Learning Subpocket Prototypes for Generalizable Structure-based Drug Design

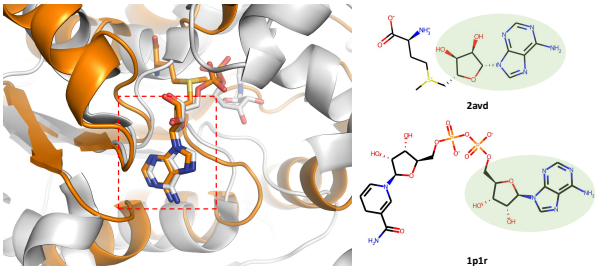
Zaixi Zhang [1 2] **Qi Liu** [1 2]

Abstract

Generating molecules with high binding affinities to target proteins (a.k.a. structure-based drug design) is a fundamental and challenging task in drug discovery. Recently, deep generative models have achieved remarkable success in generating 3D molecules conditioned on the protein pocket. However, most existing methods consider molecular generation for protein pockets independently while neglecting the underlying connections such as subpocket-level similarities. Subpockets are the local protein environments of ligand fragments and pockets with similar subpockets may bind the same molecular fragment (motif) even though their overall structures are different. Therefore, the trained models can hardly generalize to unseen protein pockets in real-world applications. In this paper, we propose a novel method DrugGPS for generalizable structure-based drug design. With the biochemical priors, we propose to learn subpocket prototypes and construct a global interaction graph to model the interactions between subpocket prototypes and molecular motifs. Moreover, a hierarchical graph transformer encoder and motif-based 3D molecule generation scheme are used to improve the model's performance. The experimental results show that our model consistently outperforms baselines in generating realistic drug candidates with high affinities in challenging out-of-distribution settings.

1. Introduction

Structure-based drug design (SBDD), i.e., designing molecules with high affinities to target protein pockets is one critical and challenging task in drug discovery (Anderson, 2003; Blundell, 1996; Verlinde & Hol, 1994; Ferreira et al., 2015; Jin et al., 2020b). Traditionally this has been achieved with virtual screening that identifies candidate molecules from molecular databases based on rules such as molecular docking (Morris & Lim-Wilby, 2008; Pagadala et al., 2017) and molecular dynamics simulations (Hansson et al., 2002; Karplus & McCammon, 2002). However, such exhaustive searches are time-consuming and infeasible to generate new molecules not existing in the database. Recently, a line of works leverage deep generative models to directly generate 3D molecules inside binding pockets (Luo & Ji, 2021; Liu et al., 2022; Peng et al., 2022; Zhang et al., 2023b).



(a) Aligned subpocket pairs (b) Ligands

Figure 1. Illustration of our motivation. (a) Two proteins (PDB ID: 2avd and 1p1r) with low sequence similarity ($\leq$10%) have similar subpockets and bind to similar ligand fragments. 2avd is colored yellow and 1p1r is color white. The subpockets are aligned and highlighted with a red dashed box. (b) The molecular graphs of ligands binding to protein 2avd and 1p1r. Similar fragments in the subpockets are marked with green ovals.

However, existing methods suffer from the generalization issue. The amount of high-quality protein-ligand complex data is rather limited and the target protein pocket may not be in the training dataset. In practice, when unpredictable events like COVID-19 occur, the generative models are required to generate molecules for new protein targets e.g., the main protease of SARS-CoV-2 (Zhang et al., 2020). Moreover, only atom-level interactions are considered and encoded in these works and the atom-by-atom generation may result in invalid molecules with unrealistic 3D structures. More discussions are included in related works.

In this paper, we propose **DrugGPS**, a structure-based **Drug** design method that is **G**eneralizable with **P**rotein **S**ubpocket prototypes to address the aforementioned challenges. Firstly, an atom-level graph and a residue-level graph are con-

[1]Anhui Province Key Lab of Big Data Analysis and Application, University of Science and Technology of China [2]State Key Laboratory of Cognitive Intelligence. Correspondence to: Qi Liu <qiliuql@ustc.edu.cn>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

structed to represent the binding context. A hierarchical 3D graph transformer is proposed to capture the hierarchical information. Secondly, to construct SBDD models that generalize well to unseen target protein pockets, we incorporate an effective biochemical prior into our model design: **although two protein pockets might be dissimilar overall, they may still bind the same fragment if they share similar subpockets** (Kalliokoski et al., 2013). Subpockets are defined as the local protein environment of the ligand fragments in protein-ligand complexes (Eguida et al., 2022). For example, in Figure. 1, two proteins (PDB ID: 2avd and 1p1r) with low sequence similarity ($\leq 10\%$) have similar subpockets and bind to similar ligand fragments. To capture the subpocket-level similarities/invariance among the binding pockets, we propose to learn subpocket prototypes and construct a global interaction graph to model the interactions between subpocket prototypes and molecular motifs (fragments) in the training process. To further highlight the subpocket-motif interactions, we employ an efficient binding analysis tool BINANA (Young et al., 2022) to identify polar contacts (hydrogen bonds). In the generation process, the context representations are enriched with a global information fusion step and ligand molecules are generated motif-by-motif. In experiments, to mimic the real-world use case, we split the dataset based on sequence-similarity and pocket-similarity and construct two out-of-distribution (OOD) settings. Experiment results demonstrate that our method can generalize well to unseen pockets in the test set. The generated molecules not only show higher binding affinities and drug-likeness but also contain more realistic substructures than the state-of-the-art baseline methods. Our key contributions include:

- In this paper, we propose **DrugGPS**, a structure-based **Drug** design method that is **G**eneralizable with **P**rotein **S**ubpocket prototypes.
- A hierarchical 3D graph transformer is proposed to encode both the atom- and residue-level information.
- We propose to construct the subpocket prototypes-molecular motif interaction graph in the training process. At the generation stage, molecules are generated motif-by-motif with the global interaction information.
- Experiments show that our model consistently outperforms baselines on generating realistic drug candidates with high affinities on challenging OOD settings.

2. Related Works

2.1. Molecule Generation

Recent years have witnessed the great success of deep generative models in molecule generation (Zhang et al., 2023a; Lee et al., 2022; Xie et al., 2021; Yang et al.). These models range from string-based (Gómez-Bombarelli et al., 2018) and graph-based methods (Jin et al., 2018; Xie et al., 2021) to recent 3D geometry-based methods (Gebauer et al., 2019; Luo & Ji, 2021). To enhance the validity of the generated molecules, some models adopt prior knowledge of molecular fragments, also known as motifs or rationales, as building blocks to generate and optimize molecules (Jin et al., 2018; 2020a; Xie et al., 2021). However, the generated molecules could hardly fit and bind to given pockets in practice if the 3D conditional information, e.g., the shape and chemical properties of the protein pockets are neglected.

2.2. Structure-based Drug Design.

Structure-based drug design (SBDD) aims to directly generate 3D molecules binding to target protein pockets. LiGAN (Ragoza et al., 2022) first uses 3D CNN to encode the protein-ligand structures and generate ligands by atom fitting and bond inference from the predicted atom densities. Some follow-up works leverage graph neural networks to encode the context information and sample atoms autoregressively (Luo & Ji, 2021; Liu et al., 2022; Peng et al., 2022). For example, GraphBP (Liu et al., 2022) adopts the framework of normalizing flow (Rezende & Mohamed, 2015) and constructs local coordinate systems to predict atom types and relative positions; Pocket2Mol (Peng et al., 2022) adopts the geometric vector perceptrons (Jing et al., 2021) and the vector-based neural network (Deng et al., 2021) as the context encoder. Some recent works also leverage fragment-based methods (Green et al., 2021; Powers et al., 2022; Zhang et al., 2023b) or pretrained models (Long et al., 2022) to generate more realistic molecules. For example, (Powers et al., 2022) expands a small molecule fragment into a larger drug-like molecule binding to a given protein pocket. However, most existing methods suffer from the generalization concern in practice where only low-quality and deficient data is available. On the contrary, DrugGPS leverage the priors of protein subpockets to build generalizable models.

2.3. Generalizable Drug Discovery

The ability to successfully apply previously acquired knowledge/data to new situations is vital to drug discovery (Ji et al., 2022; Yang et al., 2022; Zhang et al., 2022b). To this end, many works study drug discovery problems under out-of-distribution settings. For example, MoleOOD (Yang et al., 2022) builds generalizable molecule representation learning models against distribution shifts by learning invariant molecular substructure. PAR (Wang et al., 2021) uses a meta-learning strategy for few-shot molecular property prediction. MOOD (Lee et al., 2022) designs an out-of-distribution molecule generation scheme with score-based diffusion to explore chemical space. However, these methods can hardly be applied to the more challenging condi-

tional molecule generation task, i.e., structure-based drug design.

3. Methods

3.1. Overview

We first formalize the problem of structure-based drug design. Given a protein pocket-ligand complex, the 3D geometry of the ligand molecule can be represented as a set of atoms $\mathcal{G}^{mol} = \{(\boldsymbol{a}_i^{mol}, \boldsymbol{r}_i^{mol})\}$. The protein pocket (i.e., binding site) can be similarly defined as $\mathcal{G}^{pro} = \{(\boldsymbol{a}_j^{pro}, \boldsymbol{r}_j^{pro})\}$. In $\mathcal{G}^{mol}$ and $\mathcal{G}^{pro}$, $\boldsymbol{a}_i^{mol}$ and $\boldsymbol{a}_j^{pro}$ are one-hot vectors indicating the atom types and $\boldsymbol{r}_i^{mol}, \boldsymbol{r}_j^{pro} \in \mathbb{R}^3$ are the 3D cartesian coordinate vectors. Formally, our objective is to learn a conditional generative model $p(\mathcal{G}^{mol}|\mathcal{G}^{pro})$ that captures the underlying dependencies of pocket-ligand pairs for 3D ligand molecule generation.

Specifically, we formulate the generation of molecules given binding pocket as a sequential decision process. Let ϕ be the generation model and $\mathcal{G}_t^{mol}$ be the intermediate molecule at the t-th step, the generation process is defined as follows:

$$\mathcal{G}_t^{mol} = \phi(\mathcal{G}_{t-1}^{mol}, \mathcal{G}^{pro}),\ t > 1 \tag{1}$$

$$\mathcal{G}_1^{mol} = \phi(\mathcal{G}^{pro}),\ t = 1. \tag{2}$$

Note that we generate molecules motif-by-motif, i.e., a set of atoms from the new motif are included into $\mathcal{G}_t^{mol}$ at each step. Figure. 2(a) demonstrates the four main parts in one generation step, including (a) context encoding and focal motif selection, (b) next motif prediction, (c) motif attachment prediction, and (d) rotation angle prediction.

In this section, we first introduce the motif extraction procedure in Sec. 3.2. In Sec. 3.3 and Sec. 3.4, we will introduce the hierarchical context encoder and the construction of a global interaction graph, which are our main contributions to model architecture. In Sec. 3.5 and Sec. 3.6 we introduce the detailed generation procedures and derive the final training objectives.

3.2. Motif Vocabulary Construction

Motif vocabulary construction aims to extract common molecular motifs from ligand molecules in whole dataset and construct a motif vocabulary $V_{\mathcal{M}} = \{\mathcal{M}_i\}$ for the follow-up molecule generation. For the ease of motif extraction, molecules can be represented as 2D graphs $\mathcal{G}^{mol} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V}$ as atoms set and $\mathcal{E}$ as covalent bonds set. Similarly, a motif $\mathcal{M}_i = (\mathcal{V}_i, \mathcal{E}_i)$ is defined as a molecular subgraph. Each molecule can also be represented as a set of motifs: $\mathcal{V} = \bigcup_i \mathcal{V}_i$ and $\mathcal{E} = \bigcup_i \mathcal{E}_i$.

Figure. 3(a) shows the procedures to fragment molecules and construct the motif vocabulary. To extract structural motifs, we first decompose a molecule $\mathcal{G}^{mol}$ into molecular substructures $\mathcal{G}_1, \cdots, \mathcal{G}_n$ by extracting and detaching all the rotatable bonds that will not violate the chemical validity. A bond in a molecule is rotatable if cutting this bond creates two connected components of the molecule, each of which has at least two atoms. We select $\mathcal{G}_i$ as a motif if its occurrence in the whole training set is more than τ. We can select hyperparameter τ to control the size of the motif vocabulary $V_{\mathcal{M}}$ ranging from around 500 to over 2000. If $\mathcal{G}_i$ is not selected as a motif, we further decompose it into finer rings and bonds and select them as motifs. As the bond length/angles in motifs are largely fixed, we employ RDkit (Bento et al., 2020) to efficiently determine the 3D structures of motifs and trains neural networks to predict the torsion angles of rotatable bonds.

3.3. Hierachical Context Encoder

Inspired by the intrinsic hierarchical structure of protein (Stoker, 2015), we propose a hierarchical context encoder based on graph transformer (Min et al., 2022) to capture the context information of binding sites. Specifically, it includes an atom-level encoder and a residue-level encoder as described below.

3.3.1. Atom-level Encoder

For the atom-level encoding, a context 3D graph $\mathcal{C}_{t-1}^a$ is first constructed by connecting the K_a nearest neighboring atoms in $\mathcal{G}_{t-1}^{mol} \bigcup \mathcal{G}^{pro}$. The atomic attributes are firstly mapped to node embeddings $\boldsymbol{h}_k^{(0)}$ with a linear transformation layer. The edge embeddings $\boldsymbol{e}_{ij}$ are obtained by encoding pairwise distances with Gaussian functions (Schlichtkrull et al., 2018). The 3D graph transformer consists of L Transformer layers (Vaswani et al., 2017). Each Transformer layer has two parts: a multi-head self-attention (MHA) module and a position-wise feed-forward network (FFN). Particularly, in the MHA module of the l-th layer ($1 \le l \le L$), the queries are derived from the current node embeddings $\boldsymbol{h}_i^{(l)}$ while the keys and values from the relational information $\boldsymbol{r}_{ij}^{(l)} = \mathrm{Concat}(\boldsymbol{h}_j^{(l)}, \boldsymbol{e}_{ij}^{(l)})$ ($\mathrm{Concat}(\cdot)$ denotes concatenation) from neighboring nodes:

$$\boldsymbol{q}_i^{(l)} = \mathbf{W}_Q \boldsymbol{h}_i^{(l)},\ \boldsymbol{k}_{ij}^{(l)} = \mathbf{W}_K \boldsymbol{r}_{ij}^{(l)},\ \boldsymbol{v}_{ij}^{(l)} = \mathbf{W}_V \boldsymbol{r}_{ij}^{(l)}, \tag{3}$$

where $\mathbf{W}_Q, \mathbf{W}_K$ and $\mathbf{W}_V$ are learnable transformation matrices. Then, in each head $m \in \{1, 2, \ldots, M\}$ (M is the total number of heads), the scaled dot-product attention mechanism is applied:

$$\mathrm{head}_i^m = \sum_{j \in \mathcal{N}(i)} \mathrm{Softmax}\left(\frac{\boldsymbol{q}_i^{(l)\top} \cdot \boldsymbol{k}_{ij}^{(l)}}{\sqrt{d}}\right) \boldsymbol{v}_{ij}^{(l)}, \tag{4}$$

where $\mathcal{N}(i)$ denotes the neighbors of the i-th atom in $\mathcal{C}_{t-1}^a$ and d is the dimension size of embeddings. Finally, the

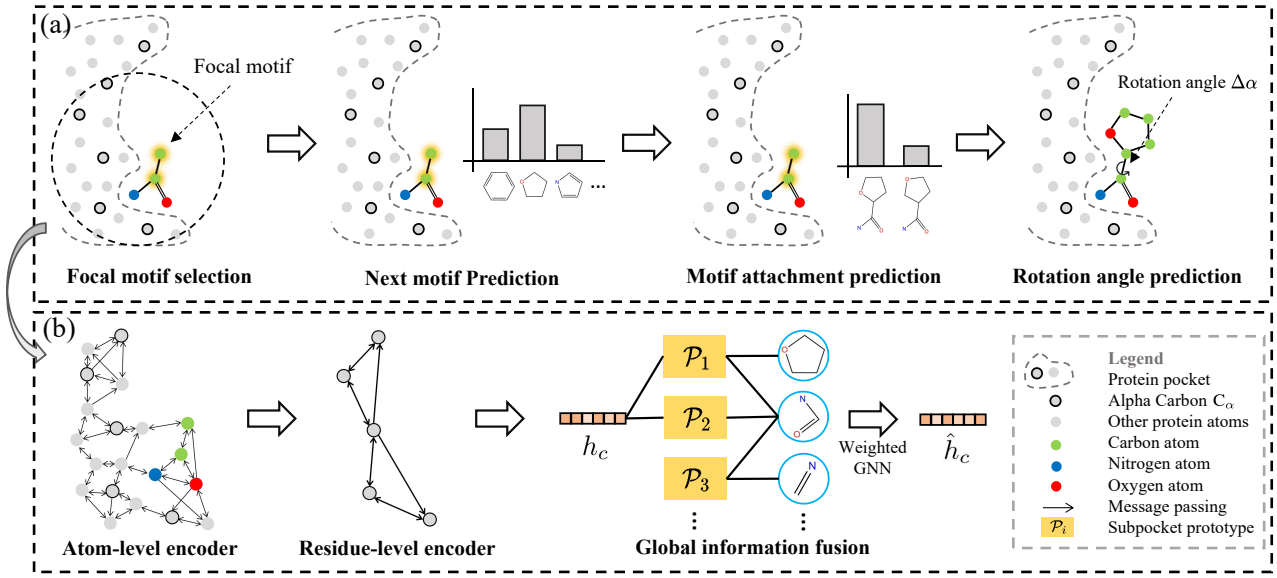


Figure 2. (a) The illustration of one generation step including four parts in our motif-based ligand generation scheme. (b) The hierarchical context encoder in DrugGPS. The global interaction information is further encoded into the subpocket embedding h_c by a weighted GNN.

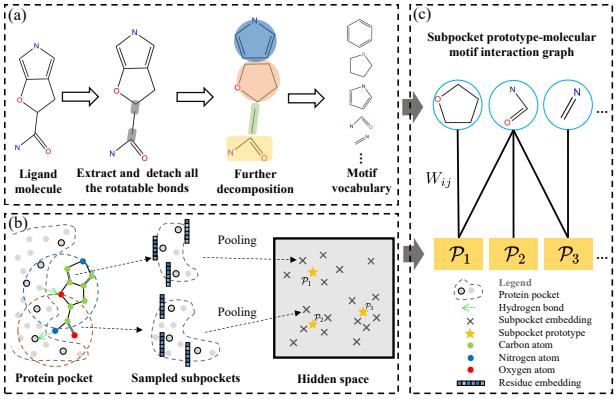


Figure 3. (a) The illustration of molecular motif extraction. (b) The sampled subpockets and subpocket prototypes. (c) The constructed Subpocket prototype-molecular motif interaction graph.

outputs from different heads are further concatenated and transformed to obtain the final output of MHA:

$$\mathrm{MHA}_i = \mathrm{Concat}\left(\mathrm{head}_i^1, \ldots, \mathrm{head}_i^M\right)\mathbf{W}_O, \quad (5)$$

where $\mathbf{W}_O$ is the output transformation matrix. The output of the atom-level encoder is a set of atom representations $\{\boldsymbol{h}_i\}$. More architectural details are shown in Appendix. A.

3.3.2. Residue-level Encoder

The residue-level encoder only keeps the C_α atom of each residue and constructs a K_r nearest neighbor graph $\mathcal{C}_{t-1}^{res}$ at the residue level. The i-th residue (res_i) can be represented by a feature vector $\boldsymbol{f}_i$ describing its geometric and chemical characteristics including its dihedral angles, volume, polarity, charge, hydropathy, and hydrogen bond interactions. We concatenate the residue features with the sum of atom-level embeddings $\boldsymbol{h}_k$ within that residue as the initial residue representation:

$$\tilde{\boldsymbol{f}}_i = \mathrm{Concat}\left(\boldsymbol{f}_i, \sum\nolimits_{k \in res_i} \boldsymbol{h}_k\right). \quad (6)$$

A local coordinate frame is built for each residue and the edge features $\boldsymbol{e}_{ij}^{res}$ between residues are computed describing the distance, direction, and orientation between neighboring residues (Ingraham et al., 2019). Lastly, the encoder takes the node and edge features into the residue-level graph transformer to compute the final representations of residues. The residue-level graph transformer architecture is similar to that of the atom-level encoder. More details of the residue-level encoder are shown in Appendix. A.

In summary, the output of our hierarchical encoder is a set of residue representations $\{\boldsymbol{f}_i\}$ and atom representations $\{\boldsymbol{h}_i\}$. Considering the distance range of pocket-ligand interactions (Marcou & Rognan, 2007), we sum all the residue representations within 6 Å of the focal atom as the subpocket representations (Figure. 3(b)). Since our encoder is based on the atom/residue attributes and pairwise relative distances, it is rotationally and translationally equivariant.

3.4. Global Interaction Graph Construction

Most existing methods consider molecular generation for protein pockets independently while neglecting the underlying connections of subpocket-level similarities (Kalliokoski et al., 2013). Here, we construct a global interaction graph

to model the interactions between subpocket and ligand fragments in the whole dataset. As there are numerous subpockets in the dataset, we propose to cluster subpocket embeddings and derive representative subpocket prototypes (Figure. 3(b)). Therefore, we have two kinds of nodes in the global interaction graph: subpocket prototype nodes and molecular motif nodes from the motif vocabulary (Figure. 3(c)). The embeddings of subpocket prototypes and molecular motifs are dynamically updated during the training process. We add an edge between a subpocket prototype node and a molecular motif node if a subpocket belonging to the cluster of the prototype binds to the motif in the training dataset. As the strength of the interaction is different, we calculate TF-IDF value as the edge weight W_{ij} between a subpocket prototype i and motif j:

$$W_{ij} = C_{ij}\left(\log\frac{1+N}{1+N_i}+1\right), \tag{7}$$

where C_{ij} is the number of times that motif j binds to subpockets belonging to prototype i, N is the total number of subpocket prototypes, and N_i is the number of subpocket prototypes binding to motif i. An edge has a larger weight if motif j has a higher co-occurrence rate with prototype i and binds to fewer other prototypes (higher specificity).

To further highlight the interactions between subpockets and molecular motifs, we employ an efficient binding analysis tool BINANA (Young et al., 2022) that is able to analyze the detailed interactions including hydrogen bonds, π-π stacking, cation-π interactions, electrostatic attraction, and hydrophobic with respect to each atom. When calculating edge weights W_{ij} of the interaction graph, we only count subpocket-molecular motifs pairs with at least one hydrogen bond, which contributes much to binding affinity.

We update the subpocket prototypes on the fly during the model training process with an online K-Means algorithm. Specifically, to stabilize the training process, we update the prototypes i.e., centroids of clusters with momentum. Algorithm 1 shows the pseudo-codes of updating subpocket prototypes with a batch of input representations H. Hyperparameter γ is used to control the momentum. FindNearest$(\cdot,\cdot)$ denotes the function to find the nearest prototypes of inputs in Euclidean space and S is the assignment matrix. Appendix. B contains more details of the global interaction graph construction.

Algorithm 1 Subpocket Prototypes Update Algorithm

Input: subpocket representations H, momentum hyperparameter γ, Prototypes $\mathcal{P}$, Count of data per cluster c
Output: Updated subpocket prototypes $\mathcal{P}$

S = FindNearest(H, $\mathcal{P}$)
$\mathcal{P} \leftarrow c\cdot\mathcal{P}\cdot\gamma + S^\top \mathrm{H}\cdot(1-\gamma)$
$c \leftarrow c\cdot\gamma + S^\top\mathbf{1}\cdot(1-\gamma)$
$\mathcal{P} \leftarrow \mathcal{P}/c$

3.5. Prototype-augmented Motif Generation

The intuition of this Prototype-augmented Motif Generation is that according to the similarity principle: **molecular motifs originating from similar subpockets are likely to bind with the target protein pocket with high affinity**.

In the generation process, we firstly obtain atom and residue embeddings from the hierarchical context encoder. The subpocket embedding $\boldsymbol{h}_c$ can be obtained by sum pooling all the residue embeddings within 6 Å of the focal atom. To leverage the knowledge from the global interaction graph, we take a global information fusion step in Figure. 2(b): we add edges between the subpocket embedding with K_p most similar subpocket prototypes in the global interaction graph with edge weights set as 1. Then we use a weighted graph neural network to propagate the global information and take the output subpocket representation as $\hat{\boldsymbol{h}}_c$. The related subpocket prototype-motif interaction information can be encoded in $\hat{\boldsymbol{h}}_c$ for the next motif prediction. We show the details of the weighted GNN in the Appendix. B.

Focal Motif Prediction: Before predicting the next motif, we first select a focal motif which the next motif attaches with. Two atom-wise MLPs are used as classifiers: protein atom classifier (for $t=1$) and molecular atom classifier (for $t\geq 2$). (1) At $t=1$, all the known context information is the protein pocket. The protein atom classifier takes the hidden representations of protein atoms as input, and predicts whether new ligand atoms can be generated within 4 Å. (2) For $t\geq 2$, the molecule atom classifier selects a focal atom from the ligand atoms generated in the previous $t-1$ steps. The motif that the focal atom belongs to is chosen as the focal motif. If no atom/motif is selected as focal, the generation process is completed.

Next Motif Prediction: Given the focal motif $\mathcal{M}_f$, the label of the next motif is predicted as:

$$P_m = \underset{\mathcal{M}\in V_{\mathcal{M}}}{\mathrm{softmax}}(\mathrm{MLP}^{\mathcal{M}}(\mathrm{e}(\mathcal{M}_f), \sum_{i\in\mathcal{M}_f}\boldsymbol{h}_i, \hat{\boldsymbol{h}}_c)\cdot\mathrm{e}(\mathcal{M})) \tag{8}$$

where P_m is the distribution over the motif vocabulary $V_{\mathcal{M}}$, $\mathrm{e}(\mathcal{M})$ denotes the motif embedding, $\sum_{i\in\mathcal{M}_f}\boldsymbol{h}_i$ is the sum of the atom embeddings in the focal motif, and $\hat{\boldsymbol{h}}_c$ is the enriched subpocket representation. We use a MLP to fuse the context information and use a dot product to score each motif. Since there is no focal motif at the first step ($t=1$), we regard *no motif* as a special motif type and also learn its embedding in training.

Motif Attachment Prediction: With the predicted motif, the next step is to attach the new motif to the generated molecule. Such a step is not deterministic since there are

potentially several attachment configurations (See Figure.2). Our goal here is to select the most appropriate attachment. Specifically, we enumerate different *valid* attachments and form a candidate set C. We employ GIN (Xu et al., 2019) to encode the candidate molecular graphs ($\mathrm{GIN}(\cdot)$) and the probability P_a of picking every molecule attachment is calculated as:

$$P_a = \underset{\mathcal{G}' \in C}{\mathrm{softmax}}(\mathrm{MLP}^a(\mathrm{GIN}(\mathcal{G}'), \hat{\boldsymbol{h}}_c)). \tag{9}$$

We merge atoms or bonds in the process of motif attachment. By pruning chemically invalid molecules and merging isomorphic graphs with RDkit (Bento et al., 2020), we have $|C| \approx 3$ on the CrossDocked dataset. Therefore, the attachment prediction is also very efficient.

Rotation Angle Prediciton: As the flexibility of molecular structures largely lie in the degree of rotatable bonds (Jing et al., 2022), we focus on predicting the rotation angles in DrugGPS. After attaching the new motif and obtaining the initial coordinates, we apply the encoder again to get the updated atom embeddings. Let X, Y denote the two end atoms of the rotatable bond (let Y denote the atom connecting the new motif). We predict the change of the torsion angle $\Delta\alpha$:

$$\Delta\alpha = \mathrm{MLP}^{\alpha}(\boldsymbol{h}_X, \boldsymbol{h}_Y, \boldsymbol{h}_{\mathcal{G}}) \mathrm{mod} 2\pi, \tag{10}$$

where $\boldsymbol{h}_X$ and $\boldsymbol{h}_Y$ indicate the embeddings of X and Y; $\boldsymbol{h}_{\mathcal{G}}$ denotes the embedding of the molecule, which is obtained with a sum pooling. $\Delta\alpha$ is also rotationally and translationally invariant since the prediction is based on the representations from the equivariant encoder. Finally, the coordinates of the atoms in the new motif are updated by rotating $\Delta\alpha$ around line XY. As for the first motif in the generation, we use a distance-based initialization for its coordinates as there is no reference ligand atoms. More details of the generation process are included in Appendix. B.

3.6. Model Training

In the training stage, the motifs of molecules are randomly masked and DrugGPS is trained to recover the masked ones. Specifically, for each pocket-ligand pair, we sample a mask ratio from the uniform distribution $U[0, 1]$ and mask the corresponding number of molecular motifs. The generation of motifs is in a breadth-first order where the root motif is set as the motif closest to the pocket. The atoms with valence bonds to the masked motifs are defined as focal atom candidates. If all molecular atoms are masked, the focal atoms are defined as protein atoms that have masked ligand atoms within 4 Å.

For the focal atom/motif prediction, we use a binary cross entropy loss $\mathcal{L}_{focal}$ for the classification of focal atoms. For the motif type and attachment prediction, we use cross entropy losses for the classification, denoted as $\mathcal{L}_{motif}$ and $\mathcal{L}_{attach}$. As for the torsion angle prediction, we fit angles with von Mises distributions with $\mathcal{L}_{\alpha}$ following (Senior et al., 2020). For the distance-based initialization, we minimize an MSE loss $\mathcal{L}_d$ with respect to the pairwise distances. In the training process, we aim to minimize the sum of the above loss functions:

$$\mathcal{L} = \mathcal{L}_{focal} + \mathcal{L}_{motif} + \mathcal{L}_{attach} + \mathcal{L}_{\alpha} + \mathcal{L}_d. \tag{11}$$

4. Experiments

4.1. Experimental Settings

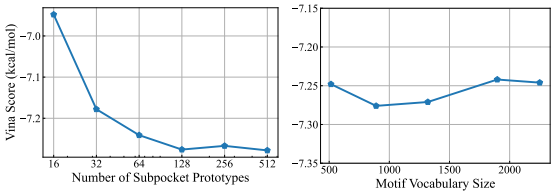
Dataset: Following previous works (Peng et al., 2022; Liu et al., 2022), we use the CrossDocked dataset (Francoeur et al., 2020) which contains 22.5 million protein-molecule pairs. We filter out data points whose binding pose RMSD is greater than 1 Å, leading to a refined subset with around 180k data points. We consider two data splitting schemes to test the generalization abilities of models: (1) **Sequence-based Clustered Split (SCS)** uses mmseqs2 (Steinegger & Söding, 2017) to cluster data at 30% sequence identity and (2) **Pocket-based Clustered Split (PCS)** uses PocketMatch (Yeturu & Chandra, 2008) to cluster data with a similarity threshold of 0.75. Specifically, PocketMatch represents pockets in a frame-invariant manner and compares pairs of sites based on the alignment of sorted distance sequences and the residue types. For both data splits, we randomly draw 100,000 protein-ligand pairs for training and 100 proteins from remaining clusters for testing. Therefore, the training and the testing set contain sequentially or structurally different pockets. For evaluation, 100 molecules are randomly sampled for each protein pocket in the test set. More details of the dataset split are shown in Appendix. C.

Baselines: DrugGPS is compared with five state-of-the-art baseline methods including LiGAN (Ragoza et al., 2022), AR (Luo et al., 2021), GraphBP (Liu et al., 2022), Pocket2Mol (Peng et al., 2022), and our previous work FLAG (Zhang et al., 2023b).

Model: The number of layers for the atom and residue-level encoder is set as 6 and 3 respectively. K_a and K_r are set as 32 and 8 respectively. The number of attention head M is set as 4; The number of weighted GNN layers is 2. The hidden dimension d is set as 256. The threshold τ in motif extraction is set to 100 and $|V_{\mathcal{M}}| = 890$ in the default setting. The number of subpocket prototypes is set as 128 in the default setting. We update the prototypes every 200 iterations with momentum $\gamma = 0.9$. In the global information fusion, the input subpocket embedding is linked with the top 4 closest prototypes measured by cosine similarity. The model is trained with the Adam optimizer with a learning rate of 0.0001. The batch size is 4 and the number of total training iterations is 1,000,000. The code of DrugGPS is at `https://github.com/zaixizhang/DrugGPS_ICML23`.

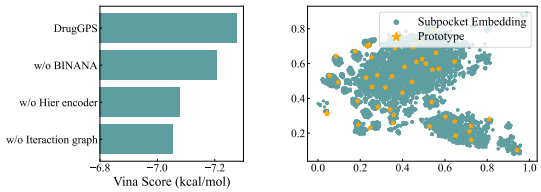
Table 1. Comparing the generated molecules' properties by different methods under the **pocket-based clustered split**. We report the means and standard deviations. The properties of the test set are shown for reference and the best results are bolded.

Methods	Vina Score (kcal/mol, ↓)	High Affinity(↑)	QED (↑)	SA (↑)	LogP	Lip. (↑)	Sim. Train (↓)	Div. (↑)	Time (↓)
Testset	-7.145±2.24	-	0.465±0.25	0.736±0.12	0.941±2.25	4.468±1.54	-	-	-
LiGAN	-6.032±1.89	0.194±0.26	0.365±0.27	0.615±0.20	-0.015±2.48	4.002±0.92	0.410±0.22	0.667±0.15	1819.8±560.7
AR	-6.114±1.66	0.235±0.23	0.483±0.18	0.662±0.19	0.210±1.76	4.688±0.45	0.394±0.21	0.650±0.13	15986.4±9851.0
GraphBP	-6.745±1.82	0.378±0.29	0.455±0.19	0.710±0.18	0.457±2.10	4.783±0.34	0.378±0.26	0.659±0.12	1162.8±438.5
Pocket2Mol	-6.869±2.19	0.413±0.23	0.524±0.24	0.726±0.21	0.830±2.17	4.892±0.22	0.364±0.19	0.695±0.17	2827.3±1456.8
FLAG	-6.956±1.92	0.445±0.22	0.552±0.20	0.737±0.19	0.745±2.09	4.904±0.14	0.388±0.18	**0.704±0.18**	1289.1±378.0
DrugGPS	**-7.276±2.14**	**0.565±0.23**	**0.613±0.22**	**0.743±0.18**	0.913±2.15	**4.917±0.12**	**0.360±0.21**	0.681±0.15	**1007.8±554.1**



(a) (b)

Figure 4. Hyperparameter analysis with respect to the (a) number of subpocket prototypes N and (b) motif vocabulary size $|V_{\mathcal{M}}|$.



(a) (b)

Figure 5. (a) Ablation studies (b) t-SNE visualization of 10000 randomly sampled subpocket embeddings and 56 prototypes.

Metrics: We choose widely-used metrics in previous works (Peng et al., 2022; Liu et al., 2022) to evaluate the sampled molecules: (1) **Vina Score** calculates the binding affinity between the generated molecules and the protein pockets with QVina (Trott & Olson, 2010; Alhossary et al., 2015); (2) **High Affinity** measures the percentage of pockets that have generated molecules with higher affinity than the references in the test set; (3) **QED** measures how likely a molecule is a potential drug candidate; (4) **Synthesizability (SA)** represents the difficulty of drug synthesis (normalized between 0 and 1 and higher values indicate easier synthesis); (5) **LogP** is the octanol-water partition coefficient and good drug candidates have LogP ranging from -0.4 to 5.6 (Ghose et al., 1999); (6) **Lipinski (Lip.)** calculates how many rules the molecule obeys the Lipinski's rule of five (Lipinski et al., 2012); (7) **Sim. Train** represents the Tanimoto similarity (Bajusz et al., 2015) with the most similar molecules in the training set; (8) **Diversity (Div.)** measures the diversity of generated molecules for a binding pocket. (9) **Time** records the time to generate 100 valid molecules for a pocket. All the generated molecules by different methods are optimized with universal force fields (Rappé et al., 1992).

4.2. Experimental Results

We show the generated molecules' properties in Table. 1 (PCS) and Table. 4 (SCS) in Appendix. D. Generally, we find PCS more challenging: the average vina scores of the generated molecules drop a lot for baseline methods from SCS to PCS (e.g., -7.288 to -6.869 for Pocket2Mol). This is not surprising since proteins with overall low sequence similarities may still have similar pockets (Eguida & Rognan, 2022) in SCS, which helps generate high-affinity molecules for pockets in the test set. Table. 7 also shows that PCS results in larger train-test performance gap. We use pocket-based clustered split as the default setting in the rest of this paper to test the generalization abilities. Thanks to the constructed global interaction graph and prototype-augmented motif generation scheme, DrugGPS can still generate molecules with high affinity in PCS and does not drop much compared with SCS (-7.276 vs. -7.345). Moreover, DrugGPS also manages to generate diverse molecules with high drug-likeliness and synthesizability, and with low similarities to the molecules in the training dataset. Note that 100% of the molecules are valid because DrugGPS explicitly filter out invalid candidates in the attachment selection step (A molecule is valid if it can be sanitized by RDkit (Bento et al., 2020)). Finally, we compare the computational efficiency for molecule generation. With the motif-based generation scheme, DrugGPS can shorten the generation steps and is more efficient than baseline methods. More results and discussions are included in Appendix. D.

In Fig.5(b), we also use t-SNE (Van der Maaten & Hinton, 2008) to visualize the sampled subpocket embeddings and their prototypes. We can observe that the prototypes can mostly occupy the centers of subpocket embeddings, which verifies the effectiveness of the learned subpocket prototype.

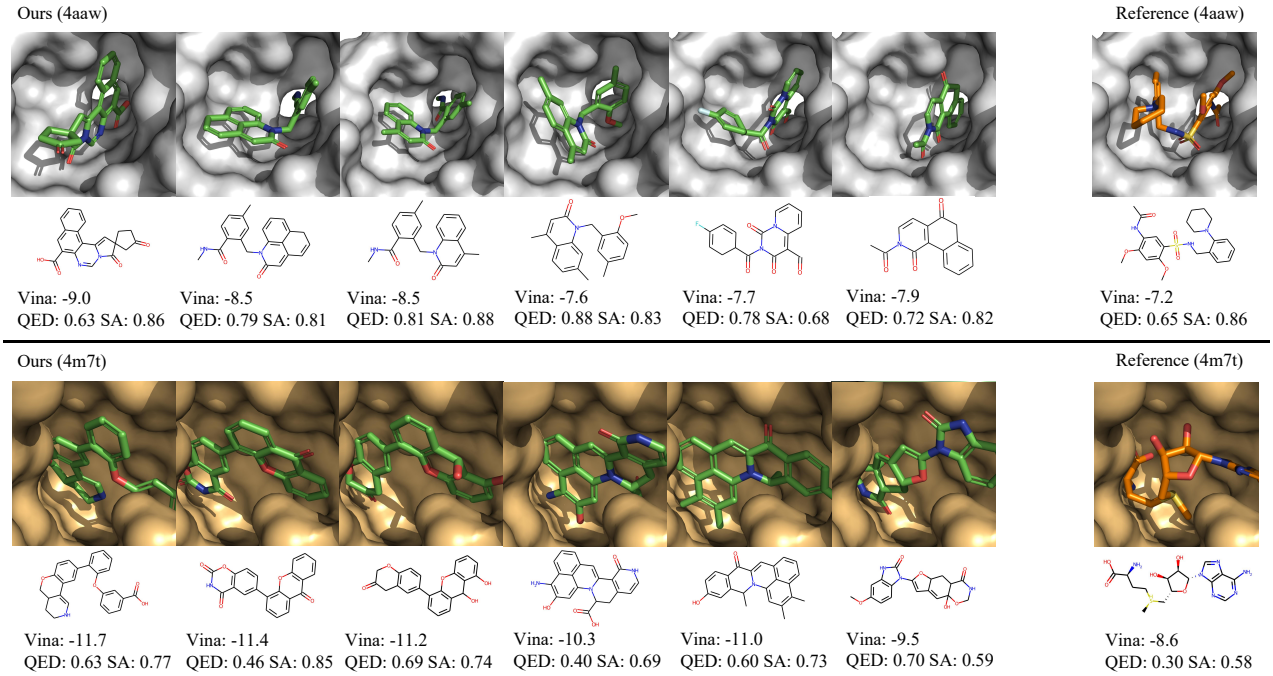


Figure 6. Examples of the generated molecules with higher binding affinities than the references. We report the vina scores, QED, and SA scores for each molecule. A lower Vina score indicates higher binding affinity.

4.3. Case Studies

In Figure 6, we further provide several generated molecule examples that have higher binding affinities (lower vina scores) than their corresponding reference molecules. Firstly, the generated molecules have novel structures that are different from the reference molecules. This implies that DrugGPS can generate novel and diverse structures. Furthermore, the generated molecules also exhibit high QED and SA scores, showing their potential to be good drug candidates. Finally, the generated molecules contain realistic substructures (e.g., benzene ring), which should attribute to the motif-based generation scheme. In DrugGPS, we focus on the prediction of rotatable bonds and use chemical tools to help determine the motif structures. We further provide quantitive substructure analysis in Appendix. D.

4.4. Hyperparameter Analysis & Ablation Studies

In Fig. 4, we explore the influence of two key hyperparameters: the number of subpocket prototypes N and the motif vocabulary size $|V_{\mathcal{M}}|$. We can control hyperparameter τ to control $|V_{\mathcal{M}}|$. With the increase of N, the average vina scores of the generated molecules decrease (higher binding affinity) and gradually stabilize. This may be because a sufficient number of subpocket prototypes is required to represent the global distributions of subpocket embeddings. When it comes to the motif vocabulary size, we find an appropriate size of $|V_{\mathcal{M}}|$ is beneficial to generate high-quality ligand molecules. Too small motif vocabulary may limits generating large complex molecules while too large vocabulary may inhibit learning good motif representations.

We further perform ablation studies to show the effectiveness of different modules in DrugGPS (Fig. 5(a)). Specifically, we remove the binding analysis tool BINANA, the residue-level encoder, and the global interaction graph in DrugGPS respectively as "w/o BINANA", "w/o Hier encoder", and "w/o Interaction graph". We find that removing these modules, especially the hierarchical encoder and global interaction graph, indeed degrades the performance of DrugGPS. This verifies that the necessity to capture the hierarchical context information and the effectiveness of the global interaction graph for generalizable ligand generation.

5. Conclusion

In this paper, we propose DrugGPS, a generalizable structure-based drug design method. Inspired by the biochemical prior of subpockets, we propose a subpocket prototype-augmented ligand molecule generation scheme that levrages the global interaction knowledge of the whole dataset. A hierarchical 3D graph transformer is also proposed to encode both the atom-level and residue-level information. Experiments show that our model consistently outperforms baselines in generating realistic drug candidates with high affinities in challenging out-of-distribution settings. Future works may include further leveraging inter-

pretable machine learning techniques (Zhang et al., 2022a), unifying protein and molecule pre-training (Zhang et al., 2021), and extending our framework to other domains such as protein design (Gao et al., 2023).

6. Acknowledgements

This research was partially supported by a grant from the National Natural Science Foundation of China (Grant No. 61922073).

References

Alhossary, A., Handoko, S. D., Mu, Y., and Kwoh, C.-K. Fast, accurate, and reliable molecular docking with quickvina 2. *Bioinformatics*, 31(13):2214–2216, 2015.

Anderson, A. C. The process of structure-based drug design. *Chemistry & biology*, 10(9):787–797, 2003.

Bajusz, D., Rácz, A., and Héberger, K. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of cheminformatics*, 7(1):1–13, 2015.

Bento, A. P., Hersey, A., Félix, E., Landrum, G., Gaulton, A., Atkinson, F., Bellis, L. J., De Veij, M., and Leach, A. R. An open source chemical structure curation pipeline using rdkit. *Journal of Cheminformatics*, 12(1):1–16, 2020.

Blundell, T. L. Structure-based drug design. *Nature*, 384 (6604 Suppl):23–26, 1996.

Crippen, G. M. and Havel, T. F. Stable calculation of coordinates from distance information. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 34(2):282–284, 1978.

Deng, C., Litany, O., Duan, Y., Poulenard, A., Tagliasacchi, A., and Guibas, L. J. Vector neurons: A general framework for so (3)-equivariant networks. In *CVPR*, pp. 12200–12209, 2021.

Eguida, M. and Rognan, D. Estimating the similarity between protein pockets. *International Journal of Molecular Sciences*, 23(20):12462, 2022.

Eguida, M., Schmitt-Valencia, C., Hibert, M., Villa, P., and Rognan, D. Target-focused library design by pocket-applied computer vision and fragment deep generative linking. *Journal of Medicinal Chemistry*, 65(20):13771–13783, 2022.

Ferreira, L. G., Dos Santos, R. N., Oliva, G., and Andricopulo, A. D. Molecular docking and structure-based drug design strategies. *Molecules*, 20(7):13384–13421, 2015.

Francoeur, P. G., Masuda, T., Sunseri, J., Jia, A., Iovanisci, R. B., Snyder, I., and Koes, D. R. 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.

Gao, Z., Tan, C., and Li, S. Z. Pifold: Toward effective and efficient protein inverse folding. *ICLR*, 2023.

Gebauer, N., Gastegger, M., and Schütt, K. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. *NeurIPS*, 32, 2019.

Ghose, A. K., Viswanadhan, V. N., and Wendoloski, J. J. A knowledge-based approach in designing combinatorial or medicinal chemistry libraries for drug discovery. 1. a qualitative and quantitative characterization of known drug databases. *Journal of combinatorial chemistry*, 1(1): 55–68, 1999.

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

Green, H., Koes, D. R., and Durrant, J. D. Deepfrag: a deep convolutional neural network for fragment-based lead optimization. *Chemical Science*, 12(23):8036–8047, 2021.

Hansson, T., Oostenbrink, C., and van Gunsteren, W. Molecular dynamics simulations. *Current opinion in structural biology*, 12(2):190–196, 2002.

Hu, L., Benson, M. L., Smith, R. D., Lerner, M. G., and Carlson, H. A. Binding moad (mother of all databases). *Proteins: Structure, Function, and Bioinformatics*, 60(3): 333–340, 2005.

Huynh, D. Q. Metrics for 3d rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, 2009.

Ingraham, J., Garg, V., Barzilay, R., and Jaakkola, T. Generative models for graph-based protein design. *Advances in neural information processing systems*, 32, 2019.

Ji, Y., Zhang, L., Wu, J., Wu, B., Huang, L.-K., Xu, T., Rong, Y., Li, L., Ren, J., Xue, D., Lai, H., Xu, S., Feng, J., Liu, W., Luo, P., Zhou, S., Huang, J., Zhao, P., and Bian, Y. DrugOOD: Out-of-Distribution (OOD) Dataset Curator and Benchmark for AI-aided Drug Discovery – A Focus on Affinity Prediction Problems with Noise Annotations. *arXiv e-prints*, art. arXiv:2201.09637, January 2022.

Jin, W., Barzilay, R., and Jaakkola, T. Junction tree variational autoencoder for molecular graph generation. In *ICML*, pp. 2323–2332. PMLR, 2018.

Jin, W., Barzilay, R., and Jaakkola, T. Hierarchical generation of molecular graphs using structural motifs. In *ICML*, pp. 4839–4848. PMLR, 2020a.

Jin, W., Barzilay, R., and Jaakkola, T. Multi-objective molecule generation using interpretable substructures. In *ICML*, pp. 4849–4859. PMLR, 2020b.

Jin, W., Barzilay, R., and Jaakkola, T. Antibody-antigen docking and design via hierarchical structure refinement. In *ICML*, pp. 10217–10227. PMLR, 2022.

Jing, B., Eismann, S., Soni, P. N., and Dror, R. O. Equivariant graph neural networks for 3d macromolecular structure. *ICML*, 2021.

Jing, B., Corso, G., Chang, J., Barzilay, R., and Jaakkola, T. Torsional diffusion for molecular conformer generation. *NeurIPS*, 2022.

Kabsch, W. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976.

Kalliokoski, T., Olsson, T. S., and Vulpetti, A. Subpocket analysis method for fragment-based drug discovery. *Journal of chemical information and modeling*, 53(1):131–141, 2013.

Karplus, M. and McCammon, J. A. Molecular dynamics simulations of biomolecules. *Nature structural biology*, 9(9):646–652, 2002.

Lee, S., Jo, J., and Hwang, S. J. Exploring chemical space with score-based out-of-distribution generation. *arXiv preprint arXiv:2206.07632*, 2022.

Lipinski, C. A., Lombardo, F., Dominy, B. W., and Feeney, P. J. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced drug delivery reviews*, 64:4–17, 2012.

Liu, M., Luo, Y., Uchino, K., Maruhashi, K., and Ji, S. Generating 3d molecules for target protein binding. *ICML*, 2022.

Long, S., Zhou, Y., Dai, X., and Zhou, H. Zero-shot 3d drug design by sketching and generating. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=H_xAgRM7I5N.

Luo, S., Guan, J., Ma, J., and Peng, J. A 3d generative model for structure-based drug design. *NeurIPS*, 34:6229–6239, 2021.

Luo, Y. and Ji, S. An autoregressive flow model for 3d molecular geometry generation from scratch. In *ICLR*, 2021.

Marcou, G. and Rognan, D. Optimizing fragment and scaffold docking by use of molecular interaction fingerprints. *Journal of chemical information and modeling*, 47(1):195–207, 2007.

Min, E., Chen, R., Bian, Y., Xu, T., Zhao, K., Huang, W., Zhao, P., Huang, J., Ananiadou, S., and Rong, Y. Transformer for graphs: An overview from architecture perspective. *arXiv preprint arXiv:2202.08455*, 2022.

Morris, G. M. and Lim-Wilby, M. Molecular docking. In *Molecular modeling of proteins*, pp. 365–382. Springer, 2008.

Pagadala, N. S., Syed, K., and Tuszynski, J. Software for molecular docking: a review. *Biophysical reviews*, 9(2):91–102, 2017.

Peng, X., Luo, S., Guan, J., Xie, Q., Peng, J., and Ma, J. Pocket2mol: Efficient molecular sampling based on 3d protein pockets. *ICML*, 2022.

Powers, A., Yu, H., Suriana, P., and Dror, R. Fragment-based ligand generation guided by geometric deep learning on protein-ligand structure. *bioRxiv*, 2022.

Ragoza, M., Masuda, T., and Koes, D. R. Generating 3d molecules conditional on receptor binding sites with deep generative models. *Chemical science*, 13(9):2701–2713, 2022.

Rappé, A. K., Casewit, C. J., Colwell, K., Goddard III, W. A., and Skiff, W. M. Uff, a full periodic table force field for molecular mechanics and molecular dynamics simulations. *Journal of the American chemical society*, 114(25):10024–10035, 1992.

Rezende, D. and Mohamed, S. Variational inference with normalizing flows. In *ICML*, pp. 1530–1538. PMLR, 2015.

Schlichtkrull, M., Kipf, T. N., Bloem, P., Berg, R. v. d., Titov, I., and Welling, M. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pp. 593–607. Springer, 2018.

Schneuing, A., Du, Y., Harris, C., Jamasb, A., Igashov, I., Du, W., Blundell, T., Lió, P., Gomes, C., Welling, M., et al. Structure-based drug design with equivariant diffusion models. *arXiv preprint arXiv:2210.13695*, 2022.

Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Žídek, A., Nelson, A. W., Bridgland, A., et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792): 706–710, 2020.

Steinegger, M. and Söding, J. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.

Stoker, H. S. *General, organic, and biological chemistry*. Cengage Learning, 2015.

Trott, O. and Olson, A. J. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Verlinde, C. L. and Hol, W. G. Structure-based drug design: progress, results and challenges. *Structure*, 2(7):577–587, 1994.

Wang, Y., Abuduweili, A., and Dou, D. Property-aware adaptive relation networks for molecular property prediction. *CoRR*, abs/2107.07994, 2021. URL `https://arxiv.org/abs/2107.07994`.

Xie, Y., Shi, C., Zhou, H., Yang, Y., Zhang, W., Yu, Y., and Li, L. Mars: Markov molecular sampling for multi-objective drug discovery. *ICLR*, 2021.

Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L., and Liu, T. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pp. 10524–10533. PMLR, 2020.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? *ICLR*, 2019.

Yang, N., Zeng, K., Wu, Q., Jia, X., and Yan, J. Learning substructure invariance for out-of-distribution molecular representations. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=2nWUNTnFijm`.

Yang, Y., Ouyang, S., Dang, M., Zheng, M., Li, L., and Zhou, H. Knowledge guided geometric editing for unsupervised drug design.

Yeturu, K. and Chandra, N. Pocketmatch: a new algorithm to compare binding sites in protein structures. *BMC bioinformatics*, 9(1):1–17, 2008.

Young, J., Garikipati, N., and Durrant, J. D. Binana 2: Characterizing receptor/ligand interactions in python and javascript. *Journal of chemical information and modeling*, 62(4):753–760, 2022.

Zhang, L., Lin, D., Sun, X., Curth, U., Drosten, C., Sauerhering, L., Becker, S., Rox, K., and Hilgenfeld, R. Crystal structure of sars-cov-2 main protease provides a basis for design of improved α-ketoamide inhibitors. *Science*, 368 (6489):409–412, 2020.

Zhang, Z., Liu, Q., Wang, H., Lu, C., and Lee, C.-K. Motif-based graph self-supervised learning for molecular property prediction. *Advances in Neural Information Processing Systems*, 34:15870–15882, 2021.

Zhang, Z., Liu, Q., Wang, H., Lu, C., and Lee, C. Protgnn: Towards self-explaining graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 9127–9135, 2022a.

Zhang, Z., Liu, Q., Zhang, S., Hsieh, C.-Y., Shi, L., and Lee, C.-K. Graph self-supervised learning for optoelectronic properties of organic semiconductors. *ICML AI for Science workshop*, 2022b.

Zhang, Z., Liu, Q., Lee, C.-K., Kim, C.-Y., and Chen, E. An equivariant generative framework for molecular graph-structure co-design. *bioRxiv*, pp. 2023–04, 2023a.

Zhang, Z., Min, Y., Zheng, S., and Liu, Q. Molecule generation for target protein binding with structural motifs. In *The Eleventh International Conference on Learning Representations*, 2023b.

A. More Details of Hierarchical Graph Transformer

Our hierarchical encoder includes the atom-level encoder and the residue-level encoder. The powerful 3D Graph Transformer is used as the model backbone. Here we first show more details of the graph transformer architecture. Then wen show the residue-level node and edge features following previous works (Ingraham et al., 2019; Jin et al., 2022).

Graph transformer architecture. The atom/residue-level encoder contains L graph transformer layers. Let $\boldsymbol{h}^{(l)}$ be the set of node representations at the l-th layer. In each graph transformer layer, there are a multi-head self-attention (MHA) and a feed-forward block (FFN). The layer normalization (LN) is applied before the two blocks (Xiong et al., 2020). The details of MHA has been shown in Sec.3.3 and the graph transformer layer is formally characterized as:

$$\boldsymbol{h}'^{(l-1)} = \mathrm{MHA}(\mathrm{LN}(\boldsymbol{h}^{(l-1)})) + \boldsymbol{h}^{(l-1)} \tag{12}$$

$$\boldsymbol{h}^{(l)} = \mathrm{FFN}(\mathrm{LN}(\boldsymbol{h}'^{(l-1)})) + \boldsymbol{h}'^{(l-1)}, \quad (0 \leq l < L). \tag{13}$$

Residue-level node features. The residue-level encoder only keeps the C_α atoms to represent residues and constructs a K_r nearest neighbor graph at the residue level. Each residue node is represented by six features: polarity $f_p \in \{0, 1\}$, hydropathy $f_h \in [-4.5, 4.5]$, volume $f_v \in [60.1, 227.8]$, charge $f_c \in \{-1, 0, 1\}$, and whether it is a hydrogen bond donor $f_d \in \{0, 1\}$ or acceptor $f_a \in \{0, 1\}$. We expand hydropathy and volume features into radial basis with interval sizes 0.1 and 10, respectively. Overall, the dimension of the residue-level feature vector $\boldsymbol{f}_i$ is 112.

Residue-level edge features. For each i-th residue, we let x_i denote the coordinate of its C_α and define its local coordinate frame $\boldsymbol{O}_i = [\boldsymbol{c}_i, \boldsymbol{n}_i, \boldsymbol{c}_i \times \boldsymbol{n}_i]$ as:

$$\boldsymbol{u}_i = \frac{\boldsymbol{x}_i - \boldsymbol{x}_{i-1}}{\|\boldsymbol{x}_i - \boldsymbol{x}_{i-1}\|}, \quad \boldsymbol{c}_i = \frac{\boldsymbol{u}_i - \boldsymbol{u}_{i+1}}{\|\boldsymbol{u}_i - \boldsymbol{u}_{i+1}\|}, \quad \boldsymbol{n}_i = \frac{\boldsymbol{u}_i \times \boldsymbol{u}_{i+1}}{\|\boldsymbol{u}_i \times \boldsymbol{u}_{i+1}\|}. \tag{14}$$

Based on the local frame, the edge features between residues i and j can be computed as:

$$\boldsymbol{e}_{ij}^{res} = \mathrm{Concat}\left(E_{\mathrm{pos}}(i-j), \quad \mathrm{RBF}(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|), \quad \boldsymbol{O}_i^\top \frac{\boldsymbol{x}_j - \boldsymbol{x}_i}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}, \quad \boldsymbol{q}(\boldsymbol{O}_i^\top \boldsymbol{O}_j)\right). \tag{15}$$

The edge feature $\boldsymbol{e}_{ij}^{res}$ contains four parts. The positional encoding $E_{\mathrm{pos}}(i-j)$ encodes the relative sequence distance between two residues. The second term $\mathrm{RBF}(\cdot)$ is a distance encoding with radial basis functions. The third term is a direction encoding corresponding to the relative direction of $\boldsymbol{x}_j$ in the local frame of i-th residue. The last term $\boldsymbol{q}(\boldsymbol{O}_i^\top \boldsymbol{O}_j)$ is the orientation encoding of the quaternion representation $\boldsymbol{q}(\cdot)$ of the spatial rotation matrix $\boldsymbol{O}_i^\top \boldsymbol{O}_j$ (Huynh, 2009). Overall, the dimension of the residue-level edge feature $\boldsymbol{e}_{ij}^{res}$ is 39.

B. More Details of Prototype-augmented Motif Generation

Global interaction graph construction. The number of subpocket prototypes is set as 128 in the default setting. For the stability of optimization, the encoder first goes through a warm-up period with 50,000 iterations. After the warm-up, the prototypes are first initialized with K-Means and further updated every 200 iterations with momentum $\gamma = 0.9$. The edge weights W_{ij} are updated in the interval of 10,000 iterations. DrugGPS is validated every 10,000 iterations and we take the checkpoint with the lowest validation loss for ligand generation. For the input subpocket, We add edges between the subpocket embedding with 4 most similar subpocket prototypes measured by cosine similarity.

Global information fusion with Weighted GNN. To enrich the subpocket representation with the global interaction information, we use a weighted GNN to propagate information on the constructed interaction graph. The subpocket prototypes and motif embeddings are firstly converted to the same hidden space with learnable transformation matrices. We use a weighted GNN adapted from GIN (Xu et al., 2019) for information propagation and aggregation:

$$x_v^l = \mathrm{MLP}(x_v^{l-1} + \frac{1}{\sum_{u \in \mathcal{N}(v)} W_{uv}} \sum_{u \in \mathcal{N}(v)} W_{uv} x_u^{l-1}), \tag{16}$$

where x_v^l denotes node v's embedding at the l-th layer and $\mathcal{N}(v)$ denotes its neighbors. W_{uv} is the edge weight calculated in Equation. 7. Finally, we take the input subpocket embedding at the final layer as $\hat{\boldsymbol{h}}_c$. In prototype-augmented motif generation, $\hat{\boldsymbol{h}}_c$ is leveraged in Equation. 8 for next motif prediction. It not only encodes the residue-level context information,

but capture the similar subpocket prototype-molecular motif interaction as well. Therefore, the next motif prediction classifier can learn to give motifs with high binding affinity higher scores.

Distance-based initialization. To decide the 3D positions of the generate motifs, it is challenging for the first motif as there is no reference ligand atoms. Following (Jin et al., 2022), we use a distance-based initialization strategy to determine the position of the first motif, which is more accurate and stable than random initialization. Specifically, a distance matrix $\boldsymbol{D} \in \mathbb{R}^{(n'+m')\times(n'+m')}$ is set as:

$$\boldsymbol{D}_{i,j} = \begin{cases} \|\boldsymbol{r}_i - \boldsymbol{r}_j\| & i,j \leq n' \\ \mathrm{MLP}^d(\boldsymbol{h}_i^{(0)}, \boldsymbol{h}_j^{(0)}) & i \leq n', j > n' \\ \|\boldsymbol{r}_i - \boldsymbol{r}_j\| & i,j > n', \end{cases} \tag{17}$$

where n' and m' denote the number of sampled protein atoms for reference and the number of atoms in the first molecular motif. $\boldsymbol{r}_i$ is the 3D coordinate of the atom. The distances within the protein atoms and motif atoms can be directly calculated. For the distances between molecular and protein atoms, we use MLP^d for prediction with the pairwise atom attributes as the input. With the distance matrix $\boldsymbol{D}$, we can obtain the coordinates of atoms by eigenvalue decomposition of its Gram matrix (Crippen & Havel, 1978):

$$\tilde{\boldsymbol{D}}_{i,j} = 0.5(\boldsymbol{D}_{i,1}^2 + \boldsymbol{D}_{1,j}^2 - \boldsymbol{D}_{i,j}^2), \quad \tilde{\boldsymbol{D}} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^\top \tag{18}$$

where $\boldsymbol{S}$ is a diagonal matrix with eigenvalues in descending order. The coordinate of each atom $\boldsymbol{r}_i$ is calculated as:

$$\tilde{\boldsymbol{r}}_i = [\boldsymbol{X}_{i,1}, \boldsymbol{X}_{i,2}, \boldsymbol{X}_{i,3}], \quad \boldsymbol{X} = \boldsymbol{U}\sqrt{\boldsymbol{S}}. \tag{19}$$

Note that even though the predicted coordinates $\{\tilde{\boldsymbol{r}}_i\}$ retain the original distance $\boldsymbol{D}$, they are located in a different coordinate system. Therefore, we apply the Kabsch algorithm (Kabsch, 1976) to find a rigid body transformation $\{\boldsymbol{R}, \boldsymbol{t}\}$ that aligns the predicted protein coordinates $\{\tilde{\boldsymbol{r}}_1, \cdots, \tilde{\boldsymbol{r}}_{n'}\}$ with the reference coordinates $\{\boldsymbol{r}_1, \cdots, \boldsymbol{r}_{n'}\}$. Lastly, the coordinates of the first motif are calculated as:

$$\boldsymbol{r}_i = \boldsymbol{R}\tilde{\boldsymbol{r}}_i + \boldsymbol{t}, \ i > n'. \tag{20}$$

For generation steps with $t > 1$, the coordinates of the attached motifs are determined and aligned similarly with RDkit (Bento et al., 2020) and Kabsch algorithm (Kabsch, 1976).

Rotation matrix. In the generation of new motifs, if the focal motif is rotatable and the rotation angle $\Delta\alpha$ is known, we use the following rotation matrix $R_{3\times3}$ and the translation vector $t_{3\times1}$ to update the coordinates of the new motif. Let X, Y denote the two end atoms of the rotatable bond (Y denote the atom connecting the new motif) and $\boldsymbol{r}_X$ and $\boldsymbol{r}_Y$ be their coordinates. Let $\boldsymbol{n}$ denotes the normalized directional vector $\frac{\boldsymbol{r}_Y - \boldsymbol{r}_X}{\|\boldsymbol{r}_Y - \boldsymbol{r}_X\|}$ and n_x, n_y and n_z be its components along the x, y and z axis. Let x_0, y_0, and z_0 be the three components of $\boldsymbol{r}_X$. The rotation matrix and translation vector are:

$$R_{3\times3} = \begin{bmatrix} n_x^2K + cos(\Delta\alpha) & n_xn_yK - n_zsin(\Delta\alpha) & n_xn_zK + n_ysin(\Delta\alpha) \\ n_xn_yK + n_zsin(\Delta\alpha) & n_y^2K + cos(\Delta\alpha) & n_yn_zK - n_xsin(\Delta\alpha) \\ n_xn_zK - n_ysin(\Delta\alpha) & n_yn_xK + n_xsin(\Delta\alpha) & n_z^2K + cos(\Delta\alpha) \end{bmatrix}, \tag{21}$$

$$t_{3\times1} = \begin{bmatrix} (x_0 - n_xM)K + (n_zy_0 - n_yz_0)sin(\Delta\alpha) \\ (y_0 - n_yM)K + (n_xz_0 - n_zx_0)sin(\Delta\alpha) \\ (z_0 - n_zM)K + (n_yx_0 - n_xy_0)sin(\Delta\alpha) \end{bmatrix}. \tag{22}$$

Here, $K = 1 - cos(\Delta\alpha)$ and $M = n_xx_0 + n_yy_0 + n_zz_0$. The coordinates $\boldsymbol{r}_i$ in the motif are updated as:

$$\boldsymbol{r}_i' = R\boldsymbol{r}_i + t \tag{23}$$

C. More Details of Experimental Settings

More training and sampling details. All the experiments are conducted on a NVIDIA Tesla V100 GPU with 32G memory. It takes around 48 hours to train DrugGPS. The implementation of DrugGPS is based on our previous work FLAG[1]. To

[1] https://github.com/zaixizhang/FLAG

Table 2. Properties of the test set molecules and the generated molecules by different methods under the **sequence-based clustered split**. We report the means and standard deviations. This is the same split used in previous works (Peng et al., 2022; Luo & Ji, 2021) and we borrow part of the baseline results from (Peng et al., 2022). The best results are bolded.

Methods	Vina Score (kcal/mol, ↓)	High Affinity(↑)	QED (↑)	SA (↑)	LogP	Lip. (↑)	Sim. Train (↓)	Div. (↑)	Time (↓)
Testset	-7.158±2.10	-	0.484±0.21	0.732±0.14	0.947±2.65	4.367±1.14	-	-	-
LiGAN	-6.114±1.57	0.238±0.28	0.369±0.22	0.590±0.15	-0.140±2.73	4.027±1.38	0.460±0.18	0.654±0.12	-
AR	-6.215±1.54	0.267±0.31	0.502±0.17	0.675±0.14	0.257±2.01	4.787±0.50	0.409±0.19	0.742±0.09	19658.56±14704
GraphBP	-7.132±1.75	0.477±0.26	0.516±0.15	0.718±0.18	0.442±2.08	4.620±0.37	0.415±0.24	0.649±0.12	1238.7±493.0
Pocket2Mol	-7.288±2.53	0.542±0.32	0.563±0.16	**0.765±0.13**	1.586±1.82	4.902±0.42	0.376±0.22	0.688±0.14	2503.51±2207
DrugGPS	**-7.345±2.42**	**0.620±0.29**	**0.592±0.21**	0.728±0.23	1.134±2.26	**4.923±0.11**	**0.370±0.26**	**0.695±0.17**	**956.3±451.6**

implement the baselines including LiGAN [2], AR[3], GraphBP[4], and Pocket2Mol[5], we use the open-source codes following their default settings. DrugGPS and baseline methods are trained on the same data split for fair comparisons.

More details of dataset split We use two OOD data split in our work. (1) **Sequence-based Clustered Split** uses mmseqs2 (Steinegger & Söding, 2017) to cluster data at 30% sequence identity, which is a popular spliting scheme in previous works (Peng et al., 2022; Luo & Ji, 2021). However, such splitting scheme only considers sequence similarity while neglecting structural similarities, which is more important in structure-based drug design. (2) **Pocket-based Clustered Split** uses PocketMatch (Yeturu & Chandra, 2008) to cluster data with a similarity threshold of 0.75. The range of similarity score in PocketMatch is from 0 to 1 and higher scores indicate higher pocket similarities. PocketMatch compares binding pockets in a frame-invariant manner: each binding site is represented by 90 lists of sorted distances capturing the geometric and chemical properties of the pocket. The pocket pairs are then aligned based on distances and residue types to obtain similarity scores (Yeturu & Chandra, 2008). Specifically in pocket-based clustered split, a set of initial centroids are sampled with pairwise similarity scores less than 0.75. The remaining pockets are compared with these centroids and assigned to the group with the higheset similarity score (> 0.75). If there is no centroid that the pocket has an over 0.75 similarity score to, the pocket is selected as a new centroid.

The output of the clustering algorithms are a set of clusters such that: (a) all centroids of clusters have similarity $< T$ to each other, and (b) all members in a cluster have similarity $\geq T$ to the centroid. T is the predefined similarity threshold. For both data splits, we randomly draw 100,000 protein-ligand pairs for training. The validation and test dataset are drawn from remaining clusters. Therefore, the training and test set contains sequentially or structurally different protein pockets.

D. More Experimental Results

Ligand molecule generation under sequence-based clustered split. We include the results under sequence-based clustered split in Table. 4. This is the same split used in previous works (Peng et al., 2022; Luo & Ji, 2021) and we borrow part of the baseline results from (Peng et al., 2022). We can observe that DrugGPS can also overperform baseline models on generating drug-like molecules with higher affinity in the popular SCS setting.

Peformance gap between training and testing set. In Table. 7, we compare the Vina scores of the generated molecules on the training and testing dataset. For the training set, we randomly sample 100 proteins and generate 100 molecules for each target protein pocket similar to that of the test set. We compare the performance of our DrugGPS with two competitive baseline methods. We also show the average vina scores from the sampled training set and the test set for reference. The vina scores of the sampled training set and the test set are roughly the same.

Generally, we can observe that the pocket-based clustered split results in larger vina score gap and is more challenging. Compared with baselines, our DrugGPS has smaller gap, which indicates its better generalization ability.

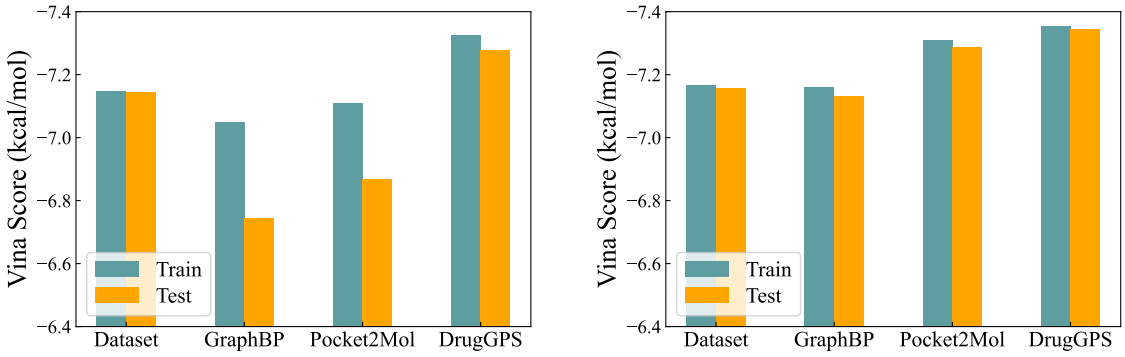
Qualitative substructure analysis. We further show qualitative substructure analysis by calculating the KL divergence of the bond angles and dihedral angles between the molecules in the test set and the generated molecules in Table. 3. We observe that DrugGPS can generate more realistic substructures than baseline methods (smaller angle distribution

[2] https://github.com/mattragoza/LiGAN

[3] https://github.com/luost26/3D-Generative-SBDD

[4] https://github.com/divelab/GraphBP

[5] https://github.com/pengxingang/Pocket2Mol



(a) Pocket-based clustered split

(b) Sequence-based clustered split

Figure 7. Comparaing the Vina scores of the generated molecules on the training and testing dataset. (a) shows the pocket-based clustered split and (b) shows the sequence-based clustered split. A lower Vina score indicates higher binding affinity.

Table 3. The KL divergence of the bond angles (upper part) and dihedral angles (lower part) between the molecules in the test set and the generated molecules by different methods. The lower letters represent the atoms in the aromatic rings. Pocket-based clustered split is used and the best results are bolded.

Angles	LiGAN	AR	GraphBP	Pocket2 Mol	FLAG	w/o BINANA	w/o Hier Encoder	w/o Interaction Graph	DrugGPS
CCC	7.16	2.31	2.09	0.93	0.59	0.71	0.86	0.75	**0.57**
CCO	7.68	2.19	1.98	0.94	0.65	**0.62**	0.67	0.64	0.63
CCN	7.71	2.58	2.46	0.65	**0.24**	0.27	0.33	0.26	0.25
CNC	6.39	3.28	1.77	0.72	0.54	0.52	0.79	0.53	**0.48**
OPO	6.08	3.73	3.71	**0.43**	0.57	0.48	0.74	0.49	0.46
CC=O	6.69	3.42	3.47	0.70	0.48	0.47	0.49	0.48	**0.45**
cccc	5.26	4.46	3.62	4.54	**0.45**	0.46	0.46	**0.45**	**0.45**
Cccc	3.46	5.05	2.28	2.53	1.41	1.40	1.71	1.34	**1.32**
CCCC	4.14	2.15	1.40	**0.69**	0.97	0.94	1.53	0.97	0.92
CCCO	3.50	2.20	1.25	1.14	0.96	0.91	1.32	0.88	**0.87**
OCCO	2.14	2.16	1.63	1.73	1.72	1.60	1.88	**1.52**	1.55
CC=CC	6.67	6.38	3.44	3.40	2.20	2.15	2.34	2.08	**2.06**

discrepancies between the generated molecules and the test set) due to its motif-based generation scheme. Moreover, we also compare DrugGPS with its variants and find the designed modules are also indispensable. Especially, the hierarchical encoder is quite important for DrugGPS to encode geometric information and predict rotation angles.

More Hyperparameter analysis. We show more hyperparameter analysis with respect to γ, K_p (number of subpocket prototypes to link with), and the subpocket size in Table. 8. We observe that DrugGPS is generally robust the choice of γ. The quality of generated molecules deteriorates slightly when too many subpocket prototypes are linked with in the global information fusion, which may lead to too much redundant information. Finally, we find choosing an appropriate subpocket size is important: some key residues may be neglected if the size is too small while some unrelated residue may be considered when the size is too large. In DrugGPS, we choose to sum up all the residue embeddings within 6 Å to obtain subpocket representations in the default setting for the best performance.

Failure Examples. Here we show some examples that the generated molecules have lower affinity (higher Vina scores) than the reference in Figure. 9. Therefore, these molecules have higher probabilities of not binding to the target pocket. The failure may be due to the following reasons: (1) Some generated molecules accidentally collide with the pocket, which is unrealistic in nature. (2) The auto-regressive generation scheme may limit its ability for overall optimization. We can observe that some generated molecules only occupy part of the pocket. In the future, we will tackle the aforementioned problems by designing penalty loss to prohibit colliding and explore generation schemes that facilitate overall optimization.

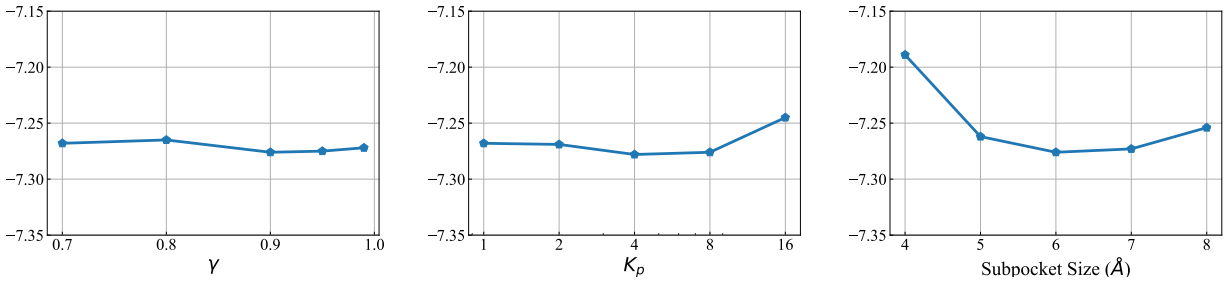


Figure 8. Hyperparameter analysis with respect to (a) γ, (b) K_p: subpocket prototypes to link with in the global knowledge fusion, and (c) the subpocket size. DrugGPS is generally robust to the choice of hyperparameters.

Structure-based drug design is a complicated conditional generation problem influenced by various factors (e.g., geometric and chemical constraints). It is common to have failure cases. However, even in the aforementioned failure situations, DrugGPS can still generate fragments aligning well with the global interaction information or the reference molecule. For example, for the first molecule in Figure 1, the first fragment generated by DrugGPS is quite similar to the reference molecule (marked with blue ovals). The fragment is also in the top-ranked fragments (listed on the leftmost) interacting with the corresponding subpocket prototype in the global interaction graph. Therefore, the prior that we impose into DrugGPS is generally helpful.

Global interaction helps ligand generation. To show the global interaction graph helps ligand generation, we include case studies showing the alignment between the generated ligand fragments and the fragment information derived from the global interaction graph in Figure. 10. Specifically, the encoded subpocked representation is fed into the global interaction graph and further mapped to the closest subpocket prototype. The molecular motifs with the largest edge weights (Equ. 7) with the subpocket prototype are shown here. In these cases, we observe that the fragment in the generated ligands (marked with blue ovals) are also within the top 6 interacted fragments from the global interaction graph (marked with black dashed boxes). These case studies support our claim that the global interaction graph helps fragment prediction and ligand generation.

The ratio of generated molecules with polycyclic structures. In the CrossDocked dataset, 55.42% of ligand molecules have polycyclic structures. Therefore the trained DrugGPS model on the CrossDocked dataset also tends to generate many polycyclic structures as shown in Figure. 6. However, the fragment-based generation scheme used in DrugGPS enables us to flexibly generate the molecules we want. For example, if we want more monocyclic compounds, we can reduce polycyclic motifs in the motif library (e.g., increase the threshold τ) and penalize merging rings at the generating stage (modified DrugGPS). In Figure. 11, we show the generated molecules with monocyclic structures by the modified DrugGPS.

Results on more datasets. We understand comprehensive evaluations on more datasets could further improve our work. Here we further evaluate our method on experimentally determined protein-ligand complexes found in Binding MOAD (Hu et al., 2005). Following (Schneuing et al., 2022), Binding MOAD is filtered and split based on the proteins' enzyme commission number, resulting in 40,354 protein-ligand pairs for training and 130 pairs for testing. In the following table, we compare DrugGPS with selected baselines including Pocket2Mol, FLAG, and DiffSBDD (Schneuing et al., 2022). For a fair comparison, the hyperparameters are finetuned for Pocket2Mol and FLAG. We use the recommended hyperparameters for Binding MOAD dataset in the original paper (Schneuing et al., 2022) for DiffSBDD. We can observe that DrugGPS can also achieve competitive performance on the Binding MOAD dataset.

Table 4. Results on the Binding MOAD dataset.

Methods	Vina Score (kcal/mol, ↓)	High Affinity(↑)	QED (↑)	SA (↑)	LogP	Lip. (↑)	Sim. Train (↓)	Div. (↑)	Time (↓)
Testset	-8.103±2.26	-	0.602±0.15	0.336±0.08	0.456±1.15	4.838±0.37	-	-	-
DiffSBDD	-6.234±1.76	0.127±0.11	0.529±0.17	0.324±0.10	0.112±1.20	4.847±0.33	0.369±0.14	0.717±0.09	**463.0±171.4**
Pocket2Mol	-7.690±2.47	0.358±0.22	0.596±0.14	**0.329±0.07**	0.697±1.66	4.750±0.28	0.375±0.15	**0.720±0.16**	2618.0±1503.4
FLAG	-7.724±2.09	0.364±0.20	0.605±0.16	0.317±0.14	0.719±1.43	4.762±0.23	0.382±0.16	0.695±0.12	1028.5±474.9
DrugGPS	**-8.215±2.29**	**0.522±0.21**	**0.623±0.18**	0.341±0.12	0.560±1.41	**4.859±0.24**	**0.363±0.16**	0.692±0.13	1007.8±554.1

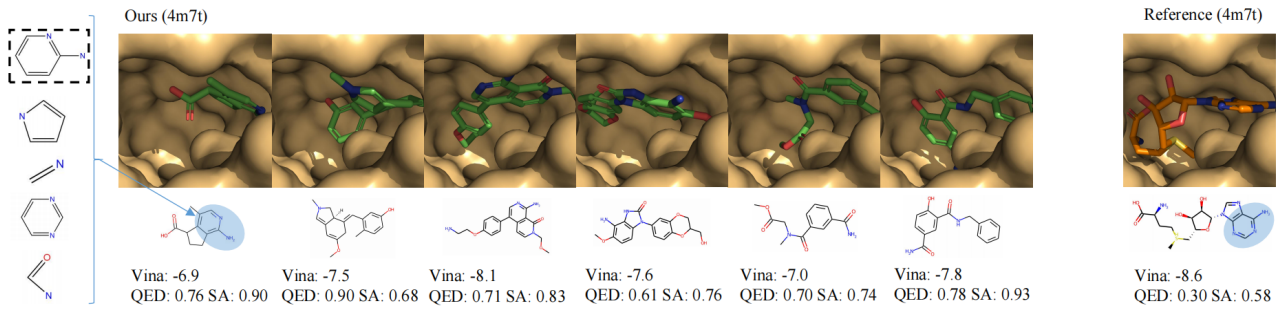


Figure 9. Generated examples with lower affinity than the reference ligands.

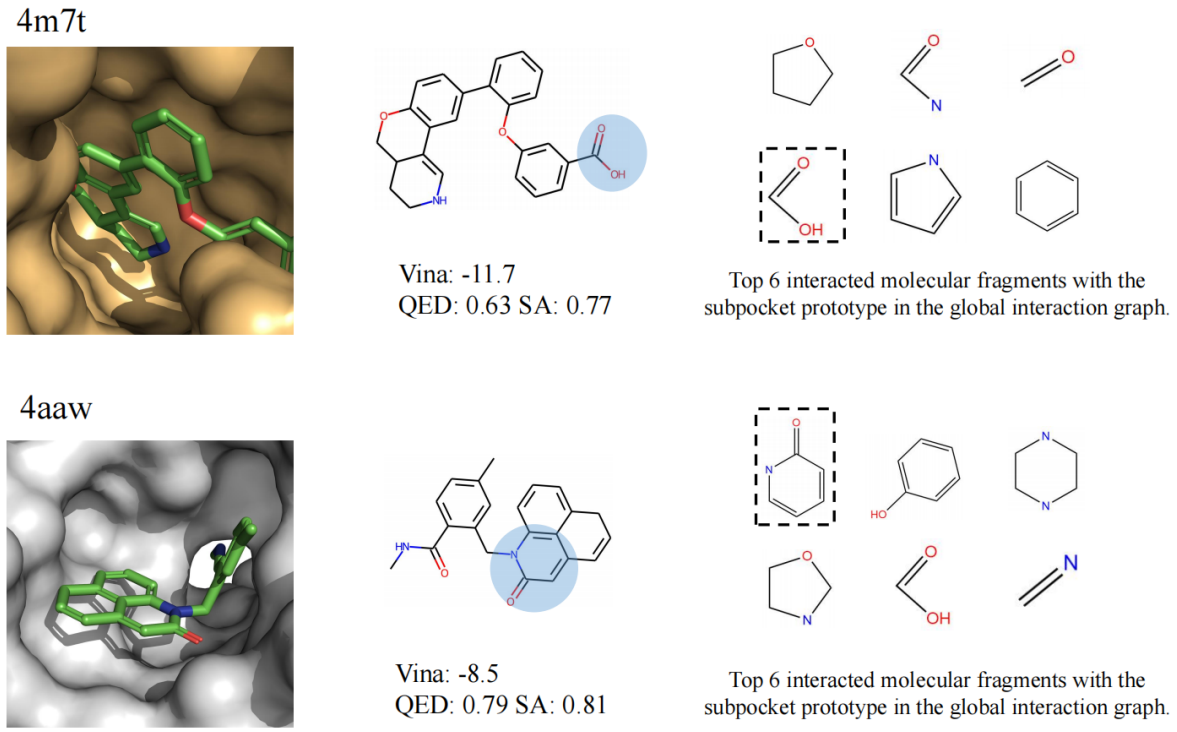


Figure 10. Global interaction helps ligand generation.

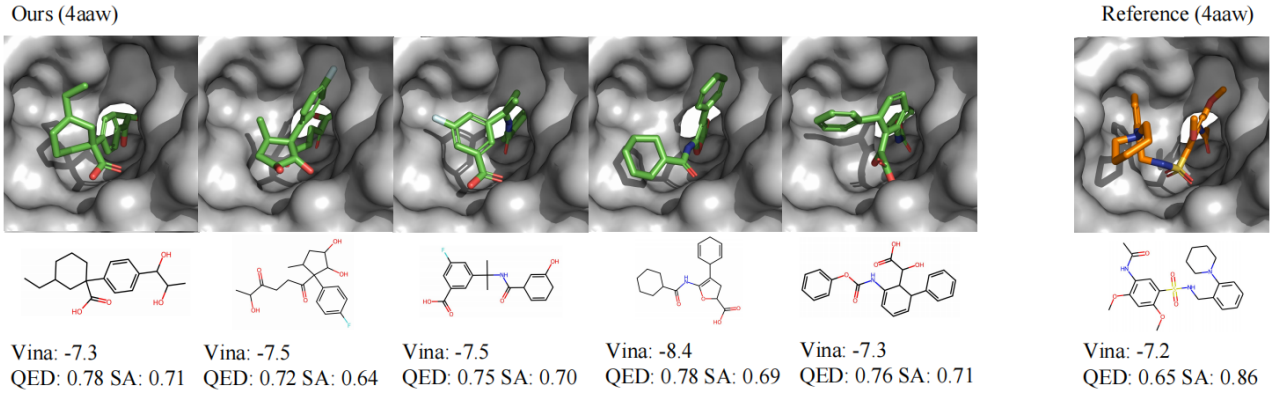


Figure 11. Generated molecules with monocyclic structures by the modified DrugGPS.