DecoyDB: A Dataset for Graph Contrastive Learning in Protein-Ligand Binding Affinity Prediction

Yupu Zhang[1] **Zelin Xu**[1] **Tingsong Xiao**[1] **Gustavo Seabra**[2]
Yanjun Li[1,2] **Chenglong Li**[2] **Zhe Jiang**[1*]

[1]Department of CISE, University of Florida, Gainesville, FL, USA
[2]Department of Medicinal Chemistry, University of Florida, Gainesville, FL, USA
{y.zhang1, zelin.xu, xiaotingsong, yanjun.li, zhe.jiang}@ufl.edu
{seabra, lic}@cop.ufl.edu

Abstract

Predicting the binding affinity of protein-ligand complexes plays a vital role in drug discovery. Unfortunately, progress has been hindered by the lack of large-scale and high-quality binding affinity labels. The widely used PDBbind dataset has fewer than 20K labeled complexes. Self-supervised learning, especially graph contrastive learning (GCL), provides a unique opportunity to break the barrier by pretraining graph neural network models based on vast unlabeled complexes and fine-tuning the models on much fewer labeled complexes. However, the problem faces unique challenges, including a lack of a comprehensive unlabeled dataset with well-defined positive/negative complex pairs and the need to design GCL algorithms that incorporate the unique characteristics of such data. To fill the gap, we propose DecoyDB[2], a large-scale, structure-aware dataset specifically designed for self-supervised GCL on protein–ligand complexes. DecoyDB consists of high-resolution ground truth complexes (≤ 2.5Å) and diverse decoy structures with computationally generated binding poses that range from realistic to suboptimal. Each decoy is annotated with a Root Mean Square Deviation (RMSD) from the native pose. We further design a customized GCL framework to pretrain graph neural networks based on DecoyDB and fine-tune the models with labels from PDBbind. Extensive experiments confirm that models pretrained with DecoyDB achieve superior accuracy, sample efficiency, and generalizability.

1 Introduction

Drug discovery is a lengthy and costly endeavor that involves identifying highly potent molecules capable of interacting with specific molecular targets, such as proteins, to treat diseases. Predicting protein–ligand binding affinity is a fundamental task in the early stages of drug discovery [Kitchen *et al.*, 2004], with two major applications. First, it enables virtual screening and *de novo* design—key computer-aided drug discovery techniques—to efficiently narrow down promising hit compounds from a vast chemical space targeting a specific protein [Li *et al.*, 2022b]. Second, it supports the subsequent optimization of these identified molecules to further improve their binding affinities and other pharmacological properties [Li *et al.*, 2020].

Classical methods typically employ theory-inspired, fixed functional forms that utilize predefined features extracted from the protein–ligand complex to estimate affinity. Although certain computational techniques such as molecular mechanics/Poisson–Boltzmann or generalized Born surface

[*]Corresponding author. Email: zhe.jiang@ufl.edu

[2]Code and data are available at: huggingface.co/datasets/jiangteam/DecoyDB.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Track on Datasets and Benchmarks.

area (MM/PBSA, MM/GBSA) [Genheden and Ryde, 2015], thermodynamic integration (TI) [Kirkwood, 1935], free energy perturbation (FEP) [Zwanzig, 1954], can offer decent precision, they are computationally intensive and impractical for use in virtual screening.

In recent years, deep learning-based affinity prediction has seen significant improvement [Rezaei *et al.*, 2022; Zhang *et al.*, 2024], with models utilizing 3D convolutional neural networks (CNNs) [Stepniewska-Dziubinska *et al.*, 2018b; Li *et al.*, 2019] and graph neural networks (GNNs) [Lim *et al.*, 2019; Yang *et al.*, 2023] to learn structural representation from ligands and proteins. Despite these advances, the accuracy of current scoring functions remains unsatisfactory, and further improvements are hindered by the limited availability of ground truth protein-ligand binding affinity labels. For instance, the widely used PDBbind dataset [Wang *et al.*, 2005] contains fewer than 20,000 labeled complexes. Such labels are typically obtained through laborious and time-consuming experiments and thus are unlikely to grow dramatically. In contrast, there is an abundance of unlabeled protein-ligand complexes in structural databases, which could potentially enhance the learning of spatial structural interaction features for binding affinity prediction.

Self-supervised learning (e.g., masked autoencoders, contrastive learning) has shown great promise for pretraining deep neural networks on vast amounts of unlabeled data, followed by fine-tuning with limited labeled samples for diverse downstream tasks such as natural language processing [Radford *et al.*, 2018] and computer vision [He *et al.*, 2022]. Among these approaches, contrastive learning has emerged as a particularly effective framework for learning representations from unlabeled graph data [He *et al.*, 2020; Chen *et al.*, 2020], including protein–ligand complexes [Luo *et al.*, 2024]. Motivated by this success, we adopt contrastive learning as a pretraining strategy to leverage large-scale unlabeled protein–ligand complexes, enabling subsequent fine-tuning on smaller labeled datasets for accurate binding affinity prediction.

However, several unique challenges arise in this context. First, there is a lack of a comprehensive unlabeled dataset containing well-defined positive and negative complex pairs, which are essential for contrastive learning. Second, preserving the original 3D structural integrity of protein–ligand complexes is critical for maintaining biochemical validity. Conventional graph contrastive learning techniques—such as generating augmented pairs through node or edge perturbations [Chen *et al.*, 2020; Wang and Qi, 2022]—may produce unrealistic conformations that violate physical and chemical constraints [Qin *et al.*, 2024]. Third, the downstream task of binding affinity prediction requires the model to capture interactions corresponding to low-energy (stable) conformations, a property that standard contrastive learning objectives typically overlook [Liu *et al.*, 2023].

To fill this gap, we propose *DecoyDB*, a comprehensive dataset of high-resolution, ground-truth 3D protein–ligand complexes filtered from the PDB, augmented with diverse decoy complexes generated computationally to produce binding poses ranging from realistic (positive pairs) to suboptimal (negative pairs). *DecoyDB* contains 61,104 ground-truth 3D complexes and 5,353,307 decoys, and each decoy is annotated with its Root Mean Square Deviation (**RMSD**) from the native pose, i.e., the spatial distances between the atoms of the decoy's ligand and the corresponding atoms in the original ligand. We further design a customized graph contrastive learning algorithm that incorporates (1) a two-category contrastive loss, where negative samples are drawn both from decoys of the same complex (with varying RMSD levels) and from different real complexes, and (2) a denoising score matching (DSM)-based regularization term in the loss. Extensive experiments demonstrate that our pretraining framework improves base models in prediction accuracy, sample efficiency, and generalizability. It is worth noting that although we focus on enhancing binding affinity prediction, *DecoyDB* also has potential for broader applications, such as molecular docking and virtual screening.

2 Related Work

Deep Learning for Protein-Ligand Binding Affinity Prediction: In recent years, deep learning has emerged as a powerful tool for predicting protein-ligand binding affinity. Earlier methods are mostly based on CNNs, including Pafnucy [Stepniewska-Dziubinska *et al.*, 2018b], OnionNet [Zheng *et al.*, 2019], and DeepAtom [Li *et al.*, 2019], which typically rasterize the binding pocket and ligand into a grid structure. However, these methods are inherently dependent on the grid resolution and do not consider the topological structure of a complex. In addition, some methods for drug target affinity prediction, such as DeepDTA [Lennox *et al.*, 2021] and GraphDTA [Nguyen *et al.*, 2020], miss the interaction structures. Recent works have shifted towards graph neural networks (GNNs) to

Table 1: Public datasets related to protein–ligand complexes for binding affinity prediction.

Category	**Dataset**	**# of complexes**	**3D structure**	**Affinity**	**Measurement**
Both affinity labels and 3D structure	PDBbind2013	10,370	Exp.	Exp.	IC_{50}, K_d, K_i
	PDBbind2016	13,189	Exp.	Exp.	IC_{50}, K_d, K_i
	PDBbind2020	19,443	Exp.	Exp.	IC_{50}, K_d, K_i
	MISATO	19,443	Exp.+QC+MD	Exp.	IC_{50}, K_d, K_i
	LP-PDBbind	18,795	Exp.	Exp.	IC_{50}, K_d, K_i
	BDB2020+	115	Exp.	Exp.	IC_{50}, K_d, K_i
	HiQBind	32,275	Exp. + Refined	Exp.	IC_{50}, K_d, K_i
	Binding MOAD	41,409	Exp.	Exp.	IC_{50}, K_d, K_i
	BioLip	48,291	Exp.	Exp.	IC_{50}, K_d, K_i
	BindingNet	69,816	Exp.+Comp.	Exp.	IC_{50}, K_d, K_i
Only affinity labels, no 3D structure	KIBA	117,657	-	Exp.	KIBA (IC_{50}, K_d, K_i)
	Davis	30,056	-	Exp.	K_d
	BindingDB	679,000	-	Exp.	IC_{50}, K_d, K_i
No affinity labels, only 3D structure	PDB (05/2025)	178,900	Exp.	-	-
	Redocked 2020	786,960	Exp. + Decoys	-	-
	DecoyDB	5,414,411	Exp. + Decoys	-	-

learn a flexible representation of 3D graph structures, such as PotentialNet [Feinberg *et al.*, 2018], IGN [Jiang *et al.*, 2021], EGNN [Satorras *et al.*, 2021], PSICHIC [Koh *et al.*, 2024], and GIGN [Yang *et al.*, 2023].

Graph Contrastive Learning: Generic graph contrastive learning (GCL) methods generate positive and negative sample pairs through random node dropping, perturbation, or subgraph sampling [You *et al.*, 2020; Xu *et al.*, 2021; Tong *et al.*, 2021], which often disrupt biochemical properties. Although recent advances have introduced GCL frameworks specifically for molecular data [Guan and Zhang, 2023; Fang *et al.*, 2023; Liu *et al.*, 2022; Sun *et al.*, 2021; Li *et al.*, 2022a], these models typically focus on a single molecular graph and fail to capture the interaction network between a protein and its ligand. Wu *et al.* [2022] proposed a self-supervised learning method but required additional molecular dynamics simulations. Ni *et al.* [2024] also evaluated their molecular foundation model on protein–ligand binding affinity prediction, but their base model was pretrained only on single molecules. Luo *et al.* [2024] introduced a two-step supervised learning framework for protein–ligand binding prediction, consisting of (1) a supervised learning step with an auxiliary contrastive loss based on decoys, and (2) a supervised parameter refinement step using true complex affinity labels only. However, this approach is not true pretraining, as the first step's loss includes pseudo-labels of binding affinity for decoys, and their decoy dataset was derived solely from PDBbind (relatively small in size). In addition, their contrastive loss does not account for varying degrees of negativeness among decoys with different deviations, and the anchors in their triplets are complex-free (lacking interaction structural information).

Protein-ligand complex datasets: Table 1 summarizes the existing public protein-ligand complex datasets. Most datasets are derived from the Protein Data Bank (PDB) [Berman *et al.*, 2000], a comprehensive repository of experimentally determined 3D biomolecular structures. PDB (as of 05/2025) provides around 178,900 protein-ligand complexes but does not provide binding affinity labels. To fill the gap, PDBbind provides fewer than 20K complexes with experimentally measured affinity and is the most widely used benchmark for binding affinity prediction. Several datasets aim to enrich or refine the PDBbind with additional structural or physical information. For example, MISATO [Siebenmorgen *et al.*, 2024] augments PDBbind entries with quantum-chemically (QC) optimized ligand geometries and 10-nanosecond molecular dynamics (MD) trajectories. LP-PDBbind [Li *et al.*, 2024a] provides a leakage-proof split of PDBbind based on structural information. In addition, the authors construct a high-quality test set, BDB2020+, by selecting new protein-ligand complexes published after PDBbind2020 for independent testing. There are several additional labeled datasets. Binding MOAD [Wagle *et al.*, 2023] provides 41,409 complexes, of which only 15,223 (37%) are annotated with affinity labels. BioLiP [Yang *et al.*, 2012] integrates Binding MOAD, PDBbind, and BindingDB, providing over 48,000 labeled complexes, but it does not apply resolution filtering and thus can contain low-quality 3D complex structures. HiQBind [Wang *et al.*, 2025] collects 32,275 protein-ligand complexes with refined structures. Despite their high-quality affinity labels, these datasets remain relatively small in scale, and there exist substantial overlaps between them. For example, BioLiP shares 26,009 complexes with Binding MOAD, and most complex structures in HiQBind are from Binding MOAD and BioLiP. To support larger-scale labels, BindingNet [Li *et al.*, 2024b] expands PDBbind to 69,816 complexes by superimposing ligands in the original complexes.

However, since the 3D complex structures are computationally generated rather than experimentally resolved, discrepancies may exist between the modeled conformations and their corresponding binding affinities. Separately, several other datasets such as KIBA [Tang *et al.*, 2014], Davis [Davis *et al.*, 2011], and BindingDB [Liu *et al.*, 2007] provide a larger number of binding affinity labels, but they do not provide corresponding 3D conformal structures. In summary, most existing datasets are designed exclusively for supervised learning and therefore cannot leverage the vast number of unlabeled 3D complexes. One existing dataset, Redocked2020 [Francoeur *et al.*, 2020], augments 3D complexes in PDBbind to 786,960 binding poses by adding decoys (via redocking ligands into protein structures) for self-supervised pretraining based on graph contrastive learning [Luo *et al.*, 2024]. However, since the 3D complexes are entirely derived from PDBbind, there exists leakage between pretraining and fine-tuning. In contrast, our DecoyDB is filtered from PDB with many complexes outside the PDBbind dataset, providing an opportunity to validate the generalizability of pretrained models during fine-tuning.

3 Problem Definition

Definition 1. *A **protein–ligand complex** refers to the binding interaction between a protein and a ligand. It can be represented as $s_k = (G_k, \mathbf{x}_k, \mathbf{a}_k)$, where $G_k = (V_k, E_k)$ is a 3D graph whose nodes ($v \in V_k$) represent atoms of the protein or ligand, and whose edges ($e \in E_k$) represent chemical interactions either within the protein, within the ligand, or between them (interaction edges). $\mathbf{x}_k \in \mathbb{R}^{|V_k| \times l}$ denotes the matrix of spatial node features ($l = 3$ for 3D coordinates). $\mathbf{a}_k \in \mathbb{R}^{|V_k| \times m}$ denotes the matrix of non-spatial node features (e.g., atom type or other chemical descriptors), where m is the number of features. $y_k \in \mathbb{R}$ is the binding affinity score associated with the complex, which is unavailable in an unlabeled dataset.*

The problem of protein-ligand binding affinity prediction can be formally defined as follows:

Input:
- A set of protein-ligand complexes $\mathcal{S} = \{s_k | 1 \leq k \leq K\}$.
- A small subset of ground truth binding affinity scores $\mathcal{Y} = \{y_k | 1 \leq k \leq K_l\}$, where $K_l << K$.
- A base GNN encoder f_θ and regression head g_ϕ: $\hat{y}_k = g_\phi(f_\theta(s_k))$

Output:
- Parameters θ of pretrained GNN encoder f_θ based on $\mathcal{S}$.
- Parameters ϕ in regression head g_ϕ based on $\mathcal{Y}$.

Objective:
- Minimize the prediction errors of the fine-tuned model.
- Maximize the label efficiency in fine-tuning.

4 Our *DecoyDB* Dataset

4.1 The Construction of *DecoyDB*

In order to use unlabeled complexes in self-supervised learning (graph contrastive learning), we need to first establish positive and negative sample pairs. Common data augmentation methods in general graph contrastive learning (e.g., edge perturbation, node dropping) are inadequate since they destroy biochemical structures. Thus, we constructed a specialized dataset to augment the original protein-ligand complexes with decoys. A decoy refers to an artificial protein-ligand complex with a computationally generated suboptimal binding pose. We name our augmented dataset *DecoyDB*.

Figure 1 illustrates our data construction process. We started by retrieving all available structures containing ligand-protein complexes from the PDB, focusing on entries obtained by X-ray crystallography with a resolution of 2.5Å or less. We refined our *DecoyDB* dataset by excluding complexes with ligands that had molecular weights outside the $(50, 700)$ range, metal clusters, monoatomic ions, common crystallization molecules, and ligands containing elements other than C (carbon), N (nitrogen), O (oxygen), H (hydrogen), S (sulfur), P (phosphorus), or X (halogens). To avoid redundancy and irrelevant protein chains, we isolated the target ligand in each remaining PDB entry and retained only those protein chains with at least one atom within 10Å of the ligand and saved a PDB file containing only the protein and ligand. This threshold captures relevant interactions, while excluding distant, noninteracting parts of the protein. In cases where multiple ligands were present

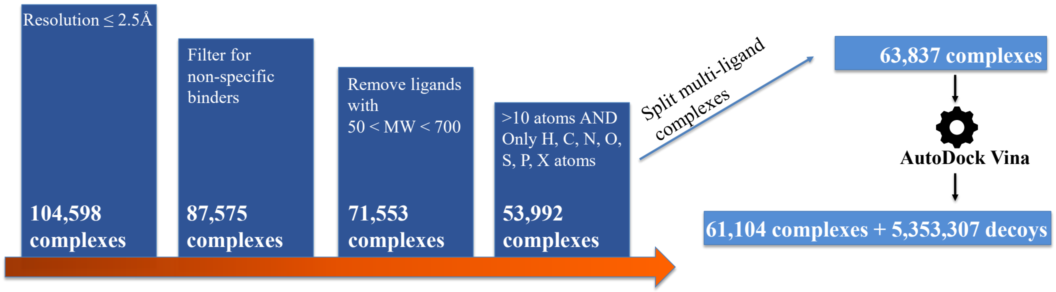
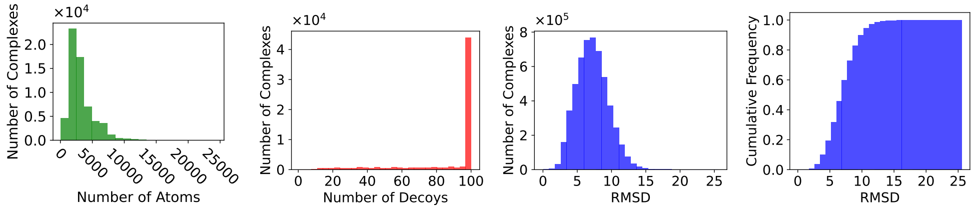

Resolution ≤ 2.5Å
104,598 complexes
Filter for non-specific binders
87,575 complexes
Remove ligands with 50 < MW < 700
71,553 complexes
>10 atoms AND Only H, C, N, O, S, P, X atoms
53,992 complexes
Split multi-ligand complexes
63,837 complexes
AutoDock Vina
61,104 complexes + 5,353,307 decoys

Figure 1: The data construction pipeline of DecoyDB.



(a) (b) (c) (d)

Figure 2: Statistical analysis of *DecoyDB*. (a) Distribution of the number of atoms in each protein-ligand complex. (b) Distribution of the number of decoys per complex. (c) Distribution of RMSD values for decoy complexes. (d) Cumulative distribution of RMSD values for decoy complexes.

within a complex, we generated a separate PDB file for each ligand-protein pair. We collected 63,837 3D complexes after the split.

To generate decoy complexes, we used AutoDock Vina 1.2 [Eberhardt *et al.*, 2021], one of the most widely used open-source programs for molecular docking. For each PDB file (a real protein-ligand complex), we defined a grid box around the ligand with a 5Å padding in each dimension to ensure sufficient space for ligand flexibility during redocking. We generated 100 different poses per ligand using an exhaustiveness parameter of 8, which balances computational efficiency with pose diversity. The final (RMSD) between the generated pose and the original crystallized ligand was calculated to quantify the deviation of the decoys from the native binding pose.

The resulting *DecoyDB* dataset contains 61,104 protein-ligand complexes and 5,353,307 decoys, with an average of 88 successfully generated decoys per complex. The data distribution is shown in Figure 2. The RMSD of the decoys ranges from 0.03Å to 25.56Å, with an average of 7.22Å. This wide range of RMSD values ensures a diverse set of decoys that represent both near-native (**positive samples**, RMSD $\leq$ 2Å) and far-from-native poses (**negative samples**, RMSD $>$ 2Å), with the 2 Å cutoff following docking success criteria [Buttenschoen *et al.*, 2024; Cole *et al.*, 2005]. Such diversity is crucial for contrastive learning, as it allows the model to learn from a broad range of binding interactions and spatial conformations, improving its generalization in predicting protein-ligand binding affinity.

4.2 A Customized Graph Contrastive Learning Framework

We now introduce a customized graph contrastive learning framework that pretrains a neural network based on two-category negative sample pairs and denoising-based regularization.

Two-category Graph Contrastive Loss with Continuous Negative Samples

DecoyDB augments the original real complexes with positive and negative sample pairs. For each real complex (anchor), we define positive pairs by matching the complex with its decoys whose RMSD is no greater than 2Å (positive samples), and define negative pairs in two categories: (1) matching the anchor complex with its decoys whose RMSD values is greater than 2Å), and (2) matching the anchor complex with a different complex. Based on the two-category negative pairs, we customize the graph contrastive loss.

Our proposed two-category InfoNCE contrastive loss is in Equation 1, where $\mathbf{z_k}$ is the embedding of a real complex (the *anchor*) from the encoder, and $\mathbf{z_i}$ is the embedding of one positive sample, $\mathbf{z_j}$ is the embedding of a negative sample, and N is the total number of samples in the minibatch. Each minibatch has one positive pair and $N-1$ negative pairs.

$$l_{k,i} = -\log \frac{\exp(sim(\mathbf{z_k}, \mathbf{z_i})/\tau)}{\sum_{j=1, j\neq i}^{N} \beta_{k,j} \exp\left(sim(\mathbf{z_k}, \mathbf{z_j})/\tau\right)} \tag{1}$$

One important factor in the above loss is $\beta_{k,j}$. It is defined in Equation 2, where $d_{\mathbf{z_k},\mathbf{z_j}}$ is the RMSD between embeddings $\mathbf{z_k}$ and $\mathbf{z_j}$, d_{max} is the max RMSD between the real complexes and their decoys (to normalize $d_{\mathbf{z_k},\mathbf{z_j}}$), sim is the cosine similarity function, and α is a hyper-parameter to balance the relative weight between two categories of negative samples. Specifically, if the negative sample $\mathbf{z_j}$ is from a decoy, we use a normalized continuous RMSD to reflect the extent of negativeness in the contrastive loss. For decoys that have more deviations from the anchor complex, our intuition is that their embeddings need to be pushed away further from the anchor. If the negative sample $\mathbf{z_j}$ is from a different complex (than the current anchor complex), then we use a binary weight $\beta = 1$.

$$\beta_{k,j} = \begin{cases} \alpha \frac{d_{\mathbf{z_k},\mathbf{z_j}}}{d_{max}} & \text{if } \mathbf{z_j} \text{ is a negative decoy of } \mathbf{z_k}, \\ 1 & \text{otherwise.} \end{cases} \tag{2}$$

Suppose there are K anchors in the complex dataset and each anchor has m positive pairs. The loss function L_1 is a sum over all complexes and their positive pairs:

$$L_1 = \frac{1}{m \times K} \sum_{k=1}^{K} \sum_{i=1}^{m} l_{k,i} \tag{3}$$

Denoising-based Regularization

Although the above two-category graph contrastive loss helps supervise the relative closeness of embedding pairs, it does not reflect the fact that each real complex has a binding pose with minimum energy. Inspired by a recent work [Jin *et al.*, 2024], we add a denoising-based regularization method. Our intuition is that the real complex structure is the most stable structure and represents the local minimum potential energy in its pose.

Specifically, we add Gaussian noise to the 3D coordinates of atoms in the complex to get perturbed complex $s' = (G, \mathbf{x}', \mathbf{a})$, i.e., $\mathbf{x}' = \mathbf{x} + \epsilon$, where $\epsilon \sim p(\epsilon) = \mathcal{N}(0, \sigma^2 I)$, $\mathbf{x}$ are the coordinates of the atoms and $\mathbf{x}'$ are perturbed coordinates of atoms. Note that we add Gaussian noise to the ligand atoms only. We apply DSM only to original complexes (not decoys). For simplicity, we reuse the same symbol of $\mathbf{x}$ to represent the coordinates of a ligand. The model's output denotes the structural energy. Our intuition is that the gradient of the model's score with respect to the input vanishes in the absence of noise. Specifically, **denoising score matching (DSM)** loss based on [Zaidi *et al.*, 2023] is:

$$L_2 = \mathbb{E}_{q_\sigma(\mathbf{x}',\mathbf{x})} \left[\frac{\partial \log f(\mathbf{s}')}{\partial \mathbf{x}'} - \frac{\mathbf{x} - \mathbf{x}'}{\sigma^2} \right] \tag{4}$$

where $q_\sigma(\mathbf{x}', \mathbf{x})$ is the joint distribution of perturbed and non-perturbed structures.

To optimize the performance on the prediction task, we integrate contrastive learning with noise-based regularization by combining two different constraints to formulate the overall objective function:

$$L = L_1 + \mu L_2 \tag{5}$$

Here, μ is a hyper-parameter adjusted to balance the contribution of contrastive learning with DSM-based regularization.

5 Experiments

Dataset description: For the pretraining phase, we used the proposed *DecoyDB* dataset and removed overlapping samples from fine-tuning binding affinity datasets. For other pretraining methods, Frad and ConBAP, we used the pretraining model provided by the authors. Specifically, Frad was pretrained on PCQM4Mv2 [Nakata and Shimazaki, 2017] and ConBAP was pretrained on Redocked2020 [Francoeur *et al.*, 2020]. For fine-tuning, we used the PDBbind2016 and PDBbind2013 datasets [Wang *et al.*, 2005], which contain 13,189 complexes and 10,370 complexes, respectively. Each sample is associated with a binding affinity. Note that we conduct pretraining only on GNN-based models. For other models, we train them directly on the PDBbind dataset without self-supervised pretraining. To ensure a fair comparison during fine-tuning, we followed the established setup by randomly selecting 11,904 training and 1,000 validation complexes for PDBbind2016, and 7,977 training and 1,000 validation complexes for PDBbind2013. For model testing, we used two independent benchmark test sets: the PDBbind2013 core set and PDBbind2016 core set, containing 107 and 285 complexes, respectively. These test sets are also removed from our pretraining dataset to conduct a rigorous evaluation of our pretraining framework.

Evaluation metrics: During pretraining, we used the early stopping strategy based on training and validation loss. During fine-tuning, we assessed the model's test performance using Root Mean Square Error (RMSE) and Pearson's correlation coefficient (R), following previous works [Stepniewska-Dziubinska *et al.*, 2018a; Zheng *et al.*, 2019].

Baselines: The baseline models include: Docking method (**AutoDock Vina** [Eberhardt *et al.*, 2021]), Drug-Target Affinity methods (**DeepDTA** [Lennox *et al.*, 2021], **GraphDTA** [Nguyen *et al.*, 2020]), CNN-based methods (**Pafnucy** [Stepniewska-Dziubinska *et al.*, 2018b], **OnionNet** [Zheng *et al.*, 2019]), GNN-based methods (**SchNet** [Schütt *et al.*, 2017], **EGNN** [Satorras *et al.*, 2021], **GIGN** [Yang *et al.*, 2023]), General GCL with edge perturbation on GIGN (**GIGN + GCL-EP**), GCL with node dropping on GIGN (**GIGN + GCL-ND**) and **pretraining for biochemistry graphs** (**Frad** [Ni *et al.*, 2024], **ConBAP** [Luo *et al.*, 2024]). We compare our customized GCL algorithm (named **OURS**) on DecoyDB against baselines. More details are provided in Appendix B.

Implementation details: For baseline models, we used the original source code provided by the authors. For the pretraining methods, we used the pretrained model parameters provided by the authors and fine-tuned them on our dataset, where ConBAP used EGNN as its base model, and Frad used TorchMD-Net [Thölke and De Fabritiis, 2022] as its base model. Our pretraining framework used these GNNs as the base encoders for complexes, with two separate dense layers during the pretraining and fine-tuning phases to make predictions. We used GIGN as the base model for the ablation study, sensitivity analysis, the impact of dataset size and the model generalization.

We used the same learning rate (5e-4) and weight decay (1e-6) for each model in two phases. During the pretraining phase, we trained each model for 20 epochs (Figure 4 (a)). In the fine-tuning phase, each model was trained for a maximum of 300 epochs, with early stopping applied if there was no improvement on the validation set within 40 epochs. The detailed setup can be found in our supplementary materials. All experiments were executed on an NVIDIA DGX-2 node with AMD EPYC 7742 64-core CPU and eight A100 GPUs. We ran each model ten times to obtain the average performance and standard deviations. Additional experimental details are provided in Appendix A.

5.1 Overall Performance

Table 2 summarizes the detailed comparison of our framework against baseline models on two datasets. Among the baselines, DTA-based methods show relatively high RMSE values, likely because they do not take into account the detailed spatial structures related to binding interactions. CNN-based models show moderate improvements, with reduced RMSE and increased R values, but their reliance on volume-based representations cannot capture the graph structural information. In contrast, GNN-based models, such as EGNN and GIGN, outperform other baselines, even without pretraining. Particularly, GIGN achieves the best accuracy among all base models. We also compared our pretraining framework with alternative contrastive learning methods (when their source codes were available) on several GNN base models. Results show that adding our pretraining framework significantly improves the base model performance. For instance, on the EGNN base model, our pretraining framework (OURS) reduces the RMSE from 1.304 to 1.250 on the PDBbind 2016, while the existing ConBAP method (which also uses decoys to augment negative samples) only reduces

Table 2: Performance comparison on PDBbind core set 2013 and PDBbind core set 2016.

	Method	Pretraining dataset	PDBbind core set 2013		PDBbind core set 2016	
			RMSE ↓	R ↑	RMSE ↓	R ↑
Docking	AutoDock Vina	-	2.400	0.570	2.350	0.600
DTA	DeepDTA	-	1.603 (0.014)	0.717 (0.016)	1.366 (0.011)	0.777 (0.013)
	GraphDTA	-	1.742 (0.039)	0.673 (0.032)	1.543 (0.033)	0.707 (0.021)
CNN	Pafnucy	-	1.544 (0.024)	0.778 (0.013)	1.423 (0.040)	0.793 (0.023)
	OnionNet	-	1.562 (0.071)	0.747 (0.031)	1.421 (0.069)	0.772 (0.024)
GNN +pretrain	TorchMD-Net	-	1.466 (0.034)	0.763 (0.016)	1.294 (0.026)	0.808 (0.011)
	+Frad	PCQM4Mv2	1.447 (0.016)	0.780 (0.008)	1.264 (0.043)	0.811 (0.006)
	SchNet	-	1.642 (0.030)	0.739 (0.016)	1.526 (0.037)	0.744 (0.014)
	+OURS	**DecoyDB**	**1.577 (0.034)**	**0.763 (0.015)**	**1.481 (0.031)**	**0.755 (0.012)**
	EGNN	-	1.496 (0.047)	0.761 (0.012)	1.334 (0.024)	0.801 (0.010)
	+ConBAP	Redocked 2020	1.479 (0.038)	0.766 (0.010)	1.300 (0.019)	0.802 (0.014)
	+OURS	**DecoyDB**	**1.437 (0.044)**	**0.781 (0.013)**	**1.267 (0.021)**	**0.813 (0.011)**
	GIGN	-	1.421 (0.038)	0.786 (0.016)	1.262 (0.032)	0.811 (0.010)
	+ GCL-EP	**DecoyDB**	1.417 (0.034)	0.789 (0.014)	1.251 (0.029)	0.814 (0.008)
	+ GCL-ND	**DecoyDB**	1.420 (0.026)	0.787 (0.011)	1.254 (0.026)	0.813 (0.013)
	+OURS	**DecoyDB**	**1.377 (0.039)**	**0.813 (0.013)**	**1.189 (0.031)**	**0.838 (0.011)**

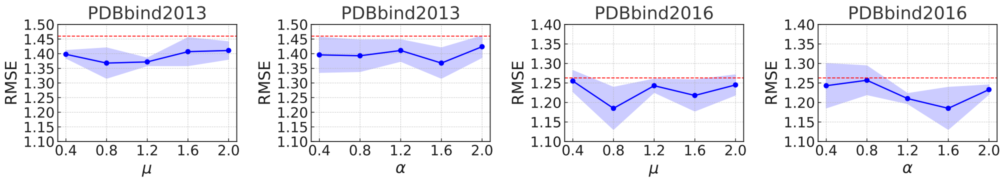
Table 3: P-values for pairwise comparisons between OURS and baselines on PDBbind 2013/2016. **Bold** denotes $p < 0.05$ (statistically significant).

Comparison	RMSE (2013)	R (2013)	RMSE (2016)	R (2016)
SchNet + OURS vs SchNet	**0.00056**	**0.0038**	**0.00015**	0.069
EGNN + OURS vs EGNN	**0.040**	**0.00069**	**0.00086**	**0.034**
EGNN + OURS vs EGNN + ConBAP	0.071	**0.00043**	**0.045**	**0.022**
GIGN + OURS vs GIGN	**0.015**	**0.0014**	**0.00018**	**0.000046**
GIGN + OURS vs GIGN + GCL-EP	**0.019**	**0.0046**	**0.00012**	**0.00065**
GIGN + OURS vs GIGN + GCL-ND	**0.011**	**0.0032**	**0.00014**	**0.00036**

the RMSE to 1.285. As another example, our pretraining method reduces the RMSE of GIGN (the best base model) from 1.460 to 1.386 on the PDBbind core set 2013 and from 1.263 to 1.188 on the PDBbind core set 2016. In contrast, general GCL does not show significant improvement on GIGN. For key comparisons, we also conducted paired t-tests on both RMSE and R. As summarized in Table 3, most improvements are statistically significant (p < 0.05). For the two borderline cases, we provide per-run results over ten runs in Appendix C.

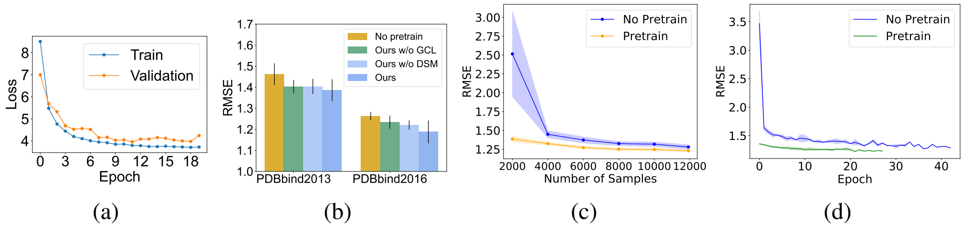
5.2 Sensitivity Analysis

In this section, we evaluate the impact of two key hyperparameters α and μ. α (Equation 2) influences the relative weight of the factor β (i.e., relative weight of decoy negative samples versus negative samples from different complexes), while μ balances the relative weight of contrastive learning and DSM. We varied both parameters ranging from 0.4 to 2.0 and observed variations in RMSE across the test datasets. As shown in Figure 3, RMSE initially decreases and then increases with rising α, but it is persistently lower than the RMSE of the baseline GIGN model (shown as the red dashed line). The results of μ follow a similar trend.



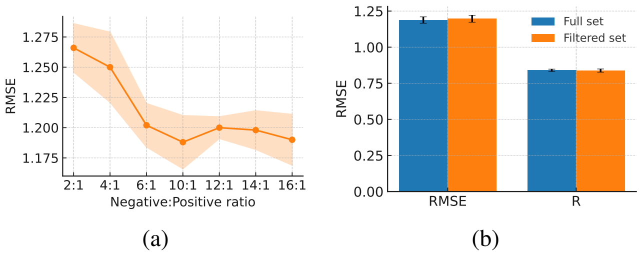
(a) (b) (c) (d)

Figure 3: Sensitivity analysis of two key hyper-parameters, α and μ, in our loss function, as shown in Equation 2 and Equation 5. (a) and (c) show the RMSE performance across different values of μ on the two datasets, PDBbind2013 and PDBbind2016. (b) and (d) show the RMSE variation with different values of α on the same two datasets. The red dashed lines show the baseline GIGN model.



(a) (b) (c) (d)

Figure 4: (a) is the training and validation loss curve during pretraining. (b) is the ablation study. (c) is the impact of fine-tuning dataset size on the binding affinity prediction. (d) is the validation curve in fine-tuning.



(a) (b)

Figure 5: (a) Negative to Positive ratio vs. RMSE; (b) Filtering out top 1% highest RMSD decoys.

5.3 Ablation Study

Table 4: Ablation study of our two-category graph contrastive loss.

Method	PDBbind core set 2013	PDBbind core set 2016
with one-category graph contrastive loss	1.406 (0.047)	1.246 (0.014)
with two-category graph contrastive loss	**1.394** (0.036)	**1.221** (0.022)

Table 5: Paired t-tests for ablations.

Ablation comparison	p-value
No pretraining vs. OURS	0.0280
OURS w/o DSM vs. OURS	0.0013
OURS w/o GCL vs. OURS	0.0325

Our ablation study evaluates the impact of two key modules, i.e., the graph contrastive learning (GCL) and denoising score matching (DSM)-based regularization in the pretraining phase. We used GIGN as the base model, starting with a full framework that included both components and then removing each component to observe the impact on model performance. As shown in Figure 4(d), removing either GCL or DSM led to a modest increase in RMSE (although their RMSE levels are still lower than the RMSE of no pretraining), indicating that both of them play a crucial role in capturing meaningful representations and improving performance. We also added an ablation study of our two-category graph contrastive loss. We pretrained two models, one model with two-category contrastive loss (both decoy negative samples and different complexes as negative samples, denoted as “with two-category graph contrastive loss”) and the other with one-category GCL loss (only different complexes as negative samples, denoted as “without decoy samples”). Results in Table 4 show that the proposed two-category graph contrastive loss slightly outperforms a one-category contrastive loss on both datasets. To analyze each component’s contribution, we run 10 times per variant and conduct paired two-sided t-tests on RMSE. As summarized in Table 5, all comparisons are statistically significant ($p < 0.05$), indicating that pretraining, DSM, and GCL contribute non-trivially to overall performance.

5.4 Varying Label Sizes in Fine-Tuning

We also assessed the effect of our pretraining on sample efficiency and learning efficiency during fine-tuning, especially when the amount of labels is limited. We changed the number of labeled samples from 2000 to 12000 during fine-tuning. As shown in Figure 4(c), the pretrained model consistently achieves lower RMSEs across all label sizes, with the most notable improvements observed with fewer labeled samples (lower mean RMSE and smaller variance). Moreover, pretraining appears to accelerate the convergence during the fine-tuning phase. To confirm this, we used an early stopping mechanism with a patience of 10 epochs. Figure 4(d) shows that the pretrained model not only starts with a lower initial RMSE but also reaches convergence more quickly (around 20 epochs) compared to the base model without our pretraining (over 40 epochs). This confirms that our pretraining enhances the label sample efficiency during fine-tuning.

5.5 Model Generalizability

It has been shown that the default split of general (training), refined (validation), and core (test) datasets in PDBbind is cross-contaminated with proteins and ligands with high similarity. To rigorously evaluate the effect of our pretraining on a model's generalizability, we used a leakage-proof split called LP-PDBbind [Li *et al.*, 2024a], which partitions PDBbind into training (10980 samples), validation (2312 samples), and test (4651 samples) sets based on high sequence and structural similarity across proteins and ligands. As a comparison, we also randomly split the data into training, validation, and test sets with the same sizes. The experimental results in Table 6 on the GIGN base model show a much more significant improvement after pretraining on the LP-PDBbind compared with a random split. This indicates that our pretraining framework enhances the generalizability of a base model across different protein and ligand structures.

Table 6: Model generalizability test on a leakage-proof split (LP-PDBbind)

	No pretrain	Pretrain
LP-PDBbind split	1.496 (0.029)	1.371 (0.006)
Random split	1.294 (0.017)	1.269 (0.015)

5.6 Impact of Decoy Dataset Size and Quality

We assessed the impact of pretraining dataset (DecoyDB) configuration over model performance through: (1) varying the number of decoys by applying different ratios of negative pairs to positive pairs (negative-to-positive ratio); and (2) by filtering out the top 1% highest-RMSD decoys (outliers). We used the GIGN backbone for pretraining and PDBBind Core 2016 for fine-tuning. The results are from ten runs. As shown in Figure 5(a), performance improves by increasing the number of decoys and peaks at 10:1, after which the gain slightly fluctuates. We used 10:1 as the default in our main experiments. Filtering out extremely high-RMSD decoys makes little difference (Figure 5(b)).

6 Conclusion and Future Works

In this paper, we propose *DecoyDB*, a comprehensive dataset of high-resolution 3D complexes augmented with decoys for pretraining graph neural networks in protein-ligand binding affinity prediction. We also design a customized GCL algorithm based on DecoyDB. Experiments show that pretraining on DecoyDB improves multiple base models in prediction accuracy, sample learning efficiency, and model generalizability.

In the future, we plan to extend the proposed framework to other tasks beyond binding affinity prediction, such as binding pose estimation. Another interesting direction is to systematically study how the choice of decoy generation tools and their parameterizations influences model performance.

Acknowledgments

This material is based upon work supported by the National Science Foundation (NSF) under Grant No. IIS-2147908, IIS-2207072, OAC-2152085, OAC-2402946, and OAC-2410884, the Bodor Professorship, as well as NIH R01CA212403 and R21EB037868.

References

Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The protein data bank. *Nucleic Acids Research*, 28(1):235–242, 01 2000.

Martin Buttenschoen, Garrett M Morris, and Charlotte M Deane. Posebusters: Ai-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical Science*, 15(9):3130–3139, 2024.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

Jason C Cole, Christopher W Murray, J Willem M Nissink, Richard D Taylor, and Robin Taylor. Comparing protein–ligand docking programs is difficult. *Proteins: Structure, Function, and Bioinformatics*, 60(3):325–332, 2005.

Mindy I Davis, Jeremy P Hunt, Sanna Herrgard, Pietro Ciceri, Lisa M Wodicka, Gabriel Pallares, Michael Hocker, Daniel K Treiber, and Patrick P Zarrinkar. Comprehensive analysis of kinase inhibitor selectivity. *Nature biotechnology*, 29(11):1046–1051, 2011.

Jerome Eberhardt, Diogo Santos-Martins, Andreas F Tillack, and Stefano Forli. Autodock vina 1.2. 0: New docking methods, expanded force field, and python bindings. *Journal of chemical information and modeling*, 61(8):3891–3898, 2021.

Yin Fang, Qiang Zhang, Ningyu Zhang, Zhuo Chen, Xiang Zhuang, Xin Shao, Xiaohui Fan, and Huajun Chen. Knowledge graph-enhanced molecular contrastive learning with functional prompt. *Nature Machine Intelligence*, 5(5):542–553, 2023.

Evan N Feinberg, Debnil Sur, Zhenqin Wu, Brooke E Husic, Huanghao Mai, Yang Li, Saisai Sun, Jianyi Yang, Bharath Ramsundar, and Vijay S Pande. Potentialnet for molecular property prediction. *ACS central science*, 4(11):1520–1530, 2018.

Paul G Francoeur, Tomohide Masuda, Jocelyn Sunseri, Andrew Jia, Richard B Iovanisci, Ian Snyder, and David R Koes. Three-dimensional convolutional neural networks and a cross-docked data set for structure-based drug design. *Journal of chemical information and modeling*, 60(9):4200–4215, 2020.

Samuel Genheden and Ulf Ryde. The mm/pbsa and mm/gbsa methods to estimate ligand-binding affinities. *Expert opinion on drug discovery*, 10(5):449–461, 2015.

Xiaoyu Guan and Daoqiang Zhang. T-mgcl: Molecule graph contrastive learning based on transformer for molecular property prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2023.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

Dejun Jiang, Chang-Yu Hsieh, Zhenxing Wu, Yu Kang, Jike Wang, Ercheng Wang, Ben Liao, Chao Shen, Lei Xu, Jian Wu, Dongsheng Cao, and Tingjun Hou. Interactiongraphnet: A novel and efficient deep graph representation learning framework for accurate protein–ligand interaction predictions. *Journal of Medicinal Chemistry*, 64(24):18209–18232, 2021. PMID: 34878785.

Wengong Jin, Siranush Sarkizova, Xun Chen, Nir Hacohen, and Caroline Uhler. Unsupervised protein-ligand binding energy prediction via neural euler's rotation equation. *Advances in Neural Information Processing Systems*, 36, 2024.

John G Kirkwood. Statistical mechanics of fluid mixtures. *The Journal of chemical physics*, 3(5):300–313, 1935.

Douglas B Kitchen, Hélène Decornez, John R Furr, and Jürgen Bajorath. Docking and scoring in virtual screening for drug discovery: methods and applications. *Nature reviews Drug discovery*, 3(11):935–949, 2004.

Huan Yee Koh, Anh TN Nguyen, Shirui Pan, Lauren T May, and Geoffrey I Webb. Physicochemical graph neural network for learning protein–ligand interaction fingerprints from sequence data. *Nature Machine Intelligence*, 6(6):673–687, 2024.

Mark Lennox, Neil Robertson, and Barry Devereux. Modelling drug-target binding affinity using a bert based graph neural network. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference*, page 4348—4353, November 2021.

Y. Li, M. A. Rezaei, C. Li, and X. Li. Deepatom: A framework for protein-ligand binding affinity prediction. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 303–310, nov 2019.

Hongjian Li, Kam-Heung Sze, Gang Lu, and Pedro J Ballester. Machine-learning scoring functions for structure-based drug lead optimization. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 10(5):e1465, 2020.

Shuangli Li, Jingbo Zhou, Tong Xu, Dejing Dou, and Hui Xiong. Geomgcl: Geometric graph contrastive learning for molecular property prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 4541–4549, 2022.

Yanjun Li, Daohong Zhou, Guangrong Zheng, Xiaolin Li, Dapeng Wu, and Yaxia Yuan. Dyscore: A boosting scoring method with dynamic properties for identifying true binders and nonbinders in structure-based drug discovery. *Journal of chemical information and modeling*, 62(22):5550–5567, 2022.

Jie Li, Xingyi Guan, Oufan Zhang, Kunyang Sun, Yingze Wang, Dorian Bagni, and Teresa Head-Gordon. Leak proof pdbbind: a reorganized dataset of protein-ligand complexes for more generalizable binding affinity prediction. *ArXiv*, pages arXiv–2308, 2024.

Xuelian Li, Cheng Shen, Hui Zhu, Yujian Yang, Qing Wang, Jincai Yang, and Niu Huang. A high-quality data set of protein–ligand binding interactions via comparative complex structure modeling. *Journal of Chemical Information and Modeling*, 64(7):2454–2466, 2024.

Jaechang Lim, Seongok Ryu, Kyubyong Park, Yo Joong Choe, Jiyeon Ham, and Woo Youn Kim. Predicting drug–target interaction using a novel graph neural network with 3d structure-embedded graph representation. *Journal of Chemical Information and Modeling*, 59(9):3981–3988, 2019. PMID: 31443612.

Tiqing Liu, Yuhmei Lin, Xin Wen, Robert N Jorissen, and Michael K Gilson. Bindingdb: a web-accessible database of experimentally determined protein–ligand binding affinities. *Nucleic acids research*, 35(suppl_1):D198–D201, 2007.

Maotao Liu, Yifan Yang, Xu Gong, Li Liu, and Qun Liu. Hiermrl: Hierarchical structure-aware molecular representation learning for property prediction. In *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 386–389. IEEE, 2022.

Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. Graphprompt: Unifying pre-training and downstream tasks for graph neural networks. In *Proceedings of the ACM web conference 2023*, pages 417–428, 2023.

Ding Luo, Dandan Liu, Xiaoyang Qu, Lina Dong, and Binju Wang. Enhancing generalizability in protein–ligand binding affinity prediction with multimodal contrastive learning. *Journal of Chemical Information and Modeling*, 64(6):1892–1906, 2024.

Gabriele Macari, Daniele Toti, Andrea Pasquadibisceglie, and Fabio Polticelli. Dockingapp rf: a state-of-the-art novel scoring function for molecular docking in a user-friendly interface to autodock vina. *International Journal of Molecular Sciences*, 21(24):9548, 2020.

Maho Nakata and Tomomi Shimazaki. Pubchemqc project: a large-scale first-principles electronic structure database for data-driven chemistry. *Journal of chemical information and modeling*, 57(6):1300–1308, 2017.

Thin Nguyen, Hang Le, Thomas P Quinn, Tri Nguyen, Thuc Duy Le, and Svetha Venkatesh. GraphDTA: predicting drug–target binding affinity with graph neural networks. *Bioinformatics*, 37(8):1140–1147, 10 2020.

Yuyan Ni, Shikun Feng, Xin Hong, Yuancheng Sun, Wei-Ying Ma, Zhi-Ming Ma, Qiwei Ye, and Yanyan Lan. Pre-training with fractional denoising to enhance molecular property prediction. *Nature Machine Intelligence*, 6(10):1169–1178, 2024.

Jiayu Qin, Jian Chen, Rohan Sharma, Jingchen Sun, and Changyou Chen. A probability contrastive learning framework for 3d molecular representation learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Mohammad A Rezaei, Yanjun Li, Dapeng Wu, Xiaolin Li, and Chenglong Li. Deep learning in drug design: Protein-ligand binding affinity prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 19(01):407–417, 2022.

Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Till Siebenmorgen, Filipe Menezes, Sabrina Benassou, Erinc Merdivan, Kieran Didi, André Santos Dias Mourão, Radosław Kitel, Pietro Liò, Stefan Kesselheim, Marie Piraud, et al. Misato: machine learning dataset of protein–ligand complexes for structure-based drug discovery. *Nature Computational Science*, pages 1–12, 2024.

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

Marta M Stepniewska-Dziubinska, Piotr Zielenkiewicz, and Pawel Siedlecki. Development and evaluation of a deep learning model for protein–ligand binding affinity prediction. *Bioinformatics*, 34(21):3666–3674, 05 2018.

Mengying Sun, Jing Xing, Huijun Wang, Bin Chen, and Jiayu Zhou. Mocl: data-driven molecular fingerprint via knowledge-aware contrastive learning from molecular graph. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 3585–3594, 2021.

Jing Tang, Agnieszka Szwajda, Sushil Shakyawar, Tao Xu, Petteri Hintsanen, Krister Wennerberg, and Tero Aittokallio. Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis. *Journal of chemical information and modeling*, 54(3):735–743, 2014.

Philipp Thölke and Gianni De Fabritiis. Equivariant transformers for neural network based molecular potentials. In *International Conference on Learning Representations*, 2022.

Zekun Tong, Yuxuan Liang, Henghui Ding, Yongxing Dai, Xinke Li, and Changhu Wang. Directed graph contrastive learning. *Advances in neural information processing systems*, 34:19580–19593, 2021.

Swapnil Wagle, Richard D Smith, Anthony J Dominic III, Debarati DasGupta, Sunil Kumar Tripathi, and Heather A Carlson. Sunsetting binding moad with its last data update and the addition of 3d-ligand polypharmacology tools. *Scientific Reports*, 13(1):3008, 2023.

Xiao Wang and Guo-Jun Qi. Contrastive learning with stronger augmentations. *IEEE transactions on pattern analysis and machine intelligence*, 45(5):5549–5560, 2022.

Renxiao Wang, Xueliang Fang, Yipin Lu, Chao-Yie Yang, and Shaomeng Wang. The pdbbind database: methodologies and updates. *Journal of medicinal chemistry*, 48(12):4111–4119, 2005.

Yingze Wang, Kunyang Sun, Jie Li, Xingyi Guan, Oufan Zhang, Dorian Bagni, Yang Zhang, Heather A Carlson, and Teresa Head-Gordon. A workflow to create a high-quality protein–ligand binding dataset for training, validation, and prediction tasks. *Digital Discovery*, 2025.

Fang Wu, Shuting Jin, Yinghui Jiang, Xurui Jin, Bowen Tang, Zhangming Niu, Xiangrong Liu, Qiang Zhang, Xiangxiang Zeng, and Stan Z Li. Pre-training of equivariant graph matching networks with conformation flexibility for drug binding. *Advanced Science*, 9(33):2203796, 2022.

Dongkuan Xu, Wei Cheng, Dongsheng Luo, Haifeng Chen, and Xiang Zhang. Infogcl: Information-aware graph contrastive learning. *Advances in Neural Information Processing Systems*, 34:30414–30425, 2021.

Jianyi Yang, Ambrish Roy, and Yang Zhang. Biolip: a semi-manually curated database for biologically relevant ligand–protein interactions. *Nucleic acids research*, 41(D1):D1096–D1103, 2012.

Ziduo Yang, Weihe Zhong, Qiujie Lv, Tiejun Dong, and Calvin Yu-Chian Chen. Geometric interaction graph neural network for predicting protein–ligand binding affinities from 3d structures (gign). *The journal of physical chemistry letters*, 14(8):2020–2033, 2023.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33:5812–5823, 2020.

Sheheryar Zaidi, Michael Schaarschmidt, James Martens, Hyunjik Kim, Yee Whye Teh, Alvaro Sanchez-Gonzalez, Peter Battaglia, Razvan Pascanu, and Jonathan Godwin. Pre-training via denoising for molecular property prediction. In *The Eleventh International Conference on Learning Representations*, 2023.

Yunjiang Zhang, Shuyuan Li, Kong Meng, and Shaorui Sun. Machine learning for sequence and structure-based protein–ligand interaction prediction. *Journal of Chemical Information and Modeling*, 64(5):1456–1472, 2024.

Liangzhen Zheng, Jingrong Fan, and Yuguang Mu. Onionnet: a multiple-layer intermolecular-contact-based convolutional neural network for protein–ligand binding affinity prediction. *ACS omega*, 4(14):15956–15965, 2019.

Robert W Zwanzig. High-temperature equation of state by a perturbation method. i. nonpolar gases. *The Journal of Chemical Physics*, 22(8):1420–1426, 1954.

A Experiment Setup

A.1 Evaluation metrics

In this section, we provide the details about our two evaluation metrics in our experiment. Root Mean Square Error (RMSE) and Pearson's correlation coefficient (R) are denoted as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{6}$$

$$R = \frac{\sum_{i=1}^{N}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2 \sum_{i=1}^{N}(\hat{y}_i - \bar{\hat{y}})^2}} \tag{7}$$

y_i and $\hat{y}_i$ represents experimental and predicted binding affinity, respectively.

A.2 Data Preprocessing

The edges E_k between atoms can be determined by the chemical bonds (covalent bonds) or a distance threshold. In our case, we used a distance threshold (5Å) to establish edges within the protein and the interactive edges between protein atoms and ligand atoms. We used the same distance threshold to filter out relevant atoms within protein binding pockets for the graphs. We kept the covalent bonds within a ligand as ligand edges.

A.3 Training Parameters

In our experiments, we trained and evaluated the DeepDTA, GraphDTA, Pafnucy, OnionNet, SchNet, EGNN, TorchMD-Net and GIGN models. The Adam optimizer was employed for all models with the following settings: the initial learning rate was set to 5×10^{-4}, weight decay was set to 1×10^{-6}. To prevent overfitting and enhance the generalization ability of the models, we reduced the learning rate by a factor of 0.1 whenever the performance on the validation set did not improve for 10 consecutive epochs. Additionally, an early stopping strategy was adopted for all models, where training was terminated if the validation performance did not improve within 40 consecutive epochs. The pretraining phase for each model was set to 20 epochs, while the maximum number of epochs for the fine-tuning phase was set to 300. In the fine-tuning phase, we set the batch size to 128. In the pretraining phase, we set the batch size to 8. Each sample in the batch contained 21 structures: 1 real data sample, 10 randomly selected decoys, and 10 perturbed data samples generated by adding Gaussian noise. In addition to performance improvements, we also examined the time cost of our framework. Using GIGN as an example, the pretraining stage took approximately 23 hours, while the fine-tuning stage required only about 30 minutes.

A.4 Model Architecture Parameters

In our experiments, we used the following parameter settings for the baseline models to ensure optimal performance. For Pafnucy, we set the channels of the three-layer 3D convolutions to 64, 128, and 256, respectively. For OnionNet, the input features were set to 3840, and there are 3 convolutional layers with 32, 64, and 128 filters. The kernel size was set to 4. The maximum length of protein sequences in DTA method was set to 1000. SchNet was configured with 3 interaction layers, each having an embedding dimension of 128. Both EGNN and GIGN models utilized a three-layer architecture with each layer having a dimension of 256. For TorchNet-MD, there are 6 layers and each layer has 256-dimensional embeddings.

B Baseline models

- **Docking Method**: **AutoDock Vina** [Eberhardt *et al.*, 2021] is a widely used program for molecular docking, which can also predict binding affinity. Since it is not a deep learning model, we directly cite the results from [Macari *et al.*, 2020]. Note that there are no repeated experiments, so there is no standard error being reported.

- **Drug-Target Affinity (DTA) methods: DeepDTA** [Lennox *et al.*, 2021] predict binding affinity based on protein and ligand sequences separately without interaction networks. **GraphDTA** [Nguyen *et al.*, 2020] is similar to DeepDTA but uses a GNN.
- **CNN-based methods: Pafnucy** [Stepniewska-Dziubinska *et al.*, 2018b] is a 3D-CNN model. **OnionNet** [Zheng *et al.*, 2019] uses 2D-CNN to learn representations.
- **GNN-based methods: TorchMD-Net** [Thölke and De Fabritiis, 2022] is a GNN-based model specially designed for the force field. **SchNet** [Schütt *et al.*, 2017] is a GNN model based on essential quantum chemical constraints. **EGNN** [Satorras *et al.*, 2021] is a GNN based on rotation and translation equivariance. **GIGN** [Yang *et al.*, 2023] is a GNN specifically designed for protein-ligand interactions. This is among the state of the art methods.
- **pretraining for general graphs:** General **GCL** based on edge perturbation and node dropping. We only tested these methods on top of the best baseline GNN model GIGN, including GCL with edge perturbation (GIGN+GCL-EP) or node dropping (GIGN+GCL-ND).
- **pretraining for biochemistry graphs:** (1) **Frad** [Ni *et al.*, 2024] designed a chemical-guided noise and utilizes denoising for pretraining. We only evaluated Frad on top of TorchMD-Net due to the availability of source codes. (2) **ConBAP** [Luo *et al.*, 2024] designs a contrastive learning strategy based on (binary) negative samples from decoys. We only evaluated ConBAP on EGNN due to the availability of source codes.
- **OURS**: This is our proposed graph contrastive learning framework. We evaluated our framework on top of GIGN, EGNN, and SchNet.

C Per-run Results

Table 7: R (2016) per-run results for SchNet vs SchNet + OURS.

Run Index	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
SchNet	0.742	0.741	0.717	0.762	0.761	0.747	0.755	0.740	0.747	0.728
SchNet + OURS	0.758	0.748	0.764	0.748	0.771	0.753	0.763	0.747	0.763	0.730

Table 8: RMSE (2013) per-run results for EGNN + OURS vs EGNN + ConBAP.

Run Index	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
EGNN + ConBAP	1.478	1.539	1.503	1.494	1.398	1.502	1.459	1.485	1.499	1.436
EGNN + OURS	1.473	1.436	1.460	1.399	1.502	1.484	1.400	1.354	1.459	1.406

D Pseudocode

Our framework is in Algorithm 1 and Algorithm 2. For simplicity, we omit the loop over epochs.

Algorithm 1 Graph Contrastive Pretraining

```
Input: Protein-ligand complexes $\mathcal{S}$, a GNN encoder $f_\theta$, DecoyDB
Output: Pretrained parameters $\theta$ of encoder $f_\theta$
for each unlabeled protein-ligand complex $s \in \mathcal{S}$ do
    With $s$ as an anchor, sample positive and negative pairs from DecoyDB
    Compute latent representation $\mathbf{z} = f_\theta(s)$ for them
    Compute contrastive loss $L_1$ (Equation 3)
    Sample Gaussian noise $\epsilon$
    Apply noise to ligand coordinates: $\mathbf{x}' = \mathbf{x} + \epsilon$
    Compute denoising loss $L_2$ (Equation 4)
    Compute total loss: $\mathcal{L}$ (Equation 5)
    Update encoder parameters: $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}$
end for
```

Algorithm 2 Fine-tuning

```
Input: Pretrained encoder $f_\theta$, labeled protein-ligand complexes $S_l$ and a new regression head $g_\phi$ s.t. $\hat{y} = g_\phi(f_\theta(s))$
Output: Fine-tuned parameters $\theta$ and $\phi$
Replace regression head $g_\phi$
for each labeled protein-ligand complex $s \in S_l$ do
    Predict binding affinity: $\hat{y} = g_\phi(f_\theta(s))$
    Compute loss: $\mathcal{L}_{\text{finetune}} = (\hat{y} - y)^2$
    Update parameters: $[\theta, \phi] \leftarrow [\theta, \phi] - \eta \nabla_{\theta,\phi} \mathcal{L}_{\text{finetune}}$
end for
```

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims are made in the abstract and introduction accurately.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [No]

   Justification: This paper focuses on the construction of a large-scale dataset and its effectiveness in model pretraining. As our primary contribution lies in data design and evaluation of pretraining performance, we did not include an explicit discussion of the dataset's limitations.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not present theoretical results requiring formal assumptions or proofs. The focus of this work is on dataset construction and empirical evaluation through experiments.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: This paper fully discloses all the necessary details required to reproduce the main experimental results that support its claims and conclusions.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We have released datasets and code with detailed documentation and instructions.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The full details can be found in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports error bars based on standard deviations over repeated trials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide information about the computational resources used in the Experiments section, with more detailed descriptions available in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research fully complies with the NeurIPS Code of Ethics. We ensured transparency, reproducibility, and responsible handling of data throughout the study.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses potential positive impacts, such as accelerating drug discovery in the introduction.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The data used in this work, experimentally resolved protein-ligand complex structures, do not pose a high risk of misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite all the origin papers of the code and data.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: All the data and codes are well documented. They can be found in our huggingface dataset and our github repo.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This research does not involve human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This research does not involve human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: This paper focus on the dataset construction and does not involve LLM.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.