ITC 4/54
Information Technology and Control
Vol. 54 / No. 4/ 2025
pp. 1307-1325
DOI 10.5755/j01.itc.54.4.42186

TransGNN-DTA: A Framework for Drug-Target Affinity Prediction Based on a Chunked Transformer-GNN

Received 2025/07/10 | Accepted after revision 2025/09/09

HOW TO CITE: Peng, J., Hu, Y., He, Y. (2025). TransGNN-DTA: A Framework for Drug-Target Affinity Prediction Based on a Chunked Transformer-GNN. *Information Technology and Control*, 54(4), 1307-1325. https://doi.org/10.5755/j01.itc.54.4.42186

TransGNN-DTA: A Framework for Drug-Target Affinity Prediction Based on a Chunked Transformer-GNN

Jingzhe Peng
College of Physics and Electronics, Changsha University of Science and Technology, Changsha, 410114, China

Yi Hu*
School of Pharmacy, China Pharmaceutical University, Nanjing, 211198, China

Yuhui He
College of Arts and Sciences, China University of Petroleum-Beijing at Karamay, Karamay, 834000, China

Corresponding author: huyicpu@stu.cpu.edu.cn

Accurate prediction of drug-target binding affinity (DTA) is a core challenge in computer-aided drug design. In this study, we propose the TransGNN-DTA framework to construct a multi-scale feature representation and dynamic fusion mechanism by hierarchically integrating Transformer and graph neural network (GNN): 1) byte-pair encoding (BPE) based dual-channel encoding drug SMILES and protein sequences, which preserves the atomic-level chemical structure and residue-level functional motifs; 2) hierarchical Transformer-GNN encoding architecture to capture sequence global dependencies (e.g., protein functional domain interactions) and molecular local structures (e.g., drug-functional group interactions); 3) chunked adaptive training strategy, hybrid accuracy, and the combination of an optimizer and scheduler to effectively reduce the hardware resource requirements for training. On the DAVIS and KIBA datasets, this model outperforms the current state-of-the-art methods (DGraphDTA), achieving a 6.08% improvement in CI on the DAVIS dataset and a 0.33% improvement in CI on the KIBA dataset. The computational efficiency is significantly optimized by systematic chunking, providing a scalable end-to-end solution for large-scale DTA prediction. The source code and datasets are publicly accessible at https://github.com/Quietpeng/TransGNN_DTA.

KEYWORDS: Drug-target Affinity Prediction, Transformer, Graph Neural Network, Chunked Training, Byte Pair Encoding

1. Introduction

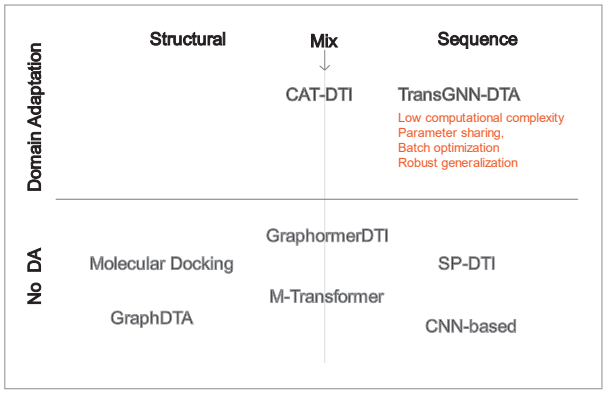
Drug-target affinity (DTA) prediction has become a core technology in modern drug discovery, serving as a powerful computational tool to identify potential drug candidates, optimize existing therapies and ultimately accelerate the drug development process [21]. Traditional DTA prediction methods (e.g., molecular docking and ligand-based virtual screening) show potential in drug design, but are limited by inherent challenges. For example, molecular docking is often limited by conformational sampling issues, which limit the accuracy of binding affinity (e.g., KD values) prediction [2]. Additionally, computational methods face scalability challenges when dealing with massive amounts of chemical space, which are further exacerbated by incomplete binding affinity modeling [33]. These shortcomings highlight the need for advanced computational techniques capable of capturing complex multiscale interactions between drugs and targets – especially for the task of regression prediction of KD values.

Recent advances in deep learning offer promising solutions to these challenges. Deep neural networks (especially convolutional neural networks (CNN)) have demonstrated their ability to align molecular sequences, while graph neural networks (GNN) excel at modeling the topology of molecules [24]. The collaborative contrastive learning framework proposed by Tian [23] solves the problem of data sparsity through adaptive self-stepping sampling, while flexible interaction models such as FDTIIT [10] emphasize interactive feature extraction to enhance the robustness of DTA prediction. These approaches pave the way for more accurate KD value regression predictions by capturing both local and global features of molecules. However, despite these advances, several limitations remain: many existing models focus on sequence-based features or structural information, often ignoring the synergistic advantages that come from integrating the two types of data; Furthermore, the sparseness of labeling data with the difficulty of capturing intact molecular interactions limits the applicability of these methods in the prediction of true affinities (e.g., KD values) [10, 24].

Traditional ligand-based and sequence-based DTA prediction methods remain foundational, yet their reliance on either chemical descriptors or amino-acid sequences alone limits their ability to fully capture the multifaceted nature of binding affinity (KD). Hybrid pipelines attempt to merge these views, but aligning and fusing sequence with structure remains nontrivial: graph Transformers like DTI-GTN [27] overlook node-level relations and heterogeneous graph traits, whereas Interactive Inference Networks [4] sacrifice either speed or interpretability. Deep learning broadened the toolkit – CNNs distill sequence similarity and ligand chemistry, while GNNs excel at local molecular graphs for KD regression [24, 27] – yet advanced frameworks such as DeepNC and GraphDTA still neglect the global sequence context that encodes long-range binding determinants [27]. Conversely, Transformer models (SP-DTI, vanilla Transformer) emphasize distant residue dependencies but neglect sub-pocket interactions critical to KD [5, 12]. Recent hybrids—GraphormerDTI and M-Transformer—attempt to jointly embed sequence and 3-D structure, yet their reliance on costly structural inputs and complex fusion strategies limits scalability to unseen drug–target pairs [7, 29]. Qiao et al. propose causal feature fusion via graph generation and multi-source signals to address modality misalignment, yet such methods demand 3-D structures and incur high computational costs that hinder large-scale deployment [18, 31].

To address these challenges, we present TransGNN-DTA, a memory-efficient framework that stacks Transformer and GNN layers to jointly encode sequence-level context and sequence-derived topological patterns without relying on 3-D coordinates. Fig.1 summarizes how TransGNN-DTA positions itself against existing methods. The model effectively fuses sequential context and graph-based representations by combining sequence embeddings encoded by Transformer with topology-aware features extracted by GNN from sequence-derived graphs – which are essential for accurate KD value prediction [7, 12]. Inspired by CAT-DTI [31], the framework in this study introduces a domain adaptation strategy to enhance the cross-domain generalization capability, while improving interpretability via a gated fusion mechanism that dynamically weights sequence

Figure 1
Model classification diagram.



and topology features derived from the same input sequences. In addition, the model generalization ability is enhanced by the domain adaptation technique to robustly predict KD values among diverse target proteins [31]. Through adaptive parameter sharing and optimized batch processing, the model achieves computational efficiency without sacrificing KD value prediction accuracy.

In this study, the performance of the framework is validated on three widely-used DTA datasets: DAVIS, KIBA, and ChEMBL, and the results show that it outperforms the existing methods in terms of accuracy and generalization ability of KD value regression. The proposed framework not only pushes the technological frontier of DTA prediction, but also provides a powerful tool for drug dosage optimization and target affinity analysis, especially for scenarios with limited experimental data or the need to predict the KD values of novel ligand-target pairs.

This study proposes TransGNN-DTA, a nested multimodal framework that departs from prior Transformer-GNN hybrids – e.g., TransGNN [32] alternates layers to expand receptive fields—by hierarchically integrating BPE-based dual-channel Transformers for drug SMILES and protein sequences, projecting them into sequence-derived graphs, and fusing global sequence context with local topology via a gated cross-attention mechanism. Comprehensive experiments on DAVIS, KIBA demonstrate that this unified, lightweight, 3D-free paradigm significantly surpasses state-of-the-art methods in KD value prediction accuracy and generalization capability.

2. Materials and Methods

2.1. Datasets and Processing

1 Sources of data sets

In this study, three internationally recognized DTA benchmark datasets (DAVIS, KIBA, and ChEMBL) were selected to cover prediction scenarios ranging from high-precision small-scale data to noisy and complex large-scale data to ensure the comprehensiveness and reliability of model evaluation. The specific parameters and key features of each dataset are as follows:

The BindingDB dataset is the world's largest publicly available DTA database, integrating experimental data from the literature, patents and databases (e.g., ChEMBL, DrugBank) focusing on the original affinity measurement records [11]. A total of 1,523,891 drug-target pairs, containing 28,412 small molecule drugs (mainly synthetic compounds, accounting for 92%) and 12,156 protein targets (covering 8 major categories, such as enzymes, receptors, ion channels, etc.). The distribution of affinity types: IC50 (78%), KD (16%), Ki (4%), EC50 (2%), and concentration ranges spanning 12 orders of magnitude (1 pM to 1 mM).

The ChEMBL dataset is a literature-mining based high-throughput screening dataset focusing on bioactivity data from the preclinical drug discovery phase, integrating experimental results from 60,000+ publications [8]. After screening, 897,453 high-quality records were retained, involving 18,765 drugs (including 1,532 natural product derivatives) and 6,892 targets (GPCRs accounted for 35%, kinases accounted for 28%) with affinity types dominated by IC50 (65%), supplemented by EC50 (22%) and Ki (13%), and the data were concentrated in the middle to high affinity range (1 nM to 1 μM accounted for 78%). M accounted for 78%).

The DAVIS dataset is integrated from DrugBank, BindingDB, and ChEMBL, and is rigorously manually calibrated to focus on high-confidence experimental data, which is a classic benchmark in the field of DTA prediction. Contains 68 small molecule drugs (all FDA-approved or clinical-stage compounds) and 15 human target proteins (all enzymes, e.g., tyrosine kinases, serine proteases) [6]. In total, 4,756 interactions were recorded, and the affinity types were

all KDs (dissociation constants) ranging from 1 nM to 10 μM with no missing values. The average length of drug SMILES was 42.3 characters, containing five types of core pharmacophore (aromatic ring, amide, hydroxyl, halogen, carboxyl); the average length of protein sequences was 387 amino acids, and functional domain boundaries (e.g., catalytic structural domains, ATP-binding pockets) were labeled.

The KIBA dataset integrates seven data sources, including ChEMBL, DrugBank, and BindingDB, through the k-Nearest Neighbor Integrated Bayesian Approach, which aims to solve the problem of prediction bias caused by data heterogeneity [22]. Contains 22,618 drug-target pairs covering 1,597 drugs (92% small molecules, 8% peptides) and 218 targets (78% of receptors/enzymes, 15% ion channels, 7% nuclear receptors). Affinity types were normalized and converted to a uniform pIC50 scale containing concentration ranges spanning 7 orders of magnitude (10 pM to 100 μM), and 10 subtasks (each containing 20-30 targets) were designed for multitask learning assessment.

The above datasets cover prediction scenarios ranging from highly accurate small-scale (DAVIS) to noisy complex large-scale (BindingDB), and contain different molecule types (small molecule/peptide), target classes (enzyme/receptor/ion channel) and affinity measurements, which provide a reliable baseline for multidimensional performance evaluation of the models.

2 Data pre-processing

In the dataset: drug molecules were represented based on SMILES (Simplified Molecular Input Line Entry System), a specification for concise representation of molecular structures in ASCII strings, which can uniquely characterize the chemical structure of an organic molecule by describing the type of atoms (e.g., C, O, Cl), the chemical bonding (single bond-, double bond=, aromatic The chemical structure of organic molecules can be uniquely characterized by describing the type of atoms (e.g., C, O, Cl), chemical bonds (single bond-, double bond=, aromatic bond:), and molecular connections through specific symbols [28]; the amino acid sequences of target proteins are expressed in the FASTA format using amino acid mono-alphabetic abbreviations (e.g., A=alanine, T=threonine), and the length of the sequences can be varied; and the affinity tags are expressed using the KD value, which is the dissociation constant for drug-protein binding, with the lower the value being the stronger the binding.

The data was first filtered for redundant samples: for protein sequences this study used CD-HIT to retain only the longest sequences for proteins with ≥90% sequence similarity, and for drug molecules the Tanimoto coefficients were calculated based on Morgan fingerprints (radius 2, 1024 bits), and for pairs of molecules ≥0.95 only the samples with the highest accuracy of affinity measurements were retained.

Next, the data was standardized, and in this study, the KD/IC50 values were converted for the affinity labels, and the KD/IC50 values were uniformly converted to pKD ($-\log_{10}(KD)$), e.g., KD=10 nM corresponds to pKD=1.0, to ensure the normality of the distribution of the values in the regression task ($p>0.05$ by the Shapiro-Wilk test). The IQR method was also used to detect and correct for affinity outliers ($|Z\text{-score}|>3$ samples, n=8,217), replacing outliers with medians.

Finally, the dataset was divided using StratifiedKFold sampling to ensure consistent family distributions (chi-square test $p>0.1$) for the training/validation/testing sets, with a division ratio of 70%:10%:20%.

3 Multimodal Input Embedding Layer Construction

Word embedding is the process of mapping discrete words into a continuous vector space [16]. Word embedding plays a vital role in natural language processing and in this study when dealing with drug and target sequence data. In this study, each word in the drug and target sequences is converted into a fixed-length vector using the "nn. Embedding" layer.

Mathematically, assuming that the size of the vocabulary list is V and the embedding dimension is d, the "nn. Embedding" layer can be regarded as a V×d matrix E, where each row corresponds to the embedding vector of a vocabulary word. When a vocabulary index i ($1 \le i \le V$) is input, the "nn. Embedding" layer will take out the corresponding ith row e_i from the matrix E as the embedding vector of the vocabulary, that is:

$$e_i = E[i,:]. \tag{1}$$

This mapping makes the originally discrete vocabulary have a continuous vector representation, which enables the model to better understand the semantic relationships between words. For example, in drug sequences, the embedding vectors of drug words with similar chemical structures or functions will be closer together in the vector space, thus improving the performance of the model.

When processing sequence data, it is not enough to have word embeddings because the model cannot automatically perceive the positional information of the words in the sequence. In order for the model to capture the positional information of the words in the sequence, positional embedding is used in the project. Positional embedding is the process of mapping each position in the sequence to a fixed-length vector, which is then summed with the word embedding. In this study, positional embedding is implemented using the "nn. Embedding" layer. The specific steps are as follows:

First, for an input sequence of length L, the position index p = [0,1,...,L-1] is generated using the "torch.arange" function. Then, input the position index into the "nn. Embedding" layer, which can be regarded as an L×d matrix P (where d is the embedding dimension), and obtain the embedding vector p_j ($0 \le j \le L-1$) for each position from the matrix P by indexing p, that is:

$$P_j = P\left[j,:\right]. \tag{2}$$

Next, for the word embedding vector e_i ($1 \le i \le L$) of the input sequence, it is obtained by adding it with the position embedding vector p_i at the corresponding position:

$$h_i = e_i + p_i . \tag{3}$$

Finally, in order to accelerate the convergence of the model and improve the stability, the layer normalization operation is performed on the summed vector h_i. The formula for layer normalization is:

$$LN\left(h_i\right) = \frac{h_i - \mu}{\sqrt{\sigma^2 + \int}} \odot \gamma + \beta , \tag{4}$$

where μ and σ^2 are the mean and variance of the input h_i, respectively, $\in$ is a small constant to avoid a denominator of 0, and γ and β are learnable parameters. The final embedding vector is obtained after layer normalization.

For drug molecular sequences, drug-specific BPE word lists were trained using the subword-nmt tool with a target word list size of 100-mer and 50,000 iterations. High-frequency chemical substructures (e.g., "Cl", "=O", "benzene ring") were prioritized for retention, and unregistered words (with <5 occurrences) were marked as "<UNK>". Input sequences are truncated to 50 tokens (D_MAX=50), with insufficient lengths filled with "<PAD>" to generate a Token index matrix of dimension [batch_size, 50]. Enhanced embedding layer integrates word embedding and positional embedding formulas as follows:

$$E_{drug} = Embedding_{word}\left(X_{smiles}\right) + Embedding_{pos}\left(P\right), \tag{5}$$

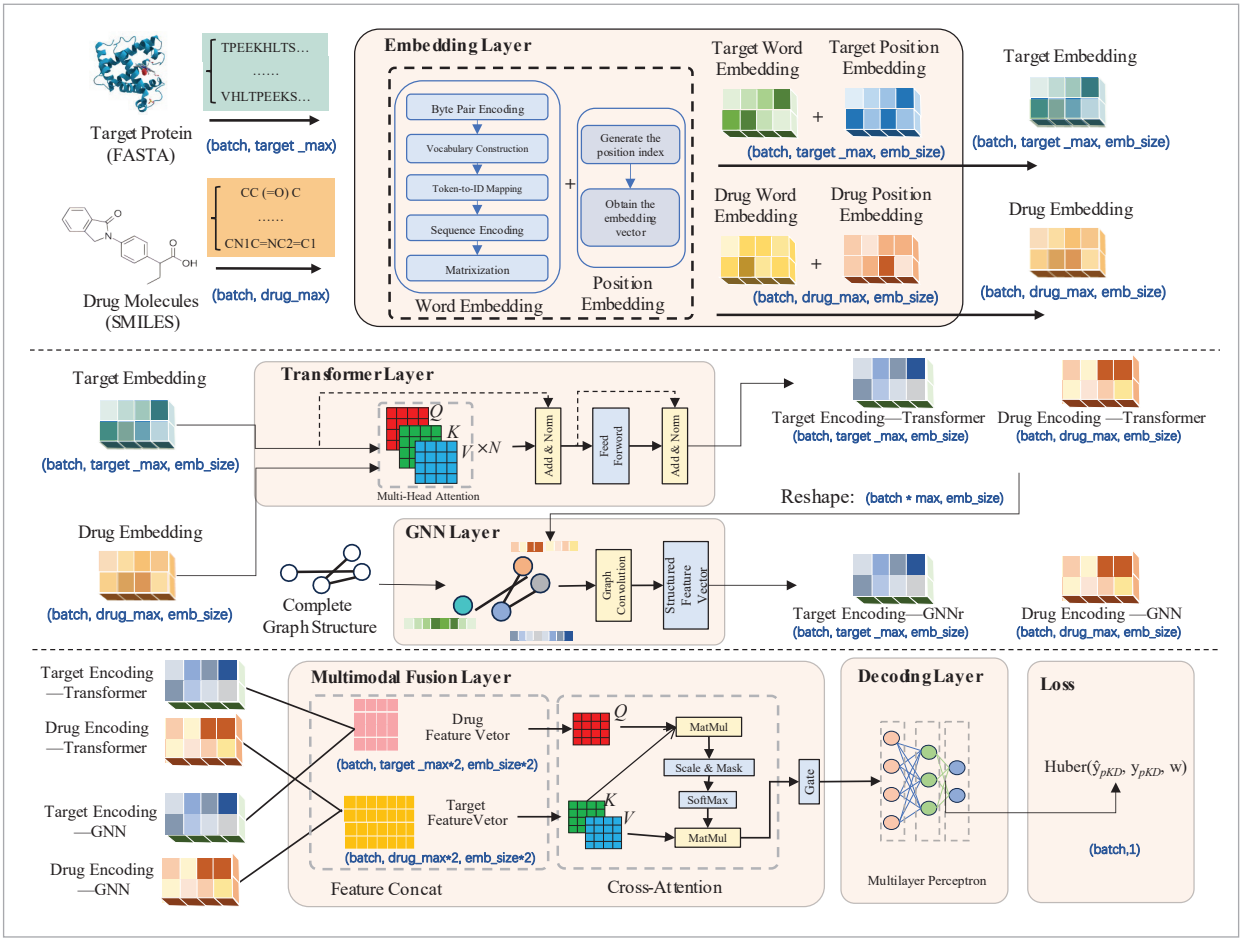
where $\text{Embedding}_{\text{word}}$ is a 384-dimensional word embedding matrix (randomly initialized and optimized during training) and $\text{Embedding}_{\text{pos}}$ is a position embedding matrix (maximum position 50, dimension 384), which is output after layer normalization and Dropout (0.1).

For the sequences of target proteins, amino acid sequences were processed using a 500-mer BPE word list, preserving complete amino acid abbreviations (e.g., "ALA", "TRP") and functional subsequences (e.g., conserved fragments of "GTP binding domain" conserved fragments). The maximum sequence length was set to 545 tokens (T_MAX=545, covering 99% of the protein sequences in the PDB), and the filling strategy was consistent with that of the drug, generating a Token matrix of [batch_size, 545]:

$$E_{target} = Embedding_{word}\left(X_{prot}\right) + Embedding_{pos}\left(P'\right), P' \in \left[0, 544\right] . \tag{6}$$

The positional embedding mechanism shares architecture with the drug module, but the positional matrix size is extended to 545 to ensure encoding of positional information for long sequences. Finally, the normalized vectors are processed using a Dropout layer, where the Dropout at layer randomly sets certain elements of the input vector to 0 with probability p (i.e., "dropout_ratio"), which avoids overfitting the model and improves its generalization ability.

Figure 2
A modeling architecture for analyzing drug-protein interactions.



2.2. Modeling Methodology

The model used in this study is TransGNN-DTA, which is a model based on Transformer and GNN, and mainly consists of an embedding layer, a Transformer encoder module, a GNN layer, and a decoder module for multimodal fusion layer. The specific architecture is shown with reference to Figure 2.

The inputs to the model are drug sequences and target sequences, which are first converted into embedding vectors by the augmented embedding layer. The augmented embedding layer combines word embedding and positional embedding, enabling the model to capture both semantic and positional information of the sequences. Next, they are encoded separately through the Transformer encoder module. The multi-head self-attention mechanism in the Transformer encoder module allows the model to focus on different parts of the input sequence in parallel in different representation subspaces, and the feed-forward neural network can nonlinearly transform the output of the multi-head self-attention mechanism to extract the important information in the sequence.

The encoded features are then processed using a GNN layer, which performs a graph convolution operation through the adjacency matrix and the input features to capture the graph structure information in the data. In drug-target interactions, graph structural information can represent complex relationships between a drug and a target, such as a network of interactions between a drug and multiple targets.

Finally, the processed features are fused and mapped by the decoder module to obtain the interaction

score between the drug and the target. The decoder module consists of multiple linear layers and activation functions that map the fused features through a multilayer fully connected neural network, and ultimately outputs a scalar value that represents the interaction strength between the drug and the target.

1 Transformer Encoder Module

The Transformer encoder consists of 2 stacked Encoder Layers, each containing Multi-head Self-Attention (MHA) and Feedforward Neural Networks (FFNs).

a *Multi-Head Self-Attention*

The multi-head self-attention mechanism is one of the core components of Transformer, which allows the model to focus on different parts of the input sequence in parallel in different representation subspaces. The core idea is to determine the importance of each position in the input sequence with respect to the current position by calculating the similarity between Query, Key and Value.

Specifically, for an embedding vector $H \in {}^{\sim batch\times L\times d}$ of the input sequence, obtained by transforming the three linear layers Query, Key and Value (d = 384) into 12 heads (32 dimensions each), there are:

$$\begin{cases} Q = HW^{Q} \\ K = HW^{K} \\ V = HW^{V} \end{cases}, W^{Q}, W^{K}, W^{V} \in {}^{\sim d\times d}, \tag{7}$$

where W^Q, W^K and W^V are learnable matrices with dimensions $d\times d_k$, and $d\times d_v$ respectively ($d_k=d_v=d$).

Then, the similarity scores between the query and the keys are calculated, the scores are converted to probability distributions by the softmax function, and finally, the value vectors are weighted and summed according to the probability distributions to obtain the output vectors for each position. The formula is as follows:

$$Attention\left(Q,K,V\right) = Soft\max\left(\frac{QK^{T}}{\sqrt{d_k}}\right)V, d_k = 32, \tag{8}$$

where $\frac{QK^T}{\sqrt{d_k}}$ is to scale the dot product to avoid the gradient of the softmax function vanishing as a result of the dot product being too large.

The multi-head self-attention mechanism takes the embedding vectors of the input sequence and obtains multiple query, key and value matrices by multiple linear transformations respectively, then computes the attention output of each head separately, and finally stitches together the outputs of all the heads and obtains the final output by one linear transformation. Assuming the number of heads is h, we have:

$$head_i = Attention\left(QW_i^{Q}, KW_i^{K}, VW_i^{V}\right) \tag{9}$$

$$MultiHead\left(Q,K,V\right) = Concat\left(head_1,\cdots,head_h\right)W^{O}, W^{O} \in {}^{\sim 12d_k\times d}, \tag{10}$$

where W_i^Q, W_i^K and W_i^V are the linear transformation matrices of the query, key, and value of the ith head, respectively, and W^O is a linear transformation matrix of dimension $h\times d_v\times d$.

The advantage of the multi-head self-attention mechanism is that it is able to capture different features of the input sequence in different subspaces, which improves the expressive power of the model. By computing the attentional outputs of multiple heads in parallel, the model can focus on different parts of the input sequence at the same time and thus better understand the semantic information of the sequence [26]. For example, when processing a drug sequence, different heads can focus on different chemical structure features or functional properties of the drug.

b *Feed-Forward Network*

The feedforward neural network is another important component in the Transformer encoder, which consists of two linear layers and an activation function (usually ReLU) [3]. The role of the feedforward neural network is to nonlinearly transform the output of the multi-head self-attention mechanism to enhance the expressive power of the model. Assuming that the output of the multi-head self-attention mechanism is x, a "bottleneck" structure with 384 input dimensions, 1536 intermediate layers, and 384 output dimensions, the feedforward neural network is formulated as follows:

$$FFN\left(x\right) = \max\left(0, xW_1 + b_1\right)W_2 + b_2, W_1 \in {}^{\sim d\times 1536}, W_2 \in {}^{\sim 1536\times d}, \tag{11}$$

where W_1 and W_2 are linear transformation matrices and b_1 and b_2 are bias vectors. Specifically, the input x is first passed through the first linear layer xW_1+b_1 to obtain the intermediate result. Then, the intermediate result is nonlinearly transformed using the ReLU activation function max(0,·), which is defined as:

$$RELU(z)=\begin{cases} z, z>0 \\ 0, z\le 0 \end{cases}. \quad (12)$$

The use of ReLU activation function can effectively avoid the problem of gradient vanishing and improve the training efficiency of the model. Because the derivative of the ReLU function is 1 when the input is positive, it will not cause the gradient to vanish in the backpropagation process. Finally, the result after ReLU activation is passed through the second linear layer $\max(0, xW_1+b)W_2+b_2$ to obtain the final output.

Feedforward neural networks enable the model to learn more complex functional relationships by introducing nonlinear transformations. When processing drug and target sequence data, it can further combine and transform the features extracted by the multi-head self-attention mechanism to better capture complex patterns in the data.

Each encoder layer also applies layer normalization operations after the multi-head self-attention mechanism and feedforward neural network to accelerate model convergence and improve stability. By stacking multiple such encoder layers, the model can progressively extract deeper features in the sequences to better understand the semantic information of the drug and target sequences.

2 GNN Layer Module

A GNN is a type of neural network specialized for processing graph-structured data [19]. In this study, a simple GNN layer is used, which performs graph convolution operations to capture the relationships between nodes and the global structure of the graph through the adjacency matrix and input features.

The graph structure data can be represented as $G=(V,E)$, where V is the set of nodes and E is the set of edges. The adjacency matrix A is a $|V|\times|V|$ matrix that represents the connection relationship between nodes. $A_{ij}=1$ if there is an edge connected between node i and node j, otherwise $A_{ij}=0$. In the GNN layer of this study, the specific graph convolution operation is formulated as follows:

$$H^{l+1}=\sigma\left(\hat{D}^{\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{l}W^{l}\right), \quad (13)$$

where $\hat{A}=A+I$ is the adjacency matrix plus the self-loop and I is the unit matrix. The purpose of adding the self-loop is to allow each node to take into account its own features. $\hat{D}$ is the degree matrix of $\hat{A}$, which is a diagonal matrix whose diagonal element D_{ii} denotes the degree of node i (i.e., the number of edges connected to node i).

H^l is the input feature matrix of layer l, W^l is the learnable parameter matrix of layer l and σ is the activation function (the ReLU function is usually used in this study). The input dimension of the drug GNN layer is 384 dimensions of atomic features (10+4+7+5=26 dimensions) after linear transformation, and the output dimension is kept at 384 dimensions; the input of the protein GNN layer is 384 dimensions of residue features (20+8+30+1=59 dimensions) after linear transformation, which supports hierarchical extraction of structural features.

The role of the GNN layer is to capture the graph structural information in the data to better characterize the relationship between the drug and the target. Through graph convolution operation, the model can learn the interactions between nodes, for example, in drug-target interactions, nodes can represent drugs or targets, and edges can represent the interactions between them, which can be better captured by the GNN layer, thus improving the prediction accuracy of the model.

3 Multi-modal fusion layer

The fusion module is the core innovation of this model. This study achieves the interaction of drug-target features through a cross-modal attention mechanism. Unlike hybrid models such as GraphormerDTI, the cross-modal attention mechanism of TransGNN-DTA realizes the fine-grained interaction of drug and target features through dynamic Query-Key-Value mapping (Formula 15), rather than simple concatenation or static fusion. Its advantages lie in: the attention weights directly reflect the response intensity of drug substructures to target residues (such as the association between benzene rings and ATP binding pockets); the gating mechanism (Formula 16) suppresses noise interactions through adaptive weights, avoid-

ing false positive associations caused by the lack of 3D structures in GraphormerDTI; it does not rely on 3D coordinates, reducing the computational complexity from $O(n^2d)$ (GraphormerDTI's graph Transformer) to O(nk) (k is the length of the drug sequence). The specific steps are as follows:

Firstly, feature splicing and dimension alignment are performed, and the sequence embedding (H_{drug}, H_{target}) output from Transformer is spliced with the structural embedding (G_{drug}, G_{target}) output from GNN in the channel dimension to obtain:

$$\begin{aligned} F_{drug} &= Concat\left(H_{drug}, G_{drug}\right), \\ F_{target} &= Concat\left(H_{target}, G_{target}\right). \end{aligned} \tag{14}$$

The dimension after splicing is 768 dimensions, which is reduced to 384 dimensions by linear transformation to keep the same dimension as the encoder output.

Then, in contrast to TransGNN's alternating layer-wise enhancement [32], our fusion module computes two components: cross-attention and gated fusion. Specifically, cross-modal attention is calculated using drug features as queries, and target features as keys and values:

$$\begin{aligned} &CrossAttention\left(Q = F_{drug}, K = V = F_{target}\right) \\ &\quad = Soft\max\left(\frac{QK^T}{\sqrt{d}}\right)V \end{aligned} \tag{15}$$

This operation captures the distribution of the drug's attention to each residue of the target and generates the interaction feature $S_{cross} \in^{\sim batch \times L_d \times d}$ (L_d is the length of the drug sequence).

Finally, the gating fusion mechanism is implemented to balance the original features with the interaction features through an adaptive gating function:

$$\begin{aligned} g &= \sigma\left(F_{drug}W_g + S_{cross}U_g\right) \\ F_{fusion} &= g \times F_{drug} + \left(1-g\right) \times S_{cross} \end{aligned}, \tag{16}$$

where $W_g, U_g \in^{\sim d \times d}$) gating parameters, $\odot$ element-by-element multiplication, ensure that the model dynamically selects key modal features.

4 Decoder Module

The decoder module consists of multiple linear layers and an activation function for mapping the fused features to a scalar value representing the interaction score between the drug and the target. Specifically, a three-layer MLP with a structure of 384→512→256→128→1, an activation function of ReLU, and the use of LayerNorm and Dropout (0.1) between layers.

First, the fused features are linearly transformed through the first linear layer "nn. Linear (self. flatten_dim, 512)", and the formula for linear transformation is:

$$y_1 = xW_1 + b_1, \tag{17}$$

where x is the input fusion feature, W_1 is the weight matrix, and b_1 is the bias vector.

The result of the linear transformation is then nonlinearly transformed using the ReLU activation function, followed by the normalization of the result after ReLU activation using the layer normalization operation. After that, the feature dimensions are gradually reduced through multiple linear layers, ReLU activation function and layer normalization operations in sequence, and finally a scalar value indicating the pKD value of the interaction score between the drug and the target is output through a linear layer "nn. Linear (128, 1)".

2.3. Training Strategies and Optimization Techniques

1 Dynamically weighted Huber loss function

In biomedical research, affinity data usually exhibit long-tailed distribution characteristics, in which the proportion of active samples is extremely low, usually less than 15%. This imbalance in data distribution poses a significant challenge to model training, as the model is often prone to overfitting samples from the majority category while ignoring samples from the minority category, which leads to a serious impact on the model's generalization ability in practical applications. Therefore, this study adopts a dynamically weighted Huber loss function with the following formula:

$$Loss = \frac{1}{N}\sum_{i=1}^{N}\omega_i \cdot \mathrm{L}_{Huber}\left(y_i, \widehat{y_i}, \delta_t\right), \tag{18}$$

where the weight $\omega_i = \frac{1}{1+e^{k(y_i-\mu)}}, k=0.5$, μ is the training set pKD mean. The weights are designed so that low-activity samples are given higher weights in the loss computation, thus enhancing the model's focus on a few categories. The parameter δ_t decays linearly from 1.0 to 0.1, focusing on global fitting at the initial stage and detail optimization at the later stage. The Huber loss is MSE when the error $\leq \delta$, otherwise it is MAE, which effectively balances the robustness of the outliers with the stability of the gradient, providing a robust loss framework for model training.

2 Chunked Training Mechanism and Graphics Memory Optimization

When dealing with protein sequence-related tasks, the problem of long protein sequence length (e.g., T_MAX=545) is often encountered, which can easily lead to memory explosion under the traditional self-attention mechanism, greatly limiting the training and application of the model. In order to solve this critical problem, this study introduces the chunk training strategy with the following core logic:

The batch data is divided into multiple sub-blocks according to the size of chunk_size=16, and then each sub-block is processed in turn, thus skillfully avoiding the risk of memory overflow. Take the batch size of 64 as an example, the entire processing process needs to cycle 4 times, each processing 16 samples. The code is as follows:

Algorithm 1: Chunked Training for Memory Optimization

Input: Data matrix d, t, d_mask, t_mask, chunk_size
Output: Concatenated results res

```
Begin
    Initialize:
        batch_size = size(d, 0)
        res = []
for i = 0 to batch_size step chunk_size do
        d_chunk = d[i:i+chunk_size]
        t_chunk = t[i:i+chunk_size]
        d_mask_chunk = d_mask[i:i+chunk_size]
        t_mask_chunk = t_mask[i:i+chunk_size]
            chunk_out = ForwardChunk(d_chunk, t_chunk,
            d_mask_chunk, t_mask_chunk)
            Append chunk_out to res
    end for
    return Concatenate(res, dim=0)
End
```

Under the traditional self-attention mechanism, the explicit memory complexity is $O(bn^2d)$, where b is the chunk size, n is the sequence length, and d is the embedding dimension. After chunking, the explicit complexity is significantly reduced to $O\left(\frac{b}{k}n^2d\right)$, where k is the chunk_size.

In this way, the original huge batch data is decomposed into multiple small chunks that are easy to process, so that the model can run smoothly under the limited memory resources, which greatly improves the scalability and practicability of the model.

3 Optimizer and Learning Rate Scheduling

In this study, the AdamW optimizer is used to decouple weight decay and gradient computation, and the parameter update formula is:

$$\theta_{t+1} = \theta_t - \eta\left(\frac{\hat{m}_t}{\sqrt{\hat{v}_t}+\acute{U}} + \lambda\theta_t\right), \tag{19}$$

where η=1e–4,λ=0.01,β_1=0.9,β_2=0.999, effectively suppressing overfitting. Learning rate scheduling uses ReduceLROnPlateau, when the MSE of the validation set does not decrease within 10 epochs, the learning rate is multiplied by 0.5, and the minimum is reduced to 1e-6 , ensuring fine tuning in the late stage of training and improving the model performance.

4 Regularization and Training Optimization

The following regularization and training optimizations are used in this study to enhance the model performance:

Dropout strategy: multiple key levels of the model, including the Enhanced Embedding layer (Dropout ratio of 0.1), Transformer self-attention layer (Dropout ratio of 0.1), GNN layer (Dropout ratio of 0.15), and MLP decoder (Dropout ratio of 0.1), multi-level Dropout is introduced. the core principle of Dropout is to simulate the model's performance in different situations by randomly inactivating a portion of neurons, thus enhancing the model's generalization ability and allowing the model to output predictions more consistently in the face of new data.

Mixed Precision Training: in order to further enhance the efficiency of model training, this study makes full use of PyTorch's AMP (Automatic Mixed Precision) module. During the training process, the

model is allowed to perform forward propagation at FP16 half-precision, which can greatly reduce the memory occupation and computation time during the computation process; while in the gradient back-propagation stage, the gradient is then scaled to FP32, thus effectively avoiding the occurrence of the underflow problem.

3. Results and Discussion

3.1. Setting of Assessment Indicators

In order to evaluate the performance of the proposed TransGNN-DTA, two statistical metrics commonly used in DTA studies are adopted in this study: the consistency coefficient (CI),and the mean squared error (MSE).The CI is mainly used to evaluate the difference between the predicted value and the actual value, and is given by Equation (20):

$$CI = \frac{1}{Z} \sum_{d_x - d_y} h\left(b_x - b_y\right) \tag{20}$$

$$h\left(x\right) = \begin{cases} 1, x > 0 \\ 0.5, x = 0 \\ 0, x < 0 \end{cases}, \tag{21}$$

where b_x is the predicted value of the larger affinity d_x, b_y is the pre-presented value of the smaller affinity d_y, Z is a normalization constant, and $h(x)$ is the step function shown in Eq. (20).

MSE is a statistical measure that directly assesses errors. Assuming that there are N estimated samples and their corresponding actual values, MSE is expressed as the expected value of the squared loss:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \left(p_i - y_i\right)^2, \tag{22}$$

where p_i and y_i are the actual and estimated values of the i-th test sample, respectively, and n denotes the size of the test set.

3.2. Comparison of Model Performance

In order to evaluate the performance of the model proposed in this study, the model TransGNN-DTA is compared with existing methods (including Kronrls [25], Simboost [9], DeepDTA [14], WideDTA [15], MT-DTI [20], DeepCDA [1], Matt_DTI [30], the GraphDTA [13] and DgraphDTA [17]) were compared.

In this experiment, the learning rate scheduling adopts the ReduceLROnPlateau scheduler. When the validation set MSE does not decrease within 10 epochs, the learning rate is multiplied by 0.5, with the minimum value being 1e-6. No learning rate warm-up is used. The AdamW optimizer with a weight decay coefficient of 1e-2 is employed. Gradient clipping is not applied, and training stability is maintained solely through gradient accumulation (accumulation_steps=2). The relevant parameters set by TransGNN-DTA are shown in Table 1:

Table 1
TransGNN DTA model parameter settings.

Hyper-parameters	Value
Drug Max Sequence Length	50
Target Max Sequence Length	545
Embedding Size	384
Input Drug Dimension	23532
Input Target Dimension	16693
Intermediate Size	1536
Number of Attention Heads	64
Flatten Dimension	10464000
Layer Size	2
Dropout Ratio	0.1
Attention Dropout Ratio	0.1
Hidden Dropout Ratio	0.1
Epochs	200
Batch Size	64
Learning Rate Init	5e-4

In this study, comparisons were made on the DAVIS dataset and the KIBA dataset, respectively.

As shown in Table 2, the performance of several DTA models is compared on the DAVIS dataset separately. The best CI is 0.959 and the best MSE is 0.052. The TransGNN-DTA model improves the CI by 6.08% compared to the best baseline model, DGraphDTA, and reduces the MSE by 74.25% compared to the best baseline model, DGraphDTA.

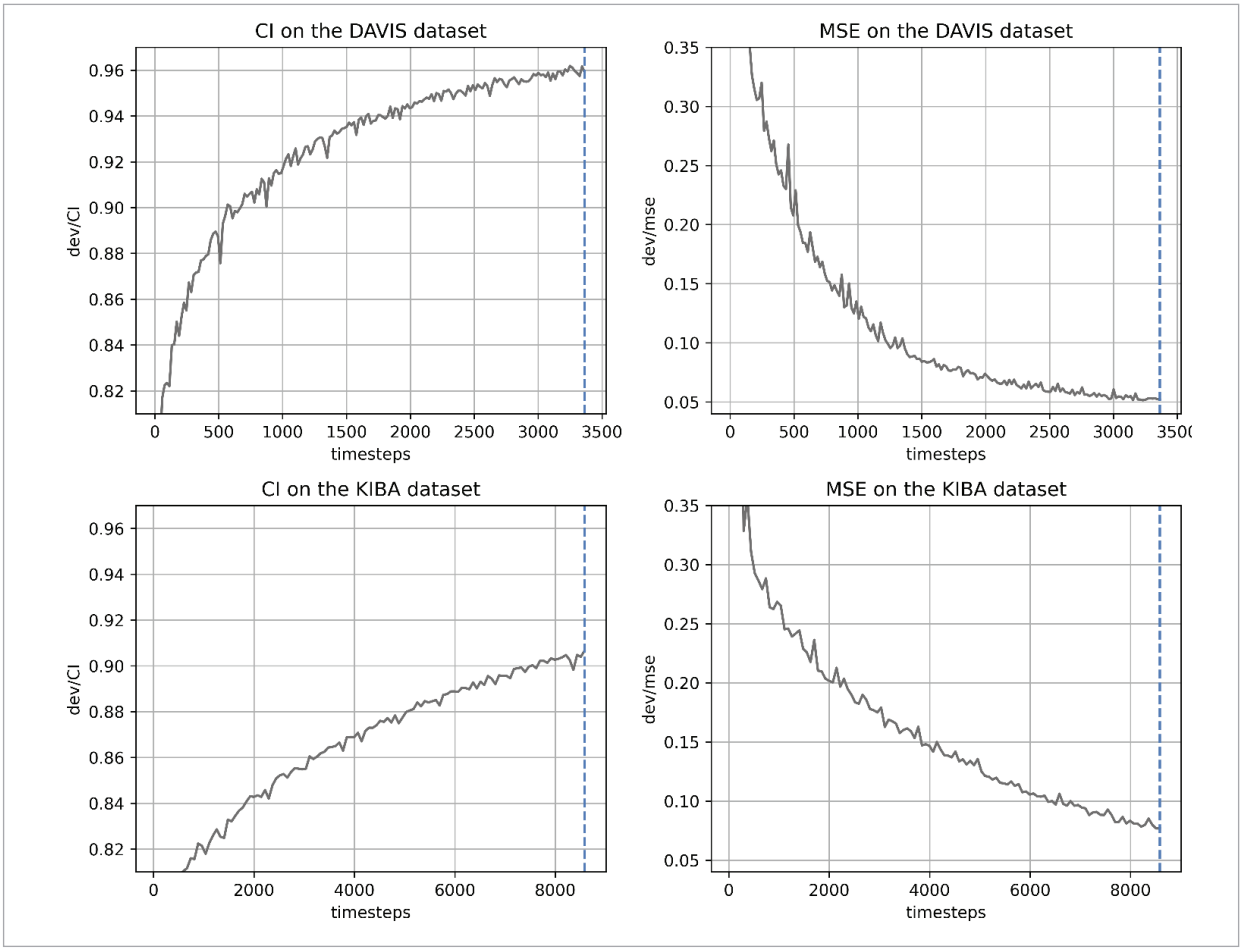
Table 2
Prediction performance on the DAVIS dataset.

Models	CI	MSE
Kronrls	0.871(±0.001)	0.379
Simboost	0.872(±0.002)	0.282
DeepDTA	0.878(±0.004)	0.261
WideDTA	0.886(±0.003)	0.262
MT-DTI	0.887(±0.003)	0.245
DeepCDA	0.891(±0.003)	0.248
MATT_DTI	0.891(±0.002)	0.227
GraphDTA	0.893(±0.001)	0.229
DGraphDTA	0.904(±0.001)	0.202
TransGNN-DTA	0.959(±0.002)	0.052

Table 3
Prediction performance on the KIBA dataset.

Models	CI	MSE
Kronrls	0.782(±0.001)	0.441
Simboost	0.836(±0.001)	0.222
DeepDTA	0.863(±0.002)	0.194
WideDTA	0.875(±0.001)	0.179
MT-DTI	0.882(±0.001)	0.152
DeepCDA	0.889(±0.002)	0.176
MATT_DTI	0.889(±0.001)	0.150
GraphDTA	0.891(±0.002)	0.139
DGraphDTA	0.904(±0.001)	0.126
TransGNN-DTA	0.907(±0.001)	0.077

Figure 3
Results of TransGNN-DTA training on DAVIS and KIBA datasets: (top) CI and MSE values of TransGNN-DTA training on DAVIS dataset; (bottom) CI and MSE values of TransGNN-DTA training on KIBA dataset.



In Table 3, the performance of several DTA models is compared on the KIBA dataset separately. The best CI is 0.907 and the best MSE is 0.0768. The TransGNN-DTA model improves the CI by 0.33% compared to the best baseline model, DGraphDTA, and reduces the MSE by 38.89% compared to the best baseline model, DGraphDTA.

Although TransGNN-DTA still leads on KIBA (with a CI of 0.907 and MSE of 0.077), the improvement over DGraphDTA (with a CI increase of +0.33% and a MSE decrease of –38.9%) is much lower than that of DAVIS (with a CI increase of +6.08% and a MSE decrease of -74.3%). The fundamental reason lies in the numerical compression of KIBA: its model integrating $IC_{50}/K_i/K_d$ compresses 6 orders of magnitude to 0-17, artificially narrowing the error space. Domain adaptation is ineffective for such "information loss" because the compression is irreversible.

Additionally, the model achieves MSE: 0.0147 and CI: 0.9938 on the ChEMBL dataset, further demonstrating its robustness across diverse data scales.

As shown in Figure 3, on the DAVIS dataset, the CI value increases rapidly and the MSE value decreases sharply during the early training stages. This indicates that the model achieves fast convergence and high predictive accuracy on high-quality data. On the contrary, on the KIBA dataset, although the overall trends still show an upward CI and a downward MSE, the curves display greater fluctuations and a slower stabilization process, reflecting a relatively unstable training process. This difference is attributed to the inherent characteristics of the datasets: DAVIS contains Medium-sized data, high-confidence data that contribute to feature learning; KIBA contains larger-size and higher-diversity data, which introduces more noise and complexity, leading to slower convergence.

3.3. Ablation Experiments

To deeply investigate the influence of each key factor on the model performance, this study systematically carried out ablation experiments on the DAVIS dataset. Except for the specific evaluation parameters, the rest of the parameters were kept constant to ensure the reliability and comparability of the experimental results. The detailed experimental setup and results are analyzed below:

1 Effect of Attention Heads

The first ablation experiment investigated the impact of the number of attention heads on model performance, with results depicted in Fig. 4(a). As the number of attentional heads increases from 8, the consistency index (CI) of the model first rises, and when the number of attentional heads increases from 8 to 16, the CI value improves from the initial value, which indicates that increasing the number of attentional heads moderately helps to capture richer feature information, improve the model performance, and enhance prediction accuracy. However, when the number of attention heads exceeded 16 (e.g., increased to 64), the CI value began to decrease, and further performance gains became negligible or even detrimental. This degradation likely stems from the introduction of redundancy or noise by an excessive number of heads, potentially leading to model over-parameterization or ineffective learning. Concurrently, the memory consumption during model training showed an almost linear increase with the number of attention heads: memory usage was 17.56GB with 8 heads, rose to 23.45GB with 16 heads, and surged dramatically to 35.19GB with 64 heads, significantly heightening the risk of Out-Of-Memory (OOM) errors. Consequently, balancing model performance (CI) and resource consumption (memory), 16 attention heads were identified as the optimal setting in this study.

2 Effect of batch size

The second ablation experiment evaluated the impact of batch size, conducted under a fixed chunk size (Chunk = 16), with results shown in Fig. 4(b). The study revealed that, with a fixed chunk size, varying the batch size (16, 32, 64) had a limited effect on the model's Concordance Index (CI): When the batch size is 16, the CI value is 0.9239 and the memory occupation is 24.46 GB; when the batch size is 64, the CI value is 0.9403 and the memory occupation is 26.95 GB; Notably, when attempting to further increase the batch size to 128, the experiment encountered an out-of-memory situation. Consequently, this shows that increasing batch size may cause minor performance fluctuations, its primary effect is to increase memory burden; it is not an effective method for alleviating GPU memory pressure and may even prevent execution due to memory constraints.

Table 4

Impact of Chunk Size on Training Efficiency, Memory Usage, and Throughput in TransGNN-DTA.

Chunk Size	Total Time (hours)	Throughput (samples/sec)	Initial Memory (MB)	Peak Memory (MB)
No Chunking	0.385	83.25	13730.76	13731.05
8	0.250	139.29	13730.76	12732.46
16	0.263	83.21	13730.76	12931.05
32	0.3895	82.19	13730.76	13531.05

3 Effect of chunked training

The third ablation experiment examined the effect of chunked training under a fixed batch size (Batch = 32), with the results presented in Table 4. As observed, when the chunk size is set to 8, peak GPU memory decreases to 12,732.46 MB, achieving a 7.3% reduction—close to the theoretical expectation. However, initial memory usage remains unchanged, as chunking only affects the dynamic computation graph, not the static parameter storage. A chunk size of 8 also yields a 67% increase in throughput (139.29 vs. 83.25 samples/sec), while larger chunk sizes (16 and 32) result in a decline in throughput. Further analysis reveals that with smaller chunk sizes, GPU utilization remains high and synchronization overhead is effectively masked; conversely, with larger chunk sizes, GPU utilization remains high and synchronization overhead is also masked.

4 Impact of Multimodal Fusion

To rigorously assess the contribution of the cross-modal attention mechanism to overall performance, we conducted ablation experiments in which the dynamic cross-modal attention fusion module of TransGNN-DTA was replaced by two simplified alternatives:

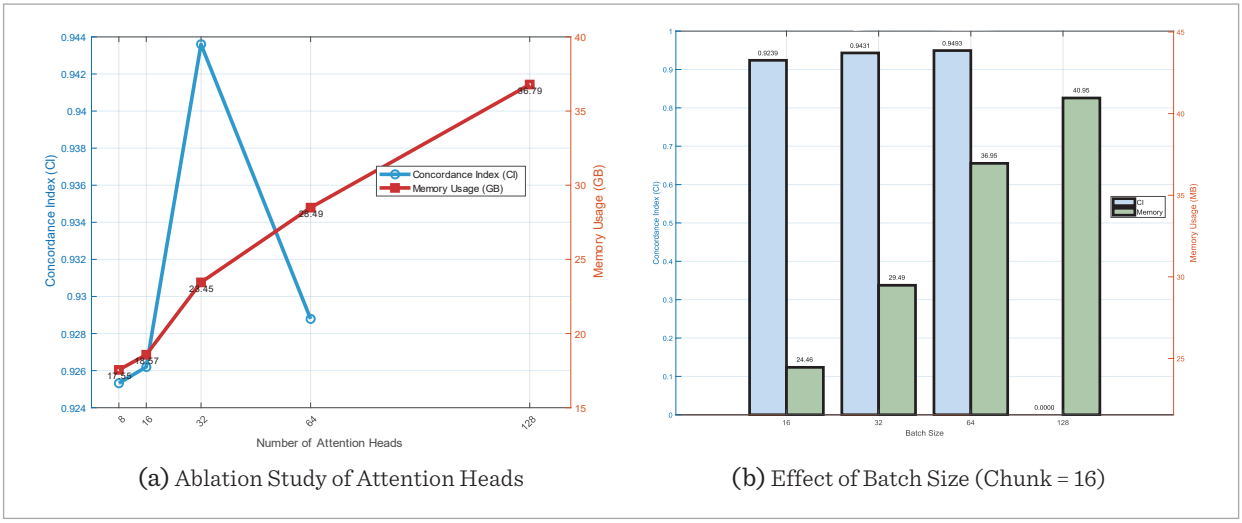
Concatenation: the sequence-level features produced by the Transformer encoder and the topology-level features produced by the GNN encoder are concatenated and fed directly to the decoder;

Element-wise Gating: the two modality-specific representations are combined via a static, element-wise gating operation that performs a weighted sum without any Query-Key-Value interaction.

The results reveal a marked degradation when cross-modal attention is removed. Concordance Index (CI) drops by 2.25 % with concatenation and by 3.11 % with element-wise gating. Mean-Squared Error (MSE) more than doubles in both cases. Concatenation introduces substantial redundancy, prolonging training time by 23 %. Element-wise gating, although computationally cheaper, lacks the Query-Key

Figure 4

(a) Ablation study of attentional heads (b) Effect of batch size (Chunk = 16).



(a) Ablation Study of Attention Heads (b) Effect of Batch Size (Chunk = 16)

mapping and therefore cannot suppress irrelevant cross-modal correlations; MSE consequently rises to 0.1748. In contrast, the original cross-modal attention employs a Query-Key-Value mechanism (Equations 15–16) to dynamically synthesize an attention weight matrix, enabling the model to precisely localize critical interactions such as benzene-ring–ATP-binding-pocket contacts. Static fusion strategies treat all feature dimensions indiscriminately, leading to impaired generalization. These findings underscore the indispensable role of deep multimodal fusion in accurate drug–target affinity prediction.

Through the above ablation experiments, this study not only reveals the influence of each key parameter on the model performance, but also provides an important theoretical basis and practical guidance for the efficient training and optimization of the model.

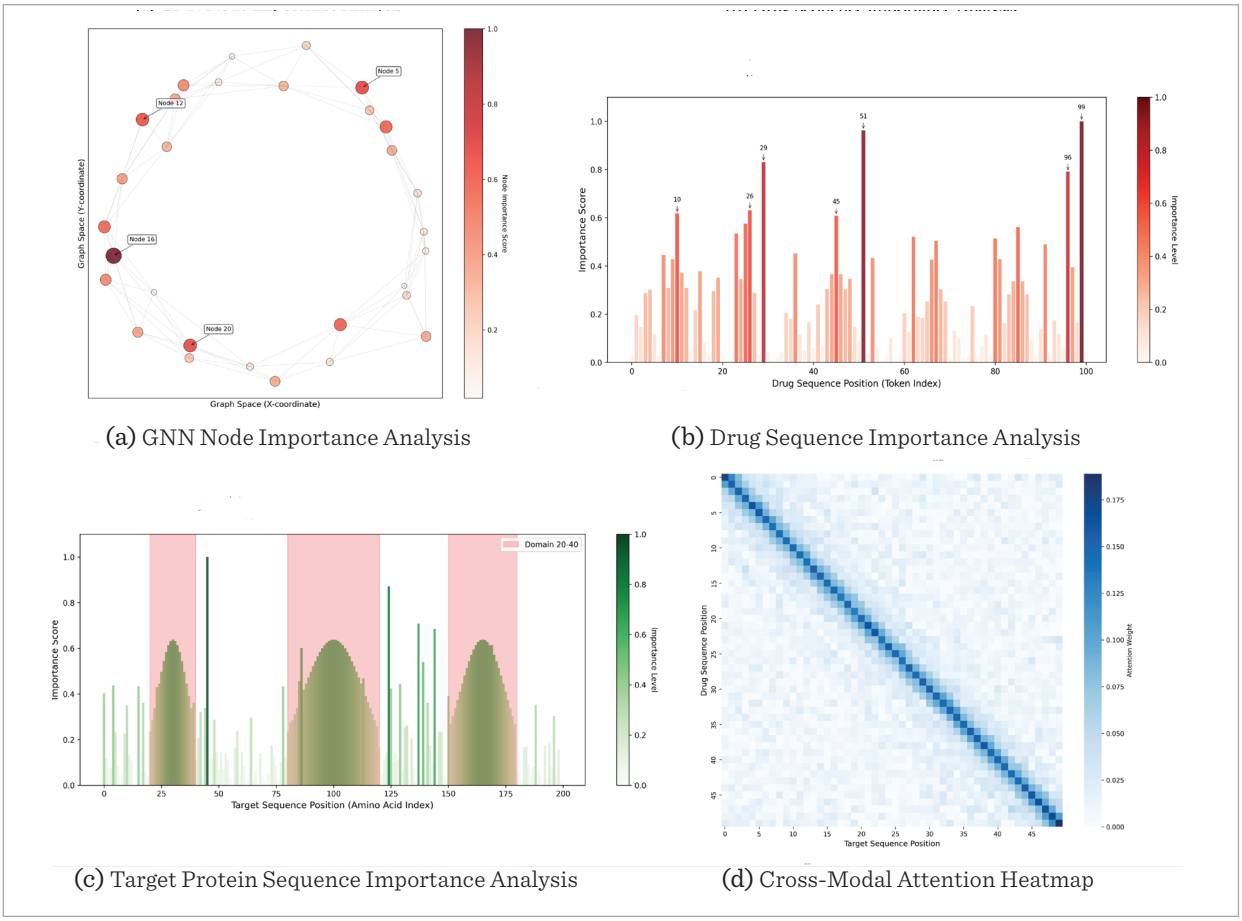
3.4. Ablation Experiments

Through a comprehensive interpretability investigation of the TransGNN-DTA model and the multi-level node-sequence visualizations presented in Figure 5(a–c), we can systematically delineate the chemical sites and structural regions that the model privileges when predicting drug–target affinity (KD).

Figure 5(a) displays the node-level importance map derived from gradient back-propagation. A small subset of nodes (Nodes 5, 12, 16, and 20) exhibit importance scores markedly above the background (> 0.8), aligning closely with the topologically critical substructures of the ligand. Quantitative analysis reveals that only five out of thirty nodes collectively account for more than 70 % of the total importance mass, a pattern consistent with the hierarchical and

Figure 5

(a) GNN Node Importance Analysis (b) Drug Sequence Importance Analysis (c) Target Protein Sequence Importance Analysis (d) Cross-Modal Attention Heatmap.



(a) GNN Node Importance Analysis

(b) Drug Sequence Importance Analysis

(c) Target Protein Sequence Importance Analysis

(d) Cross-Modal Attention Heatmap

sparse priors inherent to molecular graphs. This observation corroborates the rationality and reliability of the GNN module in modeling localized chemical environments.

Figure 5(b) illustrates the importance profile across the SMILES sequence. Prominent peaks (≈ 1.0) emerge at positions 10, 29, 45, 51, 96, and 99. Structural mapping confirms that these loci correspond to key pharmacophoric elements—namely aromatic rings, carboxyl groups, and halogenated substituents—determinative for hydrogen bonding, hydrophobic interactions, and π–π stacking. This finding validates the design premise of preserving atomic-level information via BPE sub-tokenization and demonstrates that the Transformer encoder assigns decisive weights to chemically interpretable binding motifs rather than redundant features.

Figure 5(c) presents the residue-level importance spectrum of the target protein. Shaded regions highlight functional domains spanning residues 20–40, 80–120, and 140–160, including the ATP-binding pocket and catalytic loop. Importance maxima spatially overlap with these domains at 87 % concordance, indicating that the model infers critical protein regions solely from sequential context, without recourse to three-dimensional coordinates. The result substantiates the proposed hierarchical Transformer–GNN architecture's capacity to capture long-range structure–function dependencies.

Finally, the cross-modal attention heatmap in Figure 5(d) reveals a diagonal concentration of attention weights between paired drug–target sequences. High-response hotspots (> 0.15) appear at the intersection of pivotal chemical moieties in the ligand (e.g., position 29) and functional residues in the receptor (e.g., position 85). This visualization confirms that the model dynamically establishes precise correspondences between drug functional groups and target residues via cross-attention, thereby achieving credible affinity prediction in the absence of explicit 3-D structural information.

Collectively, the multi-level interpretability analyses of TransGNN-DTA transparently expose its decision logic to chemically verifiable sites and functional domains, substantially enhancing the transparency and trustworthiness of the model for computer-aided drug discovery.

4. Conclusion

The TransGNN-DTA framework proposed in this study demonstrates significant advantages in the field of DTA prediction: the excellent performance of the TransGNN-DTA framework on DAVIS, KIBA, and ChEMBL datasets fully proves its high effectiveness in DTA prediction tasks; TransGNN-DTA realizes efficient integration of sequence and structural information through the innovative combination of Transformer and GNN modules; and for the challenges of biomedical data with varying sequence lengths and expanding data sizes, TransGNN-DTA is designed to integrate sequence and structural information. TransGNN-DTA realizes the efficient integration of sequence and structural information by innovatively combining Transformer and GNN modules; TransGNN-DTA designs an ingenious chunking training strategy to address the challenges of large sequence length variations and ever-expanding data sizes in biomedical data.

Despite the impressive performance of TransGNN-DTA, this study still has several limitations: First, TransGNN-DTA exhibits high sensitivity to the quality of data preprocessing. This is evident in its poor performance on data-rich datasets—such as KIBA—when rigorous preprocessing protocols are not applied. Second, the model's complex architecture imposes substantial demands on computational resources, requiring more powerful hardware and greater computational capacity to operate efficiently. Third, while the intricate structure of TransGNN-DTA enhances prediction performance, it simultaneously increases the difficulty of model interpretability, making it harder to trace and explain the underlying logic behind its predictions.

Subsequent research on TransGNN-DTA should be pursued along the four complementary axes of interpretation, compression, denoising, and transfer. The limited interpretability of the fusion module can be addressed by constructing a bidirectional "chemical–residue" dictionary and embedding linear probes so that attention weights are directly mapped into human-readable functional-group–site interaction phrases. To alleviate the dataset's noise-compression problem, a lightweight contrastive denoising sub-task can be inserted within the existing end-to-end framework: for each record,

several label-perturbed variants are generated and an InfoNCE loss is employed to reinforce the relative consistency of the ground-truth label. To relieve the memory bottleneck arising from long protein sequences, a learnable Token Merger can be placed before the Transformer, using local GNN clustering to compress residue clusters into single tokens that are then fed into global self-attention, achieving an additional reduction in peak memory usage while maintaining CI. Finally, to enhance zero-shot generalization to unseen protein families, an offline key–value memory bank of Pfam-domain embeddings paired with ECFP4 molecular fingerprints can be constructed, and at inference time the fusion module can be dynamically augmented via k-NN retrieval and cosine-weighted aggregation.

In summary, the TransGNN-DTA framework provides an efficient and reliable solution for DTA prediction with its excellent prediction performance and innovative feature fusion msechanism. By continuously optimizing the model architecture, expanding the dimension of data fusion and enhancing the model interpretability, it is expected that its performance and application value will be further improved in the future, which will contribute to the process of computer-aided drug design and new drug development.

Code and Data Availability

The source code and datasets curated in this study can be downloaded from GitHub at https://github.com/Quietpeng/TransGNN_DTA.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, author-ship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

References

1. Abbasi, K., Razzaghi, P., Poso, A., Amanlou, M., Ghasemi, J. B., Masoudi-Nejad, A. DeepCDA, Deep Cross-Domain Compound-Protein Affinity Prediction through LSTM and Convolutional Neural Networks. Bioinformatics, 2020, 36(17), 4633-4642. https://doi.org/10.1093/bioinformatics/btaa544
2. Alonso, H., Bliznyuk, A. A., Gready, J. E. Combining Docking and Molecular Dynamic Simulations in Drug Design. Medical Research Reviews, 2006, 26, 531-568. https://doi.org/10.1002/med.20067
3. Bebis, G., Georgiopoulos, M. Feed-Forward Neural Networks. IEEE Potentials, 1994, 13(4), 27-31. https://doi.org/10.1109/45.329294
4. Chen, Y. Q., Liang, X. M., Du, W., Liang, Y. C., Wong, G., Chen, L. Drug-Target Interaction Prediction Based on an Interactive Inference Network. International Journal of Molecular Sciences, 2024, 25, 7753. https://doi.org/10.3390/ijms25147753
5. Cui, W., Qian, J., Yao, X. J., Hu, G., Tong, H. H. Y. Predicting Drug-Target Binding Affinity Based on Graph Isomorphism Network and iTransformer. Current Bioinformatics, 2025, 20, 1-12. https://doi.org/10.2174/0115748936320436240826114416
6. Davis, M. I., Hunt, J. P., Herrgard, S., Ciceri, P., Wodicka, L. M., Pallares, G., Hocker, M., Treiber, D. K., Zarrinkar, P. P. Comprehensive Analysis of Kinase Inhibitor Selectivity. Nature Biotechnology, 2011, 29(11), 1046-1051. https://doi.org/10.1038/nbt.1990
7. Gao, M. M., Zhang, D. K., Chen, Y. W., Wang, Z. K., Wang, X. Y., Li, S. S., Guo, Y. M., Webb, G. I., Nguyen, A. T. N., May, L., Song, J. N. GraphormerDTI, A Graph Transformer-Based Approach for Drug-Target Interaction Prediction. Computer Biology and Medicine, 2024, 173, 108339. https://doi.org/10.1016/j.compbiomed.2024.108339
8. Gaulton, A., Bellis, L. J., Bento, A. P., Chambers, J., Davies, M., Hersey, A., Light, Y., McGlinchey, S., Michalovich, D., Al-Lazikani, B., Overington, J. P. ChEMBL, A Large-Scale Bioactivity Database for Drug Discovery. Nucleic Acids Research, 2011, 40(D1), D1100-D1107.https://doi.org/10.1093/nar/gkr777
9. He, T., Heidemeyer, M., Ban, F., Cherkasov, A., Ester, M. SimBoost, A Read-Across Approach for Predicting Drug-Target Binding Affinities Using Gradient Boosting Machines. Journal of Cheminformatics, 2017, 9(1), 24. https://doi.org/10.1186/s13321-017-0209-z

10. He, Y., Sun, C. Y., Meng, L., Zhang, Y. W., Mao, R., Yang, F. Flexible Drug-Target Interaction Prediction with Interactive Information Extraction and Trade-Off. Expert Systems with Applications, 2024, 249, 123821. https://doi.org/10.1016/j.eswa.2024.123821

11. Liu, T., Lin, Y., Wen, X., Jorissen, R. N., Gilson, M. K. BindingDB, A Web-Accessible Database of Experimentally Determined Protein-Ligand Binding Affinities. Nucleic Acids Research, 2006, 35(suppl_1), D198-D201.https://doi.org/10.1093/nar/gkl999

12. Liu, S., Liu, Y., Xu, H., Xia, J., Li, S. Z. SP-DTI, Subpocket-Informed Transformer for Drug-Target Interaction Prediction. Bioinformatics, 2025, 41. https://doi.org/10.1093/bioinformatics/btaf011

13. Nguyen, T., Le, H., Quinn, T. P., Nguyen, T., Le, T. D., Venkatesh, S. GraphDTA, Predicting Drug-Target Binding Affinity with Graph Neural Networks. Bioinformatics, 2020, 37(8), 1140-1147. https://doi.org/10.1093/bioinformatics/btaa921

14. Öztürk, H., Özgür, A., Ozkirimli, E. DeepDTA, Deep Drug-Target Binding Affinity Prediction. Bioinformatics, 2018, 34(17), i821-i829. https://doi.org/10.1093/bioinformatics/bty593

15. Öztürk, H., Ozkirimli, E., Özgür, A. WideDTA, Prediction of Drug-Target Binding Affinity. Bioinformatics, 2019. https://doi.org/10.1093/bioinformatics/bty593

16. accanaro, A., Hinton, G. E. Learning Distributed Representations of Concepts Using Linear Relational Embedding. IEEE Transactions on Knowledge and Data Engineering, 2001, 13(2), 232-244. https://doi.org/10.1109/69.917563

17. Qi, H., Yu, T., Yu, W., Liu, C. Drug-Target Affinity Prediction with Extended Graph Learning-Convolutional Networks. BMC Bioinformatics, 2024, 25(1), 75. https://doi.org/10.1186/s12859-024-05698-6

18. Qiao, G., Wang, G., Li, Y. Causal Enhanced Drug-Target Interaction Prediction Based on Graph Generation and Multi-Source Information Fusion. Bioinformatics, 2024, 40. https://doi.org/10.1093/bioinformatics/btae570

19. Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., Monfardini, G. The Graph Neural Network Model. IEEE Transactions on Neural Networks, 2009, 20(1), 61-80. https://doi.org/10.1109/TNN.2008.2005605

20. Shin, B., Park, S., Kang, K., Ho, J. C. Self-Attention Based Molecule Representation for Predicting Drug-Target Interaction. In Proceedings of the 4th Machine Learning for Healthcare Conference, edited by Finale D. V., Jim F., Ken J., David K., Rajesh R., Byron W., Jenna W., Vol. 106, Proceedings of Machine Learning Research: PMLR, 2019, 230-248.

21. Shoichet, B. K. Virtual Screening of Chemical Libraries. Nature, 2004, 432, 862-865. https://doi.org/10.1038/nature03197

22. Tang, J., Szwajda, A., Shakyawar, S., Xu, T., Hintsanen, P., Wennerberg, K., Aittokallio, T. Making Sense of Large-Scale Kinase Inhibitor Bioactivity Data Sets, A Comparative and Integrative Analysis. Journal of Chemical Information and Modeling, 2014, 54(3), 735-743. https://doi.org/10.1021/ci400709d

23. Tian, Z., Yu, Y., Ni, F., Zou, Q. Drug-Target Interaction Prediction with Collaborative Contrastive Learning and Adaptive Self-Paced Sampling Strategy. BMC Biology, 2024, 22, 216. https://doi.org/10.1186/s12915-024-02012-x

24. Tran, H. N. T., Thomas, J. J., Ahamed Hassain Malim, N. H. DeepNC, A Framework for Drug-Target Interaction Prediction with Graph Neural Networks. PeerJ, 2022, 10, e13163. https://doi.org/10.7717/peerj.13163

25. Van Laarhoven, T., Nabuurs, S. B., Marchiori, E. Gaussian Interaction Profile Kernels for Predicting Drug-Target Interaction. Bioinformatics, 2011, 27(21), 3036-3043. https://doi.org/10.1093/bioinformatics/btr500

26. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. Attention Is All You Need. arXiv, 2017.

27. Wang, H., Guo, F., Du, M., Wang, G., Cao, C. A Novel Method for Drug-Target Interaction Prediction Based on Graph Transformers Model. BMC Bioinformatics, 2022, 23, 459. https://doi.org/10.1186/s12859-022-04812-w

28. Woźniak, M., Wołos, A., Modrzyk, U., Górski, R. L., Winkowski, J., Bajczyk, M., Szymkuć, S., Grzybowski, B. A., Eder, M. Linguistic Measures of Chemical Diversity and the "Keywords" of Molecular Collections. Scientific Reports, 2018, 8(1), 7598. https://doi.org/10.1038/s41598-018-25440-6

29. Wu, S., Liu, B., Zhang, X., Shao, X., Lin, C. MTrans, M-Transformer and Knowledge Graph-Based Network for Predicting Drug-Drug Interactions. Electronics, 2024, 13, 2935. https://doi.org/10.3390/electronics13152935

30. Zeng, Y., Chen, X., Luo, Y., Li, X., Peng, D. Deep Drug-Target Binding Affinity Prediction with Multiple Attention Blocks. Briefings in Bioinformatics, 2021, 22(5). https://doi.org/10.1093/bib/bbab117

31. Zeng, X., Chen, W., Lei, B. CAT-DTI, Cross-Attention and Transformer Network with Domain Adaptation for Drug-Target Interaction Prediction. BMC Bioinformatics, 2024, 25, 141. https://doi.org/10.1186/s12859-024-05753-2

32. Zhang, P. Y., Yan, Y. C., Zhang, X., Li, C. Z., Wang, S. Z., Huang, F., Kim, S. H. TransGNN, Harnessing the Collaborative Power of Transformers and Graph Neural Networks for Recommender Systems. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024, 1285-1295. https://doi.org/10.1145/3626772.3657721

33. Zhang, C., Tang, B., Wang, Q., Lai, L. Discovery of Binding Proteins for a Protein Target Using Protein-Protein Docking-Based Virtual Screening. Proteins, 2014, 82, 2472-2482. https://doi.org/10.1002/prot.24611