COPOCKET: POCKET-CONDITIONED MOLECULAR GENERATION VIA MULTI-MODAL FEATURE CONSTRAINTS AND INTERACTION-AWARE CONTRASTIVE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Targeted molecular generation aims to generate molecules against protein pockets, which is a challenging task in drug discovery. Most existing methods neglect the multiscale hierarchical representation of protein pockets, and fail to sufficiently exploit critical interaction within protein-ligand binding. This paper proposes a drug design framework, namely CoPocket, focusing on different modal features of binding pockets to guide high quality molecule generation. First, we construct an atomic-level spatial interaction graph to capture the binding patterns between protein pockets and ligands. Guided by these patterns, we employ a cross-modal contrastive learning module to optimize the representation of pocket sequences. We also capture higher-order structural relations and atomic-level structural features through hypergraphs and atomic graphs of protein pockets, respectively. A cross-modal hierarchical fusion module fuses multi-modal embeddings from pocket sequences, pocket hypergraph structure, and pocket atomic graph structure. In addition, a pocket-conditioned molecule generator employs fused features as constraints for generating molecules possessing target characteristics. Experiments demonstrate that CoPocket outperforms existing methods across multiple key metrics, validating its effectiveness for molecular generation tasks.

1 INTRODUCTION

Targeted molecular generation plays a crucial role in modern drug development, with the goal of generating small-molecule ligands that bind with high affinity to specific protein pockets (Tang et al., 2024b; Du et al., 2024). In recent years, with the rapid development of artificial intelligence (Gangwal & Lavecchia, 2024), deep learning-driven generative methods have emerged as a research focus in the field of drug design (Shi et al., 2025; Garg, 2024; Tang et al., 2024a). These methods learn potential distribution patterns from known molecular and target characteristics (Özçelik et al., 2024), enabling the efficient generation of new structures with specific biological activities, thereby shortening the drug development cycle and reducing cost. According to different design strategies, current methods can be broadly divided into two categories: ligand-based methods and structure-based methods (Saini et al., 2025; Xu et al., 2025).

Ligand-based methods use existing active molecules as a basis, optimizing and modifying molecules through deep learning models to improve molecular properties (Peng et al., 2025; Ai et al., 2024; Ye, 2024). Iwata et al. (Iwata et al., 2023) employed a variogram autoencoder to learn the properties of known compound, and subsequently used Monte Carlo tree search to generate molecules. Mao et al. (Mao et al., 2024) proposed a unique IUPAC-based method to convert organic molecules into antiviral drug candidates. Xie et al. (Xie et al., 2025) integrated ligand pharmacophore fingerprinting with the GPT framework to generate bioactive ligands. These methods have advantages in terms of rationality, but they are highly dependent on the quantity and quality of known bioactive molecules, and cannot capture the binding characteristics of ligands and targets, which may lead to the omission of potential bioactive molecules.

Structure-based methods utilize structural information about target proteins as conditions to guide the generation of molecules (Qu et al., 2024; Kazi Md et al., 2025). With the rise of structure

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

prediction tools such as AlphaFold (Abramson et al., 2024), more and more methods incorporated three-dimensional structural information to generate molecules (Xiang et al., 2025; Wang et al., 2025b). Morehead et al. (Morehead & Cheng, 2024) employed a geometric diffusion framework and equivariant graph neural networks to produce molecules. Wang et al. (Wang et al., 2025a) proposed a method to optimize the structure of molecules using a discriminant module based on the chemical rule.

Zhang et al. (Zhang et al., 2023b) used a 3D graph neural network to encode protein pockets and generate fragments, and attach these fragments to the generated molecule. Zhang et al. (Zhang & Liu, 2023) further introduced the subpocket prototype learning by constructing a global interaction graph. Zhang et al. (Zhang et al., 2024) improved molecular quality and synthesizability through deep geometric processing. Zheng et al. (Zheng et al., 2025) employed equivariant 3D-Conditional diffusion models to generate ligands by a fragment-based method. Li (Li et al., 2024) produced target-specific ligand by generating synthesized molecular fragments and assembling them into complete molecules. These methods can accurately match binding sites to improve the binding affinity and expand the design space. However, these methods typically rely on high-quality structural data and employ complex modeling mechanisms, resulting in high computational resource consumption and training costs. In addition, many targets lack structural data, making them difficult to analyze and limiting their utility in drug discovery.

Although existing methods have achieved significant progress, they still face some challenges: Since ligand-based methods rely solely on ligand data without explicit target information, the generated molecules may lack precise binding specificity. Structure-based methods highly rely on high-quality target structure data, and require significant computational resources. In addition, most existing methods neglect the hierarchical features within protein pockets, and fail to sufficiently exploit the pocket-ligand interaction information, thereby limiting the expression of the pocket conformations and the potential interactions of the pocket-ligand.

To address the above issues, we proposed a target-binding pocket drug design framework (CoPocket) to generate molecules for the specific target. First, we construct an atomic-level interaction graph of pocket-ligand to capture the interaction patterns. Then, we designed a cross-modal contrastive learning module to align pocket sequence features with the interaction patterns. This framework also fuses the multi-modal features, including the pocket multi-scale structure features, the pocket sequence features, and the molecule sequence features. A cross-modal hierarchical fusion module fuses the hypergraph structure and the atom graph structure of pockets to obtain fused features, which contain global and local structural information. In addition, the fused features of pockets and molecules provide multidimensional information for pocket-conditioned molecule generator to enhance the quality of generated molecules. The main contributions are summarized as follows:

- We proposed a target-binding pocket drug design framework based on multi-modal feature constraints and cross-modal contrastive learning. The proposed framework can capture the multidimensional information of pockets to enhance the efficacy of targeted drug design.
- We constructed a cross-modal contrastive learning module to align pocket sequence features with atomic-level pocket-ligand interaction graph representations, thereby refining pocket sequence representations. We also fused cross-modal features from sequences, hypergraphs, and atom graphs of pockets. The fused features with global and local information, as constraints to guide molecule generation towards drug-like chemical space.
- Experiments show that CoPocket achieves superior performance in terms of Vina Score, SA, Diversity, demonstrating the feasibility of structure-based targeted drug design, offering a promising approach to generate molecules with high binding affinity.

2 METHOD

Figure 1 shows the overview of the CoPocket. First, for a given pocket-ligand, an interaction graph is represented, as shown in Figure 1 (a). The sequence representation of the pocket is shown in Figure 1 (b). Figure 1 (c) illustrates the process of cross-modal contrastive learning. Pocket structures are represented by hypergraphs and atom graphs, respectively, as shown in Figure 1 (d). The hypergraph structure, atom graph structure and sequence features of pockets are fused by a Cross-Modal

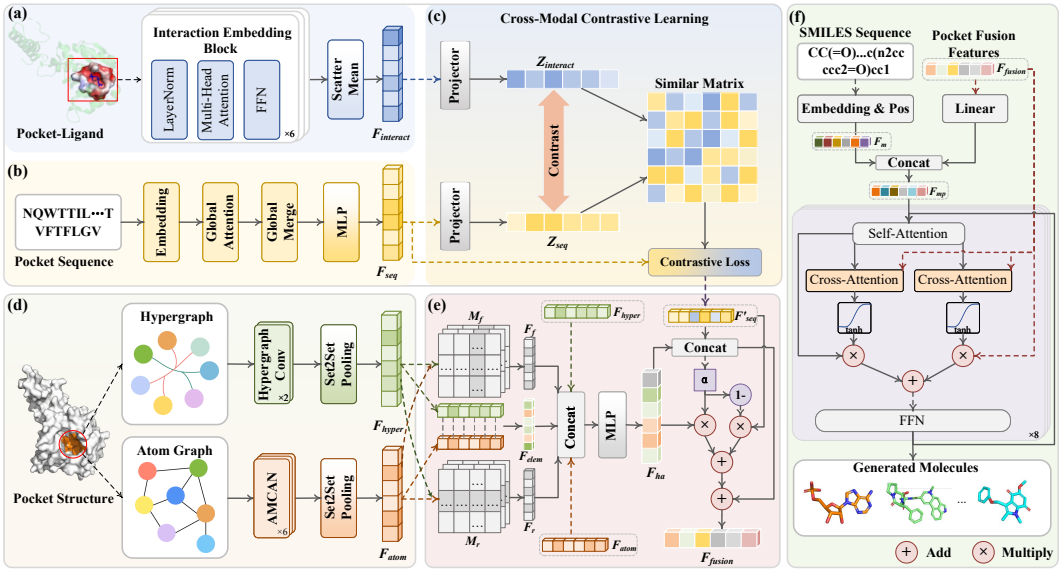

Figure 1: Framework of the proposed CoPocket. (a) Interaction Graph Representation of Pocket-Ligand. (b) Representation of Pocket Sequence. (c) Cross-Modal Contrastive Learning (CMCL). (d) Hypergraph and Atom Graph Representations of Pocket Structure. (e) Cross-Modal Feature Hierarchical Fusion. (f) Pocket-Conditioned Molecule Generator (PCMG).

Hierarchical Fusion module, as shown in Figure 1 (e). Finally, the pocket-conditional molecule generator guides the generation of molecules with the fusion features, as shown in Figure 1 (f).

2.1 Contrastive Interaction-Aware Representation of Pocket Sequence

To integrate pocket-ligand interaction information into pocket sequence representations, we propose a Contrastive Interaction-Aware Representation of Pocket Sequence module to guide sequence features toward alignment in the interaction semantic space, thereby refining sequence representations with interaction information.

2.1.1 Interaction Graph Representation of Pocket-Ligand

The pocket and ligand molecule are denoted as the set of atoms: $\mathcal{A}^{poc} = \{(a_i^{poc}, r_i^{poc})\}$, $\mathcal{A}^{lig} = \{(a_j^{lig}, r_j^{lig})\}$, respectively. Here a_i^{poc} and a_j^{lig} denote the atoms with features. $r_i^{poc} \in \mathbb{R}^3$ and $r_j^{lig} \in \mathbb{R}^3$ are their 3D coordinates. $\mathcal{A}^{poc}$ and $\mathcal{A}^{lig}$ are merged to construct the atom interaction graph $\mathcal{G}^{interact} = (\mathcal{V}^{interact}, \mathcal{E}^{interact})$, where $\mathcal{V}^{interact}$ denotes all atom nodes of a pocket-ligand, and $\mathcal{E}^{interact}$ denotes a set of spatial edges. The initial atom node features are linearly transformed to obtain the low-dimensional embedding $hg_i^{(0)}$, which is then iteratively updated by 6-layer Interaction Embedding Blocks (IEB) to obtain a final representation hg_i. More details of Interaction Embedding Block (IEB) can be found in Appendix A.1. The feature vectors of n atoms are stacked row-wise to form the atomic feature matrix: $H_{interact} = \left[g_1^\top; g_2^\top; \ldots; g_n^\top\right]^\top \in \mathbb{R}^{|\mathcal{V}^{interact}| \times d_s}$, which passes through Scatter Mean to obtain the final interaction graph representation $F_{interact} \in \mathbb{R}^{B \times d_s}$, where B is batch size, and d_s=256.

2.1.2 Representation of Pocket Sequence

Protein sequence features help compensate for the lack of structural information and provide biological semantic support at the sequence level. This module employs a pre-trained protein language model ProteinBERT (Brandes et al., 2022) to encode the amino acid sequence of the pocket for capturing the global semantic features of this pocket. After the sequence is input into the model,

a global embedding vector is extracted. Subsequently, to optimize the extracted sequence features, we project the sequence embedding features extracted from the pocket sequence P_{seq} into a unified feature space using a learnable MLP projection layer, obtaining pocket sequence-level embedding features:

$$F_{seq} = \mathrm{MLP}_{seq}(\mathrm{ProteinBert}(P_{seq})) \tag{1}$$

where $F_{seq} \in \mathbb{R}^{B\times d_s}$, B is batch size and d_s=256.

2.1.3 Cross-Modal Contrastive Learning (CMCL)

To enhance the expressive power of protein sequence features, we introduce a contrastive learning mechanism. Protein sequence features F_{seq} and atom-level interaction features $F_{interact}$ of pocket-ligand are projected into representations Z_{seq} and $Z_{interact}$, respectively. Given a mini-batch of B paired protein-ligand, the i-th sequence feature $F_{seq}^{(i)}$ and its corresponding interaction feature $F_{interact}^{(i)}$ are defined as a positive pair, while $F_{seq}^{(i)}$ and $F_{interact}^{(j)}$ with $i \neq j$ are treated as negative pairs. The similarity matrix $S \in \mathbb{R}^{B\times B}$ is then computed:

$$S_{ij} = \frac{Z_{seq}^{(i)} \cdot Z_{interact}^{(j)}}{\tau \|Z_{seq}^{(i)}\| \cdot \|Z_{interact}^{(j)}\|} \tag{2}$$

where τ is a temperature hyperparameter to adjust the distribution of similarity scores. Based on the sample alignment described above, the diagonal elements S_{ii} correspond to positive sequence-interaction pairs, while the off-diagonal elements S_{ij} $(i \neq j)$ correspond to negative pairs. During training, the sequence features are optimized by a loss $L_{seq2interact}$ from the sequence to the interaction direction, which is defined as:

$$L_{seq2interact} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(S_{ii})}{\sum_{j=1}^{B}\exp(S_{ij})} \tag{3}$$

Similarly, the interaction-to-sequence contrastive loss is defined as:

$$L_{interact2seq} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(S_{ii})}{\sum_{j=1}^{B}\exp(S_{ji})} \tag{4}$$

The final bidirectional contrastive loss is obtained by averaging the two directional losses:

$$L_{cl} = \frac{L_{seq2interact} + L_{interact2seq}}{2} \tag{5}$$

By minimizing L_{cl} through backpropagation, the sequence features are closer to their corresponding interaction features in the embedding space and pushed away from non-corresponding ones, thereby enhancing the expressive capability of the sequence features. The resulting optimized sequence features are denoted as $F'_{seq} \in \mathbb{R}^{B\times d_s}$, where B is batch size and d_s=256.

2.2 Hypergraph Representation of Pocket Structure

To capture the high-order relation with pocket structure, we construct the hypergraph to represent the spatial features of pockets. The hypergraph can be defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, X)$, where $\mathcal{V}$ is a node set representing amino acid residues. $\mathcal{E} \subseteq \mathcal{P}(\mathcal{V})$ is a hyperedge set of amino acids, where each hyperedge connects multiple amino acid nodes to capture higher-order structural relations. Let $X \in \mathbb{R}^{|\mathcal{V}|\times d_s}$ denote the node feature matrix of amino acids, where $|\mathcal{V}|$ is the number of amino acid nodes, and d_s is the feature dimension of each node.

For each residue in the binding pocket, we compute the minimum heavy-atom Euclidean distance to every other pocket residue. Hyperedges are then constructed to represent local structural clusters: each hyperedge consists of a central residue and all neighboring residues whose minimum heavy-atom distance to the central residue is below a 5 Å threshold. The hyperedge set $\mathcal{E}$ is constructed by iterating this spatial clustering process across all pocket residues. We apply hypergraph neural network Antelmi et al. (2023) to encode the constructed pocket hypergraph. The initial feature matrix X is processed by two hypergraph convolutional layers to obtain H_{hyper}, which passes through Set2Set graph pooling to obtain final represention $F_{hyper} \in \mathbb{R}^{B\times d_s}$, where B is batch size and $d_s = 256$. More details about the hypergraph representation of pocket structure can be found in Appendix A.2.

2.3 ATOM GRAPH REPRESENTATION OF POCKET STRUCTURE

To better capture both local topological relations and atomic-level features within binding pockets, this paper also constructs an atom graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the node set and $\mathcal{E}$ is the edge set. Each node $v_i \in \mathcal{V}$ represents an atom with feature encoding. Each edge $e_{ij} \in \mathcal{E}$ connects atoms via chemical bonds and spatial proximity.

We design an Adaptive Multi-Channel Aggregation Network (AMCAN) to capture the atomic-level chemical properties and spatial structural relation from atomic graph of pockets. Each layer captures topological-semantic information at different scales through neural weighted message passing. The update of the l-th layer is defined as:

$$h_i^{(l+1)} = \sigma\left(\sum_{j \in N(i)} w_{ij} h_j^{(l)} + (1 + \theta^{(l)}) \cdot h_i^{(l)}\right) \tag{6}$$

where $h_i^{(l)}$ and $h_i^{(l+1)}$ denote the embedding of node i at the l-th and $(l+1)$-th layers, respectively. w_{ij} denotes the learnable weight for edge (i, j). $N(i)$ represents the set of neighboring nodes of node i. $\theta^{(l)}$ denotes the learnable residual coefficient at layer l. To effectively integrate multi-scale node features while minimizing information loss during deep message passing, the Jumping Knowledge aggregation block is introduced to aggregate node representations across all layers, producing the final node embeddings:

$$h_i^{JK} = \max_\text{pooling}(h_i^{(1)}, h_i^{(2)}, ..., h_i^{(L)}) \tag{7}$$

All node embedding features are integrated into sequential node feature vectors to generate a node feature matrix H_{atom}. Finally, the feature matrix H_{atom} passes through Set2Set graph pooling to obtain a final representation $F_{atom} \in \mathbb{R}^{B \times d_s}$, where B is batch size, and d_s=256.

2.4 CROSS-MODAL HIERARCHICAL FUSION

In this work, we fuse multi-modal features to improve the binding specificity of generated molecules toward the target pocket. Figure 1 (e) shows the process of feature fusion by Cross-Modal Hierarchical Fusion module. It comprehensively models global, local, and element-wise interaction patterns to capture intricate protein structure-sequence relations, achieving the deep fusion of graph-based structure and sequence representations of pockets. The cross-modal hierarchical fusion algorithm is provided in Appendix A.3.

2.4.1 BIDIRECTIONAL CROSS-FUSION (BCF)

To enhance the interaction between the residual-level hypergraph feature F_{hyper} and the atom graph feature F_{atom}, the two features are first mapped to a unified low-dimensional common space through two independent linear layers:

$$F'_{hyper} = W_h(F_{hyper}), \quad F'_{atom} = W_a(F_{atom}) \tag{8}$$

where W_h and W_a are trainable parameters. $F'_{hyper} \in \mathbb{R}^{B \times d_s}$, $F'_{atom} \in \mathbb{R}^{B \times d_s}$, where B is batch size, d_s is the embedding dimension of 128.

To capture feature inter-dependencies, a forward cross matrix M_f is constructed, with atom graph embedding vectors F'_{atom} as row vectors and hypergraph embedding vectors F'_{hyper} as column vectors, which is defined as:

$$M_f = F'_{atom} \bigotimes F'_{hyper} \in \mathbb{R}^{B \times d_s \times d_s} \tag{9}$$

An inverse interaction matrix M_r is constructed to enhance symmetry modeling and capture reverse interactions from the hypergraph to the atom graph:

$$M_r = F'_{hyper} \bigotimes F'_{atom} \in \mathbb{R}^{B \times d_s \times d_s} \tag{10}$$

The flatten operation on the two matrices unifies their representations for downstream task adaptation:

$$F_f = \text{ReLU}(W_f \cdot \text{Flatten}(M_f)) \tag{11}$$

$$F_r = \mathrm{ReLU}(W_r \cdot \mathrm{Flatten}(M_r)) \tag{12}$$

In addition, an element-wise interaction operation is used to further enhance the fusion of original modality features:

$$F_{elem} = \mathrm{ReLU}(W_e \cdot (F'_{hyper} \odot F'_{atom})) \tag{13}$$

Finally, the original modality features F_{atom} and F_{hyper} are concatenated with the three interaction features (F_f, F_r, and F_{elem}), and then mapped through an MLP, resulting in the fusion feature F_{ha}:

$$F_{ha} = \mathrm{MLP}(\mathrm{Concat}(F_{atom}, F_{hyper}, F_f, F_r, F_{elem})) \tag{14}$$

2.4.2 Adaptive Gating Mechanism Fusion (AGMF)

This block can adaptively merge structure features and sequence features of protein pocket, enabling synergistic cross-modal learning. The gating parameter α adaptively balances the contributions of the fusion feature F_{ha} and sequence feature F'_{seq} of pockets:

$$\alpha = \sigma(\mathrm{MLP}(\mathrm{Concat}(F_{ha}, F'_{seq}))) \tag{15}$$

where $\sigma(\cdot)$ is the activation function. Next, the gating weights then enable adaptive weighted fusion, producing a unified representation integrating structural and sequential representations:

$$F_{fusion} = \alpha \odot F_{ha} + (1 - \alpha) \odot F'_{seq} + \lambda \cdot (F_{ha} + F'_{seq}) \tag{16}$$

where λ is a residual scaling hyperparameter used to retain the representational power of the original features.

In addition, to improve the model's ability to perceive differences between two modal representations, we introduce cosine similarity loss as a regularization term:

$$L_{diff} = 1 - \mathrm{mean}(\cos(F_{ha}, F'_{seq})) + \beta \cdot \mathrm{Std}[\cos(F_{ha}, F'_{seq})] \tag{17}$$

where β is a hyperparameter controlling the contribution of cosine-similarity variance. This loss drives the two modal representations toward high cosine similarity while penalizing batch-level variance in similarity degree, thereby stabilizing cross-modal alignment and improving overall learning ability.

2.5 Pocket-Conditioned Molecule Generator

Pocket-Conditioned Molecule Generator (PCMG) adopts a two-phase training to enhance sample efficiency: pre-training and fine tuning. During the pre-training phase, unsupervised learning is performed on the large-scale drug-like molecule dataset to acquire generalizable molecular structure representations. The detailed pre-training procedure is provided in Appendix A.4.

During the fine tuning phase, the fusion features F_{fusion} are introduced as conditional inputs based on the pre-trained model. At this stage, the molecule features F_m and the pocket fusion features F_{fusion} processed by linear mapping and other methods are concatenated as fused feature F_{mp}:

$$F_{mp} = Concat(F_m, Linear(F_{fusion})) \tag{18}$$

Through self-attention mechanisms, the intramolecular dependency patterns are captured to generate step-specific embeddings $\hat{F}^t_{pm}$ (t is generation step index) that dynamically incorporate constraints from all preceding tokens.

To achieve bidirectional interaction between pocket and molecule features, the model employs a two-layer gated cross-attention mechanism. A loss function L_{gen} is used to optimize molecule generation. In addition, two auxiliary loss functions L_{cl} and L_{diff} are introduced to strengthen the steering effect of binding pocket features on molecular generation. To dynamically balance multi-loss contributions, the total loss function $Loss$ incorporates learned parameters λ_{gate} and λ_{cl} to weight auxiliary loss terms:

$$Loss = L_{gen} + \frac{\lambda_{gate} \cdot L_{diff} + \lambda_{cl} \cdot L_{cl}}{\lambda_{gate} + \lambda_{cl} + \varepsilon} \tag{19}$$

where λ_{gate} and λ_{cl} are the learnable parameters.

Table 1: Comparison of CoPocket and other models on CrossDocked2020 dataset. (**Best**, Second Best)

Method	Vina Score (↓)	QED (↑)	SA (↑)	Lipinski (↑)	Diversity (↑)	LogP
Ref. (Test)	-7.557	0.446	0.725	4.403	-	1.076
FLAG (Zhang et al., 2023b)	-7.247	0.495	0.745	4.943	0.704	0.630
DeepBlock (Li et al., 2024)	-7.224	0.543	0.757	-	0.784	1.112
PMDM (Huang et al., 2024)	-7.572	0.594	0.611	**4.975**	0.709	0.301
GCDM (Morehead & Cheng, 2024)	-5.955	0.457	0.640	4.758	0.795	-
DTF-diffusion (Wang et al., 2025a)	-6.820	0.540	0.680	-	0.740	-
DiffFBDD (Zheng et al., 2025)	-4.269	0.429	0.327	4.882	0.619	0.039
AMDiff (Li et al., 2025)	-7.466	0.471	0.681	-	0.672	-
TSMGen (Chen et al., 2026)	-7.642	**0.659**	**0.762**	4.966	0.753	2.143
CoPocket (ours)	**-7.693**	0.606	**0.762**	4.850	**0.814**	2.627

3 Experiments and Results

3.1 Dataset and Pre-processing

ChEMBL (Zdrazil et al., 2024) dataset is used in the pre-training process. It contains 2,474,590 drug-like small molecules and SMILES sequences. Following preprocessing steps including isomer removal, duplicate elimination, and filtering out molecules with SMILES sequences shorter than 20 or longer than 120 characters, 1,127,080 drug-like molecules were obtained as the pre-trained drug-like molecule dataset. For training, CrossDocked2020 (Francoeur et al., 2020) dataset is used for comparison, which contains 22.5 million protein-ligand molecule pairs, Luo et al. (Luo et al., 2021) filtered, clustered, and randomly selected 100,000 protein-ligand pairs as the training set, and randomly selected 100 from the remaining clusters as the test set. We removed protein-ligand pairs that could not be converted into hypergraphs because of unparsable PDB files. Finally, 14,770 protein-ligand pairs are obtained for training and 52 protein-ligand pairs are obtained for testing.

3.2 Implementation Details

CoPocket has been performed based on Python 3.8 and PyTorch 1.10. The hardware setup includes an NVIDIA GeForce RTX 3090 (24GB VRAM) with CUDA and cuDNN, running on a Linux system. We set the training process to 50 epochs, with a batch size of 16, and the initial learning rate is set to 2×10^{-6}. A learning rate scheduling strategy that combines linear warm-up and decay was employed, which stabilizes early training and facilitates convergence. AdamW was used with parameter-wise weight decay to improve generalization in high-dimensional parameter spaces.

3.3 Metrics

To evaluate the performance of CoPocket, we selected widely used evaluation metrics Zhang et al. (2023a); Wu et al. (2024): (1) **Vina Score** is computed to estimate the binding affinity between the generated molecules and the targets. (2) **QED** measures the likelihood of a molecule being a potential drug candidate. (3) **SA** reflects the synthetic feasibility of a molecule, normalized between 0 and 1. (4) **LogP** indicates the octanol-water partition coefficient, with most good drug-like compounds lying between -0.4 and 5.6. (5) **Lipinski** calculates the number of generated molecules conforming to Lipinski's rule of five. (6) **Diversity** is used to measure the similarity of the generated molecules.

3.4 Comparative Experiments

Table 1 shows the results of CoPocket and other models on CrossDocked2020 dataset. CoPocket generates the molecules with high affinity (-7.693), which improves by 0.67% compared to the second-best model (-7.642). CoPocket achieved the second-best result in QED, confirming substantially improved drug-likeness of generated molecules. SA and Diversity of CoPocket are optimal among all models, showing its ability to generate synthetically favorable yet structurally novel molecules. The logP value (2.627) of CoPocket within the favorable range (-0.4 to 5.6) significantly enhances the drug candidate potential of generated molecules. Lipinski is not optimal among all

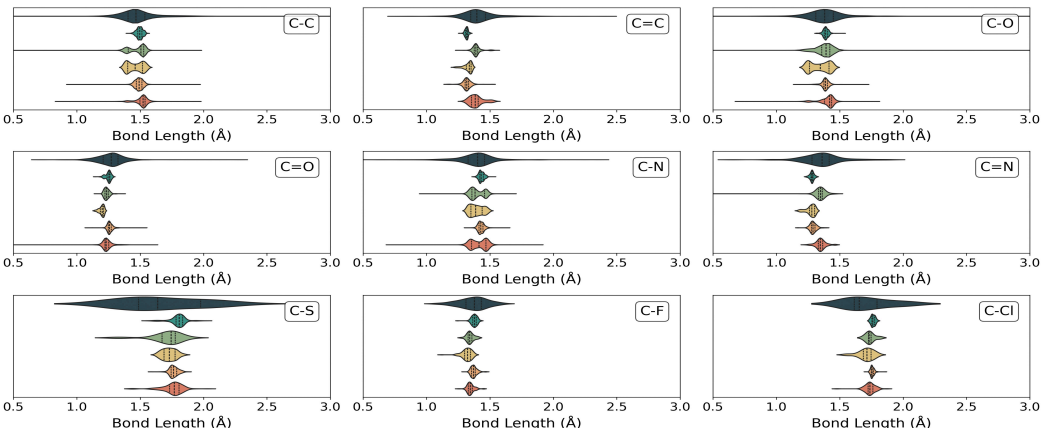

Figure 2: Distribution of chemical bond lengths in the original molecules of the CrossDocked dataset and in molecules generated by different models.

Table 2: Ablation Analysis of different features and modules. (**Best**, Second Best)

Variant	Vina Score (↓)	QED (↑)	SA (↑)	Lipinski (↑)	Diversity (↑)	LogP
CoPocket (w/o F_{hyper})	-7.691	0.592	0.755	4.844	0.814	2.622
CoPocket (w/o F_{atom})	-7.708	0.595	0.754	4.845	0.814	2.685
CoPocket (w/o F_{seq})	-7.681	0.592	0.755	4.839	0.814	2.602
CoPocket (w/o BCF)	-7.684	0.593	0.756	4.838	**0.815**	2.643
CoPocket (w/o AGMF)	-7.665	0.592	0.757	4.845	0.813	2.599
CoPocket (w/o BCF + AGMF)	**-7.730**	0.595	0.757	4.842	0.813	2.664
CoPocket (w/o CMCL)	-7.666	0.596	0.756	4.849	**0.815**	2.605
CoPocket	-7.693	**0.606**	**0.760**	**4.850**	0.814	2.627

models but exceeds that in the test set, indicating that CoPocket has promising potential. CoPocket employs contrastive learning to capture the key interaction patterns between binding pockets and ligands, effectively alleviating the computational redundancy inherent in global modeling. Additional comparisons across multiple protein targets are provided in Appendix A.5.1.

Figure 2 shows the distribution of typical bonds of of generated molecules. CoPocket exhibit a more compact distribution, with less bond length dispersion, and the generated structure has higher consistency compared to those of CrossDocked. For C-C, C=O, the peaks and ranges of CoPocket closely match those of the dataset, indicating accurate reproduction of chemical bond geometries in molecule generation. In addition, detailed analyses of ring sizes, atomic distance distributions, chemical space coverage, target affinity, and generation efficiency are provided in Appendix A.5.2.

3.5 Ablation Analysis

We performed ablation experiments from two perspectives: (1) **Feature ablation**, we remove individual modality features, including pocket hypergraph structure (F_{hyper}), pocket atom graph structure (F_{atom}) or pocket sequence (F_{seq}) to validate their impact on performance. (2) **Module ablation**, we remove key modules (BCF and AGMF, individually and together) to analyze their roles in fusion multi-modal features. In addition, CMCL module is removed to quantify its independent contribution to align the pocket sequence representation with pocket-ligand interactions. The ablation results are summarized in Table 2. CoPocket achieves superior performance, especially achieving optimal results across QED, SA, Lipinski, and Time metrics, demonstrating that multi-modal feature fusion has a promising performance. Following removal of any other component, most metrics experienced varying degrees of degradation. While simultaneously removing both BCF and AGMF modules improves Vina Score (-7.730), all other metrics exhibit declines, demonstrating that the synergistic interaction between BCF and AGMF in our model is crucial to balance multiple metrics.

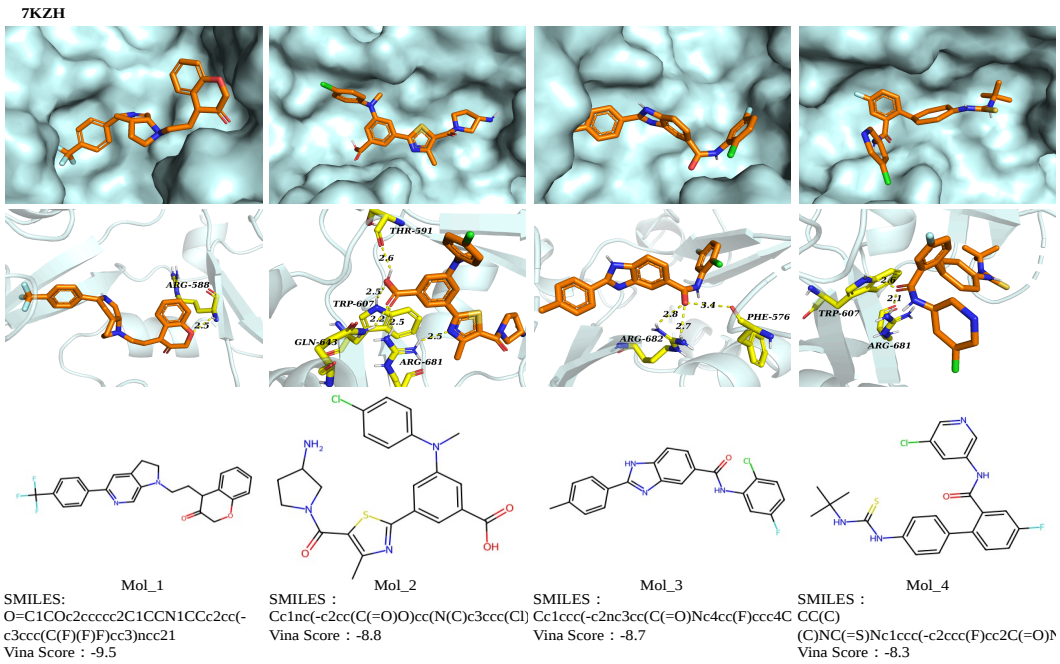

Figure 3: Binding poses, docking results, and molecular structures of four high-affinity molecules in the binding pocket of target 7KZH.

Despite the fact that CoPocket does not achieve the optimal Vina Score and Diversity, it delivers superior comprehensive performance when factoring in advantages across other metrics.

3.6 Case Study

To further validate the ability of CoPocket to generate targeted molecules, case experiments chosen Poly (ADP-ribose) polymerase-13 (PARP13) (Busa et al., 2024) as the target. We selected a central structural domains (PDB ID: 7KZH) of PARP13. Figure 3 displays the spatial positions, docking results, molecular structures, and SMILES sequences of four high-affinity synthetic molecules within the binding pockets of the target 7KZH. It can be observed that all generated molecules embed well within the protein binding pocket, exhibiting reasonable spatial conformations. Furthermore, stable hydrogen bonds form between the generated molecules and key amino acid residues, with interaction distances primarily distributed between 2.2 Å and 3.4 Å. This indicates that these molecules can form robust bonds with the protein pocket. The interaction patterns of molecules Mol_1 to Mol_4 with target 7KZH, and an extended case study of target 7TGQ are provided in Appendix A.6.

4 Conclusion

This paper proposes a target-binding pocket drug design framework (CoPocket), which captures multi-modal features of binding pockets to generate molecules against specific targets. We designed a cross-modal contrastive learning module to align pocket sequence embeddings with atomic-level interaction graph representations of pocket-ligand, thereby optimizing the representation of pocket sequences. We also constructed hypergraphs and atom graphs of protein pockets to capture higher-order and atomic-level features at global and local levels. A cross-modal hierarchical fusion module can fuse multi-modal representations from pocket sequences, pocket hypergraph structure, and pocket atom graph structure. In addition, we designed a pocket-conditioned molecule generator that facilitates bidirectional dynamic matching of pocket features and molecular features to guide the generation of molecules. Experiments show that CoPocket can achieve high-throughput generation of molecules against specific targets.

AI USE STATEMENT

AI-based tools were used during manuscript preparation for language polishing, grammar correction, and improving clarity and readability. All scientific content, including the research ideas, methodology, experiments, analysis, and conclusions, was produced and verified by the authors. The authors take full responsibility for the content and accuracy of the submitted manuscript.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

Chengwei Ai, Hongpeng Yang, Xiaoyi Liu, Ruihan Dong, Yijie Ding, and Fei Guo. Mtmol-gpt: De novo multi-target molecular generation with transformer-based generative adversarial imitation learning. *PLoS computational biology*, 20(6):e1012229, 2024.

Alessia Antelmi, Gennaro Cordasco, Mirko Polato, Vittorio Scarano, Carmine Spagnuolo, and Dingqi Yang. A survey on hypergraph representation learning. *ACM Computing Surveys*, 56 (1):24:1–24:38, 2023.

Nadav Brandes, Dan Ofer, Yam Peleg, Nadav Rappoport, and Michal Linial. ProteinBERT: A Universal Deep-Learning Model of Protein Sequence and Function. *Bioinformatics*, 38(8):2102–2110, 2022.

Veronica F. Busa, Yoshinari Ando, Stefan Aigner, Brian A. Yee, Gene W. Yeo, and Anthony K. L. Leung. Transcriptome regulation by parp13 in basal and antiviral states in human cells. *iScience*, 27(4):109251, 2024.

Yaoyu Chen, Xiaoli Lin, Ziyi Gong, and Jun Pang. TSMGen: Target-specific molecule generation via higher-order structural dependencies and context-aware bidirectional fusion. In *Forty-third International Conference on Machine Learning*, 2026.

Shaoqing Du, Xueping Hu, Luis Menendez-Arias, Peng Zhan, and Xinyong Liu. Target-based drug design strategies to overcome resistance to antiviral agents: opportunities and challenges. *Drug Resistance Updates*, 73:101053, 2024. ISSN 1368-7646.

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.

Amit Gangwal and Antonio Lavecchia. Unleashing the power of generative ai in drug discovery. *Drug Discovery Today*, 29(6):103992, 2024.

Vikas Garg. Generative ai for graph-based drug design: Recent advances and the way forward. *Current Opinion in Structural Biology*, 84:102769, 2024.

Lei Huang, Tingyang Xu, Yang Yu, Peilin Zhao, Xingjian Chen, Jing Han, Zhi Xie, Hailong Li, Wenge Zhong, Ka-Chun Wong, et al. A dual diffusion model enables 3d molecule generation and lead optimization based on target pockets. *Nature Communications*, 15(1):2657, 2024.

Hiroaki Iwata, Taichi Nakai, Takuto Koyama, Shigeyuki Matsumoto, Ryosuke Kojima, and Yasushi Okuno. Vgae-mcts: A new molecular generative model combining the variational graph auto-encoder and monte carlo tree search. *Journal of Chemical Information and Modeling*, 63(23): 7392–7400, 2023.

Fahmi Kazi Md, Shahil Yasar Haque, Eashrat Jahan, Latin Chakma, Tamanna Shermin, Asif Uddin Ahmed, Salekul Islam, Swakkhar Shatabda, and Riasat Azim. Codnet: controlled diffusion network for structure-based drug design. *Bioinformatics Advances*, 5(1):vbaf031, 2025.

G. Li, C. Jiang, Z. Gao, Y. Liu, C. Liu, J. Chen, Y. Huang, and J. Li. Molecule generation for target protein binding with hierarchical consistency diffusion model. *Chemical Science*, 16(39): 18263–18277, 2025. doi: 10.1039/d5sc02113h.

Pengyong Li, Kaihao Zhang, Tianxiao Liu, Ruiqiang Lu, Yangyang Chen, Xiaojun Yao, Lin Gao, and Xiangxiang Zeng. A deep learning approach for rational ligand generation with toxicity control via reactive building blocks. *Nature Computational Science*, 4(11):851–864, 2024.

Shitong Luo, Jiaqi Guan, Jianzhu Ma, and Jian Peng. A 3d generative model for structure-based drug design. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, pp. 6229–6239, 2021.

Jiashun Mao, Jianmin Wang, Amir Zeb, Kwang-Hwi Cho, Haiyan Jin, Jongwan Kim, Onju Lee, Yunyun Wang, and Kyoung Tai No. Transformer-based molecular generative model for antiviral drug design. *Journal of Chemical Information and Modeling*, 64(7):2733–2745, 2024.

Alex Morehead and Jianlin Cheng. Geometry-complete diffusion for 3d molecule generation and optimization. *Communications Chemistry*, 7(1):150, 2024.

Rıza Özçelik, Sarah de Ruiter, Emanuele Criscuolo, and Francesca Grisoni. Chemical language modeling with structured state space sequence models. *Nature Communications*, 15(1):6176, 2024.

Jian Peng, Jun-Lin Yu, Zeng-Bao Yang, Yi-Ting Chen, Si-Qi Wei, Fan-Bo Meng, Yao-Geng Wang, Xiao-Tian Huang, and Guo-Bo Li. Pharmacophore-oriented 3d molecular generation toward efficient feature-customized drug discovery. *Nature Computational Science*, pp. 1–17, 2025.

Yanru Qu, Keyue Qiu, Yuxuan Song, Jingjing Gong, Jiawei Han, Mingyue Zheng, Hao Zhou, and Wei-Ying Ma. Molcraft: Structure-based drug design in continuous parameter space. In *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*, pp. 41243–41257, 2024.

Manasvi Saini, Nisha Mehra, Gaurav Kumar, Rohit Paul, and Béla Kovács. Molecular and structure-based drug design: From theory to practice. In *Advances in Pharmacology*, volume 103, pp. 121–138. Elsevier, 2025.

Huanghao Shi, Zhichao Wang, Litao Zhou, Zhiwang Xu, Liangxu Xie, Ren Kong, and Shan Chang. Status and prospects of research on deep learning-based de novo generation of drug molecules. *Current Computer-Aided Drug Design*, 21(3):257–269, 2025.

Xiangru Tang, Howard Dai, Elizabeth Knight, Fang Wu, Yunyang Li, Tianxiao Li, and Mark Gerstein. A survey of generative ai for de novo drug design: New frontiers in molecule and protein generation. *Briefings in Bioinformatics*, 25(4):bbae338, 2024a.

Yidan Tang, Rocco Moretti, and Jens Meiler. Recent advances in automated structure-based de novo drug design. *Journal of Chemical Information and Modeling*, 64(6):1794–1805, 2024b.

Jianxin Wang, Yongxin Zhu, Yushuang Liu, and Bin Yu. Dtf-diffusion: A 3d equivariant diffusion generation model based on ligand-target information fusion. *Computational Biology and Chemistry*, 117:108392, 2025a.

Jike Wang, Rui Qin, Mingyang Wang, Meijing Fang, Yangyang Zhang, Yuchen Zhu, Qun Su, Qiaolin Gou, Chao Shen, Odin Zhang, et al. Token-mol 1.0: tokenized drug design with large language models. *Nature Communications*, 16(1):4416, 2025b.

Kehan Wu, Yingce Xia, Pan Deng, Renhe Liu, Yuan Zhang, Han Guo, Yumeng Cui, Qizhi Pei, Lijun Wu, Shufang Xie, et al. Tamgen: Drug design with target-aware molecule generation through a chemical language model. *Nature Communications*, 15(1):9360, 2024.

Yu-Ting Xiang, Guang-Yi Huang, Xing-Xing Shi, Ge-Fei Hao, and Guang-Fu Yang. 3d molecular generation models expand chemical space exploration in drug design. *Drug Discovery Today*, 30 (1):104282, 2025. ISSN 1359-6446.

Weixin Xie, Jianhang Zhang, Qin Xie, Chaojun Gong, Yuhao Ren, Jin Xie, Qi Sun, Youjun Xu, Luhua Lai, and Jianfeng Pei. Accelerating discovery of bioactive ligands with pharmacophore-informed generative models. *Nature Communications*, 16(1):2391, 2025.

Chengcheng Xu, Chen Zeng, Xi Yang, Yingxu Liu, Xiangzhen Ning, Lidan Zheng, Yang Liu, Qing Fan, Chao Xu, Haichun Liu, et al. Integrating generative pretrained transformer and genetic algorithms for efficient and diverse molecular generation. *Molecular Informatics*, 44(8):e202500094, 2025.

Gavin Ye. De novo drug design as gpt language modeling: large chemistry models with supervised and reinforcement learning. *Journal of Computer-Aided Molecular Design*, 38(1):20, 2024.

Barbara Zdrazil, Eloy Felix, Fiona Hunter, Emma J. Manners, James Blackshaw, Sybilla Corbett, Marleen De Veij, Harris Ioannidis, David Mendez Lopez, Juan F. Mosquera, et al. The chembl database in 2023: A drug discovery platform spanning multiple bioactivity data types and time periods. *Nucleic Acids Research*, 52(D1):D1180–D1192, 2024.

Odin Zhang, Jintu Zhang, Jieyu Jin, Xujun Zhang, RenLing Hu, Chao Shen, Hanqun Cao, Hongyan Du, Yu Kang, Yafeng Deng, et al. Resgen is a pocket-aware 3d molecular generation model based on parallel multiscale modelling. *Nature Machine Intelligence*, 5(9):1020–1030, 2023a.

Odin Zhang, Yufei Huang, Shichen Cheng, Mengyao Yu, Xujun Zhang, Haitao Lin, Yundian Zeng, Mingyang Wang, Zhenxing Wu, Huifeng Zhao, et al. Fraggen: Towards 3d geometry reliable fragment-based molecular generation. *Chemical Science*, 15(38):13715–13728, 2024.

Zaixi Zhang and Qi Liu. Learning subpocket prototypes for generalizable structure-based drug design. In *Proceedings of the International Conference on Machine Learning (ICML 2023)*, pp. 41382–41398, 2023.

Zaixi Zhang, Yaosen Min, Shuxin Zheng, and Qi Liu. Molecule generation for target protein binding with structural motifs. In *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023)*, 2023b.

Jia Zheng, Hai-Cheng Yi, and Zhu-Hong You. Equivariant 3d-conditional diffusion model for de novo drug design. *IEEE Journal of Biomedical and Health Informatics*, 29(3):1805–1816, 2025.

A Additional Method Details

A.1 Details of Interaction Embedding Block (IEB)

During the construction of interaction graphs, each atom forms connections with neighboring atoms based on spatial proximity, thereby establishing a three-dimensional interaction structure. For each pair of atom nodes i and j, the edge embedding uses a Gaussian function to encode the Euclidean distance:

$$e_{ij}^{interact} = \exp\left(-\frac{\|r_i - r_j\|^2}{2\delta^2}\right) \tag{20}$$

where r_i and r_j denote the 3D coordinates of the atoms, respectively. δ is the standard deviation of the Gaussian function, which is used to adjust the degree of influence of the distance information on the edge embedding.

The initial atom node features are linearly transformed to obtain the low-dimensional embedding $hg_i^{(0)}$, which is then iteratively updated by 6-layer Interaction Embedding Blocks (IEB). Each IEB consists of a Multi-Head Attention (MHA) module and a position FeedForward Network layer (FFN). In the l-layer MHA, the relational information between node i and j is defined as:

$$r_{ij}^{(l)} = \text{Concat}(hg_i^{(l)}, e_{ij}^{interact(l)}) \tag{21}$$

where $hg_i^{(l)}$ denotes the current embedding of node i, and $e_{ij}^{interact(l)}$ denotes the edge feature between node i and j in the interaction graph. Based on this, the query vector $q_i^{(l)}$ is computed from $hg_i^{(l)}$, while the key and value vectors $k_{ij}^{(l)}$, $v_{ij}^{(l)}$ are computed from $r_{ij}^{(l)}$, as follows:

$$q_i^{(l)} = W_Q hg_i^{(l)}, \quad k_{ij}^{(l)} = W_K r_{ij}^{(l)}, \quad v_{ij}^{(l)} = W_V r_{ij}^{(l)} \tag{22}$$

where W_Q, W_K and W_V are learnable transformation matrices. Next, the attention contribution of the neighboring nodes to the central node i is computed by scaling the dot product attention mechanism. The aggregation is represented as:

$$head_i^m = \sum_{j \in N(i)} \text{Softmax}\left(\frac{q_i^{(l)} \cdot k_{ij}^{(l)}}{\sqrt{d_h}}\right) v_{ij}^{(l)} \tag{23}$$

where $m \in \{1, 2, ..., M\}$ (M is the total number of heads). d_h is the embedding dimension used to scale the dot product and mitigate gradient explosion. $N(i)$ denotes the neighbors of the i-th atom in the graph $\mathcal{G}^{interact}$. The outputs of multiple heads are concatenated and then linearly projected to form the multi-head attention output of the current layer:

$$MHA_i = \text{Concat}(head_i^1, ..., head_i^M)W_O \tag{24}$$

where W_O is the output transformation matrix. Finally, the residual connection is utilized with a feed-forward network (FFN) to achieve inter-layer updating of the node embedding:

$$g_i^{(l+1)} = \text{FFN}(\text{LayerNorm}(g_i^{(l)} + MHA_i^{(l)})) + g_i^{(l)} \tag{25}$$

Finally, the node representation hg_i is obtained, incorporating spatial context information.

A.2 Details of Hypergraph Representation of Pocket Structure

During the construction of the pocket hypergraph, the binding pocket region is first identified from the protein, followed by extraction of all amino acid sequence information and 3D spatial coordinates within this pocket. Node positions are defined by the 3D coordinates of amino acid $C\alpha$ atoms within the identified binding pocket.

For each residue in the binding pocket, we compute the Euclidean distance between its $C\alpha$ atom and the $C\alpha$ atom of every other pocket residue. Specifically, for residues v_i and v_j, their spatial distance is defined as:

$$d_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|_2 \tag{26}$$

where $\mathbf{p}_i$ and $\mathbf{p}_j$ denote the three-dimensional coordinates of the $C\alpha$ atoms of residues v_i and v_j, respectively. Hyperedges are then constructed to represent local spatial clusters: each hyperedge e_i consists of a central residue v_i and all neighboring residues whose $C\alpha$ distance from v_i is below a 5 Å threshold:

$$e_i = \left\{ v_j \in \mathcal{V} \mid d_{ij} < 5\,\text{Å} \right\} \tag{27}$$

The hyperedge set is constructed by iterating this spatial clustering process over all pocket residues, i.e.,

$$\mathcal{E} = \{ e_i \mid v_i \in \mathcal{V} \} \tag{28}$$

We apply hypergraph neural network (Antelmi et al., 2023) to encode the constructed pocket hypergraph. HGNN consists of two cascaded supergraph convolution layers. Each layer is used to extract direct associations between local residues and cross-regional structural dependencies, respectively. During hypergraph convolution, the features of each hyperedge are aggregated from the features of all connected nodes to capture the structural relationships and local context of the residues covered by the hyperedge. $F_{edge}^{(l-1)}$ is defined as the feature of the $(l-1)$-th round of amino acid hyperedges:

$$F_{edge}^{(l-1)} = f_{edge_feature}(H, X^{(l-1)}) \tag{29}$$

where $X^{(l-1)}$ represents the node features at the $(l-1)$-th layer, the function $f_{edge_feature}(\cdot)$ represents the aggregation operation from nodes to hyperedges, and $H \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{E}|}$ is the node-hyperedge incidence matrix of the hypergraph $\mathcal{G}$:

$$H_{ij} = \begin{cases} 1, & v_i \in e_j, \\ 0, & v_i \notin e_j. \end{cases} \tag{30}$$

where each column of H represents one spatial hyperedge, and each row indicates the hyperedges associated with the corresponding amino acid residue. Subsequently, the hyperedge features are propagated back to their associated nodes to update the node features and fuse higher-order structural information:

$$X^{l} = \sigma(f_{node_update}(H^{\top}, F_{edge}^{(l-1)})) \tag{31}$$

where $H^{\top}$ denotes the transpose of the hypergraph association matrix, $f_{node_update}(\cdot)$ is the hyperedge-to-node aggregation function, and $\sigma(\cdot)$ is the nonlinear activation function. After multi-layer hypergraph convolution, the node embedding continuously aggregates the multi-body interaction information of its neighboring nodes and hyperedges to effectively capture the higher-order structural features of the protein pocket. A node feature matrix H_{hyper} of the hypergraph is finally obtained.

To obtain the overall representation of the protein pocket, the Set2Set mechanism is used to dynamically aggregate all node features. This mechanism combines recurrent neural network and content attention to dynamically aggregate all node features through multiple rounds of iterations to obtain a globally-aware graph-level representation F_{hyper}, defined as follows:

$$F_{hyper} = q_T^{*} \in \mathbb{R}^{B\times d_s} \tag{32}$$

where q_T^{*} is the graph representation obtained after the T-th iteration, which contains the splicing of the hidden state with the aggregated global contextual information. B denotes the batch size, d_s is the embedding dimension, and d_s = 256.

A.3 Details of Cross-modal Hierarchical Fusion Algorithm

Algorithm 1 shows the process of feature fusion by Cross-Modal Hierarchical Fusion module. It comprehensively models global, local, and element-wise interaction patterns to capture intricate protein structure-sequence relations, achieving the deep fusion of graph-based structure and sequence representations of pockets

A.4 Details of Pocket-Conditioned Molecule Generator

Pocket-Conditioned Molecule Generator (PCMG) adopts a two-phase training to enhance sample efficiency: pre-training and fine tuning. During the pre-training phase, unsupervised learning is

Algorithm 1 Cross-Modal Hierarchical Fusion Algorithm

Input:

F_{atom}: Pocket structure atomic graph features
F_{hyper}: Pocket structure hypergraph features
F'_{seq}: Optimized pocket sequence features

Output:

F_{fusion}: Pocket fusion features

Map F_{atom} and F_{hyper} to a low-dimensional common space:
$F'_{atom} \leftarrow \mathrm{Proj}(F_{atom})$
$F'_{hyper} \leftarrow \mathrm{Proj}(F_{hyper})$
Calculate the forward and reverse interaction representations between two modes:
$F_f \leftarrow \mathrm{CrossForward}(F'_{atom}, F'_{hyper})$
$F_r \leftarrow \mathrm{CrossReverse}(F'_{hyper}, F'_{atom})$
Calculate the element-wise interaction between two modes:
$F_{elem} \leftarrow \mathrm{ElementInteraction}(F'_{atom}, F'_{hyper})$
Structure fusion features of pockets are obtained by concatenating raw modal features and interaction representations and then mapping them:
$F_{ha} \leftarrow \mathrm{MLP}(\mathrm{Concat}(F_{atom}, F_{hyper}, F_f, F_r, F_{elem}))$
Calculate the gating weights to balance structure and pocket sequence features:
$\alpha \leftarrow \mathrm{Gate}(F_{ha}, F'_{seq})$
Final fusion with residual connections:
$F_{fusion} \leftarrow \alpha \cdot F_{ha} + (1-\alpha) \cdot F'_{seq} + \lambda(F_{ha} + F'_{seq})$

performed on the large-scale drug-like molecule dataset to acquire generalizable molecular structure representations. During the fine-tuning phase, the model undergoes supervised fine-tuning using protein-ligand complex data. By incorporating protein pocket information, it enables molecular generation under specific conditions.

A.4.1 TRAINING

In the pre-training stage, the SMILES sequences are mapped to tokens by the word-splitting process. They are converted to high-dimensional spatial representations by the word embedding layer and the position coding layer. The positional relations of the atoms in the sequences are recorded by the position coding layer, which are inputted into the stacked GPT module to predict the probability distribution of the next token. To measure the difference between the probability distribution predicted by the model and the true distribution, the cross-entropy loss function is defined as:

$$L_{gen} = -\frac{1}{B}\sum_{i=1}^{B}\sum_{c=1}^{K} y_{ic}\log(p_{ic}) \tag{33}$$

where K denotes the total number of categories (the number of all different types of characters that can occur in the SMILES sequence). y_{ic} denotes whether the true category of the i-th sample belongs to category c or not. p_{ic} denotes the predicted probability that the i-th sample belongs to category c. The model parameters are then updated by backpropagation to reduce the discrepancy.

Through pre-training, the model is able to learn syntactic laws, fragment linkage rules and contextual relations in molecular structures, laying the foundation for subsequent conditional molecule generation.

During the fine tuning phase, the fusion features F_{fusion} are introduced as conditional inputs based on the pre-trained model. At this stage, the molecule features F_m and the pocket fusion features F_{fusion} processed by linear mapping and other methods are concatenated as fused feature F_{mp}:

$$F_{mp} = \mathrm{Concat}(F_m, \mathrm{Linear}(F_{fusion})) \tag{34}$$

Through self-attention mechanisms, the intramolecular dependency patterns are captured to generate step-specific embeddings $\hat{F}^t_{pm}$ (t is generation step index) that dynamically incorporate constraints from all preceding tokens.

To achieve bidirectional interaction between pocket and molecule features, the model employs a two-layer gated cross-attention mechanism. The first layer takes the fused features of molecule and pocket as Query and pocket features as Key-Value pairs, yielding interactive features $\hat{f}_t^{(1)}$. These features are then fused via a gating mechanism to produce $f_t^{(1)}$:

$$f_t^{(1)} = \sigma(W_{s1}\hat{f}_t^{(1)}) \odot \tanh(W_{o1}\hat{f}_t^{(1)}) + (1 - \sigma(W_{s1}\hat{f}_t^{(1)})) \tag{35}$$

The second layer reverses the cross-attention: it uses pocket features as the Query and the fused features of molecule and pocket as the Key-Value pairs to produce interactive features $\hat{f}_t^{(2)}$, which are then gated to generate $f_t^{(2)}$. The outputs from both cross-attention layers are directly summed with $\hat{F}_{pm}^t$ to obtain the final fused features f_t. These combined features are fed into subsequent layers for training.

A loss function L_{gen} is used to optimize molecule generation. In addition, two auxiliary loss functions L_{cl} and L_{diff} are introduced to strengthen the steering effect of binding pocket features on molecular generation. To dynamically balance multi-loss contributions, the total loss function $Loss$ incorporates learned parameters λ_{gate} and λ_{cl} to weight auxiliary loss terms:

$$Loss = L_{gen} + \frac{\lambda_{gate} \cdot L_{diff} + \lambda_{cl} \cdot L_{cl}}{\lambda_{gate} + \lambda_{cl} + \varepsilon} \tag{36}$$

where λ_{gate} and λ_{cl} are learnable parameters.

A.4.2 Conditional Molecule Generation

During the generation phase, the target protein pocket feature F_{fusion} is input, and the SMILES sequence is generated step by step starting from the initial token. At each generation step, the currently generated partial sequence M_{seq} is processed through word embedding. The pocket feature F_{fusion} is incorporated into the molecular generation process via a cross-attention mechanism. The entire molecular generation process adopts an autoregressive strategy, where the result from the previous step is used as the current input, and the next token is iteratively generated until a termination token is reached or the maximum length limit is exceeded.

A.5 Details of Experiments

A.5.1 Additional comparisons with existing methods across multiple protein targets

We compared CoPocket with existing molecular generation methods across multiple protein targets. Results indicate that molecules generated by CoPocket generally exhibit lower Vina Scores and superior binding affinity at most targets, as shown in Table 3. For targets 1X02 and 2E6D, the DTF-diffusion method generated molecules with Vina scores of -6.2 and -9.2, respectively. The DiffFBDD method generated molecules with Vina scores of -8.6, -7.2, and -8.2 for targets 1PHK, 4F1M, and 4M7T, respectively. For targets 14GS, 2RMA, and 3AF2, molecules generated by the PMDM method yielded Vina scores of -7.5, -6.7, and -9.3, respectively. For target 4Z2G, GCDM generated molecules with a Vina score of -8.7. FLAG generated molecules with a Vina score of -9.3 for target 5BML. AMDiff generated molecules with a score of -9.2 for target 3LCS. Additionally, for target 4M7T, DrugGPS generated molecules with the lowest Vina score of -11.7, slightly lower than CoPocket's -11.3. However, for target 4AAW, CoPocket generated molecules with a Vina score of -9.0, matching DrugGPS's score. For the 2FKY target, the native molecule had the lowest Vina score of -12.1. While CoPocket's score of -10.2 was higher than the reference value, it was lower than the Vina score of the GCDM-generated molecule (-9.7), indicating CoPocket remains competitive for high-affinity targets. Except for the 4M7T and 2FKY targets, CoPocket achieved the best docking scores across all other targets: -7.8 (1X02), -10.4 (2E6D), -11.4 (1PHK), -9.5 (4F1M), -9.0 (4AAW), -9.3 (14GS), -9.6 (2RMA), -10.1 (3AF2), -11.5 (4Z2G), -9.8 (5BML), and -10.2 (3LCS). This demonstrates CoPocket's ability to effectively capture key features of protein-ligand interactions, thereby generating candidate molecules with enhanced binding capacity.

Table 3: Comparing the highest affinity of different methods at the same target.

Target (PDB ID)	Method	Vina Score (↓)
1X02	Original	-7.4
	DTF-diffusion (Wang et al., 2025a)	-6.2
	CoPocket (ours)	**-7.8**
2E6D	Original	-8.7
	DTF-diffusion (Wang et al., 2025a)	-9.2
	CoPocket (ours)	**-10.4**
1PHK	Original	-7.5
	DiffFBDD (Zheng et al., 2025)	-8.6
	CoPocket (ours)	**-11.4**
4F1M	Original	-6.4
	DiffFBDD (Zheng et al., 2025)	-7.2
	CoPocket (ours)	**-9.5**
4M7T	Original	-8.6
	DiffFBDD (Zheng et al., 2025)	-8.2
	DrugGPS (Zhang & Liu, 2023)	**-11.7**
	CoPocket (ours)	-11.3
4AAW	Original	-7.2
	DrugGPS (Zhang & Liu, 2023)	**-9.0**
	CoPocket (ours)	**-9.0**
14GS	Original	-7.5
	PMDM (Huang et al., 2024)	-7.5
	CoPocket (ours)	**-9.3**
2RMA	Original	-6.6
	PMDM (Huang et al., 2024)	-6.7
	CoPocket (ours)	**-9.6**
3AF2	Original	-7.9
	PMDM (Huang et al., 2024)	-9.3
	CoPocket (ours)	**-10.1**
2FKY	Original	**-12.1**
	GCDM (Morehead & Cheng, 2024)	-9.7
	CoPocket (ours)	-10.2
4Z2G	Original	-10.0
	GCDM (Morehead & Cheng, 2024)	-8.7
	CoPocket (ours)	**-11.5**
5BML	Original	-7.5
	FLAG (Zhang et al., 2023b)	-9.3
	CoPocket (ours)	**-9.8**
3LCS	Original	-8.7
	AMDiff (Li et al., 2025)	-9.2
	CoPocket (ours)	**-10.2**

A.5.2 Additional analyses of molecular properties and generation performance

Figure 4 shows the proportion of ring sizes among generated molecules. The generated molecules by CoPocket contain a significantly higher proportion of stable 5-atom ring and 6-atom ring compared to unstable rings, aligning with the fundamental stability requirements in drug design.

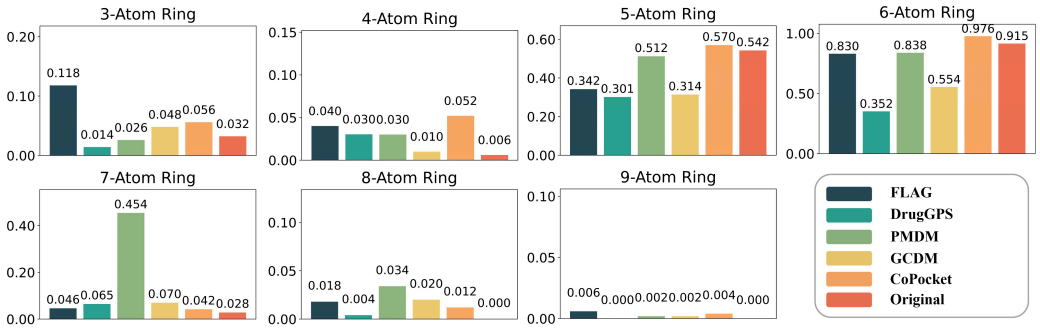

Figure 4: Proportion of different ring sizes in the original molecules of the CrossDocked dataset and in molecules generated by different models.

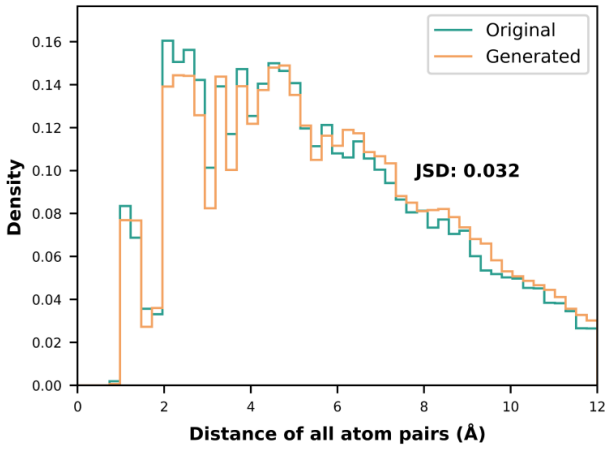

Figure 5: Comparison of the distance distribution between atoms in the original molecule and the generated molecules. The original molecule refers to the molecules in the test set.

To compare the differences in the distance distribution of atom pairs between the generated molecules and the test set, we use Jensen-Shannon divergence (JSD) to quantify the distribution differences between the two, as shown in Figure 5. JSD value of CoPocket achieves 0.032, indicating that the distance distribution of the atomic bond in the generated molecules is very close to that of the test set.

Figure 6 illustrates the chemical spatial distribution of generated molecules. These molecules exhibit a broad distribution within the embedding space, covering the primary distribution regions of real molecules. This indicates that the model-generated molecules maintain consistency with real molecules in terms of global chemical characteristics. Within the blue solid-line box region (with representative molecules from this region shown on the right), it can be observed that while the generated molecules share chemical similarities with real molecules, they exhibit distinct structural novelty. This demonstrates the model's ability to explore new chemical spaces while maintaining molecular plausibility, showcasing its capacity to investigate potential chemical structural spaces and generate structurally diverse molecules. Figure 7 shows the time required for each model to generate 100 molecules. CoPocket exhibits the highest computational efficiency among all models.

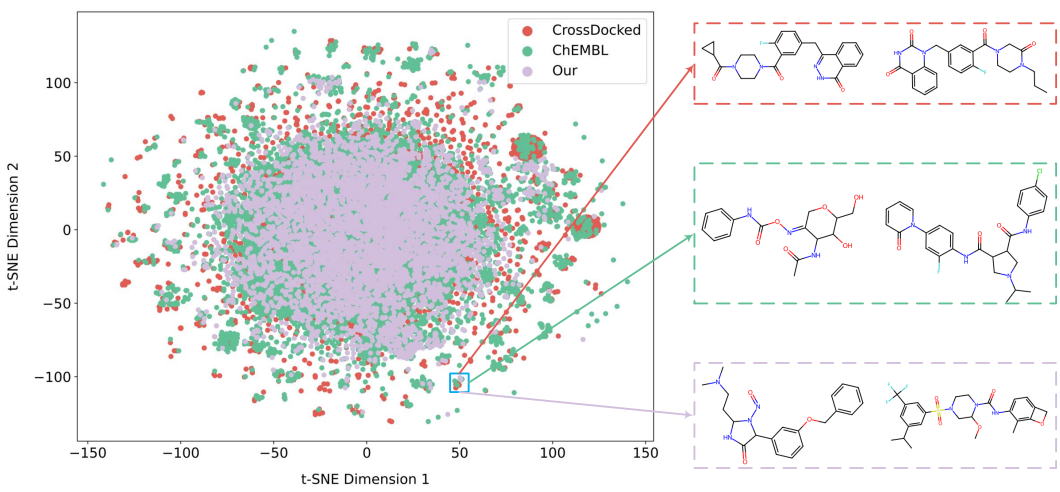

Figure 6: Molecular chemical spatial distribution. The figure displays the t-SNE distributions of CrossDocked (red), ChEMBL (green), and generated molecules (purple). The red, green, and purple boxes on the right correspond to representative examples of the three types of molecules within the blue solid-line box.

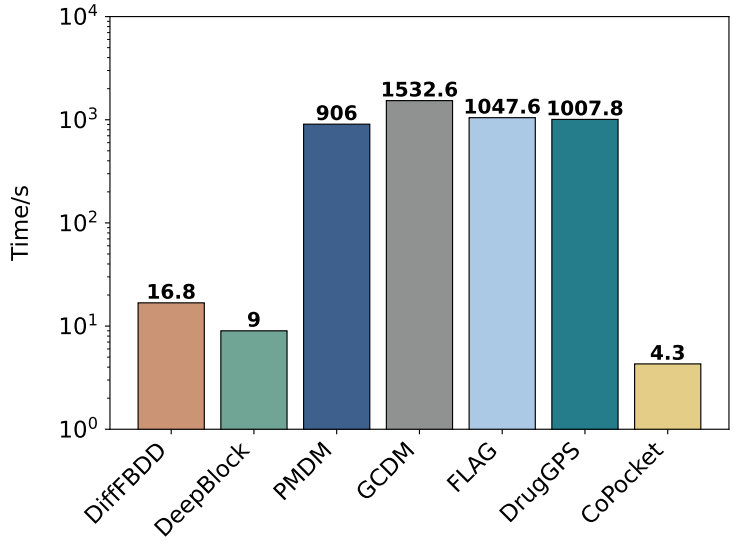

Figure 7: Comparison of molecular generation time across different models.

A.6 Extended case study of target 7TGQ and Interaction patterns

We selected another central structural domain (PDB ID: 7TGQ) of PARP13. Figure 8 displays the spatial positions, docking results, molecular structures, and SMILES sequences of four high-affinity synthetic molecules within the binding pocket of target protein 7TGQ.

Figure 9 shows the diverse interaction patterns formed by synthetic molecules Mol_1 to Mol_4 with target protein 7KZH, Mol_5 to Mol_8 with target protein 7TGQ. For target 7KZH, synthetic molecules primarily bind to the protein through hydrogen bonds, halogen bonds, π–π stacking, and carbon-hydrogen bonds. Specifically, the pyridine ring of Mol_1 exhibits π–π stacking with TYR-690, while the phenyl ring connected to the cyano group forms π–alkyl interactions with VAL-686 and PRO-535. Additionally, the oxygen and ethyl group on the pyrrole ring establish hydrocarbon bonds with SER-601 and VAL-602. Mol_2 forms multiple hydrogen bonds with ARG-588, ARG-

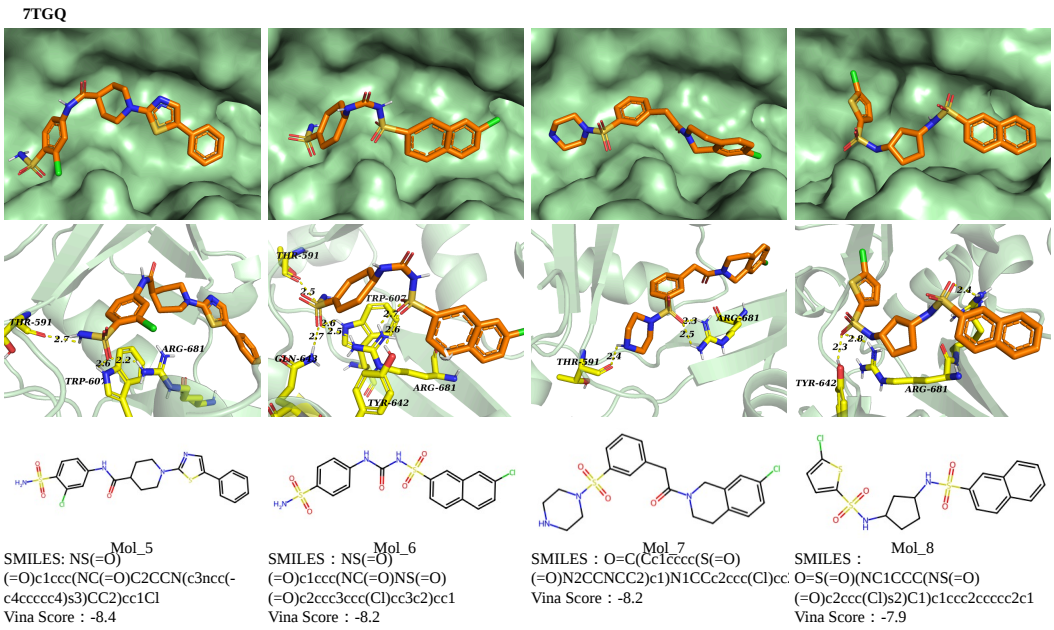

Figure 8: For the target 7TGQ, four highly affinity-binding molecules are presented. The figure depicts the spatial positioning of molecules within the protein binding pocket, docking results (orange indicates generated molecules, yellow denotes protein amino acid residues, and yellow dashed lines represent hydrogen bonds), along with corresponding molecular structures.

681, ARG-682 and PHE-663. The chlorine atom in its halogenated benzene ring participates in halogen bond formation, while π–alkyl interactions exist between the benzene ring and the protein's hydrophobic residues. The fluorine atom on the halogenated benzene ring of Mol_3 acts as an active group, forming hydrogen bonds with the protein's GLN-643, TRP-607, and ARG-681, respectively. The amide group also forms a hydrogen bond with the protein's SER-590. The aromatic ring exhibits π–π conjugation effects with TYR-559 and others. In the Mol_4 structure, the amide group forms hydrogen bonds with ARG-681 and ARG-682 of the protein. The ring system also exhibits π-cation and π-σ interactions with ARG-682 and ILE-680, while the halogen atom fluorine forms a halogen bond with PHE-663, collectively enhancing overall binding stability.

For target 7TGQ, the sulfonamide and amide groups in the synthetic molecular structure play a crucial role in binding. The sulfonamide group of Mol_5 forms hydrogen bonds with THR-591, ARG-681, and TRP-607 via its amino and sulfonyl groups, respectively. Its aromatic ring system engages in π–Cation and π–Sulfur interactions with ARG-681, ARG-682, and MET-666. The sulfonamide group in Mol_6 forms hydrogen bonds with TYR-559, THR-591, SER-593, TRP-607, and GLN-643 of the protein, while the naphthalene ring exhibits π–Cation interactions with ARG-682. The amide group and sulfonamide group in Mol_7 forms hydrogen bonds with TYR-559 and ARG-681, respectively. The amino group in the piperazine ring serves as the active site, establishing further interactions with SER-590 and THR-591. Additionally, the two benzene rings in the structure exhibit π–π and π–σ conjugation effects with TRP-607 and MET-666, respectively. In the Mol_8 structure, the two sulfonamide groups form hydrogen bonds with TYR-642, ARG-681, and ARG-682, respectively, with the sulfonamide group playing a key role in positioning during binding. It can be seen that ARG-681 and its surrounding amino acids in the protein constitute the active site residues, exhibiting strong affinity for small molecules containing a sulfonamide structure.

Overall, the generated molecules demonstrate rational binding patterns across different targets, forming stable and diverse interactions with key residues.

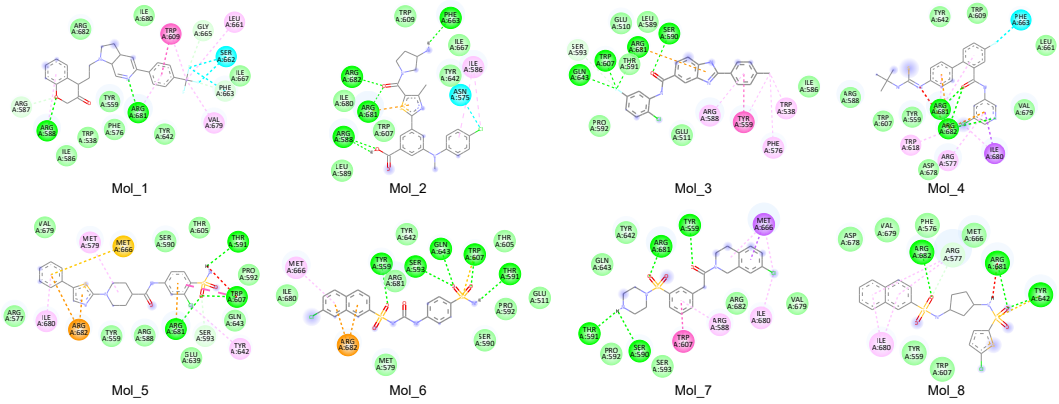

Figure 9: The binding patterns of the generated molecules with target proteins (Mol_1 to Mol_4 with 7KZH, and Mol_5 to Mol_8 with 7TGQ).