Synthon Contrastive Learning for Synthesizable 3D Molecule Generation

Nahyun Kim[*1] **Seul Lee**[*1] **Sung Ju Hwang**[1,2]
[1]KAIST, [2]DeepAuto
hyun0414@kaist.ac.kr, seull@nvidia.com, sjhwang82@kaist.ac.kr

Abstract

To consider synthesizability in structure-based drug design (SBDD), a recent approach has proposed to co-design synthetic pathways with molecular conformers of ligands. However, intermediate conformers are predicted without explicitly considering the synthons to be added later along the synthetic pathway. As a result, intermediate conformers may not adequately reflect upcoming substructures, leading to geometric mismatches and cumulative errors along the synthetic trajectory. To address this limitation, we propose SYnthon Contrastive LEarning (SYCLE), a synthesizable SBDD framework that trains the conformer predictor to align intermediate conformer representations with compatible next synthons through contrastive learning. SYCLE further leverages the learned compatibility scores to guide synthon selection during pathway generation. SYCLE achieves the best average docking scores among the evaluated baselines across all 15 LIT-PCBA targets. On CrossDocked2020, SYCLE attains a mean Vina score of -9.80 kcal/mol and an AiZynthFinder retrosynthetic planning success rate of 68.8%.

1 Introduction

Recent advances in deep learning are rapidly proliferating AI-based approaches across various stages of drug discovery, ranging from hit identification to lead optimization, establishing AI-driven drug discovery as a pivotal research paradigm. In particular, structure-based drug design (SBDD) [2] is a crucial methodology that involves designing ligands using the structural information of the binding pocket of a target protein.

However, most previous studies on SBDD molecule generation [41, 35, 45, 23, 48] have a critical limitation that they do not take molecular synthesizability into account. As a result, most of the generated candidates frequently lack clear synthetic routes or fail to satisfy practical constraints on building block availability and reaction selectivity, which significantly limits the practical impact of these methods [16, 55]. While some approaches attempt to incorporate synthesizability scores as additional optimization objectives [9, 12, 53], these heuristic proxies are often insufficient to capture the underlying complexity of chemical synthesis.

Consequently, there have been attempts to consider synthetic pathways in molecular generative models. However, they are largely restricted to 2D molecule generation and fail to integrate the 3D structural and geometric constraints of the protein binding pocket [31, 10, 50, 18, 29, 34]. 3DSynthFlow [52] was recently introduced to bridge this gap through the co-design of 3D molecular conformations and synthetic pathways. 3DSynthFlow interleaves two processes: (i) Compositional Flow, which uses a GFlowNet [3] to model the sequential addition of molecular building blocks or synthons based on synthetic pathways, and (ii) State Flow, which models the associated 3D coordinates of intermediate molecules using flow matching. Specifically, synthesis steps are sampled at discrete time intervals, while 3D conformers are refined through a continuous-time flow.

[*]Equal Contribution

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

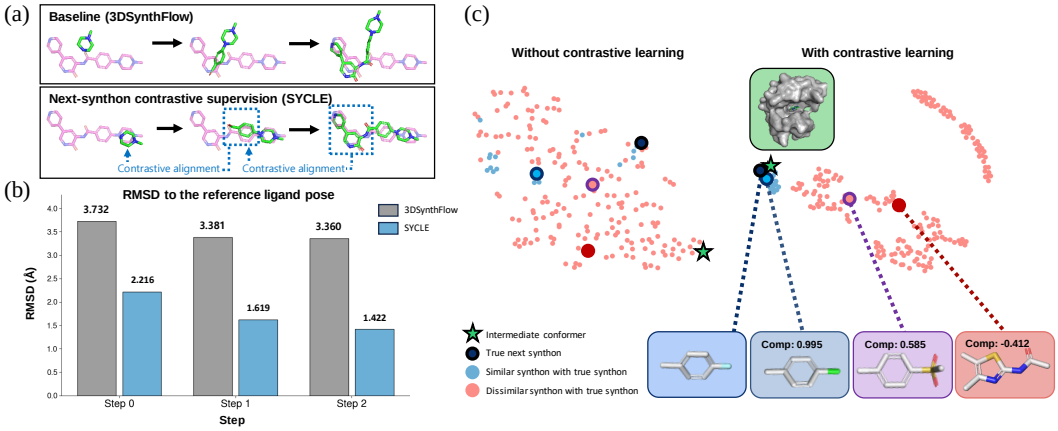

Figure 1: **Intermediate conformer prediction with contrastive fine-tuning. (a)** Predicted docking poses at each synthesis step alongside the true binding pose. SYCLE generates intermediate and final conformers that are more consistent with the true binding pose. **(b)** Mean RMSD to the reference pose across 116 three-synthon ligands from the CrossDocked2020 validation split at each synthesis step (1–3 synthons). SYCLE shows lower RMSD across all steps, supporting the effectiveness of contrastive fine-tuning for conformer prediction. **(c)** t-SNE visualization of the shared embedding space with and without contrastive learning. Synthons with a similarity score above 0.99 to the true next synthon are denoted as similar synthons, and the others are denoted as dissimilar synthons. With contrastive learning, the intermediate conformer embedding lies closer to true and similar synthons and farther from dissimilar ones, illustrating the learned compatibility structure.

Despite these advances, 3DSynthFlow remains limited in fully incorporating chemical constraints into conformer generation. State Flow learns to reconstruct conformations from partially assembled ligand structures, but its training objective does not explicitly align intermediate conformer representations with compatible next-synthon identities. Even if the intermediate molecule is identical, the final product molecule can adopt a significantly different conformation depending on which synthons are added to it. Such changes in shared-substructure binding poses have been observed in experimental fragment-growing campaigns, where they can complicate follow-up selection and require further optimization [43, 11]. Consequently, even though the intermediate molecule shares a substructure with the final product molecule, a gap can arise between the coordinates of those substructures, as shown in the top panel of Figure 1(a). Because intermediate conformations guide next-synthon selection, such geometric errors can affect subsequent pathway generation and the geometry of the final ligand. These observations motivate learning intermediate conformer representations that capture compatibility with possible subsequent synthons.

To this end, rather than simply interleaving synthetic pathway generation and 3D molecule generation, we propose a synthetic pathway-aware 3D molecule generation approach. Specifically, we propose *SYnthon Contrastive LEarning* (SYCLE), a synthesizable 3D molecule generation framework that simultaneously considers the 3D geometry of protein pockets and the sequential constraints of reaction pathways. SYCLE leverages contrastive learning [15, 38, 44] to align the representations of semantically similar pairs while contrasting them against negative samples. SYCLE extracts embeddings from partially generated intermediate 3D conformers within the pocket and computes embeddings for the candidate next synthons in a shared embedding space. During training, next-step synthons serve as positives, while synthon similarity controls repulsion from negatives, as illustrated in Figure 1(c) and Figure 2. Next-synthon identities provide additional supervision for conformer representation learning. The predicted intermediate conformers align more closely with the end product pose, as shown in the bottom panel of Figure 1(a) and Figure 1(b). Furthermore, we propose to inject the *compatibility score*, learned between the representations of the current intermediate conformer and a candidate next synthon, into the pathway generation policy to guide synthon selection. Overall, SYCLE provides a synthetic pathway-aware SBDD framework that accounts for both protein-ligand interactions and learned compatibility between intermediate conformers and candidate next synthons.

Our contributions are summarized as follows:

- We introduce a synthon contrastive learning objective that aligns intermediate 3D conformer representations with compatible next synthons, providing additional supervision for conformer prediction.

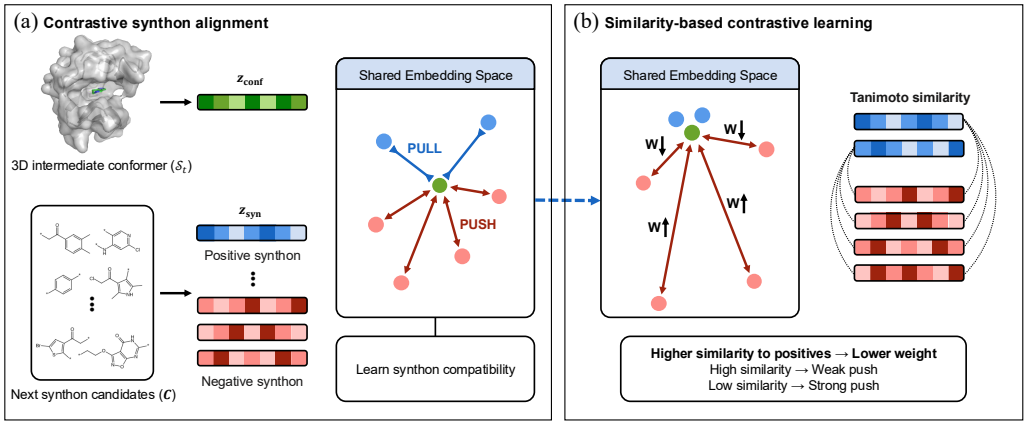

Figure 2: **The overall framework of SYCLE. (a)** The intermediate conformer and next synthon candidates are embedded in a shared embedding space. The contrastive objective pulls the embeddings of the true next synthons closer to the intermediate conformer embedding while pushing other synthon embeddings away. **(b)** Negative synthons are assigned weights that decrease with their similarity to the true next synthons, so that chemically closer negatives receive a weaker push.

- We use the learned compatibility score to augment the pathway generation policy logits and guide synthon selection within the protein pocket.
- We empirically show that SYCLE improves conformer accuracy and generated-pose physical validity, while outperforming the evaluated baselines in docking scores and retrosynthetic planning success.

2 Related Work

Structure-based drug design. Structure-based drug design (SBDD) leverages the structural information of target protein binding sites to design ligand molecules that satisfy geometric and chemical constraints while achieving high 3D structural stability and binding affinity [22]. Early methods proposed to utilize only the shape of the binding site for ligand generation [39, 1], while others directly modeled target-ligand correspondence from protein-ligand complex data through diverse generative strategies, including autoregressive models that construct molecules atom-by-atom [41, 35, 45], and diffusion-based approaches that capture the continuous distribution of 3D molecular structures [23, 48]. Despite these advances, most SBDD methods do not account for molecular synthesizability, frequently producing candidates that lack clear synthetic routes or violate practical constraints on building block availability and reaction selectivity [16], which limits their practical utility in drug discovery pipelines.

Synthesizable molecule design. Ensuring synthetic accessibility in generative molecular design has been a central challenge in drug discovery. A common approach constrains the generative process to a space defined by reaction rules and purchasable building blocks. To navigate this constrained space, a range of optimization strategies have been explored in early work, including autoencoders [5, 6], genetic algorithms [17], and reinforcement learning [20, 25]. More recently, GFlowNet-based [31, 10, 50, 14] and transformer-based [18, 29, 34] approaches have demonstrated strong performance in synthesizable molecule generation. Graph GA-ReaSyn [34] and SynGA [37] use genetic algorithms for synthesis-constrained molecular optimization. These methods primarily operate on molecular graphs or synthesis routes rather than jointly generating intermediate 3D conformations within protein pockets. Recently, Rekesh et al. [47] extended this to joint 3D coordinate generation, but their method operates unconditionally without protein pocket information. 3DSynthFlow [52] addressed this by co-designing synthesis pathways and 3D conformations within protein pockets. However, its conformer reconstruction objective does not explicitly align intermediate conformer representations with compatible next synthons. SYCLE introduces next-synthon contrastive supervision and uses the learned compatibility scores to guide synthon selection within this co-design framework.

3 Method

We introduce SYCLE, a synthesizable SBDD framework with aligned intermediate conformations, by learning synthetic pathway-aware compatibility between the 3D conformations of partial ligands and the next synthon candidates. We first introduce the GFlowNet-based generation framework and define the action space in Section 3.1. We then present the contrastive compatibility objective with similarity-based weighting in Section 3.2, followed by the forward logit augmentation for compatibility-guided generation in Section 3.3.

3.1 Preliminary: 3D Molecule and Synthesis Pathway Co-design

Following Shen et al. [52], we model a synthesizable ligand as $\boldsymbol{x} = (\mathcal{C}, \mathcal{S})$, where $\mathcal{C}$ denotes the synthetic pathway and $\mathcal{S}$ denotes the corresponding 3D conformations. Here, $\mathcal{C} = (\boldsymbol{C}^{(i)})_{i=1}^{n}$ is a sequence of n synthons to be attached (one per synthesis step), where each synthon $\boldsymbol{C}^{(i)}$ contains m_i atoms. $\mathcal{S} = (\boldsymbol{S}^{(i)})_{i=1}^{n}$ denotes the 3D coordinates of synthons, where $\boldsymbol{S}^{(i)} \in \mathbb{R}^{m_i \times 3}$ is the conformation of $\boldsymbol{C}^{(i)}$. Generation proceeds by interleaving discrete synthon attachment with continuous 3D conformation refinement over a global continuous time $t \in [0, 1]$.

Synthon attachment. Let $k(t)$ be the number of synthons that have been attached by time t, so that the compositional structure is $\mathcal{C}_t = (\boldsymbol{C}^{(i)})_{i=1}^{k(t)}$. At discrete time $t = t_{\text{gen}}^{(i)}$, we sample the next synthon $\boldsymbol{C}^{(i)}$ from the synthon attachment policy π_ϕ:

$$\boldsymbol{C}^{(i)} \sim \pi_\phi(\cdot \mid \boldsymbol{x}_t, \hat{\boldsymbol{x}}_t^1), \tag{1}$$

where $\hat{\boldsymbol{x}}_t^1$ is the model's prediction of the clean 3D conformation. Then, a transition function $T(\boldsymbol{x}_t, \boldsymbol{C}^{(i)})$ merges the new synthon into the current object. T also samples an initial conformation $\boldsymbol{S}_0^{(i)}$ for the newly introduced synthon from an isotropic Gaussian and inserts it into $\mathcal{S}_t$.

Synthesizable molecule generation requires selecting discrete actions—choosing which synthon to attach and via which reaction—such that the resulting molecules are sampled in proportion to a terminal reward $R(x)$ combining docking affinity and drug-likeness. Each action is defined over a purchasable building block library $\mathcal{B}$, where $\mathcal{B}_s \subseteq \mathcal{B}$ denotes blocks with a single attachment site and $\mathcal{B}_b \subseteq \mathcal{B}$ denotes compatible reactant blocks for a bimolecular reaction template r. Each step in the synthesis trajectory corresponds to one of two action types [50]. At the start of generation, a FIRSTSYNTHON action selects an initial synthon $\boldsymbol{C}^{(0)} \in \mathcal{B}_s$ conditioned on the protein pocket. The policy network π_ϕ computes a logit ℓ to select each action:

$$\ell = \pi_\phi(\boldsymbol{S}^{(0)}, \boldsymbol{C}^{(0)}). \tag{2}$$

All subsequent decisions are ADDSYNTHON actions, each defined as a $(r, \boldsymbol{C}^{(i)})$-pair consisting of a bimolecular reaction r compatible with $\mathcal{C}_t$ and a synthon $\boldsymbol{C}^{(i)} \in \mathcal{B}_b$. The corresponding logit ℓ is:

$$\ell = \pi_\phi(\mathcal{S}_t, (r, \boldsymbol{C}^{(i)})), \tag{3}$$

where $\mathcal{S}_t$ denotes the current conformations of the attached synthons so far at the current time t. The trajectory terminates when a terminal synthon is selected or the maximum synthesis depth is reached. The forward policy $P_F(\boldsymbol{C}^{(i)} \mid \mathcal{S}_t; \phi)$ is obtained by normalizing ℓ over all available actions at each state.

3D conformer prediction. Each synthon $\boldsymbol{C}^{(i)}$ is generated at a different time, so its conformation should be refined from the moment it is attached. To encode this temporal bias, a component-wise local time is defined as:

$$t_{\text{loc}}^{(i)} = \text{clip}\left(\frac{t - t_{\text{gen}}^{(i)}}{t_{\text{window}}}\right) \in [0, 1], \tag{4}$$

where $t_{\text{gen}}^{(i)}$ is the attachment time of the i-th synthon and t_{window} is the interpolation window, set to 1.0 in all experiments. Refinement ends at global time $t = 1$, so later-attached synthons do not reach local time 1. The same schedule is used during training and inference. An intermediate conformation is constructed via linear interpolation between the fully noisy initial conformation $\boldsymbol{S}_0^{(i)}$ and the clean target conformation $\boldsymbol{S}_1^{(i)}$, with Gaussian noise:

$$\boldsymbol{S}_t^{(i)} \sim \mathcal{N}\Big(t_{\text{loc}}^{(i)} \boldsymbol{S}_1^{(i)} + (1 - t_{\text{loc}}^{(i)}) \boldsymbol{S}_0^{(i)},\ \sigma^2 I\Big) \quad \text{for } t > t_{\text{gen}}^{(i)}. \tag{5}$$

The conformation prediction network $p^{\psi}_{1|t}$ consists of a POCKETENCODER and a LIGANDDECODER. The POCKETENCODER encodes the protein pocket using HierSemla [52], a hierarchical extension of Semla [26], producing protein features $\boldsymbol{h}^{\text{pro}}$. The LIGANDDECODER processes the partially assembled ligand through Semla layers with protein-ligand cross-attention with $\boldsymbol{h}^{\text{pro}}$, producing ligand features $\boldsymbol{h}^{\text{lig}}$. The predicted clean coordinates $\hat{\boldsymbol{S}}^{(i)}_1$ are then obtained by normalizing and projecting the equivariant component of $\boldsymbol{h}^{\text{lig}}$.

Given the current object $\boldsymbol{x}_t = (\mathcal{C}_t, \mathcal{S}_t)$, $p^{\psi}_{1|t}$ predicts the clean conformations $\hat{\boldsymbol{S}}^{(i)}_1 = p^{\psi}_{1|t}(\boldsymbol{x}_t)^{(i)}$ for all synthons attached so far. During sampling, we define a simple vector field $\hat{\boldsymbol{S}}^{(i)}_1 - \boldsymbol{S}^{(i)}_t$ and update the conformation using an Euler step:

$$\boldsymbol{S}^{(i)}_{t+\Delta t} = \boldsymbol{S}^{(i)}_t + \left(\hat{\boldsymbol{S}}^{(i)}_1 - \boldsymbol{S}^{(i)}_t\right) \cdot \kappa^{(i)} \Delta t, \tag{6}$$

where $\kappa^{(i)}$ scales the step size based on the remaining time until the interpolation for i-th synthon is complete.

Training objective. The conformer predictor $p^{\psi}_{1|t}$ is independently trained by minimizing

$$\mathcal{L}_{\text{conf}} = \mathbb{E}_{\boldsymbol{x}_1 \sim p_{\text{data}},\, t \sim \mathcal{U}(0,1)} \sum_{i=1}^{k(t)} \left\| p^{\psi}_{1|t}(\boldsymbol{x}_t)^{(i)} - \boldsymbol{S}^{(i)}_1 \right\|_2^2. \tag{7}$$

After the conformer predictor is trained, we fix $p^{\psi}_{1|t}$ and optimize the synthon attachment policy π_ϕ using a trajectory-balance objective. We optimize the trajectory balance (TB) objective of GFlowNets [42], which is equivalent to optimizing the following objective [52]:

$$\mathcal{L}_{\text{TB}} = \left(\log \frac{Z_\phi \prod_{i=0}^{n-1} P_F\left(\boldsymbol{C}^{(i)} \mid \boldsymbol{x}_{t^{(i)}_{\text{gen}}}, \hat{\boldsymbol{x}}^1_{t^{(i)}_{\text{gen}}}; \phi\right)}{R(\boldsymbol{x}_1)} \right)^2, \tag{8}$$

where Z_ϕ is a learnable normalizer, $P_F(\boldsymbol{C}^{(i)} \mid \cdot\,; \phi)$ denotes the probability of selecting synthon $\boldsymbol{C}^{(i)}$ under the policy π_ϕ, and $R(\boldsymbol{x}_1)$ is a reward defined on the final generated object. At inference time, the learned policy and the conformer predictor are executed together by alternating discrete attachment and continuous refinement until $t = 1$.

3.2 Next Synthon Contrastive Learning

Intermediate 3D conformer predictor described in Section 3.1 is trained to refine a partially assembled ligand conformation without explicitly knowing which synthon will be attached next. However, the identity of the next synthon can strongly affect the final product geometry, and these geometric constraints should propagate back to earlier intermediate conformations. To incorporate this reaction pathway signal, we introduce next synthon contrastive compatibility learning, which aligns the current intermediate conformation with the synthons that can attach in the next step.

Construction of next synthon candidate set. We first define the next synthon candidate set via a synthon-level decomposition of each ligand. Specifically, each ligand is decomposed into synthons by applying reaction-aware bond-breaking rules following Enamine synthesis protocols [21], yielding a set of synthon groups $(\boldsymbol{C}^{(i)})_{i=1}^n$. We further define a synthon connectivity graph $\mathcal{E}$, where $(\boldsymbol{C}^{(i)}, \boldsymbol{C}^{(j)}) \in \mathcal{E}$ if a chemical bond exists between $\boldsymbol{C}^{(i)}$ and $\boldsymbol{C}^{(j)}$ in the original ligand.

To determine the generation order, each synthon $\boldsymbol{C}^{(i)}$ is assigned a generation time $t^{(i)}_{\text{gen}}$ via BFS-ordered traversal of $\mathcal{E}$. At global time t, the visible and masked synthon sets are defined as:

$$\mathcal{V}_t = \{\boldsymbol{C}^{(i)} \mid t^{(i)}_{\text{gen}} \le t\}, \quad \mathcal{M}_t = \{\boldsymbol{C}^{(i)} \mid t^{(i)}_{\text{gen}} > t\}. \tag{9}$$

Throughout Section 3.2, we use square-bracket superscripts $[\cdot]$ to index samples within a mini-batch, and use round-bracket superscripts $(\cdot)$ to index synthesis steps as in Section 3.1. The next synthon candidate set consists of all masked synthons that are adjacent to the current visible set:

$$\mathcal{C}^{[m]}_t = \{\boldsymbol{C}^{[m](i)} \in \mathcal{M}_t \mid \exists\, \boldsymbol{C}^{[m](j)} \in \mathcal{V}_t,\, (\boldsymbol{C}^{[m](i)}, \boldsymbol{C}^{[m](j)}) \in \mathcal{E}\}. \tag{10}$$

The positive set for the synthon contrastive compatibility learning is defined as $\mathcal{C}^{+,[m]} := \mathcal{C}^{[m]}$. For contrastive learning, the normalization pool is formed at the mini-batch level. For a mini-batch B, we aggregate the candidate sets across all samples:

$$\mathcal{C}_{\text{mb}} = \bigcup_{[m]\in B} \mathcal{C}_t^{[m]}. \tag{11}$$

For each intermediate conformation $\mathcal{S}_t^{[m]}$, synthons in $\mathcal{C}^{+,[m]}$ are treated as positives, while in-batch negatives are defined as candidates in $\mathcal{C}^{-,[m]} := \mathcal{C}_{\text{mb}} \setminus \mathcal{C}^{+,[m]}$.

Conformer and synthon embeddings. We extract atom-wise invariant and equivariant features $\boldsymbol{h}^{\text{lig}}$ of the current intermediate ligand conformation $\mathcal{S}_t$ by passing it through the LIGANDDECODER conditioned on the pocket embedding obtained from the POCKETENCODER. Because the next reaction step is governed by the local geometry around the attachment site, we apply *attachment-aware mean pooling*:

- If attachment atoms exist, we compute the mean of features over attachment atoms.
- Otherwise, we fall back to mean pooling over all valid atoms.

Denote this pooled representation as $\boldsymbol{f}_{\text{conf}} = \text{AttPool}(\boldsymbol{h}^{\text{lig}})$. We then map it to the shared contrastive space through a projection head $\text{Proj}_{\text{conf}}$:

$$\boldsymbol{z}_{\text{conf}}(\mathcal{S}_t) = \text{Proj}_{\text{conf}}(\boldsymbol{f}_{\text{conf}}) \in \mathbb{R}^d. \tag{12}$$

The conformer embedding and coordinate predictions share the LIGANDDECODER features, allowing the contrastive objective to shape the representation used for conformer prediction. For each candidate next synthon $\boldsymbol{C} \in \mathcal{C}^{[m]}$, we first generate a 3D conformer via RDKit and represent it as a LIGANDBATCH consisting of atom types, formal charges, bond adjacency, 3D coordinates, and attachment point flags. This is encoded using a lightweight SYNTHONENCODER and produces atom-wise features $\boldsymbol{h}^{\text{syn}}$, which we pool to obtain $\boldsymbol{f}_{\text{syn}} = \text{AttPool}(\boldsymbol{h}^{\text{syn}})$. A symmetric projection head yields:

$$\boldsymbol{z}_{\text{syn}}(\boldsymbol{C}) = \text{Proj}_{\text{syn}}(\boldsymbol{f}_{\text{syn}}) \in \mathbb{R}^d. \tag{13}$$

Consequently, the current intermediate conformation and the next synthon candidates are embedded in a shared embedding space. The overall architecture of SYCLE is described in Figure 3 and Section D.

Contrastive learning objective. We define the *compatibility score* as the cosine similarity between a conformer $\mathcal{S}_t$ and a candidate next synthon $\boldsymbol{C}$ in the shared embedding space:

$$\text{Comp}(\mathcal{S}_t, \boldsymbol{C}) = \frac{\langle \tilde{\boldsymbol{z}}_{\text{conf}}(\mathcal{S}_t),\ \tilde{\boldsymbol{z}}_{\text{syn}}(\boldsymbol{C})\rangle}{\tau_c}, \tag{14}$$

where $\tilde{\boldsymbol{z}}$ denotes ℓ_2-normalized embeddings and τ_c is a temperature. We propose to use the multi-positive InfoNCE objective [30] to handle cases where multiple next synthons are valid:

$$\mathcal{L}_{\text{cont}}^{[m]} = -\log \frac{\sum_{\boldsymbol{C}\in\mathcal{C}^{+,[m]}} \exp(\text{Comp}(\mathcal{S}_t^{[m]}, \boldsymbol{C}))}{\sum_{\boldsymbol{C}\in\mathcal{C}_{\text{mb}}} \exp(\text{Comp}(\mathcal{S}_t^{[m]}, \boldsymbol{C}))}. \tag{15}$$

This objective pulls the next synthons along the synthetic pathway toward the current intermediate conformer embedding, while repelling other synthons, thereby incorporating the synthetic pathway information into the shared embedding space.

However, some negative synthons are chemically close to the true next synthon and can be alternative synthons to be attached. In this case, controlling the degree of contrastive repulsion would be more informative. Therefore, we propose to perform contrastive learning by setting *soft* negatives instead of hard ones. We introduce similarity-based weighting using the learned synthon features. We first binarize the synthon feature $\boldsymbol{f}_{\text{syn}}$ by thresholding each dimension at its batch-wise median, then compute a Tanimoto similarity $\text{Tan}(\boldsymbol{C}, \boldsymbol{C}') \in [0, 1]$ between candidate synthons. For each synthon $\boldsymbol{C} \in \mathcal{C}_{\text{mb}}$, we measure its maximum similarity to any true next synthon:

$$\text{sim}(\mathcal{C}^{+,[m]}, \boldsymbol{C}) = \max_{\boldsymbol{C}'\in\mathcal{C}^{+,[m]}} \text{Tan}(\boldsymbol{C}, \boldsymbol{C}'). \tag{16}$$

We then define a similarity-based weight:

$$w(\boldsymbol{C}) = \begin{cases} 1, & \text{if } \boldsymbol{C} \in \mathcal{C}^{+,[m]} \\ \exp\left(-\frac{\text{sim}(\mathcal{C}^{+,[m]}, \boldsymbol{C})}{\tau_w}\right), & \text{otherwise} \end{cases} \tag{17}$$

where τ_w is a temperature. We apply the weighting to Eq. (15):

$$\mathcal{L}_{\text{cont}}^{[m]} = -\log \frac{\sum_{\boldsymbol{C} \in \mathcal{C}^{+,[m]}} \exp(\text{Comp}(S_t^{[m]}, \boldsymbol{C}))}{\sum_{\boldsymbol{C} \in \mathcal{C}_{\text{mb}}} w(\boldsymbol{C}) \exp(\text{Comp}(S_t^{[m]}, \boldsymbol{C}))}. \tag{18}$$

Intuitively, synthons with a high structural similarity to the positive set receive a smaller weight $w(\boldsymbol{C})$, weakening the contrastive penalty imposed on them.

The final training objective for the conformer predictor is obtained by combining the conformation loss $\mathcal{L}_{\text{conf}}$ (Eq. (7)) and the contrastive learning loss $\mathcal{L}_{\text{cont}}$ (Eq. (18)):

$$\mathcal{L} = \mathcal{L}_{\text{conf}} + \lambda \cdot \mathcal{L}_{\text{cont}}, \tag{19}$$

where λ controls the strength of the contrastive learning signal. The contrastive objective uses next-synthon identities as additional supervision for conformer representation learning. At inference, the conformer predictor takes the current partial ligand state and protein pocket as inputs, without requiring the next-synthon identity.

3.3 Next Synthon Selection with Compatibility Score

The proposed learning scheme of the conformer and synthon embeddings is not only beneficial to the intermediate conformer generation, but also to the synthetic pathway generation itself. We propose to inject the compatibility score (Eq. (14)) into the pathway generation policy to guide synthon selection.

For AddSynthon actions, we augment the logits (Eq. (3)) with the compatibility score:

$$\ell_{\text{SYCLE}}(\mathcal{S}_t, (r, \boldsymbol{C}^{(i)})) = \ell(\mathcal{S}_t, (r, \boldsymbol{C}^{(i)})) + \beta \cdot \text{Comp}(\mathcal{S}_t, \boldsymbol{C}^{(i)}), \tag{20}$$

where $\beta > 0$ controls the strength of the compatibility score augmentation. The augmented logits define the forward policy from which the next synthon is sampled. The augmentation is not applied for FirstSynthon actions, as these actions select an initial synthon without an existing intermediate reaction context.

By utilizing the compatibility score both in the contrastive learning objective of the conformer predictor and the logits of the next synthon selection actions, SYCLE effectively couples (i) how intermediate 3D geometries are refined under the pocket structure and (ii) which synthons are selected along the synthesis pathway, yielding a synthetic pathway-aware 3D molecule generation pipeline.

After training the conformer predictor and synthon encoder, synthon embeddings $\tilde{\boldsymbol{z}}_{\text{syn}}(\boldsymbol{C})$ are precomputed for all synthons in the block library and reused during policy training and generation. Conformer embeddings $\tilde{\boldsymbol{z}}_{\text{conf}}(\mathcal{S}_t)$ are computed once per visited state and then reused to evaluate multiple candidate synthons, making the compatibility augmentation computationally lightweight.

4 Experiments

4.1 Pocket-specific Optimization

Setup. Synthetic pathways are constructed using the 38 bimolecular Enamine reaction protocols from Gao et al. [18]. The pocket-specific optimization uses a 1.2M Enamine catalog as a building block library. Building blocks with a drug-likeness score under 60 are excluded following Lee et al. [33]. Molecular complexity is bounded at two synthesis steps per molecule, in line with the scope of Enamine REAL Space [21]. Retrosynthetic planning success is evaluated via AiZynthFinder [19], a retrosynthetic planning tool run against the full Enamine catalog as the stock database. This metric measures route recovery under USPTO-derived templates and the specified search budget. Protein-ligand interactions are evaluated with PoseCheck [24], which counts hydrogen-bond acceptor/donor, van der Waals, and hydrophobic contacts between the ligand and pocket.

For reward design and evaluation, we follow the protocol of Seo et al. [50] and Shen et al. [52]. Binding affinity is estimated via Uni-Dock [56], a GPU-based docking engine, with a per-target

Table 1: **Pocket-specific optimization results on LIT-PCBA.** Average Vina docking scores for the top-100 diverse molecules across five representative targets, along with the average and median over all 15 LIT-PCBA targets. SYCLE results without † are reported as mean ± standard deviation across 3 runs with different random seeds. The results for the other 10 targets are in Table 15. For SynFlowNet, v2024.05[a] and v2024.10[b] are reported. The results for the baselines are taken from Shen et al. [52]. † denotes evaluation before redocking using the same random seed for 3DSynthFlow and SYCLE. The best results are highlighted in bold.

Category	Method	Average Vina Docking Score (kcal/mol, ↓)						
		ADRB2	ALDH1	ESR_ago	ESR_antago	FEN1	Avg.	Med.
Fragment	FragGFN	-10.19 (± 0.33)	-10.43 (± 0.29)	-9.81 (± 0.09)	-9.85 (± 0.13)	-7.67 (± 0.71)	-9.58	-9.85
	FragGFN+SA	-9.70 (± 0.61)	-9.83 (± 0.65)	-9.27 (± 0.95)	-10.06 (± 0.30)	-7.26 (± 0.10)	-9.22	-9.58
Reaction	SynNet	-8.03 (± 0.26)	-8.81 (± 0.21)	-8.88 (± 0.13)	-8.52 (± 0.16)	-6.36 (± 0.09)	-8.12	-8.52
	BBAR	-9.95 (± 0.04)	-10.06 (± 0.14)	-9.97 (± 0.03)	-9.92 (± 0.05)	-6.84 (± 0.07)	-9.35	-9.84
	SynFlowNet[a]	-10.85 (± 0.10)	-10.69 (± 0.09)	-10.44 (± 0.05)	-10.27 (± 0.04)	-7.47 (± 0.02)	-9.95	-10.34
	SynFlowNet[b]	-9.17 (± 0.68)	-9.37 (± 0.29)	-9.17 (± 0.12)	-9.05 (± 0.14)	-6.45 (± 0.13)	-8.78	-9.17
	RGFN	-9.84 (± 0.21)	-9.93 (± 0.11)	-9.99 (± 0.11)	-9.72 (± 0.14)	-6.92 (± 0.06)	-9.08	-9.91
	RxnFlow	-11.45 (± 0.05)	-11.26 (± 0.07)	-11.15 (± 0.02)	-10.77 (± 0.04)	-7.66 (± 0.02)	-10.46	-10.84
3D Reaction	3DSynthFlow	-11.96 (± 0.12)	-11.82 (± 0.03)	-11.58 (± 0.07)	-11.23 (± 0.08)	-7.79 (± 0.01)	-10.89	-11.26
	3DSynthFlow†	-9.82	-10.28	-10.62	-10.26	-6.20	-9.39	-9.54
	SYCLE	**-12.10** (± 0.09)	**-11.98** (± 0.08)	**-11.64** (± 0.21)	**-11.45** (± 0.05)	**-8.16** (± 0.10)	**-11.14**	**-11.50**
	SYCLE†	-11.68	-10.75	-11.49	-11.18	-7.03	-10.35	-10.40

Table 2: **AiZynthFinder success rates** for the 15 LIT-PCBA protein targets. The results for the baselines are taken from Shen et al. [52]. The best results are in bold.

Method	Success Rate (%, ↑)	Synthesis Steps (↓)
FragGFN+SA	3.52	3.74
SynNet	47.50	3.45
BBAR	17.92	3.68
SynFlowNet[a]	54.60	2.55
SynFlowNet[b]	58.38	2.47
RGFN	47.43	2.46
RxnFlow	65.35	2.17
3DSynthFlow	68.58	2.39
SYCLE	**69.84**	**1.58**

Table 3: **Protein-ligand interaction.** PoseCheck scores for the top-100 diverse molecules across the first 5 LIT-PCBA targets. SYCLE results without † are reported as mean ± standard deviation across 3 runs. SynFlowNet versions follow Table 1. The results for the baselines are taken from Shen et al. [52]. † denotes evaluation before redocking, as in Table 1. The best results are in bold.

Method	H-Bond Acceptors	H-Bond Donors	Van der Waals	Hydrophobic	Sum
SynFlowNet[a]	0.22 (± 0.01)	0.11 (± 0.01)	9.31 (± 0.05)	10.41 (± 0.05)	20.05
SynFlowNet[b]	0.22 (± 0.03)	0.10 (± 0.01)	8.38 (± 0.05)	9.44 (± 0.06)	18.14
RGFN	0.19 (± 0.01)	0.11 (± 0.01)	9.19 (± 0.28)	10.25 (± 0.12)	19.73
RxnFlow	0.22 (± 0.00)	0.10 (± 0.01)	9.63 (± 0.09)	10.67 (± 0.10)	20.62
3DSynthFlow	0.33 (± 0.06)	**0.17** (± 0.03)	11.04 (± 0.37)	10.79 (± 0.21)	22.32
3DSynthFlow†	0.22	0.18	9.87	10.26	20.54
SYCLE	**0.34** (± 0.02)	0.16 (± 0.03)	**11.22** (± 0.04)	**11.20** (± 0.01)	**22.91**
SYCLE†	0.23	0.16	9.50	10.58	20.47

budget of 64,000 generated molecules. QED [4] is jointly optimized as a co-objective, and molecules are generated with a maximum of 40 heavy atoms to discourage size-driven reward exploitation. The full GFlowNet reward function is provided in Section E.2. From these, we retain molecules with QED > 0.5 and select the top-100 by docking score, subject to pairwise Tanimoto similarity < 0.5 computed on RDKit fingerprints [32] to maintain chemical diversity. We additionally evaluate docking scores and PoseCheck interactions on the original generated poses of 3DSynthFlow and SYCLE before redocking.

Baselines. For pocket-specific optimization, we compare SYCLE against previous methods spanning three categories: fragment-based, reaction-based, and 3D reaction-based generation. In the fragment category, we evaluate FragGFN and FragGFN+SA, which differ in whether synthetic accessibility [12] is explicitly incorporated into the objective. The reaction-based baselines include SynNet [17], BBAR [49], two variants of SynFlowNet [10], RGFN [31], and RxnFlow [50], covering search-based, conditional generative, and GFlowNet-based methods. As the most closely related 3D reaction-based baseline, we include 3DSynthFlow as a baseline [52].

Results. Table 1 and Table 15 report docking scores after redocking across the 15 LIT-PCBA targets. Table 1 additionally reports results before redocking for 3DSynthFlow and SYCLE. SYCLE achieves the best scores on all targets and the best average docking score among the evaluated baselines. Before redocking, SYCLE also achieves lower docking scores than 3DSynthFlow, supporting its ability to generate poses with favorable docking scores. These results demonstrate that SYCLE's strategy of using next-synthon supervision for intermediate conformer prediction generates ligands with more favorable docking scores in the target protein pocket.

Table 2 reports AiZynthFinder success rates and the average number of steps in successfully planned routes. SYCLE achieves the highest planning success rate and the shortest average planned route

Table 4: **Pocket-conditional generation results on CrossDocked2020.** For each metric for valid molecules, the mean value (Avg.) and median value (Med.) are reported. Full SYCLE results are reported as mean ± standard deviation across three sampling runs using the same trained checkpoint for each ρ setting. References indicate known active molecules. Results for baselines other than SA scores and diversity are taken from Shen et al. [52]. SA scores and diversity for MolCRAFT, MolCRAFT-large, RxnFlow, and 3DSynthFlow are recomputed from author-provided SMILES; SA scores and diversity for the remaining baselines are taken from Shen et al. [51]. Best results are in bold.

Category	Method	Validity (%, ↑)	Vina (kcal/mol, ↓)		QED (↑)		SA (↑)		AiZynth. Succ Rate (%, ↑)		Div (↑)	Time (s, ↓)
		Validity	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Avg.
	Reference	–	-7.71	-7.80	0.48	0.47	0.73	0.74	36.1	–	–	–
Atom	Pocket2Mol [45]	98.3	-7.60	-7.16	0.57	0.58	0.74	0.75	29.1	22.0	0.69	2504
	TargetDiff [22]	91.5	-7.37	-7.56	0.49	0.49	0.58	0.58	9.9	3.2	0.72	3428
	DecompDiff [23]	66.0	-8.35	-8.25	0.37	0.35	0.61	0.60	0.9	0.0	0.68	6189
	DiffSBDD [48]	76.0	-6.95	-7.10	0.47	0.48	0.55	0.56	2.9	2.0	**0.76**	135
	MolCRAFT [46]	96.7	-8.05	-8.05	0.50	0.50	0.69	0.67	16.5	9.1	0.73	141
	MolCRAFT-large [46]	70.8	-9.25	-9.24	0.45	0.44	0.62	0.62	3.9	0.0	0.60	>141
Fragment	TacoGFN [51]	**100.0**	-8.24	-8.44	0.67	0.67	0.79	0.79	1.3	1.0	0.53	**4**
2D Reaction	RxnFlow [50]	**100.0**	-8.85	-9.03	0.67	0.67	0.54	0.54	34.8	34.5	0.54	**4**
3D Reaction	3DSynthFlow (low ρ) [52]	99.9	-9.14	-9.38	0.69	0.69	0.66	0.66	36.2	37.0	0.53	6
	3DSynthFlow (med ρ) [52]	99.9	-9.30	-9.62	0.72	0.71	0.66	0.66	35.1	36.0	0.49	6
	3DSynthFlow (high ρ) [52]	**100.0**	-9.42	-9.61	**0.73**	**0.73**	0.65	0.65	36.1	36.0	0.45	6
	SYCLE (low ρ)	**100.0**	-9.41 (± 0.00)	-9.51 (± 0.04)	0.69 (± 0.00)	0.69 (± 0.00)	0.77 (± 0.00)	0.77 (± 0.00)	58.4 (± 0.64)	59.5 (± 1.32)	0.58 (± 0.00)	9
	SYCLE (med ρ)	**100.0**	-9.63 (± 0.01)	-9.80 (± 0.01)	0.70 (± 0.00)	0.70 (± 0.00)	0.79 (± 0.00)	0.79 (± 0.00)	64.4 (± 0.59)	64.8 (± 1.13)	0.55 (± 0.00)	9
	SYCLE (high ρ)	**100.0**	**-9.80** (± 0.01)	**-9.96** (± 0.04)	0.70 (± 0.00)	0.70 (± 0.00)	**0.81** (± 0.00)	**0.81** (± 0.00)	**68.8** (± 1.72)	**69.8** (± 1.04)	0.52 (± 0.00)	9
Ablation	- similarity weighting	100.0	-9.68	-9.84	0.69	0.69	0.79	0.79	65.6	66.5	0.54	9
	- logit augmentation	100.0	-9.61	-9.87	0.69	0.69	0.78	0.78	57.3	57.0	0.55	9
	- sim. weighting & logit aug.	100.0	-9.50	-9.70	0.70	0.70	0.78	0.78	53.1	53.0	0.55	9

length among the evaluated baselines. Fewer synthesis steps are generally preferred, as synthetic complexity is known to correlate with practical considerations such as yield and cost [9].

Table 3 summarizes PoseCheck metrics. After redocking, SYCLE demonstrates higher interaction scores overall compared to the baseline models, suggesting that its improved docking performance is accompanied by more extensive protein–ligand contacts. Before redocking, SYCLE achieves total interaction counts comparable to those of 3DSynthFlow.

4.2 Pocket-conditional Generation

Setup. We also evaluate SYCLE in a pocket-conditional generation setting, where a single policy generates ligands for previously unseen protein pockets without relying on extra oracle evaluations [45, 22, 48]. We use the same pocket-conditional protocol and reward definition as 3DSynthFlow [52], where the TacoGFN [51] proxy trained on CrossDocked2020 is used to estimate the reward. Synthetic pathway construction follows the same protocol as in Section 4.1, except that a 300k Enamine stock subset is used as the building block library.

Evaluation is conducted on the CrossDocked2020 benchmark [13] using 100 test pockets, and each method generates 100 molecules per pocket. In addition to the metrics used in the pocket-specific setting, we report the following metrics: **validity** measures the fraction of valid and unique molecules, **diversity** is the mean pairwise Tanimoto distance computed over RDKit fingerprints [32], and **time** refers to the wall-clock time to generate 100 molecules.

Baselines. We compare SYCLE with a diverse set of baselines covering atom-based, fragment-based, and reaction-based molecular generation. The atom-based baselines include Pocket2Mol [45], TargetDiff [22], DecompDiff [23], DiffSBDD [48], MolCRAFT, and MolCRAFT-large [46]. We include TacoGFN [51] as a fragment-based method and RxnFlow [50] as a 2D reaction-based method. Following 3DSynthFlow [52], SYCLE is tested with three reward regimes: low $\rho \sim U(1, 64)$, medium $\rho \sim U(16, 64)$, and high $\rho \sim U(32, 64)$, where ρ controls the exploration-exploitation trade-off during sampling (see Section E.5 for further details).

Results. Table 4 summarizes the pocket-conditional generation results on the CrossDocked2020 benchmark. SYCLE improves Vina docking scores over 3DSynthFlow across all corresponding ρ settings, with SYCLE (high ρ) outperforming all baselines in average docking score. Notably, SYCLE outperforms the baselines in retrosynthetic planning success by a large margin. Not only does SYCLE achieve the highest SA score, but it also achieves a mean AiZynthFinder success rate of 68.8%, compared with the highest baseline rate of 36.2%. These results demonstrate that SYCLE improves both docking performance and retrosynthetic planning success.

4.3 Ablation Study

We conduct ablation studies of SYCLE on the pocket-conditional generation benchmark with the high ρ setting.

Effect of contrastive learning. To assess the contribution of contrastive learning, we train a contrastive-removed baseline (CR) using the same initialization, data, and epoch budget as SYCLE, removing the contrastive loss and synthon encoder. Compatibility-based logit augmentation is also disabled in CR and all other settings remain unchanged. Evaluation is conducted on the CrossDocked2020 benchmark using 100 test pockets and 100 molecules generated per pocket by each method, with all poses evaluated before docking or refinement. As shown in Table 5, CR improves over the released 3DSynthFlow checkpoint, indicating a benefit from additional fine-tuning. SYCLE further reduces clashes and improves raw Vina scores and PoseBusters [7] validity over CR. These results support the contribution of the contrastive learning module and its use in synthon selection to generating higher-quality poses beyond additional fine-tuning alone.

Table 5: **Effect of contrastive learning.** Results are evaluated before docking or refinement in the pocket-conditional setting. CR denotes the contrastive-removed baseline. Clashes and raw Vina scores are medians; PB-valid is the percentage passing all PoseBusters checks.

Method	Clashes ↓	Raw Vina (kcal/mol) ↓	PB-valid (%) ↑
Crystal Reference	6	–	95.0
3DSynthFlow	22	-2.0	8.8
CR	13	-5.1	32.3
SYCLE	**9**	**-5.9**	**39.3**

Effect of similarity weighting. To examine the effect of the proposed similarity-based weighting, we compare SYCLE against a variant that treats all negative synthons as hard negatives, removing the similarity-based weight w in Eq. (18). As shown in Table 4, removing the similarity weighting lowers the AiZynthFinder success rate, with a smaller change in SA score, supporting the contribution of similarity weighting to retrosynthetic planning success. This result suggests that chemically close negatives may represent plausible alternative synthons and should receive a weaker contrastive push rather than being uniformly penalized.

Effect of logit augmentation. To examine the effect of the logit augmentation for synthon selection, we compare SYCLE against a variant that does not use the compatibility score in the action logits (i.e., $\beta = 0$ in Eq. (20)), and relies solely on the GFlowNet policy to select synthons. As shown in Table 4, removing the logit augmentation leads to a notable drop in the AiZynthFinder success rate, demonstrating the contribution of compatibility-guided synthon selection to retrosynthetic planning success. This result suggests that the proposed compatibility score provides useful guidance beyond the learned policy alone.

5 Conclusion

In this work, we present SYCLE, a synthesizable SBDD framework that uses next-synthon contrastive supervision to improve intermediate conformer modeling. By aligning intermediate conformer representations with compatible next synthons and incorporating the learned compatibility score into the pathway generation policy, SYCLE improves conformer prediction and guides synthon selection. Experiments show improved conformer accuracy and generated-pose physical validity, together with improved docking scores and retrosynthetic planning success compared with the evaluated baselines, supporting its potential for synthesis-aware SBDD.

Acknowledgements

This work was supported by the Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2019-II190075, Artificial Intelligence Graduate School Program (KAIST)); the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00256259); the "Advanced GPU Utilization Support Program" funded by the Government of the Republic of Korea (Ministry of Science and ICT); the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Ministry of Science and ICT (MSIT) of the Republic of Korea in connection with the Global AI Frontier Lab International Collaborative Research (No. RS-2024-00469482 and RS-2024-00509279); and the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2022-II220713, Meta-learning Applicable to Real-world Problems).

References

[1] Keir Adams and Connor W Coley. Equivariant shape-conditioned generation of 3d molecules for ligand-based drug design. *arXiv preprint arXiv:2210.04893*, 2022.

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

[3] Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J Hu, Mo Tiwari, and Emmanuel Bengio. Gflownet foundations. *Journal of Machine Learning Research*, 24(210):1–55, 2023.

[4] G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90–98, 2012.

[5] John Bradshaw, Brooks Paige, Matt J Kusner, Marwin Segler, and José Miguel Hernández-Lobato. A model to search for synthesizable molecules. *Advances in Neural Information Processing Systems*, 32, 2019.

[6] John Bradshaw, Brooks Paige, Matt J Kusner, Marwin Segler, and José Miguel Hernández-Lobato. Barking up the right tree: an approach to search over molecule synthesis dags. *Advances in neural information processing systems*, 33:6852–6866, 2020.

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

[8] Max W Chang, Christian Ayeni, Sebastian Breuer, and Bruce E Torbett. Virtual screening for hiv protease inhibitors: a comparison of autodock 4 and vina. *PloS one*, 5(8):e11955, 2010.

[9] Connor W Coley, Luke Rogers, William H Green, and Klavs F Jensen. Scscore: synthetic complexity learned from a reaction corpus. *Journal of chemical information and modeling*, 58 (2):252–261, 2018.

[10] Miruna Cretu, Charles Harris, Ilia Igashov, Arne Schneuing, Marwin Segler, Bruno Correia, Julien Roy, Emmanuel Bengio, and Pietro Liò. Synflownet: Design of diverse and novel molecules with synthesis constraints. *arXiv preprint arXiv:2405.01155*, 2024.

[11] Lindsey G DeRatt, E Christine Pietsch, Justin S Cisar, Edgar Jacoby, Faraz Kazmi, Rosalie Matico, Paul Shaffer, Alexandra Tanner, Weixue Wang, Ricardo Attar, et al. Discovery of alternative binding poses through fragment-based identification of dhodh inhibitors. *ACS Medicinal Chemistry Letters*, 15(3):381, 2024.

[12] Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1(1):8, 2009.

[13] 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.

[14] Piotr Gaiński, Oussama Boussif, Andrei Rekesh, Dmytro Shevchuk, Ali Parviz, Mike Tyers, Robert A Batey, and Michał Koziarski. Scalable and cost-efficient de novo template-based molecular generation. *arXiv preprint arXiv:2506.19865*, 2025.

[15] Bowen Gao, Bo Qiang, Haichuan Tan, Yinjun Jia, Minsi Ren, Minsi Lu, Jingjing Liu, Wei-Ying Ma, and Yanyan Lan. Drugclip: Contrastive protein-molecule representation learning for virtual screening. *Advances in Neural Information Processing Systems*, 36:44595–44614, 2023.

[16] Wenhao Gao and Connor W Coley. The synthesizability of molecules proposed by generative models. *Journal of chemical information and modeling*, 60(12):5714–5723, 2020.

[17] Wenhao Gao, Rocío Mercado, and Connor W Coley. Amortized tree generation for bottom-up synthesis planning and synthesizable molecular design. *arXiv preprint arXiv:2110.06389*, 2021.

[18] Wenhao Gao, Shitong Luo, and Connor W Coley. Generative artificial intelligence for navigating synthesizable chemical space. *arXiv preprint arXiv:2410.03494*, 2024.

[19] Samuel Genheden, Amol Thakkar, Veronika Chadimová, Jean-Louis Reymond, Ola Engkvist, and Esben Bjerrum. Aizynthfinder: a fast, robust and flexible open-source software for retrosynthetic planning. *Journal of cheminformatics*, 12(1):70, 2020.

[20] Sai Krishna Gottipati, Boris Sattarov, Sufeng Niu, Yashaswi Pathak, Haoran Wei, Shengchao Liu, Simon Blackburn, Karam Thomas, Connor Coley, Jian Tang, et al. Learning to navigate the synthetically accessible chemical space using reinforcement learning. In *International conference on machine learning*, pages 3668–3679. PMLR, 2020.

[21] Oleksandr O Grygorenko, Dmytro S Radchenko, Igor Dziuba, Alexander Chuprina, Kateryna E Gubina, and Yurii S Moroz. Generating multibillion chemical space of readily accessible screening compounds. *Iscience*, 23(11), 2020.

[22] Jiaqi Guan, Wesley Wei Qian, Xingang Peng, Yufeng Su, Jian Peng, and Jianzhu Ma. 3d equivariant diffusion for target-aware molecule generation and affinity prediction. *arXiv preprint arXiv:2303.03543*, 2023.

[23] Jiaqi Guan, Xiangxin Zhou, Yuwei Yang, Yu Bao, Jian Peng, Jianzhu Ma, Qiang Liu, Liang Wang, and Quanquan Gu. Decompdiff: diffusion models with decomposed priors for structure-based drug design. *arXiv preprint arXiv:2403.07902*, 2024.

[24] Charles Harris, Kieran Didi, Arian R Jamasb, Chaitanya K Joshi, Simon V Mathis, Pietro Lio, and Tom Blundell. Benchmarking generated poses: How rational is structure-based drug design with generative models? *arXiv preprint arXiv:2308.07413*, 2023.

[25] Julien Horwood and Emmanuel Noutahi. Molecular design in synthetically accessible chemical space via deep reinforcement learning. *ACS omega*, 5(51):32984–32994, 2020.

[26] Ross Irwin, Alessandro Tibo, Jon Paul Janet, and Simon Olsson. Efficient 3d molecular generation with flow matching and scale optimal transport. In *ICML 2024 AI for Science Workshop*, 2024.

[27] Micael Jacobsson and Anders Karlén. Ligand bias of scoring functions in structure-based virtual screening. *Journal of chemical information and modeling*, 46(3):1334–1343, 2006.

[28] Moksh Jain, Sharath Chandra Raparthy, Alex Hernández-Garcıa, Jarrid Rector-Brooks, Yoshua Bengio, Santiago Miret, and Emmanuel Bengio. Multi-objective gflownets. In *International conference on machine learning*, pages 14631–14653. PMLR, 2023.

[29] Zygimantas Jocys, Zhanxing Zhu, Henriette MG Willems, and Katayoun Farrahi. Synthformer: Equivariant pharmacophore-based generation of synthesizable molecules for ligand-based drug design. *Artificial Intelligence in the Life Sciences*, page 100148, 2025.

[30] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

[31] Michał Koziarski, Andrei Rekesh, Dmytro Shevchuk, Almer van der Sloot, Piotr Gaiński, Yoshua Bengio, Chenghao Liu, Mike Tyers, and Robert Batey. Rgfn: Synthesizable molecular generation using gflownets. *Advances in Neural Information Processing Systems*, 37:46908–46955, 2024.

[32] Greg Landrum et al. Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling. *Greg Landrum*, 8(31.10):5281, 2013.

[33] Kyunghoon Lee, Jinho Jang, Seonghwan Seo, Jaechang Lim, and Woo Youn Kim. Drug-likeness scoring based on unsupervised learning. *Chemical Science*, 13(2):554–565, 2022.

[34] Seul Lee, Karsten Kreis, Srimukh Prasad Veccham, Meng Liu, Danny Reidenbach, Saee Paliwal, Weili Nie, and Arash Vahdat. Exploring synthesizable chemical space with iterative pathway refinements. *International conference on learning representations*, 2026.

[35] Meng Liu, Youzhi Luo, Kanji Uchino, Koji Maruhashi, and Shuiwang Ji. Generating 3d molecules for target protein binding. *arXiv preprint arXiv:2204.09410*, 2022.

[36] Tairan Liu, Misagh Naderi, Chris Alvin, Supratik Mukhopadhyay, and Michal Brylinski. Break down in order to build up: decomposing small molecules for fragment-based drug design with e molfrag. *Journal of chemical information and modeling*, 57(4):627–631, 2017.

[37] Alston Lo, Connor Coley, and Wojciech Matusik. A genetic algorithm for navigating synthesizable molecular spaces. In *International Conference on Learning Representations*, volume 2026, pages 37665–37693, 2026.

[38] Frederieke Lohmann, Stephan Allenspach, Kenneth Atz, Carl CG Schiebroek, Jan A Hiss, and Gisbert Schneider. Protein binding site representation in latent space. *Molecular Informatics*, 44(1):e202400205, 2025.

[39] Siyu Long, Yi Zhou, Xinyu Dai, and Hao Zhou. Zero-shot 3d drug design by sketching and generating. *Advances in Neural Information Processing Systems*, 35:23894–23907, 2022.

[40] Shitong Luo, Jiaqi Guan, Jianzhu Ma, and Jian Peng. A 3d generative model for structure-based drug design. *Advances in Neural Information Processing Systems*, 34:6229–6239, 2021.

[41] Youzhi Luo and Shuiwang Ji. An autoregressive flow model for 3d molecular geometry generation from scratch. In *International conference on learning representations*, 2022.

[42] Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in gflownets. *Advances in Neural Information Processing Systems*, 35:5955–5967, 2022.

[43] Alexander Metz, Jan Wollenhaupt, Steffen Glöckner, Niki Messini, Simon Huber, Tatjana Barthel, Ahmed Merabet, H-D Gerber, Andreas Heine, Gerhard Klebe, et al. Frag4lead: growing crystallographic fragment hits by catalog using fragment-guided template docking. *Biological Crystallography*, 77(9):1168–1182, 2021.

[44] Rebecca Manuela Neeser, Ilia Igashov, Arne Schneuing, Michael M Bronstein, Philippe Schwaller, and Bruno Correia. Flow-based fragment identification via contrastive learning of binding site-specific latent representations. In *AI for Accelerated Materials Design-ICLR 2025*, 2025.

[45] Xingang Peng, Shitong Luo, Jiaqi Guan, Qi Xie, Jian Peng, and Jianzhu Ma. Pocket2mol: Efficient molecular sampling based on 3d protein pockets. In *International conference on machine learning*, pages 17644–17655. PMLR, 2022.

[46] 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. *arXiv preprint arXiv:2404.12141*, 2024.

[47] Andrei Rekesh, Miruna Cretu, Dmytro Shevchuk, Vignesh Ram Somnath, Pietro Liò, Robert A Batey, Mike Tyers, Michał Koziarski, and Cheng-Hao Liu. Syncogen: Synthesizable 3d molecule generation via joint reaction and coordinate modeling. *arXiv preprint arXiv:2507.11818*, 2025.

[48] Arne Schneuing, Charles Harris, Yuanqi Du, Kieran Didi, Arian Jamasb, Ilia Igashov, Weitao Du, Carla Gomes, Tom L Blundell, Pietro Lio, et al. Structure-based drug design with equivariant diffusion models. *Nature Computational Science*, 4(12):899–909, 2024.

[49] Seonghwan Seo, Jaechang Lim, and Woo Youn Kim. Molecular generative model via retrosynthetically prepared chemical building block assembly. *Advanced Science*, 10(8):2206674, 2023.

[50] Seonghwan Seo, Minsu Kim, Tony Shen, Martin Ester, Jinkyoo Park, Sungsoo Ahn, and Woo Youn Kim. Generative flows on synthetic pathway for drug design. *arXiv preprint arXiv:2410.04542*, 2024.

[51] Tony Shen, Seonghwan Seo, Grayson Lee, Mohit Pandey, Jason R Smith, Artem Cherkasov, Woo Youn Kim, and Martin Ester. Tacogfn: Target-conditioned gflownet for structure-based drug design. *arXiv preprint arXiv:2310.03223*, 2023.

[52] Tony Shen, Seonghwan Seo, Ross Irwin, Kieran Didi, Simon Olsson, Woo Youn Kim, and Martin Ester. Compositional flows for 3d molecule and synthesis pathway co-design. *arXiv preprint arXiv:2504.08051*, 2025.

[53] Amol Thakkar, Veronika Chadimová, Esben Jannik Bjerrum, Ola Engkvist, and Jean-Louis Reymond. Retrosynthetic accessibility score (rascore)–rapid machine learned synthesizability classification from ai driven retrosynthetic planning. *Chemical science*, 12(9):3339–3349, 2021.

[54] Viet-Khoa Tran-Nguyen, Célien Jacquemard, and Didier Rognan. Lit-pcba: an unbiased data set for machine learning and virtual screening. *Journal of chemical information and modeling*, 60(9):4263–4273, 2020.

[55] W Patrick Walters and Regina Barzilay. Critical assessment of ai in drug discovery. *Expert opinion on drug discovery*, 16(9):937–947, 2021.

[56] Yuejiang Yu, Chun Cai, Jiayue Wang, Zonghua Bo, Zhengdan Zhu, and Hang Zheng. Uni-dock: Gpu-accelerated docking enables ultralarge virtual screening. *Journal of chemical theory and computation*, 19(11):3336–3345, 2023.

A Limitations

Although SYCLE shows strong empirical performance, there are several limitations. First, the contrastive objective considers only the immediately following synthon, leaving the framework partially myopic to longer-range synthetic dependencies. Second, the accessible chemical space is restricted by the fixed reaction protocols, Enamine building-block libraries, and the limit of two synthesis steps. Within this space, SYCLE shows some preference for certain reaction types, leaving room to improve reaction diversity in the generated synthetic routes. Encouraging broader exploration of reaction types is an important direction for future work. Finally, the generation order is fixed such that a FIRSTSYNTHON action always initiates the trajectory, which may not reflect the optimal order for some target pockets. Allowing pocket-conditioned adaptive ordering is left as future work.

Our evaluation relies on docking scores and retrosynthetic planning success, which do not establish experimental binding affinity or synthesis success. Although SYCLE improves generated-pose quality, physical validity remains below that of crystal reference structures, leaving room for further improvement.

B Impact Statement

This work has the potential to accelerate early-stage drug discovery by generating molecules with high binding affinity and feasible synthetic routes, reducing the time and cost of identifying viable drug candidates. As with generative molecular design broadly, misuse for harmful purposes remains a societal risk that warrants responsible deployment. We also caution that practical synthesis success depends on additional factors not captured computationally, such as reaction conditions, reagent availability, and cost.

C Theoretical Background

C.1 Trajectory Balance with Deterministic State Transitions

Following Shen et al. [52], when the initial conformation $\boldsymbol{S}_0^{(i)}$ for each newly added synthon is sampled under a fixed random seed, the state $x_{t_{\text{gen}}^{(i)}}$ is uniquely determined by the sequence $(\boldsymbol{C}^{(1)}, \ldots, \boldsymbol{C}^{(k(t))})$. This determinism allows the forward transition probability to be simplified as:

$$P_F\Big(x_{t_{\text{gen}}^{(i+1)}} \mid x_{t_{\text{gen}}^{(i)}}, \hat{\boldsymbol{x}}^1_{t_{\text{gen}}^{(i)}}; \phi, p^{\psi}_{1|t}\Big) = P_F\Big(\boldsymbol{C}^{(i)} \mid x_{t_{\text{gen}}^{(i)}}, \hat{\boldsymbol{x}}^1_{t_{\text{gen}}^{(i)}}; \phi\Big), \tag{21}$$

and the TB objective in Eq. (8) decomposes into a tractable form over the sequence of synthon selections.

C.2 Convergence of Compatibility-Augmented Trajectory Balance

SYCLE augments the GFlowNet action logits with the compatibility score (Eq. 20):

$$\ell_{\text{SYCLE}}\Big(\mathcal{S}_t, (r, \boldsymbol{C}^{(i)})\Big) = \ell\Big(\mathcal{S}_t, (r, \boldsymbol{C}^{(i)})\Big) + \beta \cdot \text{Comp}\Big(\mathcal{S}_t, \boldsymbol{C}^{(i)}\Big), \tag{22}$$

where ℓ is the learnable GFlowNet logit (Eq. (29)) and $\text{Comp}(\mathcal{S}_t, \boldsymbol{C}^{(i)})$ is the compatibility score (Eq. (14)) computed from the frozen conformer predictor $p^{\psi}_{1|t}$. The resulting forward policy conditioned on pocket $\mathcal{P}$ is:

$$P_F^{\text{SYCLE}}\Big(\boldsymbol{C}^{(i)} \,\Big|\, \mathcal{S}_t, \mathcal{P}\Big) = \frac{\exp\Big(\ell\big(\mathcal{S}_t, (r, \boldsymbol{C}^{(i)})\big) + \beta \cdot \text{Comp}\big(\mathcal{S}_t, \boldsymbol{C}^{(i)}\big)\Big)}{\sum_{\boldsymbol{C}' \in \mathcal{A}(\mathcal{C}_t)} \exp\Big(\ell(\mathcal{S}_t, (r, \boldsymbol{C}')) + \beta \cdot \text{Comp}(\mathcal{S}_t, \boldsymbol{C}')\Big)}. \tag{23}$$

We now show that the TB objective with P_F^{SYCLE} retains a valid fixed point where $p(x_1) \propto R(x_1)$.

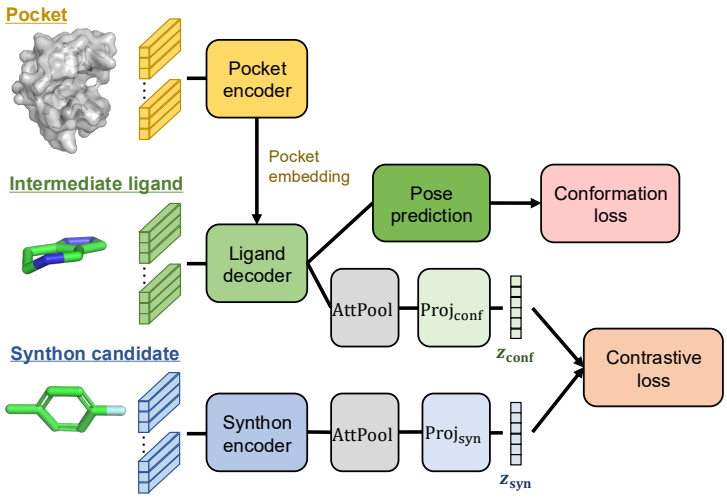

Figure 3: **The overall architecture of SYCLE.**

Proposition. *Let* $c\big(\mathcal{S}_t, \boldsymbol{C}^{(i)}\big) = \beta \cdot \mathrm{Comp}\big(\mathcal{S}_t, \boldsymbol{C}^{(i)}\big)$ *be any fixed, bounded function of the current state and candidate synthon. Then the TB objective in Eq.* (8) *with* P_F^{SYCLE} *achieves* $\mathcal{L}_{\mathrm{TB}} = 0$ *for all trajectories* τ*, with the optimal learnable logit given by:*

$$\ell^*\Big(\mathcal{S}_t,\ (r, \boldsymbol{C}^{(i)})\Big) = \log P^*\Big(\boldsymbol{C}^{(i)} \,\Big|\, \mathcal{S}_t,\ \mathcal{P}\Big) - c\Big(\mathcal{S}_t,\ \boldsymbol{C}^{(i)}\Big) + \mathrm{const}, \tag{24}$$

where P^* *is the reward-proportional forward policy.*

proof Fix any trajectory $\tau = (\boldsymbol{C}^{(1)}, \ldots, \boldsymbol{C}^{(n)})$. By Section C.1, the states $\mathcal{S}_{t_{\mathrm{gen}}^{(i)}}$ are uniquely determined by τ independently of ϕ. Therefore, $c(\mathcal{S}_t, \boldsymbol{C}^{(i)})$ is independent of ϕ along any fixed trajectory.

For any target distribution $P^*\big(\boldsymbol{C}^{(i)} \mid \mathcal{S}_t, \mathcal{P}\big)$ satisfying $\prod_i P^*\big(\boldsymbol{C}^{(i)}\big) \propto R(x_1)$, we can always find ℓ^* such that $\mathrm{softmax}(\ell^* + c) = P^*$ by setting:

$$\ell^*\Big(\mathcal{S}_t,\ (r, \boldsymbol{C}^{(i)})\Big) = \log P^*\Big(\boldsymbol{C}^{(i)} \mid \mathcal{S}_t, \mathcal{P}\Big) - c\Big(\mathcal{S}_t, \boldsymbol{C}^{(i)}\Big) + \mathrm{const},$$

since $c(\cdot)$ cancels in the numerator and denominator of the softmax:

$$\mathrm{softmax}(\ell^* + c)_j = \frac{\exp(\log P_j^* - c_j + c_j)}{\sum_k \exp(\log P_k^* - c_k + c_k)} = \frac{P_j^*}{\sum_k P_k^*} = P_j^*.$$

Setting $Z_\phi = \sum_{x_1} R(x_1)$ then gives $Z_\phi \prod_i P_F^{\mathrm{SYCLE}}(\boldsymbol{C}^{(i)}) = R(x_1)$ for all trajectories τ, and hence $\mathcal{L}_{\mathrm{TB}} = 0$.

D SYCLE Architecture

SYCLE consists of a conformer predictor $p_{1|t}^{\psi}$ and a pathway generation policy π_ϕ. The conformer predictor internally integrates a synthon encoder to learn the compatibility between intermediate conformers and candidate next synthons. Figure 3 illustrates the training architecture of the conformer predictor, which is jointly optimized with the conformation loss $\mathcal{L}_{\mathrm{conf}}$ and the contrastive loss $\mathcal{L}_{\mathrm{cont}}$ (Eq. (19)).

D.1 Conformer Predictor

The conformer predictor consists of a PocketEncoder and a LigandDecoder. The PocketEncoder produces pocket embeddings that condition the LigandDecoder, which in turn outputs both the predicted conformer and the conformer embedding z_{conf}(Eq. (12)) used for contrastive learning.

Pocket encoder. The protein pocket is encoded using *HierSemla*, a hierarchical extension of Semla [26] introduced in Shen et al. [52]. Rather than processing all protein atoms jointly, HierSemla first applies intra-residue attention to obtain per-atom invariant and equivariant features, then collapses them into a single residue-level representation $(\boldsymbol{h}_i^{(\text{pro})}, \boldsymbol{x}_i^{(\text{pro})})$ through mean pooling. Limiting self-attention to atoms within the same residue makes the encoder scalable to full-atom protein pockets without prohibitive memory overhead.

Ligand decoder. The partial ligand is encoded by a sequence of Semla layers [52] that incorporate pocket context via cross-attention, yielding atom-level invariant and equivariant features $\boldsymbol{h}^{(\text{lig})}$. At each layer, cross-attention messages are computed over all ligand atom–pocket residue pairs $(i, j) \in \mathcal{V} \times \mathcal{P}$, where $\mathcal{V}$ and $\mathcal{P}$ denote the sets of ligand atoms and protein residues, respectively:

$$(\boldsymbol{m}_{i,j}^{\text{inv}},\, \boldsymbol{m}_{i,j}^{\text{equi}}) = \Omega_\theta^{(\text{pro-lig})}\left(\tilde{h}_i,\, \tilde{h}_j,\, \tilde{x}_i \cdot \tilde{x}_j\right), \tag{25}$$

where $\tilde{h}_i$ and $\tilde{x}_i$ are the layer-normalized invariant and coordinate-normalized equivariant projections of node i, respectively. The cross-attention messages are combined with intra-ligand messages before the subsequent feature update, allowing each ligand atom to incorporate structural context from the surrounding binding site. Each atom is additionally annotated with a binary flag `IS_ATTACHMENT_POINT` indicating the reactive site at which the next synthon will be attached. This provides explicit geometric information for the conformer predictor to reserve sufficient space for the incoming synthon at the attachment site. The predicted clean coordinates $\hat{\boldsymbol{S}}_1^{(i)}$ are obtained by normalizing and projecting the equivariant component of $\boldsymbol{h}^{(\text{lig})}$.

Synthon encoder. The SYNTHONENCODER processes a synthon $\boldsymbol{C}$ through Semla layers without pocket conditioning to produce atom-wise features. Given atom types, charges, adjacency, and 3D coordinates of the synthon, the encoder computes invariant and equivariant features as:

$$\left(\boldsymbol{h}^{(\text{syn})}, \boldsymbol{x}^{(\text{syn})}\right) = \text{SYNTHONENCODER}(\boldsymbol{C}), \tag{26}$$

where $\boldsymbol{h}^{(\text{syn})} \in \mathbb{R}^{L \times d_{\text{inv}}}$ and $\boldsymbol{x}^{(\text{syn})} \in \mathbb{R}^{L \times 3 \times d_{\text{equi}}}$ denote the invariant and equivariant features per atom respectively, with equivariant features defined over the three spatial directions. These dimensions are kept identical to those of the LIGANDDECODER to ensure compatibility in the shared embedding space. Two key differences distinguish it from the LIGANDDECODER. First, it uses no pocket conditioning, as synthons are encoded solely based on their intrinsic chemical structure. Second, it employs a lighter layer stack of 3 layers compared to the 8 layers of the LIGANDDECODER. The SYNTHONENCODER also includes an attachment point embedding $\boldsymbol{e}_{\text{att}}$ identical to that in the LIGANDDECODER, providing explicit information about the attachment site without time or size conditioning.

Projection heads. Both the conformer embedding $\boldsymbol{z}_{\text{conf}}$ and the synthon embedding $\boldsymbol{z}_{\text{syn}}$ are projected into the shared contrastive space from a common feature dimension. Before projection, atom-wise invariant features $\boldsymbol{h}^{(\cdot)} \in \mathbb{R}^{d_{\text{inv}}}$ and equivariant features $\boldsymbol{x}^{(\cdot)} \in \mathbb{R}^{3 \times d_{\text{equi}}}$ are pooled via ATTPOOL and concatenated into a single vector of dimension $d_{\text{inv}} + 3d_{\text{equi}}$. Both $\text{Proj}_{\text{conf}}$ and Proj_{syn} share the same architecture:

$$\text{Proj}(\boldsymbol{f}) = W_2\, \text{ReLU}(\text{LayerNorm}(W_1 \boldsymbol{f}))\,, \tag{27}$$

where $W_1 \in \mathbb{R}^{d_h \times (d_{\text{inv}} + 3d_{\text{equi}})}$ and $W_2 \in \mathbb{R}^{d \times d_h}$, with d_h denoting the hidden dimension of the projection head.

D.2 Pathway Generation Policy

The pathway generation policy π_ϕ reuses the architecture of 3DSynthFlow [52], with a graph transformer as backbone f_θ and an MLP g_θ for action embedding. The protein pocket is modeled as a residue-level graph whose nodes correspond to C_α atoms, with edges connecting every pair of residues. Each ligand atom attends to all protein nodes via cross-edges, and pairwise distances are projected into edge feature vectors to provide geometric context during message passing.

At each step of the synthesis trajectory, the policy chooses between two action types, conditioned on temperature ρ and multi-objective weights following Shen et al. [52].

For FIRSTSYNTHON, the logit for selecting $C^{(0)} \in \mathcal{B}_s$ is computed as:

$$\ell = \mathrm{MLP}_{\textsc{FirstSynthon}}\big(f_\theta(\mathcal{S}^{(0)}, \mathcal{P})\big) \odot g_\theta(C^{(0)}), \tag{28}$$

where f_θ encodes the current state and protein pocket $\mathcal{P}$, and $g_\theta(C^{(0)})$ is the synthon embedding.

For each subsequent ADDSYNTHON step, the logit for selecting $(r, C^{(i)}) \in \mathcal{R}(\mathcal{S}_t) \times \mathcal{B}_b$, where $\mathcal{R}(\mathcal{S}_t)$ denotes the set of reactions applicable to the current state $\mathcal{S}_t$, is:

$$\ell = \mathrm{MLP}_{\textsc{AddSynthon}}\big(f_\theta(\mathcal{S}_t, \mathcal{P})\big) \odot g_\theta(r, C^{(i)}), \tag{29}$$

where the reaction type r is jointly encoded with the synthon embedding by $g_\theta(r, C^{(i)})$.The full definition of the action space is provided in Appendix E.1.

The compatibility score augmentation (Eq. (20)) is applied only to ADDSYNTHON logits, as FIRSTSYNTHON operates without an existing intermediate conformation. The conformer embedding $\boldsymbol{z}_{\text{conf}}(\mathcal{S}_t)$ from the frozen conformer predictor is passed as an additional input to evaluate the compatibility score at each ADDSYNTHON step.

E Experimental Details

E.1 Action Space

Following Shen et al. [52], we model each synthesis step as a forward transition in the GFlowNet trajectory, so that a complete synthetic pathway corresponds to a full generation trajectory. Rather than treating each building block as a fully formed molecule, we adopt the synthon representation of Liu et al. [36], in which each fragment carries one or two reactive sites that define where the next reaction can occur. A fragment with one reactive site is a *brick*: once selected, no further extension is possible and the trajectory ends. A fragment with two reactive sites is a *linker*: it introduces an additional reactive site, allowing the assembly to continue. Valid connections between fragments are governed by the reaction rule set $\mathcal{R}$.

The building block library $\mathcal{B}$ is the complete synthon set, with $\mathcal{B}' \subseteq \mathcal{B}$ denoting the brick synthon set and $\mathcal{B}_r \subseteq \mathcal{B}$ the set of synthons compatible with reaction r. The allowable action space $\mathcal{A}(\mathcal{C})$ at compositional state $\mathcal{C}$ is defined as:

$$\mathcal{A}(\mathcal{C}) = \begin{cases} \mathcal{B}' & \text{if } t = 1, \\ \bigcup_{r \in \mathcal{R}(\mathcal{C})} \mathcal{B}_r & \text{otherwise}, \end{cases} \tag{30}$$

where $\mathcal{R}(\mathcal{C}) \subseteq \mathcal{R}$ denotes the subset of reactions applicable to the current state. Generation is capped at two synthesis steps, consistent with the tractable range of Enamine REAL Space [21].

E.2 Reward Function

Pocket-specific optimization. Following Seo et al. [50], the reward is a weighted sum of drug-likeness and binding affinity:

$$R(x) = w_1 \mathrm{QED}(x) + w_2 \widehat{\mathrm{Vina}}(x), \tag{31}$$

where $\widehat{\mathrm{Vina}}(x)$ denotes the normalized Vina score. The weights w_1 and w_2 act as multi-objective conditioning weights [28]. Also, for baselines that do not use GFlowNet, it is set to 0.5.

Pocket-conditional generation. We build on the reward formulation of Seo et al. [50], dropping the SA score term from Shen et al. [51] because synthesizability is enforced structurally through the pathway generation process. The affinity component r_{affinity} rescales the proxy docking score via a piecewise linear mapping. r_{QED} saturates at a QED of 0.7. The final reward is divided by the cube root of heavy atom count to offset the size bias in docking scores, discouraging the policy from inflating molecular size as a shortcut to higher reward.

$$r_{\text{affinity}}(x) = \begin{cases} 0 & \text{if } 0 \le \mathrm{Proxy}(x) \\ -0.04 \times \mathrm{Proxy}(x) & \text{if } -8 \le \mathrm{Proxy}(x) \le 0 \\ -0.2 \times \mathrm{Proxy}(x) - 1.28 & \text{if } -13 \le \mathrm{Proxy}(x) \le -8 \\ 1.32 & \text{if } \mathrm{Proxy}(x) \le -13 \end{cases} \tag{32}$$

$$r_{\text{QED}}(x) = \begin{cases} \text{QED}(x)/0.7 & \text{if QED}(x) \leq 0.7 \\ 1 & \text{otherwise} \end{cases} \tag{33}$$

$$\text{SYCLE-Reward}(x) = \frac{r_{\text{affinity}}(x) \times r_{\text{QED}}(x)}{\sqrt[3]{\text{HeavyAtomCounts}(x)}} \tag{34}$$

E.3 Datasets

Building block library. We use 38 bimolecular Enamine synthesis protocols for constructing synthetic pathways [18]. For pocket-specific optimization, we use the Enamine comprehensive catalog (v2024.06.10) containing approximately 1.2M building blocks after filtering by drug-likeness score [33] and chemical validity. For pocket-conditional generation, we use a 300k Enamine stock set with the same filtering applied.

LIT-PCBA pockets. The LIT-PCBA benchmark [54], comprising 15 protein targets with experimentally validated active and inactive compounds, is used for pocket-specific optimization. Each binding pocket is constructed by collecting all residues containing heavy atoms within 10 Å of the co-crystallized ligand, consistent with the pocket extraction procedure used for CrossDocked2020. The PDB entries and target names are listed in Table 6.

Table 6: **Target information for the LIT-PCBA dataset.**

Target	PDB Id	Target name
ADRB2	4ldo	Beta2 adrenoceptor
ALDH1	5l2m	Aldehyde dehydrogenase 1
ESR_ago	2p15	Estrogen receptor α with agonist
ESR_antago	2iok	Estrogen receptor α with antagonist
FEN1	5fv7	FLAP Endonuclease 1
GBA	2v3d	Acid Beta-Glucocerebrosidase
IDH1	4umx	Isocitrate dehydrogenase 1
KAT2A	5h86	Histone acetyltransferase KAT2A
MAPK1	4zzn	Mitogen-activated protein kinase 1
MTORC1	4dri	PPIase domain of FKBP51, Rapamycin
OPRK1	6b73	Kappa opioid receptor
PKM2	4jpg	Pyruvate kinase muscle isoform M1/M2
PPARG	5y2t	Peroxisome proliferator-activated receptor γ
TP53	3zme	Cellular tumor antigen p53
VDR	3a2i	Vitamin D receptor

CrossDocked2020. CrossDocked2020 [13] is a large-scale dataset of protein-ligand complexes constructed by cross-docking ligands into non-native protein structures. Following the data split of Luo et al. [40] and Peng et al. [45], we used 99,000 complexes for training and 1,000 for validation, excluding all LIT-PCBA targets from the training set. The conformer predictor and the pocket-conditional policy are both trained on this split, with the 100 test pockets reserved exclusively for evaluating pocket-conditional generation.

E.4 Baselines

Pocket-specific optimization. We use the redocked baseline results reported in Shen et al. [52] and additionally evaluate 3DSynthFlow before redocking under the same protocol as SYCLE.

Pocket-conditional generation. We used the baseline results as detailed in the caption of Table 4.

Evaluation before redocking. Vina scores are computed on the original generated poses using `score_only`, without coordinate refinement. We reselect the top-100 molecules using these scores while retaining the same diversity criterion. PoseCheck metrics are also computed on the original generated coordinates.

Table 7: **Hyperparameters for the conformer predictor.**

Hyperparameter	Value
Architecture	
Number of protein layers	4
Number of ligand layers	8
Noise prior	Gaussian
Time distribution	Beta(1.0, 1.0)
Time Scheduling	
Time per action	0.3
Interpolation window	1.0
Maximum decomposed parts	3
Minimum decomposed fragment size (atoms)	5
Minimum trajectory length	2
Maximum trajectory length	3

E.5 Hyperparameters

Conformer predictor hyperparameters. We describe the hyperparameters used for the conformer predictor as below. All other architectural hyperparameters follow the default settings of SemlaFlow [26].

1. **Number of protein layers**: Number of HierSemla layers for protein message passing within each residue.
2. **Number of ligand layers**: Number of Semla layers for ligand decoding with protein-ligand cross-attention.
3. **Noise prior**: Distribution from which the initial conformation $\mathbf{S}_0^{(i)}$ is sampled.
4. **Time distribution**: Distribution used to sample the global time t during training.
5. **Time per action**: Time interval per synthon addition, setting the attachment time as $t_{\text{gen}}^{(i)} = 0.3 \cdot (i-1)$, following 3DSynthFlow [52].
6. **Interpolation window**: Time window over which conformation refinement is performed. With $t_{\text{window}} = 1.0$, all synthon conformations are refined jointly through $t = 1$.
7. **Maximum decomposed parts**: Maximum number of synthons per molecule during data preparation.
8. **Minimum decomposed fragment size**: Minimum per-synthon atom count when fragmenting training ligands, filtering out trivially small fragments unlikely to appear in real synthesis steps.
9. **Minimum / Maximum trajectory length**: Lower and upper bounds on synthon attachment steps during generation, corresponding to 1–2 reaction steps.

Pathway generation policy hyperparameters. We define the hyperparameters used for the pathway generation policy below.

1. **GFN temperature (ρ)**: The reward exponentiation parameter [28] that reshapes the sampling distribution as $\pi(x) \propto R(x)^{\rho}$. Sampling ρ from a wider range (e.g., $\mathcal{U}(1, 64)$) encourages broader exploration, while restricting ρ to a narrower high-value range (e.g., $\mathcal{U}(32, 64)$) concentrates sampling toward high-reward molecules.
2. **Sampling tau (EMA factor)**: The exponential moving average factor applied to the policy parameters during data collection. For pocket-specific optimization, $\tau = 0.9$ is used to stabilize training over long optimization runs. For pocket-conditional generation, $\tau = 0.0$ is used to ensure the policy adapts rapidly across diverse pockets.

Table 8: **Hyperparameters for the pathway generation policy.**

Hyperparameter	Value
GFlowNet Training	
GFN temperature (ρ)	Uniform(1, 64)
Sampling tau (pocket-specific)	0.9
Sampling tau (pocket-conditional)	0.0
Learning rate (Z)	10^{-2}
Learning rate (P_F)	10^{-4}
Model	
State embedding dimension	64
Action embedding dimension	64
Compatibility score coefficient β	10.0
Action Space	
Minimum trajectory length	2 (min. reaction steps: 1)
Maximum trajectory length	3 (max. reaction steps: 2)
Action space subsampling ratio	1%
Train random action probability	5%

Table 9: **Hyperparameters for the contrastive learning objective.**

Hyperparameter	Value
Contrastive loss weight λ	0.1
Contrastive temperature τ_c	0.5
Similarity temperature τ_w	0.3
Number of synthon layers	3
Projection dimension	256

3. **Learning rate** (Z): The learning rate for updating the partition function Z in the TB objective.
4. **Learning rate** (P_F): The learning rate for updating the forward policy P_F.
5. **State embedding dimension**: The hidden dimension of the graph transformer used to encode the protein-ligand complex state.
6. **Action embedding dimension**: The hidden dimension of the MLP used to embed each candidate synthon action.
7. **Compatibility score coefficient** β: The scaling coefficient applied to the compatibility score when augmenting the AddSynthon logits (Eq. (20)).
8. **Minimum / Maximum trajectory length**: The minimum and maximum number of synthon attachment steps during generation, corresponding to 1-2 reaction steps.
9. **Action space subsampling ratio**: The fraction of candidate synthons randomly subsampled for each FirstSynthon and bi-molecular reaction template at each step. A minimum subsampling size of 100 is enforced for reactions with small reactant block sets to avoid high variance.
10. **Train random action probability**: The probability of selecting a uniformly random action during training, ensuring that rare action categories are sufficiently explored.

Contrastive learning We define the hyperparameters used for the contrastive learning objective below.

1. **Contrastive loss weight** λ: The weighting coefficient for $\mathcal{L}_{\text{cont}}$ in the joint training objective of Eq. (19).

Table 10: **Summary of training time for SYCLE components.**

Model Component	Hardware	Batch Size	Iterations	Training Time
Conformer Predictor	4×RTX A5000 (24GB)	dynamic	30 epochs	∼80 hours
Policy (Pocket-Specific)	1×RTX A5000 (24GB)	64	1,000 steps	18–24 hours
Policy (Pocket-Conditional)	1×RTX A5000 (24GB)	64	–	up to 72 hours

2. **Contrastive temperature** τ_c: The temperature parameter that scales cosine similarities in the contrastive loss. A lower temperature sharpens the distribution over candidate synthons.
3. **Similarity-weight temperature** τ_w: The temperature parameter that controls similarity-based weighting of negative synthons. A lower temperature assigns smaller weights to negatives with nonzero similarity to positive synthons, weakening their contrastive repulsion.
4. **Number of synthon encoder layers**: The number of layers in the SYNTHONENCODER used to compute synthon embeddings z_{syn}.
5. **Projection dimension**: The output dimension of the projection heads $\text{Proj}_{\text{conf}}$ and Proj_{syn}, which map conformer and synthon features into the shared contrastive embedding space.

E.6 Training Details

Conformer predictor training. The conformer predictor $p_\psi^{1|t}$ is initialized from the pretrained pose predictor of 3DSynthFlow [52] and fine-tuned on the CrossDocked dataset [13] for 30 epochs by minimizing the joint objective (Eq. (19)). All parameters, including the backbone, are jointly optimized during fine-tuning. Each CrossDocked ligand is fragmented into at most three synthons by applying 38 bimolecular Enamine reaction SMARTS, allowing the conformer predictor to train on intermediate ligand structures at different points along the synthesis trajectory. Training samples are constructed by drawing a random assembly order and a random global time t for each molecule, yielding snapshots of the ligand at different stages of incompletion. This curriculum trains the model to produce geometrically consistent binding poses even when only a subset of the final synthons has been attached.

Pocket-specific training. Following 3DSynthFlow [52], the conformer predictor is frozen after Stage 1, and the pathway generation policy π_ϕ is trained online for 1,000 iterations using the TB objective with a batch size of 64.

Pocket-conditional training. For pocket-conditional generation, the policy is trained following the same protocol as 3DSynthFlow [52], the conformer predictor is frozen and the pathway generation policy π_ϕ is trained for pocket-conditional generation. The checkpoint is selected based on sampled reward and training loss monitored on 50 randomly held-out pockets from the training set, with training running for up to 72 hours on a single NVIDIA RTX A5000 (24GB) GPU.

Computational cost. Table 10 summarizes the training time for each component. The conformer predictor is fine-tuned on 4×NVIDIA RTX A5000 (24GB) GPUs, requiring approximately 80 hours. As the conformer predictor is trained once and shared across all targets, no per-target retraining is required. The policy is trained on a single NVIDIA RTX A5000 (24GB) GPU, taking 18–24 hours per target depending on the protein pocket.

F Additional Experimental Results

F.1 Effect of contrastive fine-tuning

We decompose ligands from the CrossDocked2020 validation split using reaction-aware bond-breaking rules derived from the 38 bimolecular Enamine reaction protocols and retain the 116 ligands that decompose into exactly three synthons. For each ligand, we construct a synthon connectivity graph, with edges indicating bonds between synthons in the original molecule, and determine the assembly order through breadth-first traversal. We measure mean RMSD between the predicted

Table 11: **Conformer prediction accuracy.** Mean RMSD (Å) to the corresponding atoms of the reference ligand pose on 116 ligands from the CrossDocked2020 validation split that decompose into three synthons. Columns indicate the number of assembled synthons. CR denotes the contrastive-removed baseline. The best results are in bold.

Method	1 synthon	2 synthons	3 synthons
3DSynthFlow	3.73	3.38	3.36
CR	2.58	2.06	1.85
SYCLE	**2.22**	**1.62**	**1.42**

Table 12: **Conformer predictor hyperparameters.** Mean RMSD (Å) evaluated on the same validation ligands as in Table 11. Columns indicate the number of assembled synthons. The contrastive temperature τ_c is fixed at 0.5. The default setting is shaded.

Setting	1 synthon	2 synthons	3 synthons
Default ($d = 256$, $\tau_w = 0.3$)	2.22	1.62	1.42
Dimension ablation ($d = 128$, $\tau_w = 0.3$)	2.35	1.59	1.42
Temperature ablation ($d = 256$, $\tau_w = 0.1$)	2.40	1.56	1.30

Table 13: **Compatibility augmentation strength.** Pocket-conditional generation results on CrossDocked2020 under the high ρ setting for different values of β. Average values are reported for 4 metrics. The default setting is shaded, and the best results are in bold.

β	Vina (kcal/mol, ↓)	SA (↑)	AiZynthFinder success rate (%, ↑)	Diversity (↑)
0	-9.61	0.78	57.3	**0.55**
5	-9.69	0.79	59.6	0.52
10	**-9.80**	**0.81**	**68.8**	0.52
15	-9.77	0.77	63.8	0.54

coordinates and the corresponding atoms of the reference ligand pose after one, two, and three synthons have been assembled.

We evaluate the conformer predictor directly using intermediate structures constructed from reference ligands, rather than molecules sampled by the pathway generation policy. We compare the conformer predictors of SYCLE and a contrastive-removed baseline (CR) with the released 3DSynthFlow conformer predictor. CR is initialized from the same pretrained checkpoint as SYCLE and fine-tuned on the same data for 30 epochs, but without the contrastive loss or synthon encoder. The remaining conformer-predictor training settings are identical, allowing us to assess the contribution of contrastive learning beyond additional fine-tuning alone.

As shown in Table 11, CR improves over the released checkpoint, while SYCLE further reduces RMSD at all three stages. These results support the effectiveness of next-synthon contrastive supervision in improving intermediate and final conformer prediction.

F.2 Hyperparameter analysis

Conformer predictor hyperparameters. We examine the effects of the projection dimension d and similarity-weight temperature τ_w on conformer prediction. Starting from the default setting of $d = 256$ and $\tau_w = 0.3$, we separately reduce d to 128 or τ_w to 0.1, while keeping the remaining settings unchanged. We use the same 116 validation ligands and RMSD evaluation protocol described in Section F.1. As shown in Table 12, both variants yield RMSD values close to the default setting, suggesting limited sensitivity of conformer prediction accuracy to these choices over the tested range.

Compatibility augmentation strength. We evaluate the effect of the compatibility score coefficient $\beta \in \{0, 5, 10, 15\}$ on the pocket-conditional generation benchmark under the high ρ setting. As shown in Table 13, the default value $\beta = 10$ achieves the best average Vina score, SA score,

Table 14: **Results by molecular size.** Vina scores and AiZynthFinder success rates for generated molecules grouped by heavy-atom count in the pocket-conditional setting. Vina scores are in kcal/mol, and success rates are percentages. * indicates SYCLE groups containing fewer than 20 molecules. All summarizes the full sample used in this analysis.

	3DSynthFlow		SYCLE	
Heavy atoms	Vina (↓)	AiZynth. (↑)	Vina (↓)	AiZynth. (↑)
≤ 24	-8.07	38.5	-9.36	68.4
25–27	-8.84	37.0	-9.76	75.0
28–30	-9.16	32.9	-9.82	67.9
31–33	-9.49	36.6	-9.99	55.0
34–36	-9.55	37.3	-9.81*	38.5*
≥ 37	-9.81	35.8	-8.75*	50.0*
All	-9.42	35.9	-9.81	69.3

and AiZynthFinder success rate among the tested values. Increasing β to 15 does not provide further improvement, indicating that stronger compatibility augmentation is not necessarily beneficial. Molecular diversity remains similar across the tested settings.

F.3 Analysis by molecular size

Vina scoring can favor compounds with higher molecular weights [27, 8], motivating an analysis of performance across molecular sizes. This analysis uses molecules generated in a single sampling run under the high ρ setting. Across all molecules in this analysis, the mean heavy-atom counts are 28.2 for SYCLE and 32.0 for 3DSynthFlow. SYCLE therefore achieves better average Vina scores despite generating smaller molecules, indicating that its docking advantage is not explained by a larger average molecular size.

We further group the generated molecules by heavy-atom count. As shown in Table 14, SYCLE achieves lower Vina scores and higher AiZynthFinder success rates in each of the 25–27, 28–30, and 31–33 heavy-atom bins. These results show that its advantages persist when comparing molecules within similar size ranges. For molecules with at least 37 heavy atoms, SYCLE has a less favorable mean Vina score, but this group contains fewer than 20 SYCLE molecules, limiting the interpretation of this difference.

F.4 Pocket-specific results

We report the full results across all 15 LIT-PCBA targets in Table 15, Table 16, Table 17, and Table 18. SYCLE consistently outperforms all baselines in docking score and ligand efficiency across most of the targets.

F.5 Visualization of sample ligand generation

Figure 4 illustrates how the predicted final binding conformer $\hat{x}_1$ evolves as synthesis progresses in SYCLE, evaluated against the ALDH1 target. At each autoregressive step, a new synthon is added to the growing molecule (shown in pink in the synthesis pathway), and the flow matching model re-estimates the final 3D binding conformer conditioned on all synthons assembled so far. The predicted conformer progressively refines with each additional synthon, converging to the final conformation at $t = 1.0$. The bottom row shows the corresponding synthesis pathway.

Table 15: **Average Vina docking score.** Results for top-100 diverse molecules across all 15 LIT-PCBA targets. The best results are in bold.

Category	Method	Average Vina Docking Score (kcal/mol, ↓)				
		ADRB2	ALDH1	ESR_ago	ESR_antago	FEN1
Fragment	FragGFN	-10.19 (± 0.33)	-10.43 (± 0.29)	-9.81 (± 0.09)	-9.85 (± 0.13)	-7.67 (± 0.71)
	FragGFN+SA	-9.70 (± 0.61)	-9.83 (± 0.65)	-9.27 (± 0.95)	-10.06 (± 0.30)	-7.26 (± 0.10)
Reaction	SynNet	-8.03 (± 0.26)	-8.81 (± 0.21)	-8.88 (± 0.13)	-8.52 (± 0.16)	-6.36 (± 0.09)
	BBAR	-9.95 (± 0.04)	-10.06 (± 0.14)	-9.97 (± 0.03)	-9.92 (± 0.05)	-6.84 (± 0.07)
	SynFlowNet[a]	-10.85 (± 0.10)	-10.69 (± 0.09)	-10.44 (± 0.05)	-10.27 (± 0.04)	-7.47 (± 0.02)
	SynFlowNet[b]	-9.17 (± 0.68)	-9.37 (± 0.29)	-9.17 (± 0.12)	-9.05 (± 0.14)	-6.45 (± 0.13)
	RGFN	-9.84 (± 0.21)	-9.93 (± 0.11)	-9.99 (± 0.11)	-9.72 (± 0.14)	-6.92 (± 0.06)
	RxnFlow	-11.45 (± 0.05)	-11.26 (± 0.07)	-11.15 (± 0.02)	-10.77 (± 0.04)	-7.66 (± 0.02)
3D Reaction	3DSynthFlow	-11.96 (± 0.12)	-11.82 (± 0.03)	-11.58 (± 0.07)	-11.23 (± 0.08)	-7.79 (± 0.01)
	SYCLE	**-12.10** (± 0.09)	**-11.98** (± 0.08)	**-11.64** (± 0.21)	**-11.45** (± 0.05)	**-8.16** (± 0.10)
		GBA	IDH1	KAT2A	MAPK1	MTORC1
Fragment	FragGFN	-8.76 (± 0.46)	-9.91 (± 0.32)	-9.27 (± 0.20)	-8.93 (± 0.18)	-10.51 (± 0.31)
	FragGFN+SA	-8.92 (± 0.27)	-9.76 (± 0.64)	-9.14 (± 0.43)	-8.28 (± 0.40)	-10.14 (± 0.30)
Reaction	SynNet	-7.60 (± 0.09)	-8.74 (± 0.08)	-7.64 (± 0.38)	-7.33 (± 0.14)	-9.30 (± 0.45)
	BBAR	-8.70 (± 0.05)	-9.84 (± 0.09)	-8.54 (± 0.06)	-8.49 (± 0.07)	-10.07 (± 0.16)
	SynFlowNet[a]	-9.27 (± 0.06)	-10.40 (± 0.08)	-9.41 (± 0.04)	-8.92 (± 0.05)	-10.84 (± 0.03)
	SynFlowNet[b]	-8.28 (± 0.15)	-9.18 (± 0.35)	-8.06 (± 0.15)	-7.89 (± 0.13)	-9.60 (± 0.16)
	RGFN	-8.48 (± 0.06)	-9.49 (± 0.13)	-8.53 (± 0.11)	-8.22 (± 0.15)	-9.89 (± 0.06)
	RxnFlow	-9.62 (± 0.04)	-10.95 (± 0.05)	-9.73 (± 0.03)	-9.30 (± 0.01)	-11.39 (± 0.09)
3D Reaction	3DSynthFlow	-9.90 (± 0.14)	-11.28 (± 0.15)	-10.17 (± 0.37)	-9.61 (± 0.11)	-11.91 (± 0.01)
	SYCLE	**-10.05** (± 0.01)	**-11.69** (± 0.08)	**-10.43** (± 0.09)	**-9.76** (± 0.05)	**-12.16** (± 0.03)
		OPRK1	PKM2	PPARG	TP53	VDR
Fragment	FragGFN	-10.28 (± 0.15)	-11.24 (± 0.27)	-9.54 (± 0.12)	-7.90 (± 0.02)	-10.96 (± 0.06)
	FragGFN+SA	-9.58 (± 0.44)	-10.83 (± 0.34)	-9.19 (± 0.29)	-7.61 (± 0.27)	-10.66 (± 0.61)
Reaction	SynNet	-8.70 (± 0.36)	-9.55 (± 0.14)	-7.47 (± 0.34)	-5.34 (± 0.23)	-10.98 (± 0.57)
	BBAR	-9.84 (± 0.10)	-11.39 (± 0.08)	-8.69 (± 0.10)	-7.05 (± 0.09)	-11.07 (± 0.04)
	SynFlowNet[a]	-10.34 (± 0.07)	-11.98 (± 0.12)	-9.40 (± 0.05)	-7.90 (± 0.10)	-11.62 (± 0.13)
	SynFlowNet[b]	-9.36 (± 0.25)	-10.64 (± 0.19)	-8.25 (± 0.10)	-6.84 (± 0.06)	-10.32 (± 0.07)
	RGFN	-9.61 (± 0.11)	-10.96 (± 0.18)	-8.53 (± 0.07)	-7.07 (± 0.06)	-10.86 (± 0.11)
	RxnFlow	-10.84 (± 0.03)	-12.53 (± 0.02)	-9.73 (± 0.02)	-8.09 (± 0.06)	-12.30 (± 0.07)
3D Reaction	3DSynthFlow	-11.26 (± 0.41)	-13.36 (± 0.03)	-10.00 (± 0.04)	-8.41 (± 0.17)	-12.98 (± 0.10)
	SYCLE	**-11.58** (± 0.06)	**-13.55** (± 0.13)	**-10.21** (± 0.12)	**-9.09** (± 0.03)	**-13.29** (± 0.14)

Table 16: **Average ligand efficiency.** Results for top-100 diverse molecules across all 15 LIT-PCBA targets. The best results are in bold.

Category	Method	Average Ligand Efficiency (↑)				
		ADRB2	ALDH1	ESR_ago	ESR_antago	FEN1
Fragment	FragGFN	0.410 (± 0.006)	0.368 (± 0.007)	0.347 (± 0.003)	0.358 (± 0.002)	0.246 (± 0.004)
	FragGFN+SA	0.406 (± 0.007)	0.374 (± 0.023)	0.369 (± 0.003)	0.345 (± 0.024)	0.210 (± 0.004)
Reaction	SynNet	0.274 (± 0.041)	0.272 (± 0.006)	0.317 (± 0.005)	0.289 (± 0.020)	0.196 (± 0.003)
	BBAR	0.412 (± 0.006)	0.401 (± 0.008)	0.380 (± 0.001)	0.387 (± 0.003)	0.257 (± 0.003)
	SynFlowNet[a]	0.401 (± 0.006)	0.380 (± 0.007)	0.361 (± 0.003)	0.361 (± 0.004)	0.247 (± 0.004)
	SynFlowNet[b]	0.380 (± 0.021)	0.362 (± 0.016)	0.351 (± 0.008)	0.349 (± 0.005)	0.234 (± 0.007)
	RGFN	0.393 (± 0.005)	0.357 (± 0.004)	0.346 (± 0.002)	0.344 (± 0.002)	0.241 (± 0.001)
	RxnFlow	0.412 (± 0.005)	0.396 (± 0.005)	0.375 (± 0.002)	0.380 (± 0.004)	0.246 (± 0.001)
3D Reaction	3DSynthFlow	0.448 (± 0.019)	0.395 (± 0.006)	0.391 (± 0.005)	0.398 (± 0.016)	**0.252** (± 0.005)
	SYCLE	**0.460** (± 0.010)	**0.409** (± 0.010)	**0.399** (± 0.006)	**0.410** (± 0.007)	0.247 (± 0.002)
		GBA	IDH1	KAT2A	MAPK1	MTORC1
Fragment	FragGFN	0.333 (± 0.018)	0.367 (± 0.009)	0.322 (± 0.008)	0.302 (± 0.002)	0.354 (± 0.005)
	FragGFN+SA	0.318 (± 0.005)	0.369 (± 0.020)	0.298 (± 0.020)	0.294 (± 0.015)	0.355 (± 0.027)
Reaction	SynNet	0.244 (± 0.013)	0.281 (± 0.016)	0.294 (± 0.042)	0.226 (± 0.004)	0.316 (± 0.035)
	BBAR	0.336 (± 0.002)	0.382 (± 0.005)	0.332 (± 0.003)	0.320 (± 0.002)	0.385 (± 0.004)
	SynFlowNet[a]	0.330 (± 0.004)	0.368 (± 0.002)	0.327 (± 0.003)	0.305 (± 0.002)	0.368 (± 0.002)
	SynFlowNet[b]	0.324 (± 0.007)	0.360 (± 0.013)	0.309 (± 0.004)	0.297 (± 0.011)	0.361 (± 0.009)
	RGFN	0.310 (± 0.002)	0.351 (± 0.003)	0.310 (± 0.003)	0.298 (± 0.002)	0.346 (± 0.004)
	RxnFlow	0.327 (± 0.004)	0.378 (± 0.001)	0.330 (± 0.001)	0.313 (± 0.001)	0.370 (± 0.001)
3D Reaction	3DSynthFlow	0.340 (± 0.018)	0.388 (± 0.007)	0.340 (± 0.016)	0.322 (± 0.009)	0.387 (± 0.008)
	SYCLE	**0.346** (± 0.002)	**0.411** (± 0.011)	**0.344** (± 0.007)	**0.324** (± 0.004)	**0.393** (± 0.007)
		OPRK1	PKM2	PPARG	TP53	VDR
Fragment	FragGFN	0.352 (± 0.004)	0.442 (± 0.008)	0.319 (± 0.007)	0.307 (± 0.005)	0.394 (± 0.006)
	FragGFN+SA	0.327 (± 0.014)	0.440 (± 0.009)	0.303 (± 0.013)	0.248 (± 0.025)	0.390 (± 0.020)
Reaction	SynNet	0.298 (± 0.039)	0.296 (± 0.005)	0.253 (± 0.031)	0.211 (± 0.031)	0.359 (± 0.015)
	BBAR	0.370 (± 0.006)	0.442 (± 0.004)	0.326 (± 0.007)	0.288 (± 0.005)	0.409 (± 0.002)
	SynFlowNet[a]	0.359 (± 0.004)	0.427 (± 0.003)	0.317 (± 0.002)	0.287 (± 0.008)	0.393 (± 0.003)
	SynFlowNet[b]	0.355 (± 0.008)	0.410 (± 0.018)	0.298 (± 0.010)	0.282 (± 0.005)	0.380 (± 0.005)
	RGFN	0.349 (± 0.001)	0.405 (± 0.002)	0.307 (± 0.002)	0.271 (± 0.001)	0.381 (± 0.002)
	RxnFlow	0.369 (± 0.007)	0.436 (± 0.005)	0.319 (± 0.002)	0.289 (± 0.003)	0.405 (± 0.002)
3D Reaction	3DSynthFlow	0.389 (± 0.006)	0.444 (± 0.010)	0.321 (± 0.005)	0.294 (± 0.006)	0.416 (± 0.009)
	SYCLE	**0.421** (± 0.005)	**0.460** (± 0.018)	**0.326** (± 0.011)	**0.329** (± 0.002)	**0.433** (± 0.002)

Table 17: **Average AiZynthFinder success rate.** Results for top-100 diverse molecules across all 15 LIT-PCBA targets. The best results are in bold.

Category	Method	AiZynthFinder Success Rate (%, ↑)				
		ADRB2	ALDH1	ESR_ago	ESR_antago	FEN1
Fragment	FragGFN	4.00 (± 3.54)	3.75 (± 1.92)	1.00 (± 1.00)	3.75 (± 1.92)	0.25 (± 0.43)
	FragGFN+SA	5.75 (± 1.48)	6.00 (± 2.55)	4.00 (± 2.24)	1.00 (± 0.00)	0.00 (± 0.00)
Reaction	SynNet	54.17 (± 7.22)	50.00 (± 0.00)	50.00 (± 0.00)	25.00 (± 25.00)	50.00 (± 0.00)
	BBAR	21.25 (± 5.36)	19.50 (± 3.20)	17.50 (± 1.50)	19.50 (± 3.64)	20.00 (± 2.12)
	SynFlowNet[a]	52.75 (± 1.09)	57.00 (± 6.04)	53.75 (± 9.52)	56.50 (± 2.29)	53.00 (± 8.92)
	SynFlowNet[b]	56.50 (± 6.58)	56.00 (± 3.08)	61.00 (± 2.74)	64.50 (± 9.86)	60.75 (± 3.77)
	RGFN	46.75 (± 6.86)	47.50 (± 4.06)	50.25 (± 2.17)	49.25 (± 4.38)	48.50 (± 6.58)
	RxnFlow	60.25 (± 3.77)	63.25 (± 3.11)	71.25 (± 4.15)	66.50 (± 4.03)	**65.50** (± 4.09)
3D Reaction	3DSynthFlow	**85.00** (± 7.87)	69.33 (± 3.86)	**80.00** (± 4.24)	69.67 (± 14.06)	60.00 (± 15.64)
	SYCLE	81.00 (± 3.00)	**72.33** (± 3.51)	75.67 (± 3.51)	**74.00** (± 3.46)	58.67 (± 6.03)

		GBA	IDH1	KAT2A	MAPK1	MTORC1
Fragment	FragGFN	5.00 (± 4.24)	4.50 (± 1.66)	1.25 (± 0.83)	0.75 (± 0.83)	2.75 (± 1.30)
	FragGFN+SA	3.00 (± 1.00)	4.50 (± 4.97)	1.50 (± 0.50)	3.25 (± 1.48)	3.50 (± 2.50)
Reaction	SynNet	50.00 (± 0.00)	50.00 (± 0.00)	45.83 (± 27.32)	50.00 (± 0.00)	54.17 (± 7.22)
	BBAR	17.75 (± 2.28)	19.50 (± 1.50)	18.75 (± 1.92)	16.25 (± 3.49)	18.75 (± 3.90)
	SynFlowNet[a]	58.00 (± 4.64)	59.00 (± 4.06)	55.50 (± 10.23)	47.25 (± 6.61)	57.00 (± 7.58)
	SynFlowNet[b]	61.50 (± 3.84)	60.50 (± 3.91)	57.25 (± 4.97)	44.50 (± 9.29)	62.00 (± 1.22)
	RGFN	48.00 (± 1.22)	43.00 (± 2.74)	49.00 (± 1.22)	42.00 (± 3.00)	44.50 (± 4.03)
	RxnFlow	66.00 (± 1.58)	64.00 (± 5.05)	66.50 (± 2.06)	**63.00** (± 4.64)	**70.50** (± 2.87)
3D Reaction	3DSynthFlow	64.67 (± 3.09)	65.67 (± 13.91)	**68.00** (± 19.51)	54.67 (± 10.50)	64.67 (± 15.69)
	SYCLE	**68.33** (± 6.43)	**68.67** (± 8.39)	59.00 (± 3.46)	60.67 (± 6.81)	61.00 (± 5.57)

		OPRK1	PKM2	PPARG	TP53	VDR
Fragment	FragGFN	2.50 (± 2.29)	8.75 (± 3.11)	0.75 (± 0.43)	4.25 (± 1.64)	3.50 (± 2.18)
	FragGFN+SA	3.25 (± 1.79)	9.75 (± 2.28)	1.25 (± 1.09)	2.25 (± 1.92)	3.75 (± 2.77)
Reaction	SynNet	54.17 (± 7.22)	50.00 (± 0.00)	54.17 (± 7.22)	29.17 (± 18.16)	45.83 (± 7.22)
	BBAR	13.75 (± 3.11)	20.00 (± 0.71)	15.50 (± 2.29)	18.50 (± 3.28)	12.25 (± 3.34)
	SynFlowNet[a]	56.50 (± 7.63)	50.75 (± 1.09)	53.50 (± 5.68)	55.50 (± 9.94)	53.50 (± 1.80)
	SynFlowNet[b]	56.25 (± 2.49)	58.00 (± 7.00)	57.00 (± 5.74)	66.50 (± 6.80)	53.50 (± 3.84)
	RGFN	48.00 (± 2.55)	48.50 (± 3.20)	47.00 (± 5.83)	53.25 (± 3.63)	46.50 (± 2.69)
	RxnFlow	**72.25** (± 2.05)	62.00 (± 3.24)	65.50 (± 4.03)	67.50 (± 2.96)	66.75 (± 2.28)
3D Reaction	3DSynthFlow	67.33 (± 20.37)	**78.67** (± 8.73)	65.67 (± 8.18)	61.33 (± 11.73)	**74.00** (± 17.15)
	SYCLE	71.33 (± 9.87)	75.33 (± 3.79)	**69.33** (± 6.66)	**79.00** (± 7.55)	73.33 (± 6.51)

Table 18: **Average number of synthesis steps.** Results for top-100 diverse molecules across all 15 LIT-PCBA targets. The best results are in bold.

Category	Method	Average Number of Synthesis Steps (↓)				
		ADRB2	ALDH1	ESR_ago	ESR_antago	FEN1
Fragment	FragGFN	3.60 (± 0.10)	3.83 (± 0.08)	3.76 (± 0.20)	3.76 (± 0.16)	3.34 (± 0.18)
	FragGFN+SA	3.73 (± 0.09)	3.66 (± 0.04)	3.66 (± 0.07)	3.67 (± 0.21)	3.79 (± 0.19)
Reaction	SynNet	3.29 (± 0.36)	3.50 (± 0.00)	3.00 (± 0.00)	4.13 (± 0.89)	3.50 (± 0.00)
	BBAR	3.60 (± 0.17)	3.62 (± 0.19)	3.76 (± 0.04)	3.72 (± 0.11)	3.59 (± 0.14)
	SynFlowNet[a]	2.64 (± 0.07)	2.48 (± 0.07)	2.60 (± 0.25)	2.45 (± 0.09)	2.56 (± 0.29)
	SynFlowNet[b]	2.42 (± 0.10)	2.48 (± 0.10)	2.38 (± 0.10)	2.34 (± 0.30)	2.41 (± 0.14)
	RGFN	2.88 (± 0.21)	2.65 (± 0.09)	2.78 (± 0.19)	2.91 (± 0.23)	2.76 (± 0.17)
	RxnFlow	2.42 (± 0.23)	2.19 (± 0.12)	1.95 (± 0.20)	2.15 (± 0.18)	2.23 (± 0.18)
3D Reaction	3DSynthFlow	1.77 (± 0.37)	2.23 (± 0.16)	2.02 (± 0.20)	2.43 (± 0.61)	2.73 (± 0.50)
	SYCLE	**1.27** (± 0.06)	**1.50** (± 0.27)	**1.54** (± 0.30)	**1.40** (± 0.12)	**1.97** (± 0.23)
		GBA	IDH1	KAT2A	MAPK1	MTORC1
Fragment	FragGFN	3.94 (± 0.11)	3.74 (± 0.10)	3.78 (± 0.09)	3.72 (± 0.18)	3.84 (± 0.18)
	FragGFN+SA	3.94 (± 0.15)	3.84 (± 0.23)	3.66 (± 0.18)	3.69 (± 0.21)	3.94 (± 0.08)
Reaction	SynNet	3.38 (± 0.22)	3.38 (± 0.22)	3.46 (± 0.95)	3.50 (± 0.00)	3.29 (± 0.36)
	BBAR	3.71 (± 0.12)	3.68 (± 0.02)	3.63 (± 0.05)	3.73 (± 0.05)	3.77 (± 0.09)
	SynFlowNet[a]	2.48 (± 0.18)	2.61 (± 0.13)	2.45 (± 0.37)	2.81 (± 0.24)	2.44 (± 0.27)
	SynFlowNet[b]	2.45 (± 0.08)	2.46 (± 0.12)	2.45 (± 0.12)	2.83 (± 0.27)	2.39 (± 0.17)
	RGFN	2.77 (± 0.20)	2.97 (± 0.15)	2.78 (± 0.10)	2.86 (± 0.19)	2.92 (± 0.06)
	RxnFlow	2.10 (± 0.08)	2.16 (± 0.11)	2.29 (± 0.05)	2.29 (± 0.11)	2.05 (± 0.09)
3D Reaction	3DSynthFlow	2.47 (± 0.24)	2.49 (± 0.42)	2.47 (± 0.61)	2.70 (± 0.33)	2.42 (± 0.61)
	SYCLE	**1.67** (± 0.05)	**1.35** (± 0.14)	**1.92** (± 0.44)	**1.65** (± 0.47)	**1.60** (± 0.25)
		OPRK1	PKM2	PPARG	TP53	VDR
Fragment	FragGFN	3.82 (± 0.13)	3.71 (± 0.12)	3.73 (± 0.24)	3.73 (± 0.23)	3.75 (± 0.06)
	FragGFN+SA	3.62 (± 0.12)	3.84 (± 0.21)	3.71 (± 0.04)	3.66 (± 0.05)	3.67 (± 0.25)
Reaction	SynNet	3.29 (± 0.36)	3.50 (± 0.00)	3.29 (± 0.36)	3.67 (± 0.91)	3.63 (± 0.22)
	BBAR	3.70 (± 0.17)	3.61 (± 0.05)	3.72 (± 0.13)	3.65 (± 0.05)	3.77 (± 0.16)
	SynFlowNet[a]	2.49 (± 0.33)	2.62 (± 0.10)	2.56 (± 0.12)	2.51 (± 0.27)	2.55 (± 0.09)
	SynFlowNet[b]	2.50 (± 0.11)	2.52 (± 0.20)	2.53 (± 0.06)	2.34 (± 0.10)	2.51 (± 0.17)
	RGFN	2.81 (± 0.12)	2.82 (± 0.10)	2.82 (± 0.18)	2.64 (± 0.10)	2.84 (± 0.18)
	RxnFlow	2.00 (± 0.09)	2.34 (± 0.19)	2.21 (± 0.06)	2.12 (± 0.12)	2.12 (± 0.12)
3D Reaction	3DSynthFlow	2.34 (± 0.53)	2.20 (± 0.41)	2.61 (± 0.39)	2.73 (± 0.47)	2.25 (± 0.78)
	SYCLE	**1.42** (± 0.13)	**1.55** (± 0.18)	**1.90** (± 0.28)	**1.35** (± 0.09)	**1.55** (± 0.20)

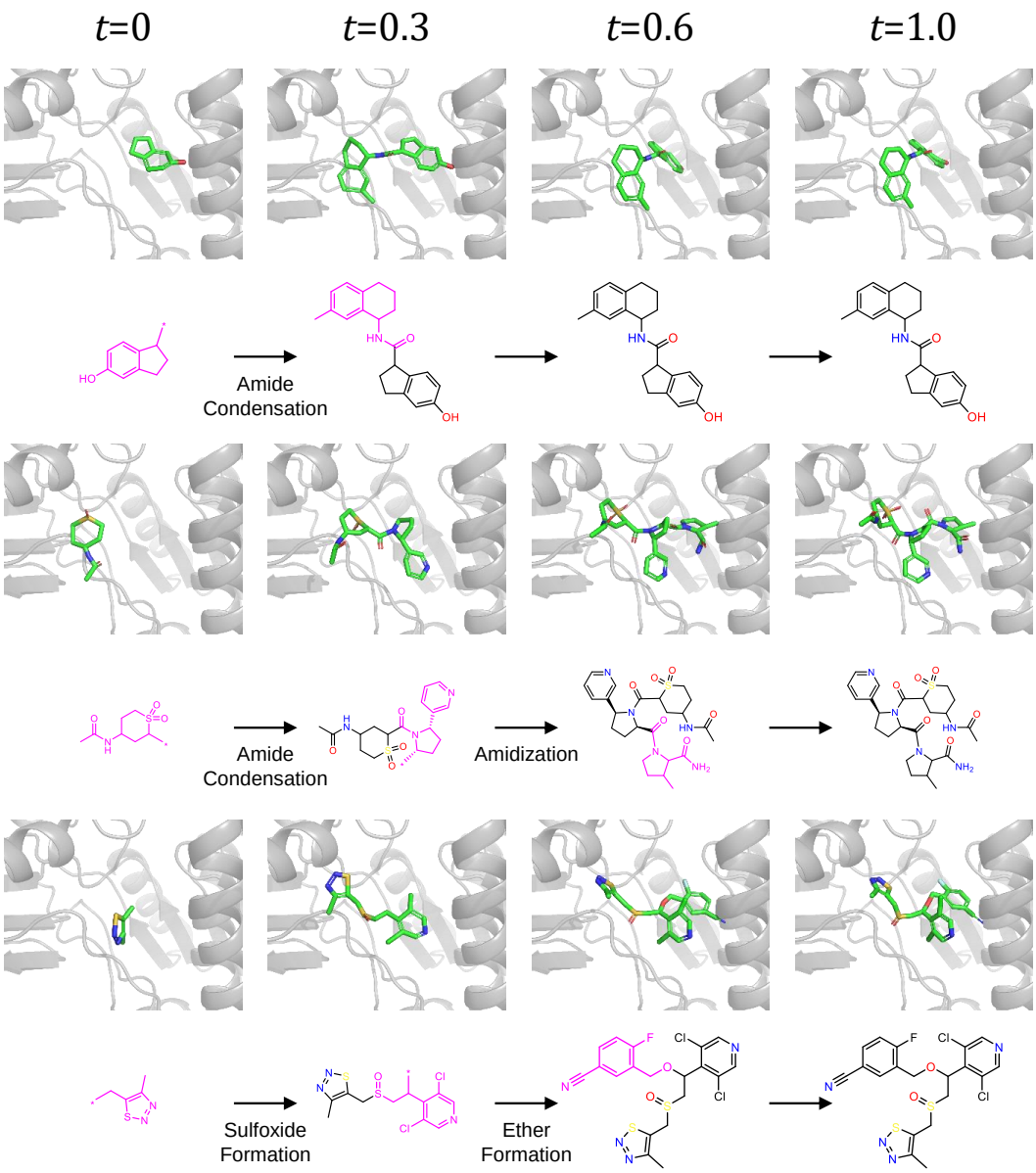

Figure 4: **Ligand generation trajectories of SYCLE.**

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and Section 1 clearly state the three contributions, which are validated by main tables.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Limitations are discussed in Section A.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: Proposition with full proof is provided in Appendix C.2. The underlying determinism assumption is stated in Appendix C.1.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Full training details, hyperparameters, and dataset descriptions are provided in Appendix E.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: Source code, training scripts, and configuration files are provided in the supplementary material. We use publicly available datasets described in Appendix E.3. Instructions for accessing and preparing the data and running the experiments are included with the code.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: Full hyperparameters and training/evaluation protocols are described in Section 4, Appendix E.5 and E.6

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: We report mean $\pm$ standard deviation for SYCLE across three independent training runs in the pocket-specific setting (Tables 1 and 3) and three sampling runs from the same trained checkpoint in the pocket-conditional setting (Table 4). Before-redocking comparisons in Tables 1 and 3 use a single shared random seed. Error bars are not reported for these comparisons or the additional ablation analyses.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Compute resources are summarized in Table 10, including GPU type, batch size, and training duration for each component.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: This research conforms to the NeurIPS Code of Ethics. Our work uses only publicly available datasets (CrossDocked2020, LIT-PCBA) with no human subjects or sensitive data involved. The proposed method targets drug discovery applications with potential positive societal impact.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: Broader impacts including potential misuse are discussed in Section B.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [N/A]

Justification: The proposed model is designed for structure-based drug design and does not pose significant risks for misuse beyond standard scientific use cases.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: All external datasets and tools (CrossDocked2020, Enamine catalog, AiZynthFinder, RDKit) are properly cited.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: The SYCLE codebase is provided in the supplementary material, together with documentation, training configurations, and instructions for reproducing the experiments.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: No human subjects research was conducted.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs were not used as part of the core methodology.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.