DTI-RME: a Robust and Multi-Kernel Ensemble Approach for Drug-Target Interaction Prediction

YUQING QIAN, XIN ZHANG, YIZHENG WANG, QUAN ZOU, CHEN CAO, YIJIE DING AND XIAOYI GUO

CONTENTS

1. SUPPLEMENTARY TEXTS

A. The proposed model

The objective function of the proposed model aims to minimize the data fitting errors while simultaneously enforcing regularization and smoothness to control model complexity and avoid overfitting, specifically accounting for the potential structures within the data. The objective can be formulated as follows:

$$\begin{aligned}
\underset{\hat{Y},A,B,\alpha,U,V,w,\beta^d,\beta^t}{\arg\min} & \sum_{i=1}^{N}\sum_{j=1}^{M}\mathcal{L}_{L_2-C}\left(Y_{ij}-\hat{Y}_{ij}\right) \\
& +\lambda_1 \mathrm{E}\left(w,\hat{Y},A,B,\alpha,U,V\right) \\
& +\lambda_2 \mathrm{R}\left(A,B,\alpha,U,V\right) \\
& +\lambda_3\left\|w\right\|_2^2+\lambda_4\left(\left\|\beta^d\right\|_2^2+\left\|\beta^t\right\|_2^2\right), \\
s.t. K_{opt}^d = & \sum_{i=1}^{v^d}\beta_i^d K_i^d,\ K_{opt}^t=\sum_{i=1}^{v^t}\beta_i^t K_i^t, \\
\sum_{i=1}^{v^d}\beta_i^d =& 1,\ \beta_i^d\geq 0,\ i=1,\ldots,v^d, \\
\sum_{i=1}^{v^t}\beta_i^t =& 1,\ \beta_i^t\geq 0,\ i=1,\ldots,v^t, \\
\sum_{i=1}^{4} w_i =& 1,\ w_i\geq 0,\ i=1,\ldots,4.
\end{aligned} \tag{S1}$$

$\mathcal{L}_{L_2-C}$ is L_2-C loss function, specifically designed for reconstructing the DTI interaction matrix. The L_2-C Loss combines the benefits of the traditional L_2 loss and the robustness of the C-loss. The formulation is as follows:

$$\mathcal{L}_{L_2-C}\left(\epsilon\right)=\begin{cases}1-\exp\left(-\frac{\epsilon^2}{2\sigma^2}\right), & \text{if } \epsilon>0\\ \epsilon^2, & \text{if } \epsilon\leq 0\end{cases}. \tag{S2}$$

In Equation S1, the first term represents the data fitting loss, while the second term $\mathrm{E}\left(\cdot\right)$ is the error based on multiple possible data structures. The third term $\mathrm{R}\left(\cdot\right)$ introduces regularization to constrain the complexity of the learned parameters. The fourth and fifth terms prevent model parameters from overfitting, while λ_1, λ_2, λ_3, and λ_4 represent hyperparameters that control the weights of these respective terms.

Because the structure of the new data is unknown and could potentially involve multiple types, we assume a mixture of four types of substructures: drug data structure, target data structure, drug-target pair data structure, and low-rank structure. This assumption leads to a combination of losses weighted by w_1, w_2, w_3, and w_4:

$$\begin{aligned}
\mathrm{E}\left(w,\hat{Y},A,B,\alpha,U,V\right) = & w_1\mathcal{L}_{\mathrm{d}}\left(\hat{Y},A\right) \\
& +w_2\mathcal{L}_{\mathrm{t}}\left(\hat{Y},B\right) \\
& +w_3\mathcal{L}_{\mathrm{p}}\left(\hat{Y},\alpha\right) \\
& +w_4\mathcal{L}_{\mathrm{rank}}\left(\hat{Y},U,V\right).
\end{aligned} \tag{S3}$$

Each of the four potential structures can be explicitly formulated as follows:

$$\mathcal{L}_{\mathrm{d}}\left(\hat{Y},A\right)=\left\|\hat{Y}-K_{opt}^d A\right\|_F^2, \tag{S4a}$$

$$\mathcal{L}_{\mathrm{t}}\left(\hat{Y},B\right)=\left\|\hat{Y}-BK_{opt}^t\right\|_F^2, \tag{S4b}$$

$$\mathcal{L}_{\mathrm{p}}\left(\hat{Y},\alpha\right)=\left\|\mathrm{vec}\left(\hat{Y}\right)-K_{opt}^p\alpha\right\|_F^2, \tag{S4c}$$

$$\mathcal{L}_{\mathrm{rank}}\left(\hat{Y},U,V\right)=\left\|\hat{Y}-UV\right\|_F^2. \tag{S4d}$$

To avoid overfitting and to regularize the learned parameters associated with each substructure, we also introduce a composite regularization term for the parameters A, B, α, U, and V:

$$\begin{aligned} \mathrm{R}(A, B, \alpha, U, V) = & \left\| K_{opt}^{d} A \right\|_{\mathcal{H}}^{2} \\ & + \left\| B K_{opt}^{t} \right\|_{\mathcal{H}}^{2} \\ & + \left\| K_{opt}^{p} \alpha \right\|_{\mathcal{H}}^{2} \\ & + \|U\|_{F}^{2} + \|V\|_{F}^{2}, \end{aligned} \tag{S5}$$

where $\|\cdot\|_{\mathcal{H}}$ denotes the Hilbert-Schmidt norm [1] and $\|\cdot\|_{F}$ is the Frobenius norm.

Additionally, to adequately balance the various kernel components, an optimal kernel is computed by linearly combining the base kernels using a weighted sum. For the drug data structure, the optimal kernel is calculated as:

$$K_{opt}^{d} = \sum_{i=1}^{v^{d}} \beta_{i}^{d} K_{i}^{d}. \tag{S6}$$

For the target data structure, the optimal kernel is calculated as:

$$K_{opt}^{t} = \sum_{i=1}^{v^{t}} \beta_{i}^{t} K_{i}^{t}. \tag{S7}$$

The objective function S1 includes 9 parameters ($\hat{Y}$, A, B, α , U, V, w, β^{d} and β^{t}) need to be estimated. We divided them into four groups according their meanings. The first group includes A, B, α, U and V, which is for learning the four types of potential structures. The second group consists of w that represents the combination weights of structures. The third group are β^{d} and β^{t} that represents the kernels of weights. The forth group only includes $\hat{Y}$ representing the predicted interaction matrix. Generally, the objective function S1 is non-convex, and it is difficult to be minimized directly. Therefore, we propose an effective alternating optimization algorithm, where we iteratively update one group of parameters while holding the other groups fixed.

B. Initializations of the proposed method

The initialization of the parameters is a critical step in ensuring the effectiveness and convergence of the proposed alternating optimization algorithm. As mentioned, the objective function S1 involves multiple parameter groups, and finding a global minimizer is challenging due to the overall non-convex nature of the problem. To address this, we adopt a reasonable initialization strategy for the various parameter groups to ensure that the optimization process begins with a good starting point.

We initialize the second group for the four potential data structures equally, assigning $w_i = 1/4$ for $i = 1, 2, 3, 4$.

For the third group, we initialize the kernel weights to be uniform across the base kernels. Specifically, we set $\beta_{i}^{d} = 1/v^{d}$ for each base drug kernel and $\beta_{i}^{t} = 1/v^{t}$ for each base target kernel, assuming an equal contribution from each kernel initially.

The initial reconstruction of the drug-target interaction matrix (the forth group) is set directly from the observed interaction matrix, $\hat{Y} = Y$.

C. Solution to the proposed model

Next, we describe the step-by-step optimization process for each parameter group, starting with the first group, which consists of the parameters responsible for learning the four potential structures.

C.1. Optimize the first group parameters.

We first calculate the first group parameters A, B, α, U and V while holding the other group parameters fixed. When we compute A, the objective function S1 can be rewritten as

$$\arg\min_{A} \left\| \hat{Y} - K_{opt}^{d} A \right\|_{F}^{2} + \lambda_{2} \left\| K_{opt}^{d} A \right\|_{\mathcal{H}}^{2}. \tag{S8}$$

With some simple calculation, we also have

$$\left\| K_{opt}^{d} A \right\|_{\mathcal{H}}^{2} = \left\| A^{T} K_{opt}^{d} A \right\|_{F}^{2}. \tag{S9}$$

Hence, the problem S8 becomes

$$\arg\min_{A} \left\| \hat{Y} - K_{opt}^{d} A \right\|_{F}^{2} + \lambda_2 \left\| A^{T} K_{opt}^{d} A \right\|_{F}^{2}. \tag{S10}$$

By taking the derivative with respective to A, we have,

$$K_{opt}^{d} \left(-\hat{Y} + K_{opt}^{d} A + \lambda_2 A \right) = 0. \tag{S11}$$

This implies

$$A = \left(K_{opt}^{d} + \lambda_2 I \right)^{-1} \hat{Y} \tag{S12}$$

where I is the identity matrix and we obtained the solution.

When we compute B, the objective function S1 can be rewritten as

$$\arg\min_{B} \left\| \hat{Y}^{T} - K_{opt}^{t} B^{T} \right\|_{F}^{2} + \lambda_2 \left\| K_{opt}^{t} B^{T} \right\|_{\mathcal{H}}^{2}, \tag{S13}$$

which has the same form as the Equation S8. Thus, we can solve B in the same way solving A and get the solution

$$B^{T} = \left(K_{opt}^{t} + \lambda_2 I \right)^{-1} B^{T}. \tag{S14}$$

When we compute α, the objective function S1 can be reformulated as:

$$\arg\min_{\alpha} \left\| \mathrm{vec}\left(\hat{Y} \right) - K_{opt}^{p} \alpha \right\|_{F}^{2} + \lambda_2 \left\| K_{opt}^{p} \alpha \right\|_{\mathcal{H}}^{2}. \tag{S15}$$

Minimizing Equation S15 with respect to α leads to the following system of linear equations:

$$\left(K_{opt}^{p} + \lambda_2 I \right) \alpha = \mathrm{vec}\left(\hat{Y} \right). \tag{S16}$$

From Equation S16, we can directly compute $K_{opt}^{p} \alpha$ as:

$$\mathrm{vec}(K_{opt}^{p} \alpha) = K_{opt}^{p} \left(K_{opt}^{p} + \lambda_2 I \right)^{-1} \mathrm{vec}\left(\hat{Y} \right), \tag{S17}$$

where $K_{opt}^{p} = K_{opt}^{t} \otimes K_{opt}^{d}$ is a Kronecker product kernel. Direct computation of Equation Eq. (S17) requires inverting an $nm \times nm$ matrix, which incurs a computational cost of $O(n^3 m^3)$. To avoid this, we employ a more efficient approach using the vec-trick algorithm [2] and Singular Value Decomposition (SVD) [3].

Theorem 1. *Let the eigenvalues of the matrices $K_o^d pt$ and $K_o^t pt$ be, respectively, $\left\{ \Lambda_l^d \right\}_{l=1}^{n}$ and $\left\{ \Lambda_p^t \right\}_{p=1}^{m}$, with the corresponding eigenvectors given as, respectively, the columns of the matrices Q^d and Q^t. Then the eigenvectors of the $K^t \otimes K^d$ is the same $Q^t \otimes Q^d$, while the eigenvalues are $\left\{ \Lambda_l^t \Lambda_p^d \right\}_{l=1,p=1}^{n,m}$ for the Pairwise kernel.*

Since the kernel matrices are positive semidefinite, they can be eigendecomposed (we omit the decomposition for K_{opt}^{t}) as:

$$K_{opt}^{d} = Q^{d} \, \mathrm{diag}\left(\Lambda_1^d, \ldots, \Lambda_n^d \right) \left(Q^{d} \right)^{T}. \tag{S18}$$

Let $\Lambda_{l,p} = \Lambda_l^d \Lambda_p^t$ and $Q = Q^t \otimes Q^d$. Using *Theorem 1* and the fact that $QQ^T = I$, we can rewrite the inverse matrix in Equation Eq. (S17) as:

$$\left(K_{opt}^{p} + \lambda_2 I \right)^{-1} = Q \left(\mathrm{diag}(\mathrm{vec}(\Lambda)) + \lambda_2 I \right)^{-1} Q^{T}. \tag{S19}$$

Here, $\text{diag}(\text{vec}(\Lambda)) + \lambda_2 I$ is diagonal, and its inverse is easily computed as:

$$Z_{l,p} = \left(\Lambda_{l,p} + \lambda_2\right)^{-1}. \tag{S20}$$

Substituting the terms in Equation Eq. (S17) with Z and $K^p_{opt} = Q\,\text{diag}(\text{vec}(\Lambda))\,Q^T$, we can rewrite Equation S17 as:

$$\text{vec}(K^p_{opt}\alpha) = \left(Q^t \otimes Q^d\right)\,\text{diag}\left(\text{vec}\left(\Lambda \odot Z\right)\right)\,\left(Q^t \otimes Q^d\right)^T\,\text{vec}(W). \tag{S21}$$

Theorem 2. *Let $A \in R^{n\times m}$, $B \in R^{m\times l}$ and $C \in \mathbb{R}^{l\times p}$ be matrices. Then*

$$\left(C^T \otimes A\right)\text{vec}\left(B\right) = \text{vec}\left(ABC\right). \tag{S22}$$

It is obvious that the right-hand side of Equation S22 is considerably faster to compute than the left-hand side, because it avoids the direct computation of the large Pairwise kernel. Equation S22 is called vec-trick algorithms [2].

Using *Theorem 2* and taking out $\text{vec}\left(\cdot\right)$ operation, we can further simplify Equation S21 as:

$$(K^p_{opt}\alpha) = Q^d\left(\Lambda \odot Z \odot \left(\left(Q^d\right)^T W Q^t\right)\right)\left(Q^t\right)^T. \tag{S23}$$

When we compute U and V, the objective function S1 can be rewritten as

$$\underset{U,V}{\arg\min}\left\|\hat{Y} - UV\right\| + \lambda_2\left(\|U\|_F^2 + \|V\|_F^2\right). \tag{S24}$$

Equation S24 is non-convex with respect to U and V jointly, but it is convex with respect to each of them individually when the other is fixed. We alternate between fixing U and solving for V, and fixing V and solving for U.

When V is fixed, the objective function S24 is a standard ridge regression problem. Differentiating the above expression with respect to U and setting the gradient to zero, we get the following normal equation:

$$U\left(VV^\top + \lambda_2 I\right) = \hat{Y}V^\top. \tag{S25}$$

Solving for U:

$$U = \hat{Y}V^\top\left(VV^\top + \lambda_2 I\right)^{-1}. \tag{S26}$$

Similarly, when U is fixed, solving for V:

$$V = \left(U^\top U + \lambda_2 I\right)^{-1}U^\top\hat{Y}. \tag{S27}$$

C.2. Optimize the second group parameters.

Given that all other parameters $(\hat{Y}, A, B, \alpha, U, V, \beta^d, \beta^t)$ are fixed, we can focus on minimizing the objective function with respect to w. The simplified part of the objective function involving w is:

$$\begin{aligned}&\underset{w}{\arg\min}\, w^T\begin{bmatrix}\lambda_1\mathcal{L}_{\text{d}}\left(\hat{Y}, A\right)\\ \lambda_1\mathcal{L}_{\text{t}}\left(\hat{Y}, B\right)\\ \lambda_1\mathcal{L}_{\text{p}}\left(\hat{Y}, \alpha\right)\\ \lambda_1\mathcal{L}_{\text{rank}}\left(\hat{Y}, U, V\right)\end{bmatrix} + \lambda_3 w^T w\\ &s.t.\ \sum_{i=1}^{4} w_i = 4, w_i \geq 0, i = 1, \ldots, 4.\end{aligned} \tag{S28}$$

It is a standard quadratic programming (QP) problem [4]. The optimization method used for the objective function S31 is the interior-point-convex algorithm [5] implemented in MATLAB.

C.3. Optimize the third group parameters.

When optimizing with respect to the third group parameters (β^d and β^t) while keeping all other parameters fixed, the objective function S1 simplifies to two separate optimization problems-one for β^d and one for β^t, since the two are not coupled in the objective.

The simplified part of the objective function involving β^d is:

$$\begin{aligned} &\arg\min_{\beta^d} \left\| \hat{Y} - \sum_{i=1}^{v^d} \beta_i^d K_i^d A \right\|_F^2 + \left\| \mathrm{vec}\left(\hat{Y}\right) - K_{opt}^t \otimes \left(\sum_{i=1}^{v^d} \beta_i^d K_i^d \right) \alpha \right\|_F^2 + \frac{\lambda_4}{\lambda_1} \left\| \beta^d \right\|_2^2 \\ &s.t. \sum_{i=1}^{v^d} \beta_i^d = 1,\ \beta_i^d \geq 0,\ i = 1, \ldots, v^d. \end{aligned} \tag{S29}$$

According to [6, 7], we know that objective function S29 is a QP problem. Here, we also use the interior-point-convex algorithm implemented in MATLAB. While the optimization of β^t is similar to that of β^d, and its objective function is

$$\begin{aligned} &\arg\min_{\beta^t} \left\| \hat{Y} - A \sum_{i=1}^{v^t} \beta_i^t K_i^t \right\|_F^2 + \left\| \mathrm{vec}\left(\hat{Y}\right) - \left(\sum_{i=1}^{v^t} \beta_i^t K_i^t \right) \otimes K_{opt}^t \alpha \right\|_F^2 + \frac{\lambda_4}{\lambda_1} \left\| \beta^t \right\|_2^2 \\ &s.t. \sum_{i=1}^{v^t} \beta_i^t = 1,\ \beta_i^t \geq 0,\ i = 1, \ldots, v^t. \end{aligned} \tag{S30}$$

C.4. Optimize the forth group parameters.

After calculating the first, the second and the third group parameters, we can then compute $\hat{Y}$ while holding them fixed. The objective function S1 can be rewritten as

$$\begin{aligned} \arg\min_{\hat{Y}} \sum_{i=1}^{N} \sum_{j=1}^{M} \mathcal{L}_{L_2-C}\left(Y_{ij} - \hat{Y}_{ij}\right) &+ \lambda_1 w_1 \left\| \hat{Y} - K_{opt}^d A \right\|_F^2 \\ &+ \lambda_1 w_2 \left\| \hat{Y} - B K_{opt}^t \right\|_F^2 \\ &+ \lambda_1 w_2 \left\| \hat{Y} - \mathrm{unvec}\left(K_{opt}^p \alpha\right) \right\|_F^2 \\ &+ \lambda_1 w_3 \left\| \hat{Y} - UV \right\|_F^2. \end{aligned} \tag{S31}$$

where $\mathcal{L}_{L_2-C}$ is Equation S2. $\mathcal{L}_{L_2-C}$ is non-convex and non-quadratic loss. Hence, we firstly use the half-quadratic optimization algorithm to rewrite the objective function. By introducing additional auxiliary variable, it reformulates a non-quadratic loss function as an augmented objective function in an enlarged parameter space. According to the conjugate function [8] and half-quadratic theory [9], for a fixed $E_{ij} = Y_{ij} - \hat{Y}_{ij}$ and $E_{ij} > 0$, the following equations holds

$$\mathcal{L}_{L_2-C}\left(E_{ij}\right) = \min_{W_{ij}} Q\left(E_{ij}, W_{ij}\right) + \phi\left(W_{ij}\right), \tag{S32}$$

where $\phi\left(W_{ij}\right)$ is the conjugate function of $l\left(E_{ij}\right)$, W_{ij} is the corresponded auxiliary variable, and $Q(\cdot,\cdot) : \mathbb{R} \to \mathbb{R}$ is a quadratic term for E_{ij} and W_ij. Considering the quadratic term of multiplicative form [9]:

$$Q\left(E_{ij}, W_{ij}\right) = W_{ij} E_{ij}^2. \tag{S33}$$

The loss function in Equation S32 can be optimized by the following alternating minimization scheme:

1. When E_{ij} are fixed, the minimization of the objective function in Equation S32 becomes convex with respect to W_{ij}. The explicit optimum is given by

$$W_{ij} = \frac{\partial \mathcal{L}_{L_2-C}\left(E_{ij}\right)}{\partial E_{ij}} = \exp\left(-\frac{E_{ij}^2}{2\sigma^2}\right). \tag{S34}$$

   It is expected that outliers often cause large fitting errors and the corresponding weights W_{ij} should be small.

2. When W_{ij} is fixed, the minimization of the objective function in S32 reduces to the weighted loss:

$$\mathcal{L}_{L_2-C}\left(E_{ij}\right) = W_{ij}E_{ij}^2. \tag{S35}$$

That is, Equation S2 is rewritten as

$$\mathcal{L}_{L_2-C}\left(E_{ij}\right) = \begin{cases} W_{ij}{E_{ij}}^2, & \text{if } E_{ij} > 0 \\ {E_{ij}}^2, & \text{if } E_{ij} \leq 0 \end{cases}. \tag{S36}$$

Then, substitute Equation S36 into Equation S31, which is therefore convex with respect to $\hat{Y}$, and its solution is

$$\hat{Y} = \frac{W \odot Y + \lambda_1 w_1 K_{opt}^d A + \lambda_1 w_2 B K_{opt}^t + \lambda_1 w_3 \text{unvec}(K_{opt}^p \alpha) + \lambda_1 w_3 UV}{W + \lambda_1} \tag{S37}$$

The update sequence generated by the above scheme will converges.

D. Stop criteria

Our algorithm stops when it detects a relative change in $\hat{Y}$ that does not exceed the maximum tolerance. The default value of maximum tolerance is 0.01.

E. Baseline methods

The following baseline methods are used for comparison:

1. **KronRLS-MKL** [6]: KronRLS-MKL is an extension of the KronRLS algorithm under the MKL framework. It automatically selects the more relevant kernels by returning weights indicating their importance in the drug-target prediction at hand.

2. **DLapRLS with HSIC-MKL** [10]: Inspired by NetLapRLS [11], DLapRLS is proposed for predicting DTI. The method consists of three key steps: (**a**) multiple drug and target kernels are constructed based on two distinct feature spaces; (**b**) these corresponding kernels are linearly combined using the Hilbert–Schmidt Independence Criterion-based Multiple Kernel Learning (HSIC-MKL) approach in each feature space separately; and (**c**) DLapRLS is then applied to build the predictive model for DTIs.

3. **KronRLS with CKA-MKL** [12]: KronRLS with CKA-MKL first determines the mixture weights of the input pairwise kernels, and then learns the pairwise prediction function. Both steps are performed efficiently without explicit computation of massive pairwise matrices, making the method applicable to solving large pairwise learning problems.

4. **MK-TCMF** [13]: Multiple kernel matrices are integrated via MKL. And the original adjacency matrix of DTI could be decomposed into three matrices. These include the latent feature matrix of the drug space, the latent feature matrix of the target space and the bi-projection matrix. To obtain better prediction performance, the MKL algorithm can regulate the weight of each kernel matrix according to prediction error.

5. **GRGMF** [14]: A generalized matrix factorization framework is designed to capture the latent representations of both drugs and targets, while adaptively incorporating neighborhood information for each node. This allows the model to account for local network structure during learning. Additionally, GRGMF introduces two graph-based regularization terms, which leverage affinity information from external databases to further refine the latent representations, ensuring that the resulting model is more robust and accurate in predicting potential drug-target interactions.

6. **MVGCN** [15]: The method proceeds in several steps: (**a**) It begins by constructing a multi-view heterogeneous network (MVHN) that integrates multiple similarity networks with the original biomedical bipartite network, providing a comprehensive multi-view structure; (**b**) A self-supervised learning strategy is then applied to the bipartite network to extract node attributes, which serve as the initial node embeddings; (**c**) To refine these embeddings, a Neighborhood Information Aggregation (NIA) layer is designed to iteratively update node embeddings by aggregating information from both inter- and intra-domain neighbors

across each view of the MVHN; (**d**) The embeddings from multiple NIA layers across all views are then combined, and the multiple views are integrated to produce the final node embeddings; (**e**) Finally, these node embeddings are fed into a discriminator to predict the existence of links in the network, enabling accurate link prediction.

7. **MIDTI** [16]: The method includes the following key components: (**a**) MIDTI constructs multi-view similarity networks for both drugs and targets, leveraging diverse sources of information to represent their relationships. These multi-view networks are then effectively integrated using an unsupervised fusion strategy to capture comprehensive similarity information for both drugs and targets. (**b**) The embeddings of drugs and targets are jointly learned from the multi-type networks, simultaneously capturing their complex associations. (**c**) MIDTI employs a deep interactive attention mechanism, which further refines these embeddings by learning more discriminative representations, considering known DTI relationships. (**d**) Finally, the learned drug and target embeddings are input into a multilayer perceptron model, which predicts potential drug–target interactions, allowing for more accurate and biologically meaningful predictions.

F. Deep learning-based methods

The following deep learning-based methods are used for comparison:

1. **DeepDTIs** [17]: The features of drugs and targets were automatically extracted from simple chemical substructure and sequence order information and then applies known label pairs of interaction to build a classification model.

2. **MDADTI** [18]: The method applies random walk with restart method and positive point-wise mutual information to calculate topological similarity matrices of drugs and targets. This captures the global structure information of similarity measures. MDADTI utilizes multimodal deep autoencoder to fuse multiple topological similarity matrices of drugs and targets. It automatically learn low-dimensional features of drugs and targets, and applies deep neural network to predict DTI.

3. **DAEFCNN** [19]: The method exploits the capabilities of Convolutional Neural Networks (CNN) to obtain 1D representations of protein sequences and compounds from SMILES strings. These representations can be interpreted as features that express local dependencies or patterns that can then be used in a Fully Connected Neural Network (FCNN), acting as a binary classifier.

4. **DeepDTA** [20]: The method employs CNN blocks to learn representations from raw protein sequences and SMILES strings. It combines these representations to feed into a fully connected layer block.

5. **DeepConvDTI** [21]: The method applies convolutional filters across the entire sequence of a protein to capture local residue patterns. These patterns are pivotal to drug–target interactions. These local patterns highlight the main protein residues involved in interaction. The model then uses a max-pooling operation on the CNN results, identifying the strongest matching local residue patterns within the protein sequence that contribute to DTIs. Once local residue patterns are identified, the model abstracts and organizes them into meaningful protein features. These features serve as input for higher layers within the network, allowing the model to construct richer representations of proteins. CNN protein features are then concatenated with drug features, which are derived from drug molecular fingerprints. This combined representation of proteins and drugs predicts DTI probability.

6. **GAEMSDNN** [22]: This method involves a symmetric graph auto-encoder (SGAE) [23] and a new designed multisubspaces deep neural network (MSDNN): the SGAE is used to learn low-level features, which protect the manifold and reconstruction information at the same time. MSDNN contains a subspace and an ensemble layer. The subspace layer can obtain many different strong feature subsets, and the ensemble layer can comprehensively utilize these strong feature subsets in a unified optimization frame-work. MSDNN can extract more effective features from features of drugs and targets, as well as increase the gradient signal to improve prediction performance.

G. Molecular dynamic simulation

The molecular dynamics (MD) simulations were performed using the AmberTools software package, applying both the AMBER ff19SB [24] and AM1-BCC force fields [25] to the top-ranked compound-protein complexes. The complex system was solved in a cubic water box with the TIP3P water model, ensuring a 10 buffer on each side. Periodic boundary conditions (PBC) were employed, and the net charge was neutralized by adding 0.15 M NaCl. The system underwent two rounds of energy minimization: the first involved 10,000 steps of steep descent minimization to resolve steric clashes, followed by 10,000 steps of conjugate gradient minimization for further refinement. The system was then equilibrated through 200 ps of NPT (constant number of particles, pressure, and temperature) ensemble simulations, followed by 500 ps of NVT (constant number of particles, volume, and temperature) ensemble. Temperature regulation (300 K) was achieved using a Berendsen thermostat with a coupling delay of 1 ps. The pressure was controlled at 1 atm via a Monte Carlo Barostat with a relaxation time of 1 ps. A 100 ns production simulation in the NVT ensemble, without bonds or angle constraints, was then carried out. CPPTRAJ was used for trajectory analysis.

The structural stability of the ligand-protein complex was assessed by calculating the root mean square deviation (RMSD). To examine the compactness of the protein, the radius of gyration (Rg) was computed, while hydrogen bonding interactions within the protein were analyzed using the H-bonds module. The flexibility of individual residues was evaluated by calculating the root mean square fluctuation (RMSF). All graphical representations of the results were generated using Origin academic.

The RMSD of particle coordinates is one measure of distance r, or dissimilarity, between molecular conformations. Each structure should have matching elementwise atoms i of numbers of atoms M in the same order, as the distance between them is calculated and summed for the final result. It is calculated between coordinate arrays $\mathrm{r_i(t)}$ and $\mathrm{r_i(0)}$ according to the equation below:

$$\mathrm{RMSD}(t) = \left[\frac{1}{M} \sum_{i=1}^{N} m_i |\mathrm{r}_i\,(t) - \mathrm{r}_i\,(0)|^2 \right]^{\frac{1}{2}} \tag{S38}$$

The root-mean-square-fluctuation (RMSF) of a structure is the time average time T of the RMSD. It is calculated according to the below equation, where $\vec{r}_1\,(t)$ is the coordinates of particle i, and $\langle \vec{r}_1 \rangle$ is the ensemble average position of i.

$$\mathrm{RMSF}_i = \sqrt{\frac{1}{T} \sum_{t=1}^{T} \left(\vec{r}_1\,(t) - \langle \vec{r}_1 \rangle\right)^2} \tag{S39}$$

The radius of gyration of a structure measure how compact it is. In MD study, it is usually calculated as follows:

$$R_g = \sqrt{\frac{\sum_{i=1}^{N} m_i (\mathrm{r}_i\,(t) - \mathrm{r}_i\,(0))^2}{\sum_{i=1}^{N} m_i}} \tag{S40}$$

where m_i is the mass of atom $\mathrm{r_i(t)}$ and $\mathrm{r_i(0)}$ is the position of atom i, relative to the center-of-mass of the selection.

H. Computational complexity analysis

In this section, we analyze the computational complexity of the optimization algorithms presented in Section C. We employ an alternating optimization algorithm, in which we iteratively update one group of parameters while keeping the others fixed. The computational complexity of each update step is analyzed as follows:

1. **First group of parameters**: The update required solving Equation S12, S14, S23, S26, and S27. These involve matrix inversions and decompositions, leading to a time complexity of $\mathcal{O}(n^3 + m^3)$, where n and m denote the number of drugs and targets, respectively.

2. **Second group of parameters**: This step involves solving a QP problem, which can be done in constant time with complexity $\mathcal{O}(1)$, due to the fixed problem size.

3. **Third group of parameters**:The update requires solving two QP problems, namely Equation S29 and S30, with time complexity $\mathcal{O}((v^t)^3 + (v^d)^3)$, where v^t and v^d are the numbers of target and drug kernels, respectively.

4. **Fourth group of parameters**: This step involves computing Equation S37, with a time complexity of $\mathcal{O}(nm)$.

Assuming the number of iterations is $Iter$, the overall complexity becomes $\mathcal{O}(Iter(n^3 + m^3 + (v^t)^3 + (v^d)^3 + 1))$, In practical scenarios, $Iter$, v^t, and v^d, are typically much smaller than n and m. Therefore, the dominant computational cost arises from the first group update, and the overall time complexity can be approximated as $\mathcal{O}(n^3 + m^3)$.

2. SUPPLEMENTARY FIGURES

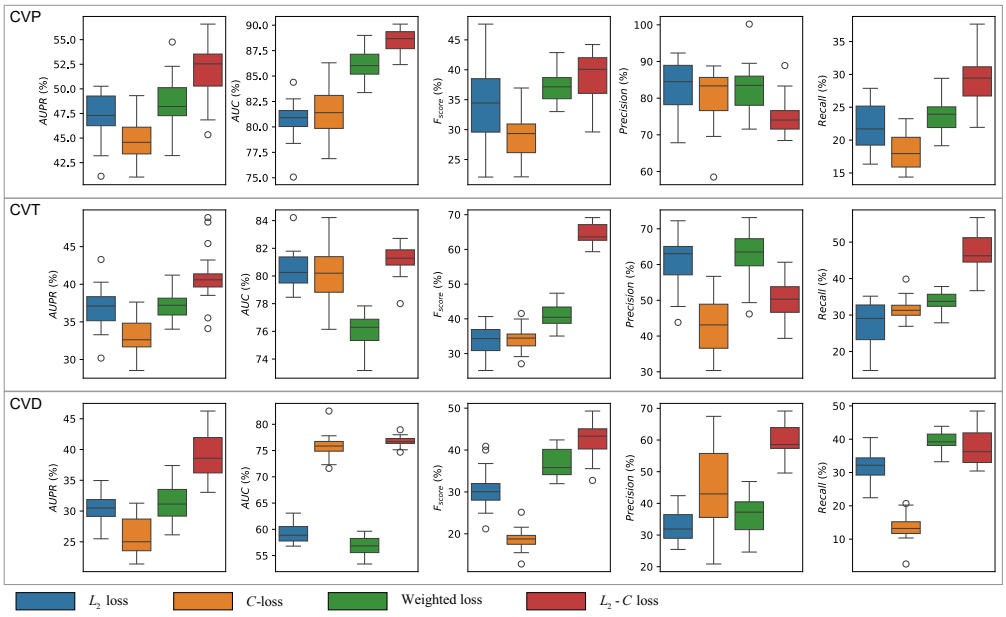

Fig. S1. Box plot of five metrics for CVP, CVD, and CVT settings on NR dataset, showing the performance different loss function.

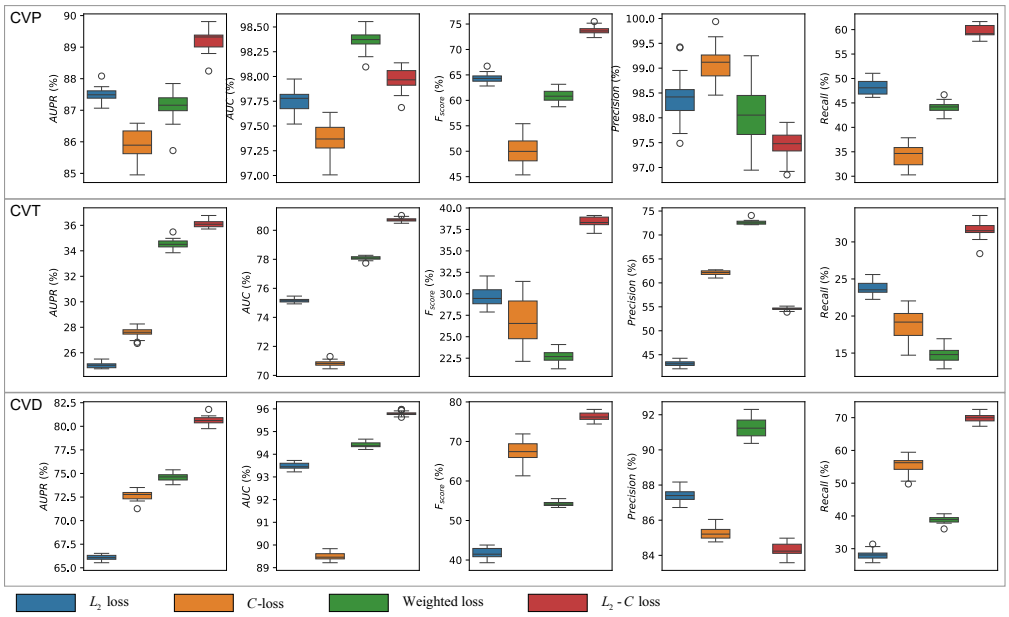

Fig. S2. Box plot of five metrics for CVP, CVD, and CVT settings on IC dataset, showing the performance different loss function.

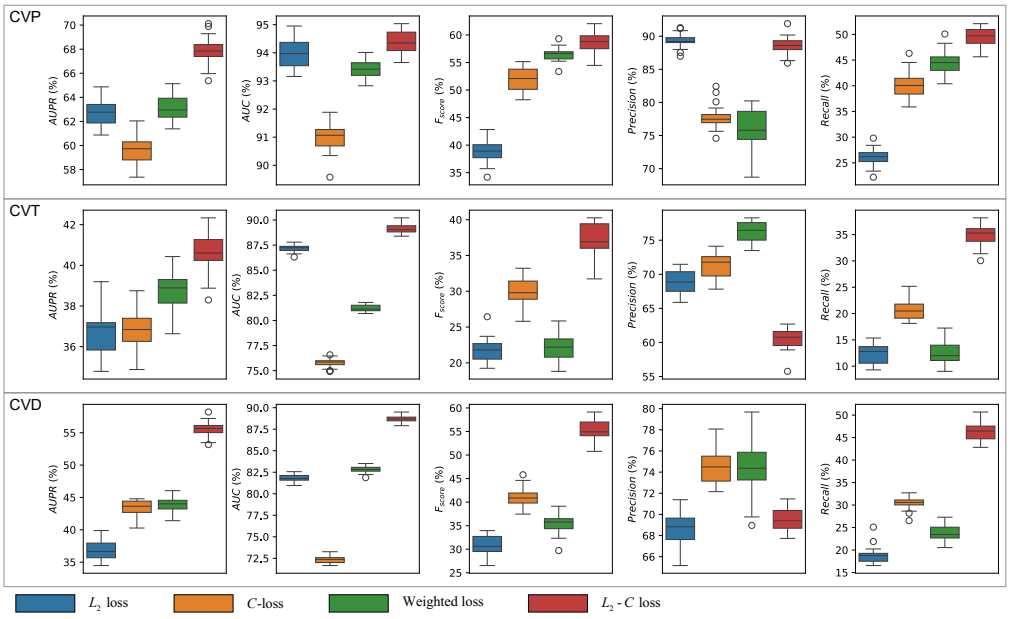

Fig. S3. Box plot of five metrics for CVP, CVD, and CVT settings on GPCR dataset, showing the performance different loss function.

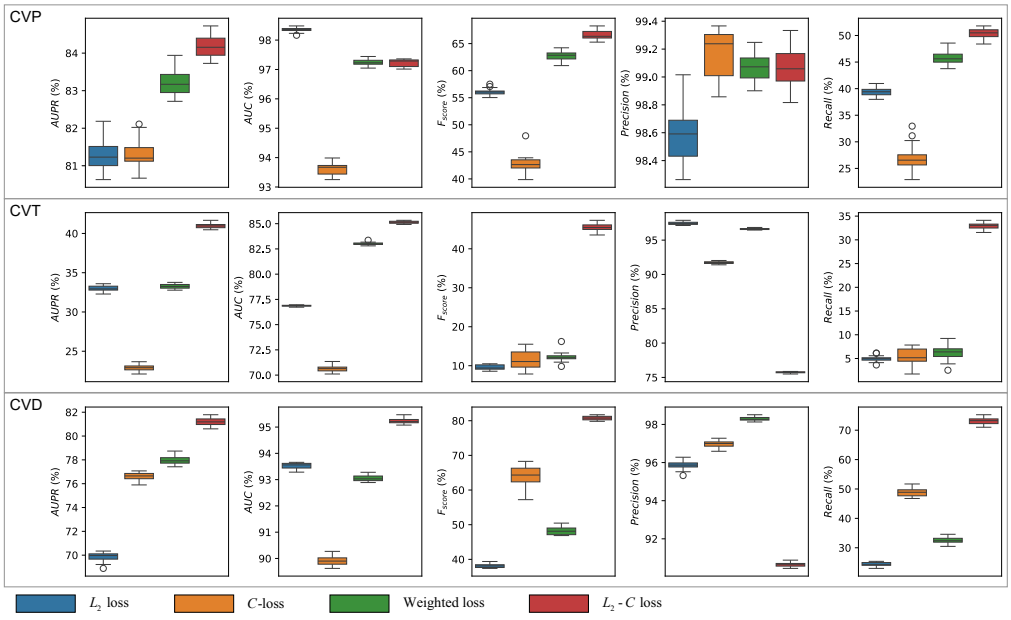

Fig. S4. Box plot of five metrics for CVP, CVD, and CVT settings on E dataset, showing the performance different loss function.

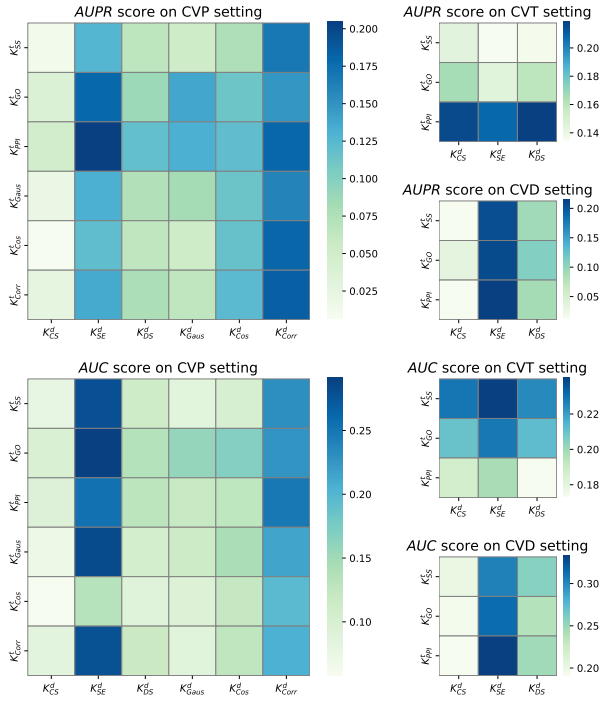

Fig. S5. Heatmaps of *AUPR* and *AUC* results for CVP, CVD, and CVT settings on NR dataset, showing the performance degradation of single-kernel compared to the MKL method. Deeper colors indicate a larger decrease in performance metrics relative to the MKL method.

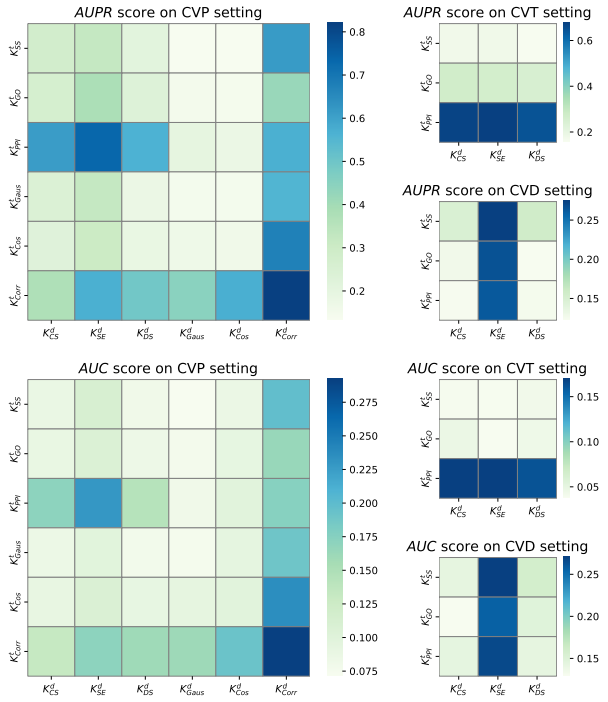

Fig. S6. Heatmaps of *AUPR* and *AUC* results for CVP, CVD, and CVT settings on IC dataset, showing the performance degradation of single-kernel compared to the MKL method. Deeper colors indicate a larger decrease in performance metrics relative to the MKL method.

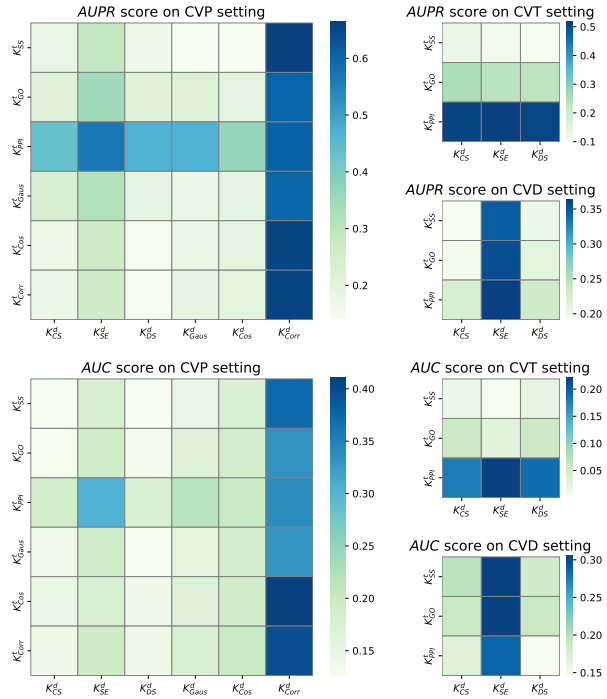

Fig. S7. Heatmaps of *AUPR* and *AUC* results for CVP, CVD, and CVT settings on GPCR dataset, showing the performance degradation of single-kernel compared to the MKL method. Deeper colors indicate a larger decrease in performance metrics relative to the MKL method.

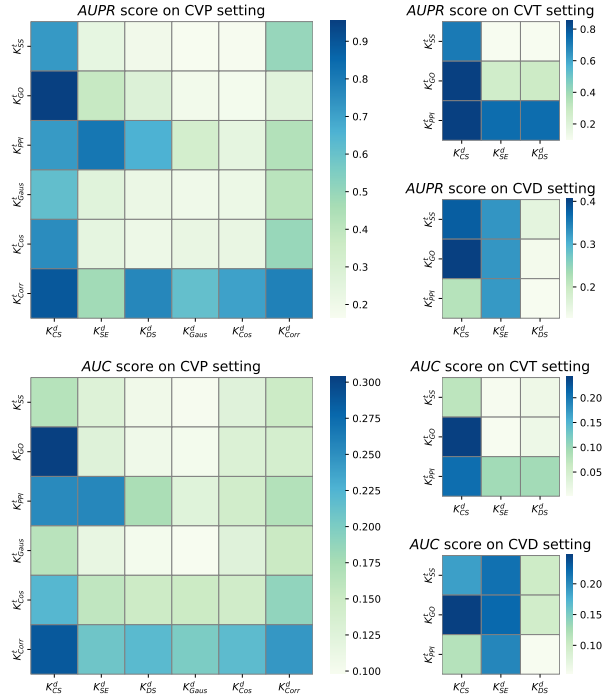

Fig. S8. Heatmaps of *AUPR* and *AUC* results for CVP, CVD, and CVT settings on E dataset, showing the performance degradation of single-kernel compared to the MKL method. Deeper colors indicate a larger decrease in performance metrics relative to the MKL method.

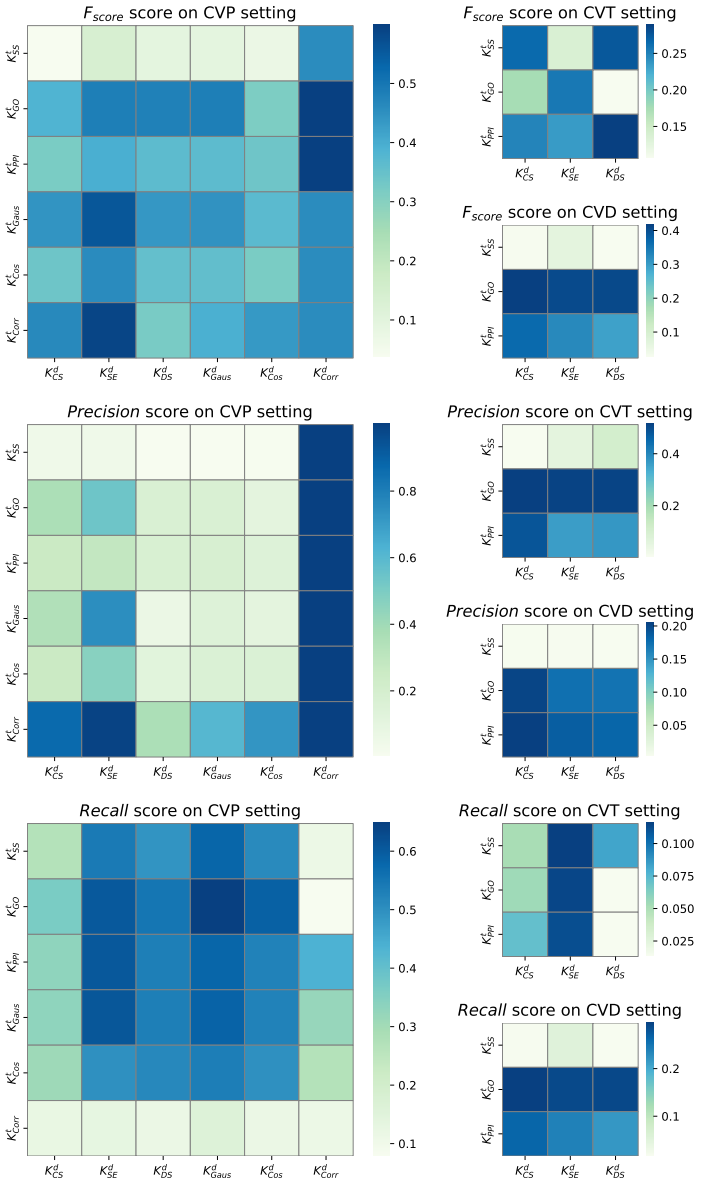

Fig. S9. Heatmaps of F_{score}, *Precision* and *Recall* results for CVP, CVD, and CVT settings on Luo dataset, showing the performance degradation of single-kernel compared to the MKL method. Deeper colors indicate a larger decrease in performance metrics relative to the MKL method.

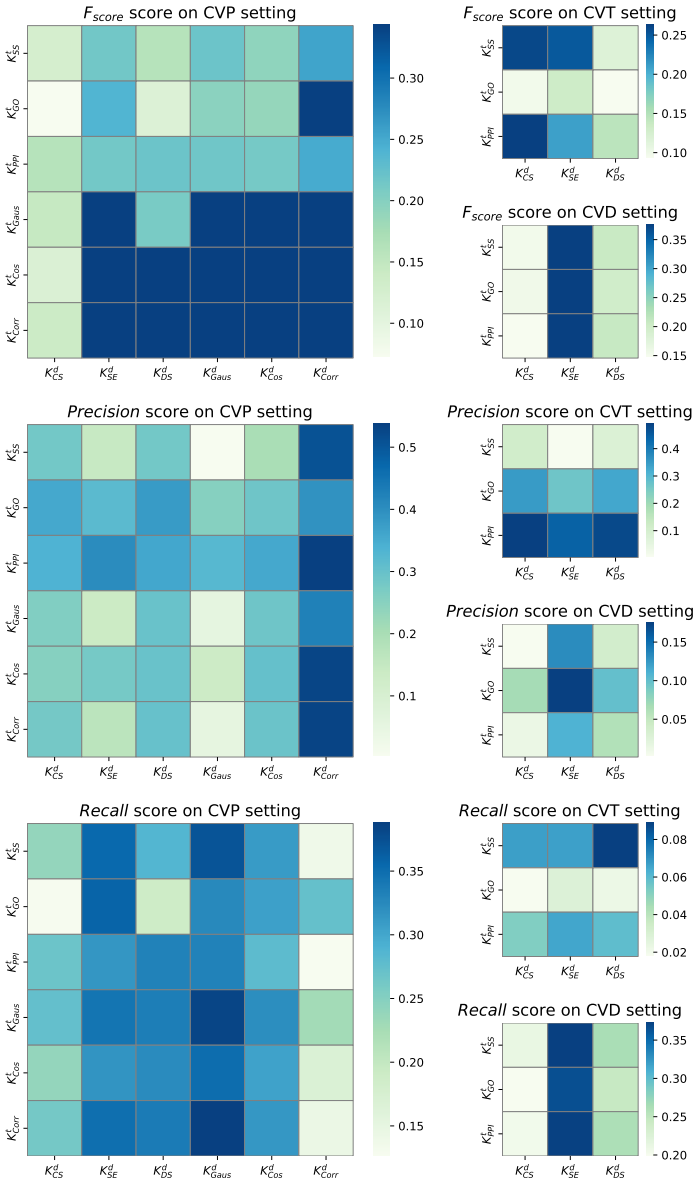

Fig. S10. Heatmaps of F_{score}, *Precision* and *Recall* results for CVP, CVD, and CVT settings on NR dataset, showing the performance degradation of single-kernel compared to the MKL method. Deeper colors indicate a larger decrease in performance metrics relative to the MKL method.

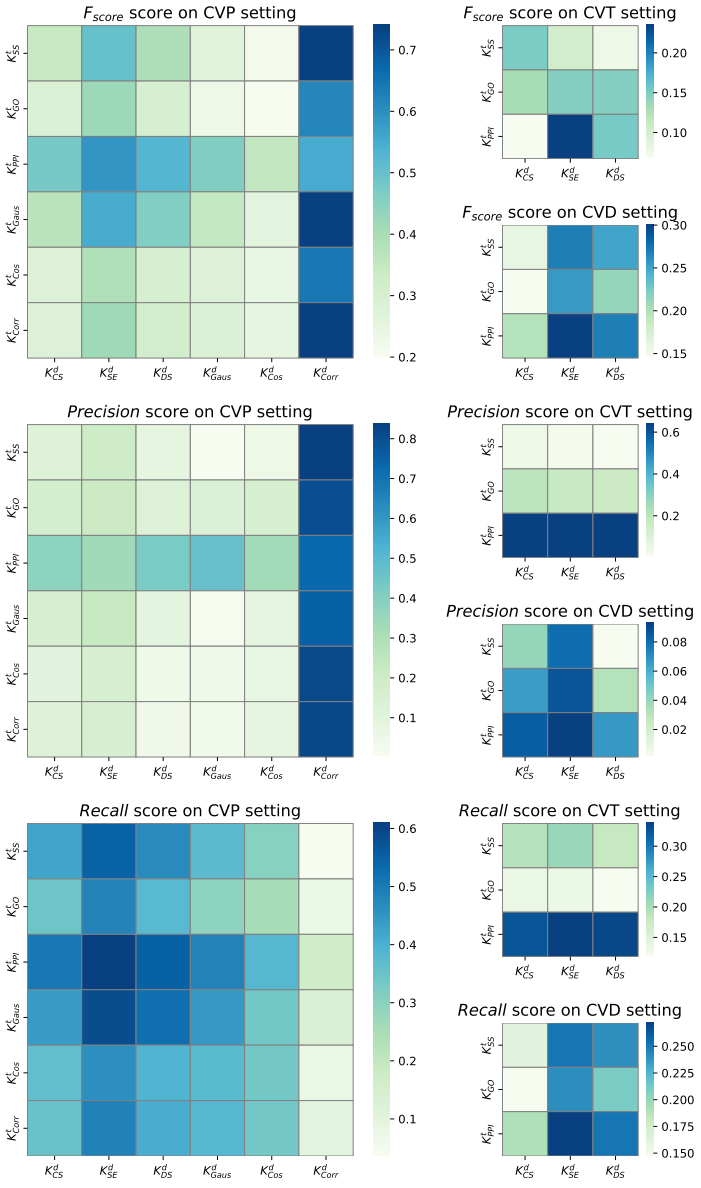

Fig. S11. Heatmaps of F_{score}, *Precision* and *Recall* results for CVP, CVD, and CVT settings on GPCR dataset, showing the performance degradation of single-kernel compared to the MKL method. Deeper colors indicate a larger decrease in performance metrics relative to the MKL method.

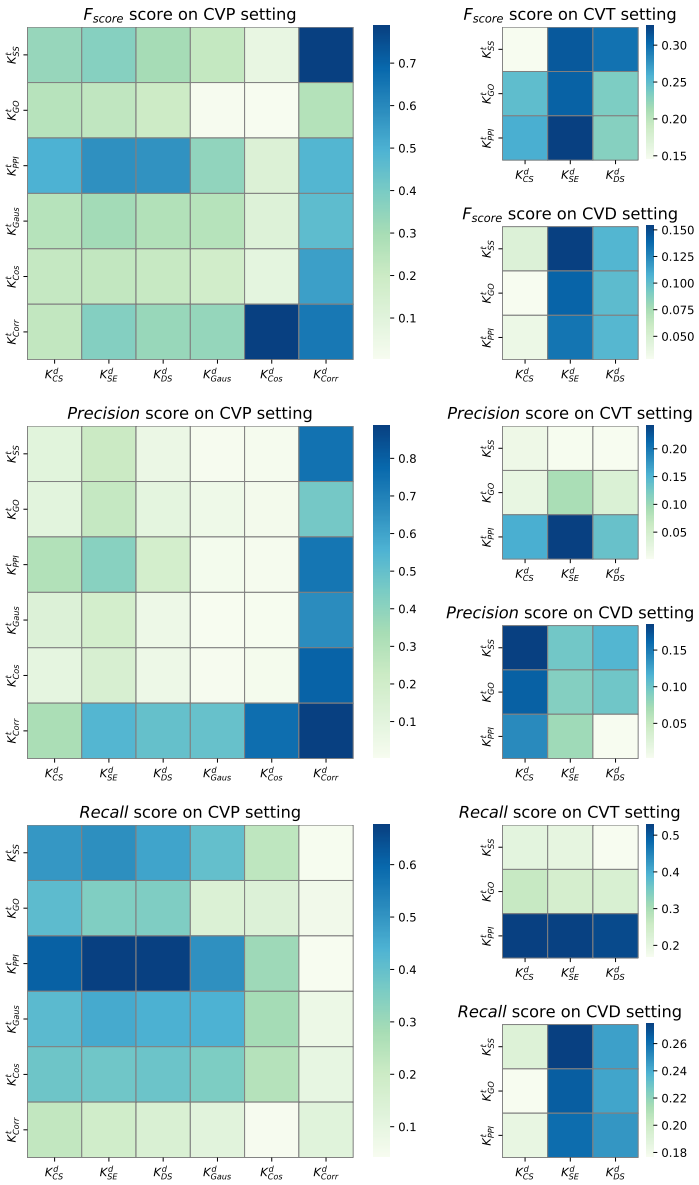

Fig. S12. Heatmaps of F_{score}, *Precision* and *Recall* results for CVP, CVD, and CVT settings on IC dataset, showing the performance degradation of single-kernel compared to the MKL method. Deeper colors indicate a larger decrease in performance metrics relative to the MKL method.

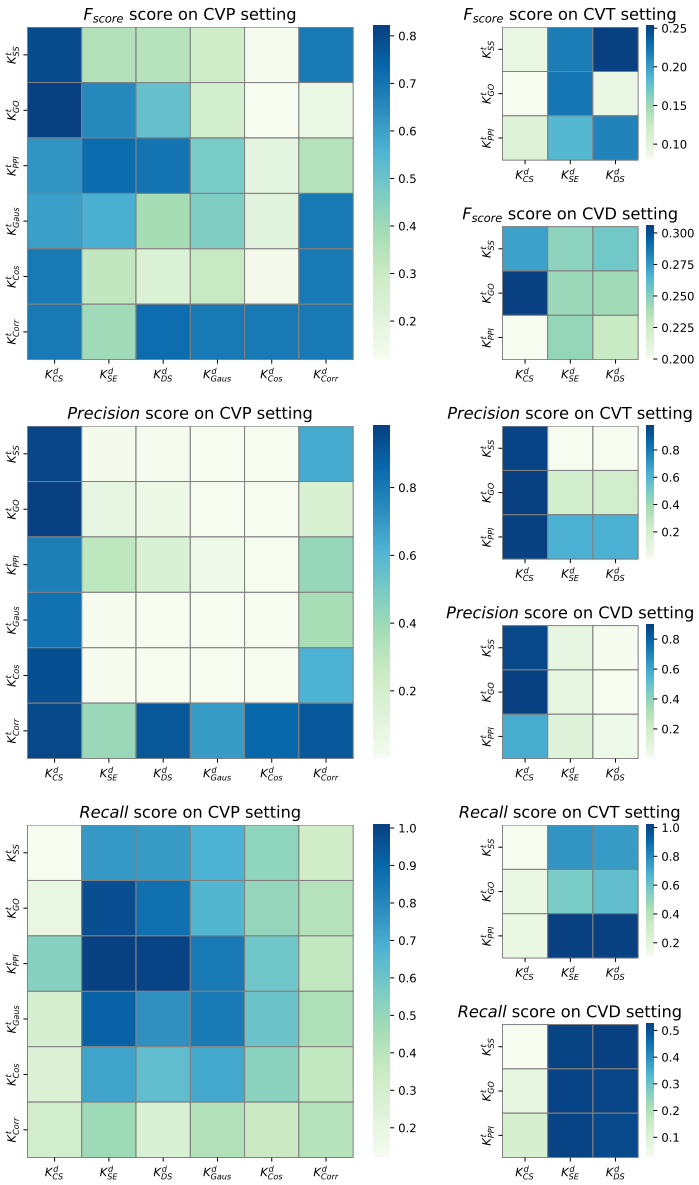

Fig. S13. Heatmaps of F_{score}, *Precision* and *Recall* results for CVP, CVD, and CVT settings on E dataset, showing the performance degradation of single-kernel compared to the MKL method. Deeper colors indicate a larger decrease in performance metrics relative to the MKL method.

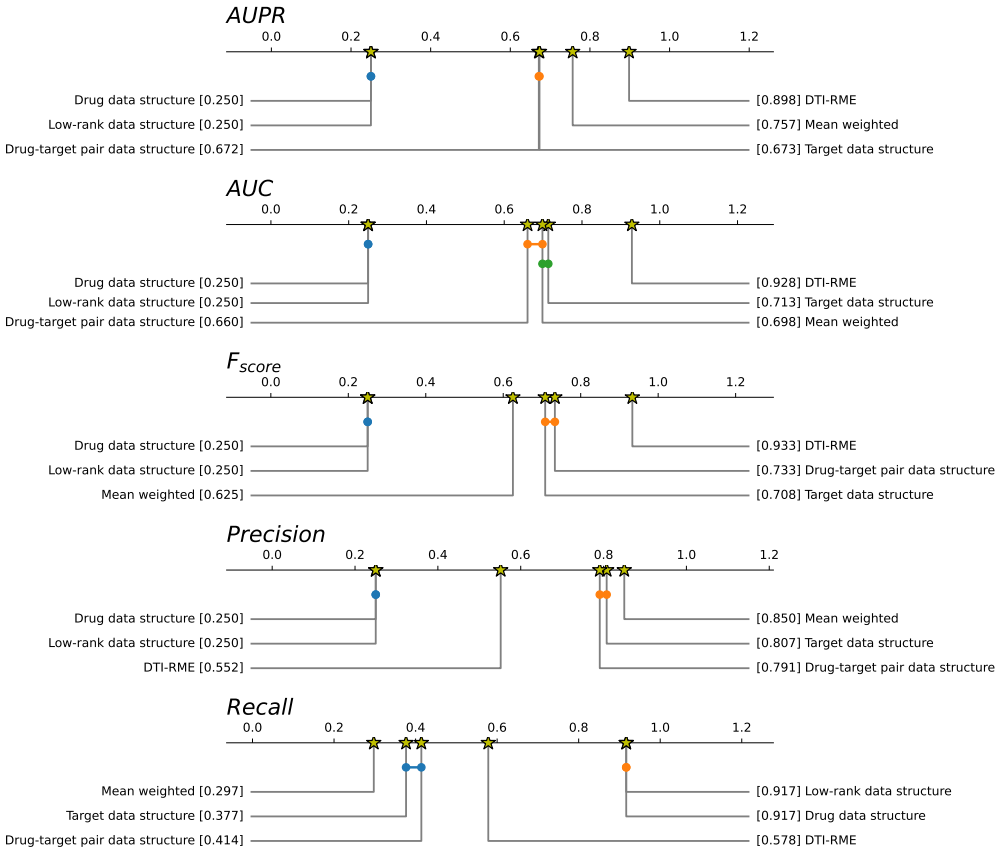

Fig. S14. Critical difference diagram showing the average score ranks of different data structure across 20 runs of CVD setting on five DTI datasets. A crossbar is over each group of methods that do not show a statistically significant difference among themselves.

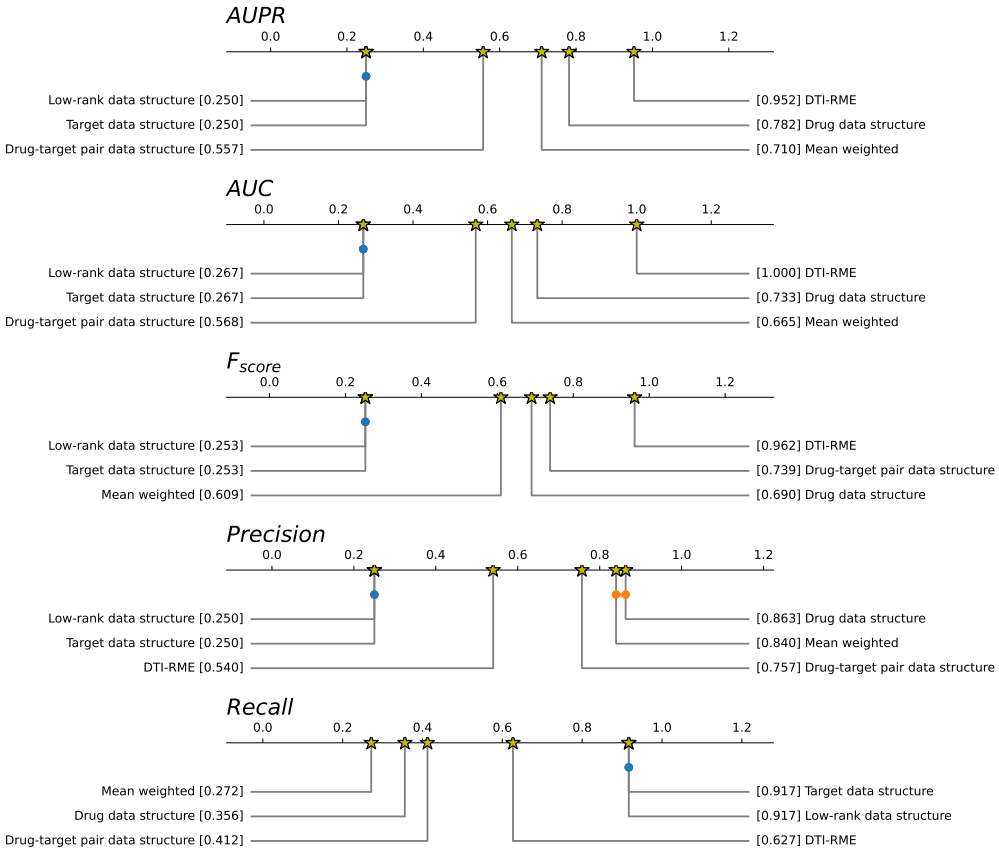

Fig. S15. Critical difference diagram showing the average score ranks of different data structure across 20 runs of CVT setting on five DTI datasets. A crossbar is over each group of methods that do not show a statistically significant difference among themselves.

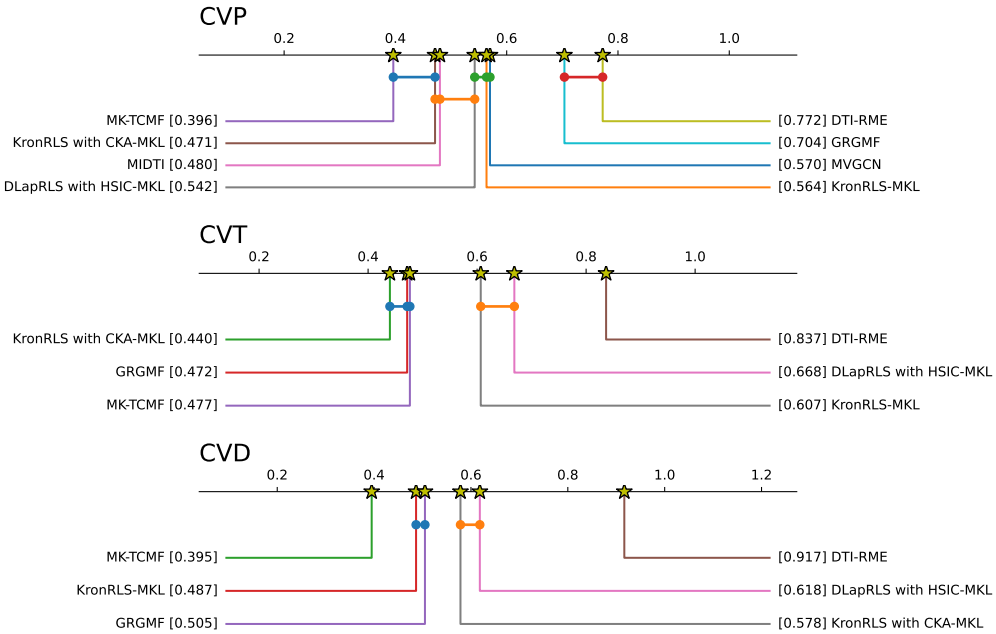

Fig. S16. Critical difference diagram showing the average score ranks of different baseline methods across all metrics under CVP, CVT, and CVD setting on NR datasets. A crossbar is over each group of methods that do not show a statistically significant difference among themselves.

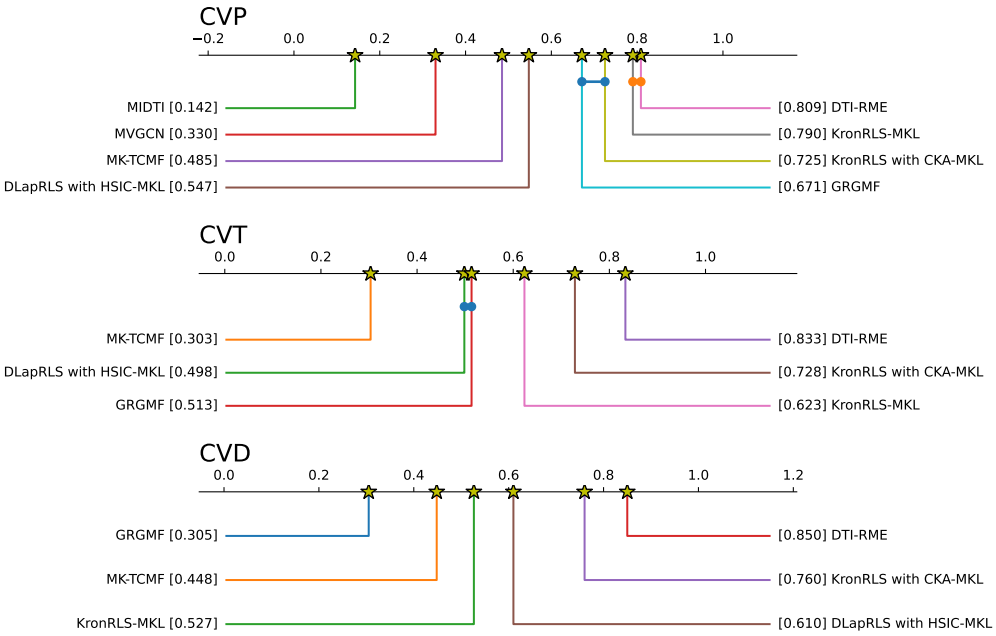

Fig. S17. Critical difference diagram showing the average score ranks of different baseline methods across all metrics under CVP, CVT, and CVD setting on IC datasets. A crossbar is over each group of methods that do not show a statistically significant difference among themselves.

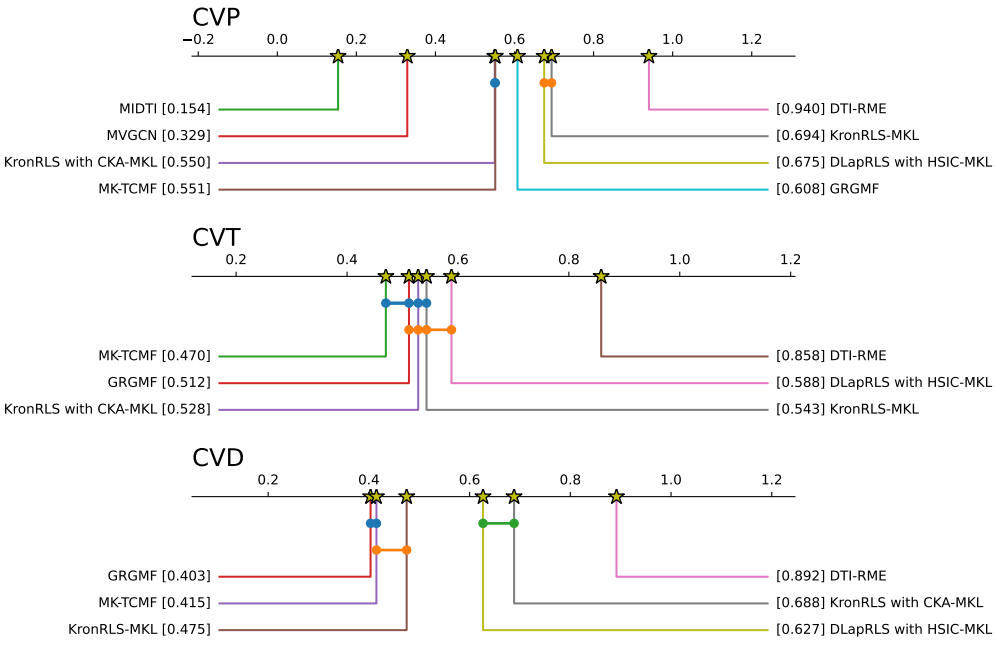

Fig. S18. Critical difference diagram showing the average score ranks of different baseline methods across all metrics under CVP, CVT, and CVD setting on GPCR datasets. A crossbar is over each group of methods that do not show a statistically significant difference among themselves.

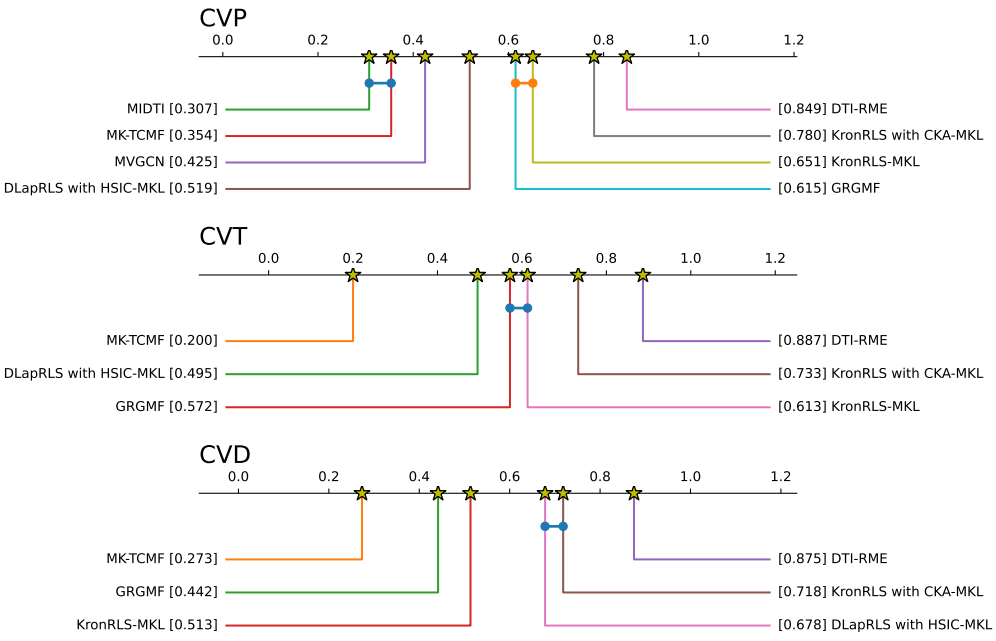

Fig. S19. Critical difference diagram showing the average score ranks of different baseline methods across all metrics under CVP, CVT, and CVD setting on E datasets. A crossbar is over each group of methods that do not show a statistically significant difference among themselves.

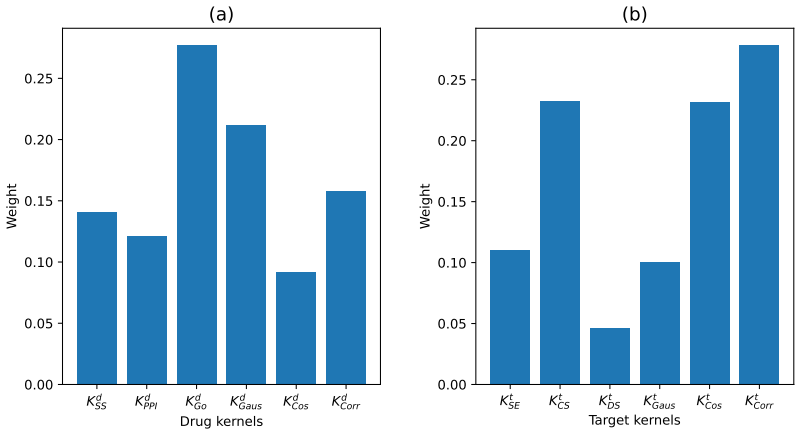

Fig. S20. Visualization of model interpretability components on NR dataset. **(a)**: Weights of drug kernels (β^d); **(b)**: Weights of target kernels (β^t).

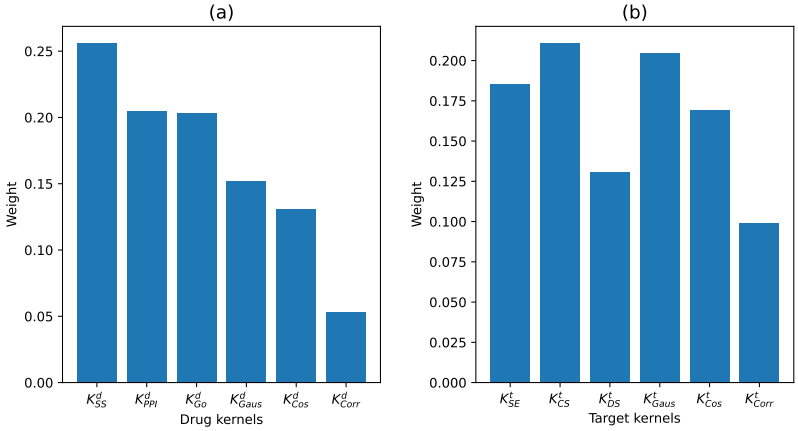

Fig. S21. Visualization of model interpretability components on IC dataset. **(a)**: Weights of drug kernels (β^d); **(b)**: Weights of target kernels (β^t).

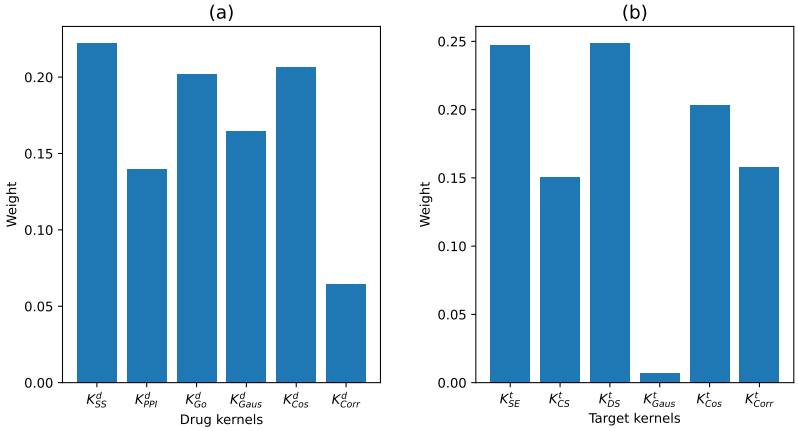

Fig. S22. Visualization of model interpretability components on GPCR dataset. **(a)**: Weights of drug kernels (β^d); **(b)**: Weights of target kernels (β^t).

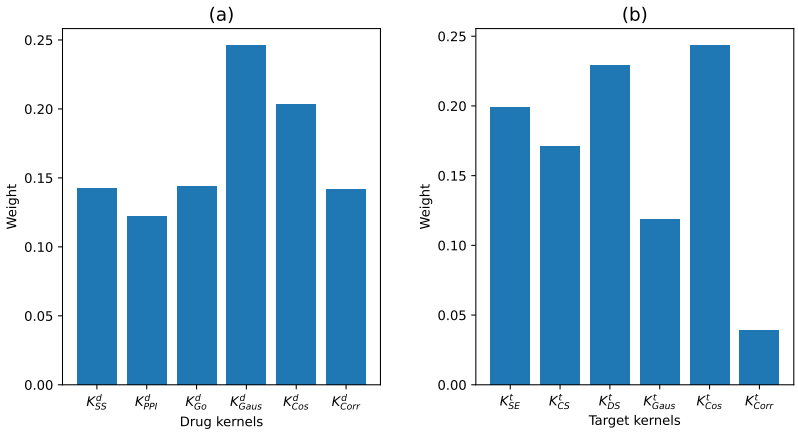

Fig. S23. Visualization of model interpretability components on E dataset. **(a)**: Weights of drug kernels (β^d); **(b)**: Weights of target kernels (β^t).

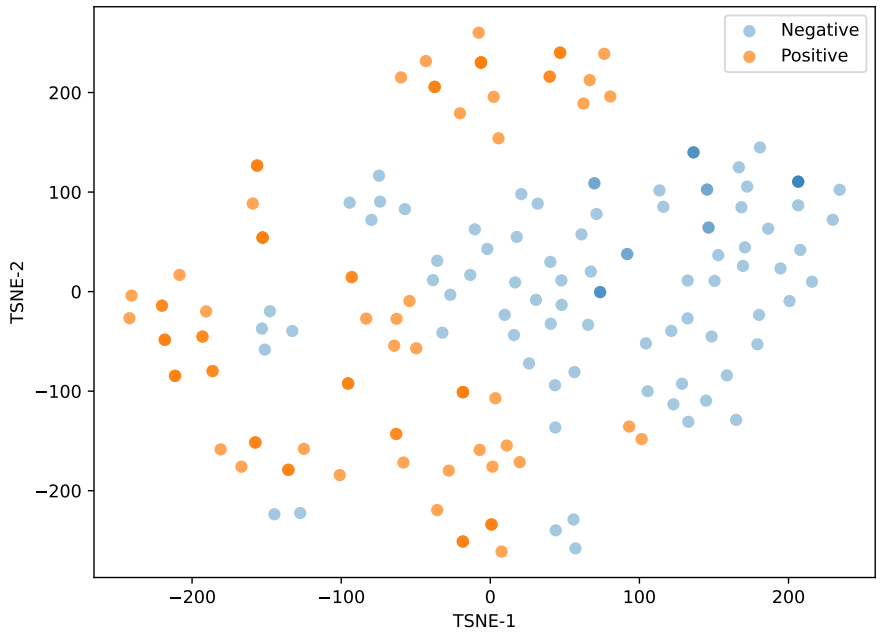

Fig. S24. t-SNE visualization of drug-target pair representations ($U - V$) on NR dataset.

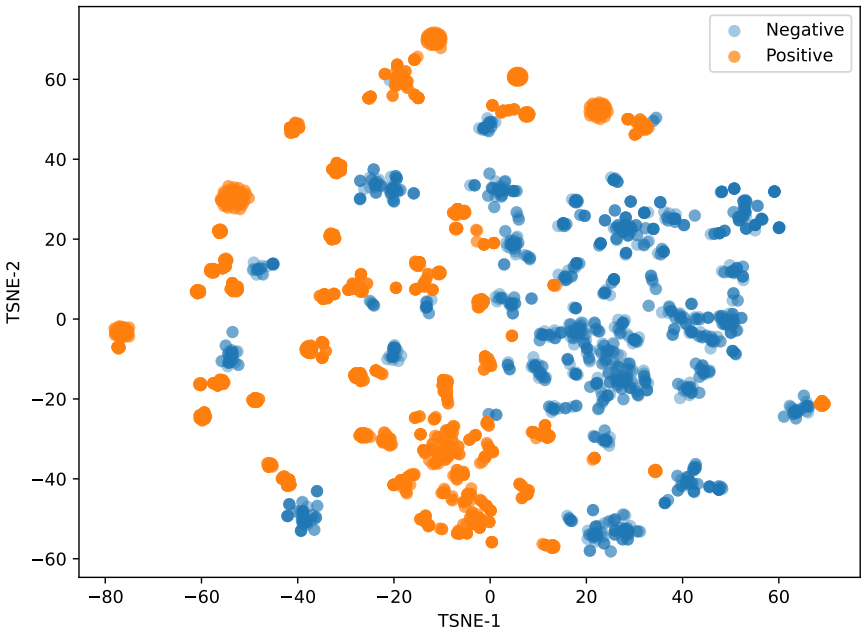

Fig. S25. t-SNE visualization of drug-target pair representations ($U - V$) on IC dataset.

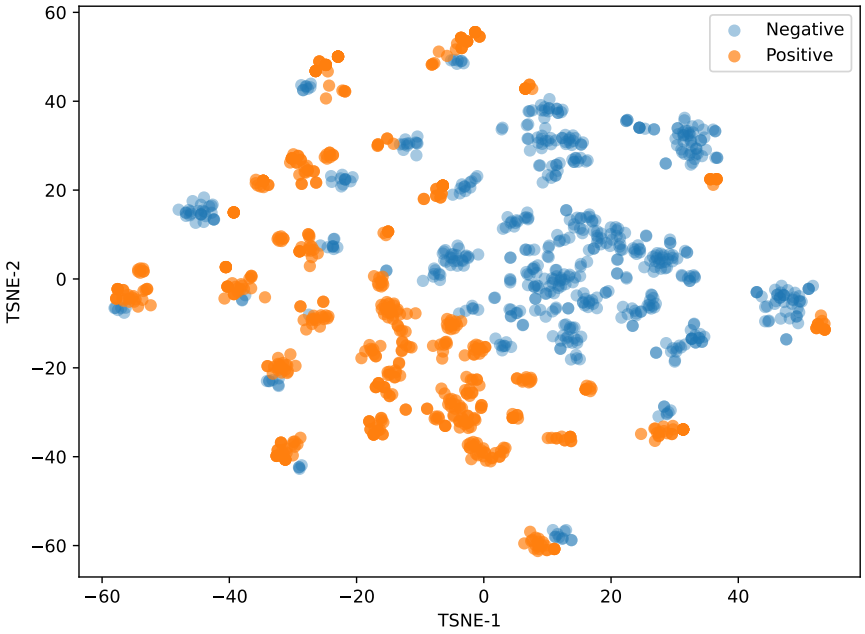

Fig. S26. t-SNE visualization of drug-target pair representations ($U - V$) on GPCR dataset.

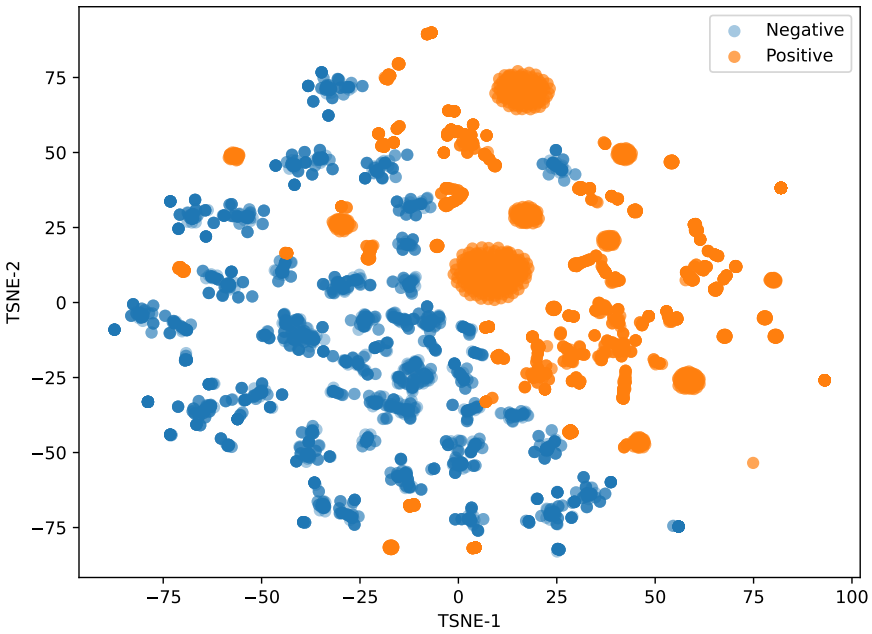

Fig. S27. t-SNE visualization of drug-target pair representations ($U - V$) on E dataset.

3. SUPPLEMENTARY TABLES

Table S1. Comparison of DTI-RME (Ours) and baseline methods across CV settings on NR dataset.

CV setting	Method	$AUPR$(%)	AUC(%)	F_{score}(%)	$Precision$(%)	$Recall$(%)
CVP	KronRLS-MKL	47.44±2.83	80.94±1.71	34.53±4.74	82.94±7.13	23.22±4.46
	DLapRLS with HSIC-MKL	48.75±2.46	81.83±2.31	32.38±3.47	81.13±6.88	21.61±3.00
	KronRLS with CKA-MKL	41.36±2.58	82.27±1.04	35.00±3.18	70.60±5.59	24.56±2.84
	MK-TCMF	44.86±2.56	81.15±2.38	28.51±3.63	77.34±9.90	18.78±3.16
	GRGMF	48.85±2.88	85.81±1.53	36.87±3.07	**84.17±5.70**	24.31±2.52
	MVGCN	39.86±4.21	81.73±2.91	44.16±4.55	58.84±6.37	**37.06±4.77**
	MIDTI	36.76±3.67	80.50±2.59	40.66±4.32	57.36±4.51	33.37±5.20
	DTI-RME (Ours)	**52.85±3.43**	**88.34±1.23**	**40.65±4.50**	74.60±5.36	29.33±4.35
CVD	KronRLS-MKL	36.62±2.36	79.87±2.59	35.62±2.75	62.11±5.17	27.01±3.19
	DLapRLS with HSIC-MKL	37.33±2.52	76.43±2.42	40.94±3.16	60.97±4.17	33.00±3.68
	KronRLS with CKA-MKL	31.95±2.07	80.39±1.51	33.96±2.98	43.39±3.56	30.20±3.64
	MK-TCMF	32.40±2.97	73.55±3.50	35.83±3.19	**62.49±5.90**	28.20±3.76
	GRGMF	31.63±2.67	81.14±1.46	30.18±2.21	23.29±2.28	46.86±4.34
	DTI-RME (Ours)	**40.20±2.99**	**81.70±1.58**	**63.72±2.26**	51.75±4.11	**48.27±2.83**
CVT	KronRLS-MKL	29.84±2.60	59.01±3.63	31.24±3.13	32.88±3.89	32.36±4.41
	DLapRLS with HSIC-MKL	31.40±3.83	56.34±3.49	35.81±3.13	35.65±3.63	39.60±5.17
	KronRLS with CKA-MKL	29.82±3.62	57.69±5.24	34.43±2.26	43.84±5.08	30.68±5.00
	MK-TCMF	26.56±4.07	52.74±4.27	30.46±3.64	38.36±8.77	30.41±4.29
	GRGMF	26.81±3.76	75.45±2.69	18.96±3.02	46.37±8.96	13.06±2.63
	DTI-RME (Ours)	**37.10±4.30**	**76.42±3.41**	**41.68±4.04**	**59.60±8.39**	**36.68±5.68**

Table S2. Comparison of DTI-RME (Ours) and baseline methods across CV settings on GPCR dataset.

CV setting	Method	$AUPR$(%)	AUC(%)	F_{score}(%)	$Precision$(%)	$Recall$(%)
CVP	KronRLS-MKL	63.25±0.95	93.43±0.32	56.17±1.77	76.16±2.38	44.94±2.15
	DLapRLS with HSIC-MKL	64.85±1.29	91.25±0.64	46.84±3.26	87.75±2.52	32.56±3.05
	KronRLS with CKA-MKL	60.73±1.47	91.00±0.43	44.21±2.64	87.42±2.35	30.03±2.34
	MK-TCMF	60.16±1.29	91.15±0.60	51.86±1.91	77.91±1.84	39.17±1.99
	GRGMF	62.31±1.19	93.80±0.42	39.52±2.22	**89.48±1.56**	25.49±1.81
	MVGCN	56.76±1.54	89.95±0.73	38.26±4.09	80.70±2.52	25.82±3.71
	MIDTI	50.82±1.54	87.77±0.71	22.93±4.96	75.64±2.81	14.25±4.03
	DTI-RME (Ours)	**67.81±1.20**	**94.34±0.46**	**58.58±1.69**	88.22±1.55	**49.85±1.91**
CVD	KronRLS-MKL	38.27±1.39	81.21±1.18	21.93±1.76	**76.50±4.22**	13.13±1.22
	DLapRLS with HSIC-MKL	36.92±1.34	75.85±0.93	30.01±2.60	71.47±3.26	19.99±2.25
	KronRLS with CKA-MKL	36.61±1.41	87.28±0.67	21.94±1.89	68.89±3.10	13.46±1.43
	MK-TCMF	32.83±0.96	80.17±1.03	26.71±2.36	67.17±2.51	17.19±1.93
	GRGMF	26.22±1.58	82.91±0.94	31.05±1.83	35.11±2.03	28.77±2.24
	DTI-RME (Ours)	**40.72±1.68**	**89.11±0.63**	**37.33±2.25**	59.91±3.06	**35.03±2.15**
CVT	KronRLS-MKL	36.57±2.06	81.84±1.63	30.13±1.56	68.47±2.29	19.62±1.27
	DLapRLS with HSIC-MKL	43.17±1.99	72.42±1.09	41.25±3.40	74.34±5.42	30.46±3.94
	KronRLS with CKA-MKL	43.74±2.53	82.77±1.03	35.38±3.15	74.24±3.29	23.65±2.65
	MK-TCMF	33.52±2.02	77.77±1.43	13.34±3.81	**78.67±5.65**	7.94±2.86
	GRGMF	23.98±2.01	83.79±0.85	7.67±2.37	70.14±10.66	4.24±1.51
	DTI-RME (Ours)	**55.84±2.33**	**88.65±0.70**	**54.71±2.74**	69.91±3.84	**46.43±2.85**

Table S3. Comparison of DTI-RME (Ours) and baseline methods across CV settings on IC dataset.

CV setting	Method	*AUPR*(%)	*AUC*(%)	F_{score}(%)	*Precision*(%)	*Recall*(%)
CVP	KronRLS-MKL	88.39±0.30	**98.02±0.14**	69.29±1.31	98.18±0.41	53.69±1.59
	DLapRLS with HSIC-MKL	86.28±0.48	96.25±0.32	56.56±2.73	**99.16±0.30**	39.88±2.65
	KronRLS with CKA-MKL	87.35±0.27	97.82±0.14	64.57±1.22	98.60±0.52	48.15±1.44
	MK-TCMF	85.88±0.45	97.27±0.20	51.06±2.62	99.00±0.39	34.71±2.42
	GRGMF	87.00±0.40	98.41±0.11	60.61±0.94	98.09±0.44	43.90±0.98
	MVGCN	78.24±0.37	94.73±0.40	50.83±3.96	98.19±0.63	34.83±3.61
	MIDTI	74.78±0.46	94.00±0.38	27.34±5.58	96.63±1.75	17.12±4.75
	DTI-RME (Ours)	**89.34±0.41**	97.97±0.15	**73.65±0.91**	97.36±0.40	**59.36±1.18**
CVD	KronRLS-MKL	34.51±2.15	78.07±1.38	23.09±2.18	72.77±5.93	14.82±1.81
	DLapRLS with HSIC-MKL	27.68±1.23	70.87±1.03	26.08±3.82	62.12±5.16	18.31±3.81
	KronRLS with CKA-MKL	35.36±2.05	**81.84±0.90**	26.12±6.80	**74.72±6.80**	17.29±3.72
	MK-TCMF	18.98±0.80	65.67±1.37	9.78±1.49	72.96±4.97	5.33±0.89
	GRGMF	25.02±1.91	75.15±1.78	29.61±1.92	43.05±2.85	24.15±1.78
	DTI-RME (Ours)	**36.22±1.28**	80.70±1.18	**38.23±2.52**	54.57±3.86	**31.39±2.44**
CVT	KronRLS-MKL	66.15±1.96	93.52±0.50	41.84±2.76	87.28±2.02	28.18±2.35
	DLapRLS with HSIC-MKL	72.57±0.93	89.54±0.42	66.96±2.08	85.32±1.32	55.76±3.01
	KronRLS with CKA-MKL	74.57±1.35	94.43±0.30	54.06±2.74	91.24±1.30	39.10±2.59
	MK-TCMF	63.31±1.18	90.22±0.41	29.10±2.46	89.76±1.74	17.78±1.87
	GRGMF	32.54±2.11	77.11±1.62	9.25±1.87	**92.81±3.94**	4.92±1.07
	DTI-RME (Ours)	**80.41±0.60**	**95.79±0.20**	**76.05±1.21**	84.28±1.59	**69.73±1.76**

Table S4. Comparison of DTI-RME (Ours) and baseline methods across CV settings on E dataset.

CV setting	Method	$AUPR$(%)	AUC(%)	F_{score}(%)	$Precision$(%)	$Recall$(%)
CVP	KronRLS-MKL	83.29±0.86	94.47±0.66	58.89±1.92	98.95±0.17	42.05±1.95
	DLapRLS with HSIC-MKL	81.25±0.37	93.66±0.23	41.94±2.41	99.24±0.18	26.78±1.95
	KronRLS with CKA-MKL	83.22±0.32	97.21±0.12	62.45±1.21	99.02±0.12	45.68±1.31
	MK-TCMF	80.37±0.46	95.92±0.22	18.32±2.86	98.40±0.63	10.31±1.87
	GRGMF	81.41±0.36	**98.36±0.09**	56.16±0.68	98.58±0.23	39.29±0.65
	MVGCN	77.58±0.36	91.51±0.19	40.67±4.21	99.36±0.28	26.07±3.39
	MIDTI	76.56±0.38	91.10±0.23	18.37±5.53	**99.57±0.36**	11.08±4.05
	DTI-RME (Ours)	**84.17±0.33**	**97.24±0.13**	**66.56±0.86**	98.99±0.14	**50.17±0.98**
CVD	KronRLS-MKL	32.98±1.38	76.85±1.25	9.55±1.53	**97.43±1.82**	5.08±0.85
	DLapRLS with HSIC-MKL	22.94±0.72	70.69±1.11	10.95±1.47	91.63±2.06	5.88±0.85
	KronRLS with CKA-MKL	33.26±1.27	83.04±1.28	11.99±1.46	96.61±2.43	6.45±0.84
	MK-TCMF	9.99±0.42	65.32±1.16	5.22±0.59	68.23±4.71	2.75±0.35
	GRGMF	23.62±1.94	75.16±1.87	30.70±2.62	56.56±4.82	21.68±2.19
	DTI-RME (Ours)	**40.98±1.21**	**85.13±1.17**	**45.40±1.72**	75.73±3.04	**33.08±1.67**
CVT	KronRLS-MKL	69.95±1.07	93.54±0.93	38.08±2.41	95.86±1.30	24.29±1.98
	DLapRLS with HSIC-MKL	76.47±0.70	89.90±0.63	63.42±2.68	97.05±0.47	47.68±2.73
	KronRLS with CKA-MKL	77.90±1.11	93.07±0.97	48.23±1.72	**98.28±0.17**	32.34±1.51
	MK-TCMF	60.55±0.95	89.35±1.10	9.21±2.67	92.07±2.14	5.13±1.76
	GRGMF	40.30±1.85	92.89±0.83	46.81±1.61	47.16±1.91	47.61±2.86
	DTI-RME (Ours)	**81.22±1.43**	**95.23±0.34**	**80.74±1.52**	90.65±1.50	**73.07±1.65**

REFERENCES

1. I. Gohberg, S. Goldberg, M. A. Kaashoek, *et al.*, "Hilbert-schmidt operators," Classes Linear Oper. Vol. I pp. 138–147 (1990).
2. T. Van Laarhoven, S. B. Nabuurs, and E. Marchiori, "Gaussian interaction profile kernels for predicting drug–target interaction," Bioinformatics **27**, 3036–3043 (2011).
3. A. J. Laub, *Matrix analysis for scientists and engineers* (SIAM, 2004).
4. J. Nocedal and S. J. Wright, "Quadratic programming," Numer. optimization pp. 448–492 (2006).
5. P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, "A practical anti-cycling procedure for linearly constrained optimization," Math. Program. **45**, 437–474 (1989).
6. A. C. Nascimento, R. B. Prudêncio, and I. G. Costa, "A multiple kernel learning algorithm for drug-target interaction prediction," BMC bioinformatics **17**, 1–16 (2016).
7. H. Yang, Y. Ding, J. Tang, and F. Guo, "Drug–disease associations prediction via multiple kernel-based dual graph regularized least squares," Appl. Soft Comput. **112**, 107811 (2021).
8. S. Boyd and L. Vandenberghe, *Convex optimization* (Cambridge university press, 2004).
9. M. Nikolova and R. H. Chan, "The equivalence of half-quadratic minimization and the gradient linearization iteration," IEEE Transactions on Image Process. **16**, 1623–1627 (2007).
10. Y. Ding, J. Tang, and F. Guo, "Identification of drug–target interactions via dual laplacian regularized least squares with multiple kernel fusion," Knowledge-Based Syst. **204**, 106254 (2020).
11. Z. Xia, L.-Y. Wu, X. Zhou, and S. T. Wong, "Semi-supervised drug-protein interaction prediction from heterogeneous biological spaces," in *BMC systems biology,* vol. 4 (Springer, 2010), pp. 1–16.
12. A. Cichonska, T. Pahikkala, S. Szedmak, *et al.*, "Learning with multiple pairwise kernels for drug bioactivity prediction," Bioinformatics **34**, i509–i518 (2018).
13. Y. Ding, J. Tang, F. Guo, and Q. Zou, "Identification of drug–target interactions via multiple kernel-based triple collaborative matrix factorization," Briefings Bioinforma. **23**, bbab582 (2022).
14. Z.-C. Zhang, X.-F. Zhang, M. Wu, *et al.*, "A graph regularized generalized matrix factorization model for predicting links in biomedical bipartite networks," Bioinformatics **36**, 3474–3481 (2020).
15. H. Fu, F. Huang, X. Liu, *et al.*, "Mvgcn: data integration through multi-view graph convolutional network for predicting links in biomedical bipartite networks," Bioinformatics **38**, 426–434 (2022).
16. W. Song, L. Xu, C. Han, *et al.*, "Drug–target interaction predictions with multi-view similarity network fusion strategy and deep interactive attention mechanism," Bioinformatics **40** (2024).
17. M. Wen, Z. Zhang, S. Niu, *et al.*, "Deep-learning-based drug–target interaction prediction," J. proteome research **16**, 1401–1409 (2017).
18. H. Wang, J. Wang, C. Dong, *et al.*, "A novel approach for drug-target interactions prediction based on multimodal deep autoencoder," Front. pharmacology **10**, 1592 (2020).
19. N. R. Monteiro, B. Ribeiro, and J. P. Arrais, "Drug-target interaction prediction: end-to-end deep learning approach," IEEE/ACM transactions on computational biology bioinformatics **18**, 2364–2374 (2020).
20. H. Öztürk, A. Özgür, and E. Ozkirimli, "Deepdta: deep drug–target binding affinity prediction," Bioinformatics **34**, i821–i829 (2018).
21. I. Lee, J. Keum, and H. Nam, "Deepconv-dti: Prediction of drug-target interactions via deep learning with convolution on protein sequences," PLoS computational biology **15**, e1007129 (2019).
22. Q. Ye, X. Zhang, and X. Lin, "Drug-target interaction prediction via graph auto-encoder and multi-subspace deep neural networks," IEEE/ACM Transactions on Comput. Biol. Bioinforma. **20**, 2647–2658 (2022).
23. J. Park, M. Lee, H. J. Chang, *et al.*, "Symmetric graph convolutional autoencoder for unsupervised graph representation learning," in *Proceedings of the IEEE/CVF international conference on computer vision,* (2019), pp. 6519–6528.
24. D. A. Case, H. M. Aktulga, K. Belfon, *et al.*, "Ambertools," J. chemical information modeling **63**, 6183–6191 (2023).
25. C. Tian, K. Kasavajhala, K. A. Belfon, *et al.*, "ff19sb: amino-acid-specific protein backbone parameters trained against quantum mechanics energy surfaces in solution," J. chemical theory computation **16**, 528–552 (2019).