BindingDB and Davis datasets, we assessed model performance for both classification and regression. However, the results can only account for naïve and shallow DTIs prediction performance. This does not ensure generalizability of the model. In addition, because Davis has a severely skewed label and only less samples, which are sparse compared to the feature space, we should interpret the result carefully. We randomly split the data into a 7:1:2 ratio (train/valid/test) five times with the same random state value and split function to ensure an identically split dataset for a fair comparison. Table 3 and Fig. S5 shows the regression results for BindingDB.

The proposed model (KNU-DTI) achieved the best performance in terms of RMSE and CI. We have included the ensemble model results in the table to show the best performance. However, for a fair comparison, we used a model without an ensemble as the criterion. We also tested DeepAffinity because it is a comparative model that uses SPS (Table S2). Table 4 and Fig. S6 shows the classification results for the same dataset. The proposed model achieved the best AUROC, AUPRC, accuracy, and precision values. Although HyperAttentionDTI achieved the best recall score, when it comes to the drug discovery task, precision is more important than recall because a low precision score means a high false-

Table 2
Datasets details.

Dataset	Compound	Protein	Positive pair	Negative Pair	Total	Threshold (pK_i)
BindingDB	83,756	1614	75,606	53,501	129,107	6.5
Davis	68	361	4026	20,522	24,548	6
PDBbind	194	186	106	112	218	6.5
ER	287	6	338	157	495	6.5
GPCR	51,182	323	49,115	23,927	73,042	6.5
Kinase	2367	48	2570	641	3211	6.5
Ion channel	6838	78	4402	3149	7548	6.5

Table 3
Comparison results of the proposed model and baselines on the BindingDB regression.

Models	RMSE (↓)	CI (↑)
HyperAttentionDTI	0.8220 (0.005)	0.8309 (0.001)
GraphDTA (GINConvNet)	0.8279 (0.004)	0.8273 (0.001)
Moltrans	1.008 (0.025)	0.7895 (0.010)
ELECTRA_DTA	0.8062	0.8370 (0.002)
PerceriverCPI	0.8702 (0.011)	0.8193 (0.003)
KNU-DTI (Base)	0.9594 (0.004)	0.7956 (0.001)
KNU-DTI (Full)	**0.7804 (0.002)**	**0.8410 (0.000)**
KNU-DTI (Full, ensemble3)	0.7443 (0.004)	0.8505 (0.000)

positive rate, which can result in meaningless cost and time consumption. The GINConvNet closely followed our model. This implies that representing a compound as a graph presents a promising approach. Although the proposed model did not achieve the best performance for the Davis dataset. However, this does not mean that our model has a lower prediction performance than the other models, because of the data instability of Davis (Table S3). For detailed information regarding the instability of the Davis dataset, please refer to section 5 of the supplementary materials.

3.3. *Ablation test*

To validate the first assumption defined in Section 2.3, we calculated the cosine similarity between the representation vectors obtained from each learning module. The cosine similarity values are summarized in Table 5. Except for the representation vectors learned using the ECFP and D-MPNN, all other vectors exhibited near-zero similarity, confirming their orthogonality. This result substantiates the validity of the first assumption. Furthermore, as stated in the second assumption, we posited that if each learning module effectively captures its target features, the learned representation vectors could be integrated into a single knowledge-united vector through an elemental-wise addition operation. This unified vector is expected to encapsulate richer feature information for DTI prediction. If the newly constructed knowledge-united vector effectively captures more comprehensive information for DTI prediction, its contribution can be analyzed through ablation tests. These tests were conducted to verify both the second and third assumptions in section 2.3. Both of the protein and compound transformer encoder modules were used as the base modules, whereas the others were used as additional features as ablation conditions. Ablation test results show that the ECFP representation vector has a significant influence on model performance (Table 6). Furthermore, as expected, we could further improve our model performance by using each additional feature, respectively. This suggests that the representation vectors learned by all learning modules successfully capture information necessary for improving predictive performance. However, the ablation tests revealed one notable exception. When the ECFP representation vector was combined with the D-MPNN representation vector, no performance improvement was observed compared to using the ECFP representation vector alone. Interestingly, this aligns with the observation that the ECFP and D-MPNN representation vectors were the only pair failing to exhibit orthogonality based on cosine similarity. This indicates that the information learned through ECFP and the D-MPNN learning module is not entirely independent and includes some redundant features. Considering that D-MPNN focuses on the topological features of 2D compound graphs, while ECFP emphasizes molecular structural features through algorithmic descriptors, it is plausible that these two modules could capture overlapping information.

3.4. *Protein class dependent test*

We assessed the generalizability of our model using a protein class-dependent test. The same BindingDB dataset (n = 129,107) used in the model performance test was randomly split into training and validation sets with a 0.95:0.05 ratio. To maximize the amount of data available for training, only 5 % of the data was allocated for validation. The hyperparameters were determined through the 3.2 Model performance test, and the test set was prepared as an external dataset rather than being split from the training data. This allowed us to allocate more data for training without compromising the evaluation process. The protein class-labeled dataset, selected based on GO terms, was used as the test set. Although the training and test sets originated from the same database, the dependency between them was significantly reduced due to their distinct selection methods: the training set was curated through

Table 5
Cosine similarities between representation vectors.

Representation vector pair		Cosine similarity
Protein	O_{sps}/O_{Tp}	0.07
Compound	O_{Tc}/O_{Gc}	−0.04
	O_{Tc}/O_{Mc}	0.04
	O_{Gc}/O_{Mc}	**0.23**

Table 4
Comparison results of the proposed model and baselines on the BindingDB classification.

Models	AUROC	AUPRC	ACC	Recall	Precision
TransformerCPI	0.8816 (0.003)	0.9054 (0.002)	0.8036 (0.001)	0.8187 (0.013)	0.8458 (0.007)
HyperAttentionDTI	0.9050 (0.001)	0.9249 (0.003)	0.8350 (0.002)	**0.9011 (0.005)**	0.8319 (0.005)
GraphDTA (GINConvNet)	0.9065 (0.003)	0.9232 (0.003)	0.8341 (0.004)	0.8455 (0.006)	0.8672 (0.002)
Moltrans	0.8525 (0.011)	0.8815 (0.012)	0.7807 (0.012)	0.8606 (0.008)	0.7866 (0.014)
perceriverCPI	0.8801 (0.001)	0.9053 (0.003)	0.8142 (0.020)	0.8331 (0.020)	0.8337 (0.008)
KNU-DTI (Full)	**0.9121 (0.001)**	**0.9319 (0.001)**	**0.8383 (0.005)**	0.8490 (0.016)	**0.8724 (0.004)**
KNU-DTI (Full, ensemble3)	0.9230 (0.001)	0.9413 (0.001)	0.8496 (0.003)	0.8583 (0.014)	0.8827 (0.006)
KNU-DTI (baseline)	0.8730 (0.002)	0.8965 (0.003)	0.7989 (0.003)	0.8165 (0.011)	0.8363 (0.004)

Table 6
Ablation test with Binding DB.

Ablation Settings			MSE (↓)	RMSE (↓)	CI (↑)	R2 (↑)
SPS	D-MPNN	ECFP				
X	X	X	0.9282 (0.007)	0.9634 (0.003) (−19.3 %)	0.7954 (0.000)	0.6178 (0.003)
O	X	X	0.8780 (0.015)	0.9370 (0.008) (−17.0 %)	0.8004 (0.002)	0.6385 (0.006)
X	O	X	0.8531 (0.010)	0.9236 (0.006) (−15.8 %)	0.8030 (0.001)	0.6487 (0.004)
X	X	O	0.6393 (0.006)	0.7995 (0.004) (−2.7 %)	0.8357 (0.001)	0.7368 (0.002)
O	O	X	0.8218 (0.005)	0.9065 (0.003) (−14.2 %)	0.8080 (0.001)	0.6616 (0.002)
O	X	O	0.6113 (0.006)	0.7819 (0.004) (−0.5 %)	**0.8419 (0.000)**	0.7483 (0.002)
X	O	O	0.6391 (0.005)	0.7994 (0.003) (−2.7 %)	0.8365 (0.001)	0.7368 (0.002)
O	O	O	**0.6053 (0.003)**	**0.7780 (0.002)** (0.0 %)	0.8413 (0.000)	**0.7508 (0.001)**

*Note: The third value in the RMSE column indicates a declining ratio relative to the top score.

algorithmic filtering of invalid data, whereas the test set was constructed based on GO term classifications. Additionally, Smith-Waterman alignment scores between the training and test datasets showed an average sequence similarity of less than 30 %, further demonstrating the reduced overlap between them. These observations suggest that the protein class-dependent test provides a partial evaluation of the model's generalizability. Protein class-dependent test results are shown in Table 7 and Fig. S7. The proposed model achieved the best performance for kinase, GPCR, and ion channel classes. However, the ER class results show that the simple transformer-encoder-based models achieved the best performance. Notably, the ER class had only six protein samples, which was significantly lower than the other classes. This can introduce a high variance into the results, causing inaccuracies.

3.5. *Extra domain test*

As mentioned above, we cannot ensure model generalizability by using only a protein class-dependent test. Therefore, we conducted an extra domain test. We used the same training/validation set used in the protein class-dependent test, but used PDBbind data as a test set. Because a test set which has low sequence similarity compared to the training set was used (Section 2.6), the generalizability of our model can now be ensured (Table 8 and Fig. S8). Our model achieved the best performance overall. This indicates that our model has better generalizability than other baseline models. Although our model generally achieved the best performance, the extra-domain test RMSE was significantly higher than that of the model performance test (Section 3.2). This implies that the generalizability of the proposed model is better than that of the baseline models; however, we cautiously suggest that the absolute generalizability of our model is sufficient for application in real-world studies.

3.6. *Case study*

We demonstrated that the proposed model effectively predicted DTIs. However, from the perspective of drug discovery, the DTI prediction model can be combined with a molecular generative model to create novel compound sequences based on target proteins. Sequence-based DTI prediction models present the advantage that they are easy to understand and apply to novel drug-target protein pairs. Furthermore, because they promptly return the prediction results, they do not cause bottlenecks. In other words, they can be adopted as binding-affinity evaluation modules for generative molecular models. However, as mentioned previously, sequence information retrieval remains a challenge. Three-dimensional (3D) structure-based docking simulators use sophisticated 3D interaction information obtained from real-world experiments. Thus, they are more precise than sequence-based DTIs prediction methods. However, the docking simulators incur high computational costs resulting in a high time complexity. Thus, although a docking simulator can be applied to high-throughput analysis that

Table 7
Protein class dependent test.

Protein Class	Model	MSE (↓)	RMSE (↓)	CI (↑)	R2 (↑)	PCC (↑)
Kinase (n = 3211)	Moltrans	4.5670	2.1371	0.4941	−1.3538	−0.0142
	GraphDTA(GINConvNet)	3.8158	1.9534	0.6012	−0.9676	0.3189
	HyperAttentionDTI	3.4217	1.8498	0.5437	−0.7644	0.1370
	perceiverCPI	4.9380	2.2222	0.5973	−1.5463	0.3239
	KNU-DTI(FULL)	**3.2089**	**1.7913**	**0.6270**	**−0.6547**	**0.3791**
	KNU-DTI(Baseline)	3.9074	1.9767	0.4975	−1.0138	−0.0205
GPCR (n = 73,042)	Moltrans	2.0176	1.4204	0.5376	−0.2486	0.1213
	GraphDTA(GINConvNet)	2.2113	1.4870	0.5557	−0.3684	0.1846
	HyperAttentionDTI	1.9587	1.3995	0.5384	−0.2121	0.1278
	perceiverCPI	2.0906	1.4459	**0.5833**	−0.2937	**0.2805**
	KNU-DTI(FULL)	**1.8318**	**1.3534**	0.5822	**−0.1336**	0.2766
	KNU-DTI(Baseline)	2.3590	1.5359	0.5211	−0.4599	0.0710
ER (n = 495)	Moltrans	2.2962	1.5153	0.5483	**−0.2570**	0.1655
	GraphDTA(GINConvNet)	3.6589	1.9128	0.5578	−1.0122	**0.1903**
	HyperAttentionDTI	2.7541	1.6596	0.4382	−0.5147	−0.1777
	perceiverCPI	3.1235	1.7673	**0.5618**	−0.7178	0.1900
	KNU-DTI(FULL)	2.6881	1.6395	0.5375	−0.4783	0.1148
	KNU-DTI(Baseline)	**2.2943**	**1.5147**	0.5145	−0.2995	0.0662
Channel (n = 7548)	Moltrans	2.2828	1.5109	0.5393	−0.2437	0.1269
	GraphDTA(GINConvNet)	2.3999	1.5492	0.5435	−0.3063	0.1196
	HyperAttentionDTI	2.2810	1.5103	0.5456	−0.2415	0.1336
	perceiverCPI	2.8678	1.6935	0.5820	−0.5609	0.2628
	KNU-DTI(FULL)	**2.1519**	**1.4669**	**0.5883**	**−0.1713**	**0.2709**
	KNU-DTI(Baseline)	2.7516	1.6588	0.5038	−0.5002	0.0384

Table 8
PDB extra domain test.

Protein Class	Model	MSE(↓)	RMSE(↓)	CI(↑)	R2(↑)	PCC(↑)
PDB (n = 218)	Moltrans	6.7882	2.6054	0.5913	−1.1249	0.1381
	GraphDTA	3.6071	1.8992	0.5343	−0.1085	0.0709
	HyperAttention	3.2008	1.7891	0.5967	0.0164	0.3327
	perceiverCPI	3.2657	1.8071	0.6134	−0.0035	0.3290
	KNU-DTI(FULL)	**2.7839**	**1.6685**	**0.6604**	**0.1445**	**0.4383**
	KNU-DTI(Baseline)	5.2026	2.2809	0.4898	−0.5987	−0.0310

does not require immediate responses, in the end-to-end generative models, it can be a rate-limiting step in the workflow. In this section, a case study showing DTIs prediction score correlations between a 3D docking method and the proposed sequence-based model is presented. We demonstrated correlation heat maps using the following three scores.

1) PDB pK_i/pK_d labels from real-world experiments.
2) Vina docking simulator using Gibbs-free energy score.
3) DTA predicted using pK_i values from the proposed model.

The Vina docking score represents the Gibbs free energy between the drug and target protein. A negative value indicates that a spontaneous reaction is dominant, positive value indicates that a non-spontaneous reaction is dominant, and a zero value indicates equilibrium. We used the UCSF Chimera [40] and AutoDock Vina docking simulators [41] and the same PDBbind dataset as in the extra-domain test (section 3.5). However, during docking process a sample was dropped out because we could not obtain a conclusive docking score. In total, 217 samples were used in this study.

The results of the case study are visualized in a scatter plot (Fig. 2). The samples were divided into three groups, namely monomer, homopolymer, and heteropolymer. The best PCC value between the Vina docking score and the label was −0.38 in the heteropolymer group. In contrast, the best PCC between the predicted pK_i value and the label was 0.49 in the homopolymer group. This can be interpreted as the values predicted by the proposed model, with a positive tendency to be labeled. In addition, the sequence-based predicted pK_i value correlation was higher than the docking score correlation. Unlike the general concept, these results show that in some cases, the sequence-based prediction model has a competitive edge over the docking simulator.

4. Conclusion

In this study, we proposed a sequence-based knowledge-uniting model, KNU-DTI, which learns structural information using algorithmically extracted features such as SPS and ECFPs. We show that our model has the best DTIs prediction ability and generalizability compared with baseline models. Moreover, we demonstrated that these extracted features affect prediction performance and that knowledge uniting can be an effective method using an ablation test. In addition, a case study showed that the sequence-based DTI prediction model presents a competitive edge over the docking simulator in some cases.

However, our model had some limitations. A key limitation of our study lies in the inability to quantitatively establish the relationship between orthogonality and the true independence of information, as well as to precisely assess the degree of redundancy between representation vectors. Secondly, while we demonstrated that training on diverse representation vectors can enhance performance by uniting them into knowledge vectors, this approach also leads to increased model complexity and a heightened risk of overfitting. Addressing these limitations will be a central focus of future work, aiming to provide a more rigorous understanding of the interactions and contributions of individual representation vectors to predictive performance while simultaneously improving information capture and reducing model complexity.

CRediT authorship contribution statement

Ryong Heo: Writing – original draft, Validation, Software, Methodology, Data curation, Conceptualization. **Dahyeon Lee:** Visualization, Validation, Software, Data curation. **Byung Ju Kim:** Writing – original draft, Investigation, Funding acquisition, Conceptualization. **Sangmin Seo:** Formal analysis, Data curation, Conceptualization. **Sanghyun Park:** Project administration, Investigation, Funding acquisition, Conceptualization. **Chihyun Park:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Resources, Project administration, Investigation, Funding acquisition, Conceptualization.

Data availability

Dataset and code for running models used in this study are available on our github page https://github.com/DBpackage/KNU_DTA.

Ethical Statement for solid state ionics

1) This material is the authors' own original work, which has not been previously published elsewhere.

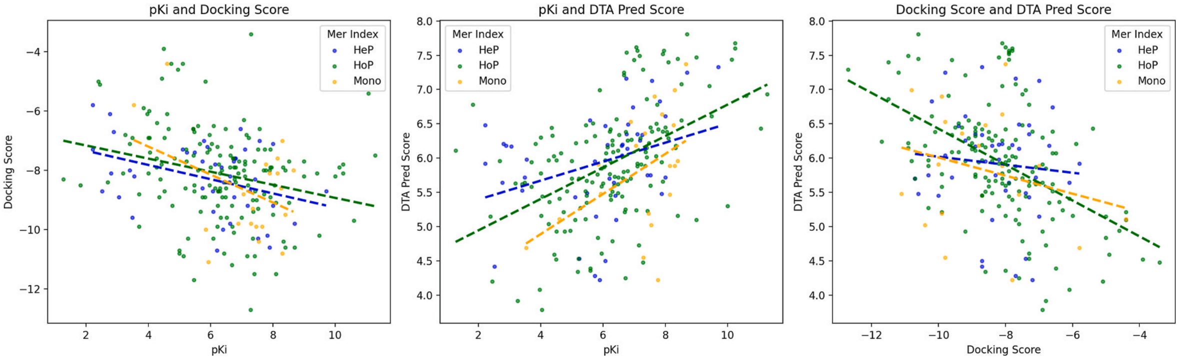

Fig. 2. Case study scatter plots. (a) pK_i(label) and docking scores have a slightly negative correlation. (b) pK_i and predicted pK_i value (DTA) exhibit a positive correlation. (c) Docking score and predicted pK_i value exhibit a negative correlation.

2) The paper is not currently being considered for publication elsewhere.
3) The paper reflects the authors' own research and analysis in a truthful and complete manner.
4) The paper properly credits the meaningful contributions of co-authors and co-researchers.
5) The results are appropriately placed in the context of prior and existing research.
6) All sources used are properly disclosed (correct citation). Literally copying of text must be indicated as such by using quotation marks and giving proper reference.
7) All authors have been personally and actively involved in substantial work leading to the paper, and will take public responsibility for its content.

The violation of the Ethical Statement rules may result in severe consequences.

To verify originality, your article may be checked by the originality detection software iThenticate. See also http://www.elsevier.com/editors/plagdetect.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used ChatGTP v4.0 in order to revise the grammar and improve the readability of some paragraphs. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the published article.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00229822, No. NRF-2022R1F1A1075186, No. RS-2023-00242528).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.compbiomed.2025.109927.

References

[1] D. Sun, W. Gao, H. Hu, S. Zhou, Why 90% of clinical drug development fails and how to improve it? Acta Pharm. Sin. B 12 (7) (2022) 3049–3062.

[2] X. Chen, C.C. Yan, X. Zhang, X. Zhang, F. Dai, J. Yin, et al., Drug–target interaction prediction: databases, web servers and computational models, Briefings Bioinf. 17 (4) (2016) 696–712.

[3] A.C. Nascimento, R.B. Prudêncio, I.G. Costa, A multiple kernel learning algorithm for drug-target interaction prediction, BMC Bioinf. 17 (2016) 1–16.

[4] T. He, M. Heidemeyer, F. Ban, A. Cherkasov, M. Ester, SimBoost: a read-across approach for predicting drug–target binding affinities using gradient boosting machines, J. Cheminf. 9 (1) (2017) 1–14.

[5] M. Karimi, D. Wu, Z. Wang, Y. Shen, DeepAffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks, Bioinformatics 35 (18) (2019) 3329–3338.

[6] Independently recurrent neural network (indrnn): building a longer and deeper rnn, in: S. Li, W. Li, C. Cook, C. Zhu, Y. Gao (Eds.), Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[7] T. Nguyen, H. Le, T.P. Quinn, T. Nguyen, T.D. Le, S. Venkatesh, GraphDTA: predicting drug–target binding affinity with graph neural networks, Bioinformatics 37 (8) (2021) 1140–1147.

[8] F. Scarselli, M. Gori, A.C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE Trans. Neural Network. 20 (1) (2008) 61–80.

[9] D. Weininger, SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules, J. Chem. Inf. Comput. Sci. 28 (1) (1988) 31–36.

[10] K. Huang, C. Xiao, L.M. Glass, J. Sun, MolTrans: molecular interaction transformer for drug–target interaction prediction, Bioinformatics 37 (6) (2021) 830–836.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, et al., Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).

[12] K. Huang, C. Xiao, L. Glass, J. Sun (Eds.), Explainable Substructure Partition Fingerprint for Protein, Drug, and More, NeurIPS Learning Meaningful Representation of Life Workshop, 2019.

[13] J. Mistry, A. Bateman, R.D. Finn, Predicting active site residue annotations in the Pfam database, BMC Bioinf. 8 (1) (2007) 1–14.

[14] L. Chen, X. Tan, D. Wang, F. Zhong, X. Liu, T. Yang, et al., TransformerCPI: improving compound–protein interaction prediction by sequence-based deep learning with self-attention mechanism and label reversal experiments, Bioinformatics 36 (16) (2020) 4406–4414.

[15] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient Estimation of Word Representations in Vector Space, 2013 arXiv preprint arXiv:13013781.

[16] Language modeling with gated convolutional networks, in: Y.N. Dauphin, A. Fan, M. Auli, D. Grangier (Eds.), International Conference on Machine Learning, PMLR, 2017.

[17] Q. Zhao, H. Zhao, K. Zheng, J. Wang, HyperAttentionDTI: improving drug–protein interaction prediction by sequence-based deep learning with attention mechanism, Bioinformatics 38 (3) (2022) 655–662.

[18] N.-Q. Nguyen, G. Jang, H. Kim, J. Kang, Perceiver CPI: a nested cross-attention network for compound–protein interaction prediction, Bioinformatics 39 (1) (2023) btac731.

[19] K. Yang, K. Swanson, W. Jin, C. Coley, P. Eiden, H. Gao, et al., Analyzing learned molecular representations for property prediction, J. Chem. Inf. Model. 59 (8) (2019) 3370–3388.

[20] D. Rogers, M. Hahn, Extended-connectivity fingerprints, J. Chem. Inf. Model. 50 (5) (2010) 742–754.

[21] Z. Zhu, Z. Yao, G. Qi, N. Mazur, P. Yang, B. Cong, Associative learning mechanism for drug-target interaction prediction, CAAI. Trans. Intell. Technol. 8 (4) (2023) 1558–1577.

[22] Z. Zhu, Z. Yao, X. Zheng, G. Qi, Y. Li, N. Mazur, et al., Drug–target affinity prediction method based on multi-scale information interaction and graph optimization, Comput. Biol. Med. 167 (2023) 107621.

[23] B.-Y. Sun, H.-L. Wang, W.-B. Li, H.-J. Wang, J. Li, Z.-Q. Du, Constructing and combining orthogonal projection vectors for ordinal regression, Neural Process. Lett. 41 (2015) 139–155.

[24] J. Cheng, A.Z. Randall, M.J. Sweredoski, P. Baldi, SCRATCH: a protein structure and structural feature prediction server, Nucleic. acids. Res. 33 (suppl_2) (2005) W72–W76.

[25] P. Gage, A new algorithm for data compression, C Users J. 12 (2) (1994) 23–38.

[26] T. Lin, Y. Wang, X. Liu, X. Qiu, A Survey of Transformers. AI Open, 2022.

[27] M. Krenn, F. Häse, A. Nigam, P. Friederich, A. Aspuru-Guzik, Self-referencing embedded strings (SELFIES): a 100% robust molecular string representation, Mach. Learn.: Sci. Technol. 1 (4) (2020) 045024.

[28] A.M. Bran, P. Schwaller, Transformers and large language models for chemistry and drug discovery. Drug Development Supported by Informatics, Springer, 2024, pp. 143–163.

[29] J. Gilmer, S.S. Schoenholz, P.F. Riley, O. Vinyals, G.E. Dahl (Eds.), Neural Message Passing for Quantum Chemistry. International Conference on Machine Learning, PMLR, 2017.

[30] A. Gaulton, L.J. Bellis, A.P. Bento, J. Chambers, M. Davies, A. Hersey, et al., ChEMBL: a large-scale bioactivity database for drug discovery, Nucleic. acids. Res. 40 (D1) (2012) D1100–D1107.

[31] H.L. Morgan, The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service, J. Chem. Doc. 5 (2) (1965) 107–113.

[32] G. Landrum, RDKit: Open-Source Cheminformatics, 2006.

[33] Linguistic regularities in continuous space word representations, in: T. Mikolov, W-t Yih, G. Zweig (Eds.), Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2013.

[34] J. Tang, A. Szwajda, S. Shakyawar, T. Xu, P. Hintsanen, K. Wennerberg, et al., Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis, J. Chem. Inf. Model. 54 (3) (2014) 735–743.

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

[36] J. Wang, N. Wen, C. Wang, L. Zhao, L. Cheng, ELECTRA-DTA: a new compound-protein binding affinity prediction model based on the contextualized sequence encoding, J. Cheminf. 14 (1) (2022) 1–14.

[37] M.I. Davis, J.P. Hunt, S. Herrgard, P. Ciceri, L.M. Wodicka, G. Pallares, et al., Comprehensive analysis of kinase inhibitor selectivity, Nat. Biotechnol. 29 (11) (2011) 1046–1051.

[38] R. Wang, X. Fang, Y. Lu, S. Wang, The PDBbind database: collection of binding affinities for protein– ligand complexes with known three-dimensional structures, J. Med. Chem. 47 (12) (2004) 2977–2980.

[39] K. Wang, R. Zhou, Y. Li, M. Li, DeepDTAF: a deep learning method to predict protein–ligand binding affinity, Briefings Bioinf. 22 (5) (2021) bbab072.

[40] E.F. Pettersen, T.D. Goddard, C.C. Huang, G.S. Couch, D.M. Greenblatt, E.C. Meng, et al., UCSF Chimera—a visualization system for exploratory research and analysis, J. Comput. Chem. 25 (13) (2004) 1605–1612.

[41] O. Trott, A.J. Olson, AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading, J. Comput. Chem. 31 (2) (2010) 455–461.