TurboHopp: Accelerated Molecule Scaffold Hopping with Consistency Models

Kiwoong Yoo[1], **Owen Oertell**[2], **Junhyun Lee**[3], **Sanghoon Lee**[1,3], and **Jaewoo Kang**[*1,3]

[1]AIGEN Sciences
[2]Cornell University
[3]Korea University

kiwoong.yoo@aigensciences.com, ojo2@cornell.edu, ljhyun33@korea.ac.kr,
sanghoon.lee@aigensciences.com, kangj@korea.ac.kr

Abstract

Navigating the vast chemical space of druggable compounds is a formidable challenge in drug discovery, where generative models are increasingly employed to identify viable candidates. Conditional 3D structure-based drug design (3D-SBDD) models, which take into account complex three-dimensional interactions and molecular geometries, are particularly promising. Scaffold hopping is an efficient strategy that facilitates the identification of similar active compounds by strategically modifying the core structure of molecules, effectively narrowing the wide chemical space and enhancing the discovery of drug-like products. However, the practical application of 3D-SBDD generative models is hampered by their slow processing speeds. To address this bottleneck, we introduce TurboHopp, an accelerated pocket-conditioned 3D scaffold hopping model that merges the strategic effectiveness of traditional scaffold hopping with rapid generation capabilities of consistency models. This synergy not only enhances efficiency but also significantly boosts generation speeds, achieving up to 30 times faster inference speed as well as superior generation quality compared to existing diffusion-based models, establishing TurboHopp as a powerful tool in drug discovery. Supported by faster inference speed, we further optimize our model, using Reinforcement Learning for Consistency Models (RLCM), to output desirable molecules. We demonstrate the broad applicability of TurboHopp across multiple drug discovery scenarios, underscoring its potential in diverse molecular settings. The code is provided at https://github.com/orgw/TurboHopp

1   Introduction

The vast chemical space, with up to 10^{63} potential molecules (molecular weight under $500\,\mathrm{Da}$) (Kirkpatrick and Ellis [2004]), presents immense challenges in drug discovery, where the process often extends over a decade and incurs costs in the billions. Recent strides in molecular design have introduced advanced generative models that excel in de-novo generation and optimization within 3D molecular space (Isert et al. [2023]). Despite these technological advances, challenges persist, such as ensuring synthesizability (Gao et al. [2021], Fialková et al. [2021]) and maintaining desirable drug-like properties. The scarcity of protein-ligand complex data further complicates these demanding issues. While 1D/2D models attempt to address these challenges though various methods (Loeffler et al. [2024]), they often fall short in representing true 3D structures, and typically

---

[*]Corresponding author.

necessitating additional steps for proper conformation generation. On the other hand, 3D models (Ragoza et al. [2022], Luo et al. [2022], Baillif et al. [2023], Peng et al. [2022]), which directly incorporate spatial structures, show greater promise in activity prediction regarding 3D structures but are hindered by significantly slower inference speeds (Baillif et al. [2023]), which limit their practical applications such as fine-tuning via Reinforcement Learning (Ding and Jin [2024], Baillif et al. [2023], Wallace et al. [2023]). Recent research efforts, such as those by Lee and Cho [2024], have started fine-tuning 3D-autoregressive models with specialized reward functions to align with these stringent standards. Yet, notably absent are attempts involving Structure-Based Drug Design Diffusion Models(3D-SBDD DMs), as the prolonged inference times remain a formidable barrier, precluding the adoption of similar methodologies. In this work, we address this gap by successfully applying reinforcement learning to 3D-SBDD DMs, overcoming the challenge of prolonged inference times and demonstrating the feasibility of such approaches in practical drug design scenarios.

3D-SBDD-DMs (Guan et al. [2023, 2024], Schneuing et al. [2022], Torge et al. [2023], Igashov et al. [2024]) exhibit powerful generative capabilities but suffer from slow processing speeds due to their reliance on iterative sampling. Addressing these speed limitations is critical for several reasons. Primarily, faster processing speeds are essential for accelerating hit discovery phases, which could significantly increase the likelihood of identifying quality hits within a shorter time period. Additionally, the high computational demands of these models not only escalate costs but also limit access to advanced optimization techniques, thereby restricting the generation of molecules with desirable properties such as synthesizability, high binding affinity, and drug-likeness. Furthermore, real-time applications, such as interactive modeling sessions with human experts, require rapid model inference to provide effective feedback and facilitate iterative testing (Sundin et al. [2022]). By enhancing the speed of these models, it may be possible to enable real-time applications, allowing for on-the-fly modifications and adaptations during interactive sessions. Addressing the speed limitations of diffusion models is thus essential to fully realize these objectives and leverage the full potential of 3D-SBDD-DMs in practical applications.

To address slow inference of diffusion models, various strategies have been adopted to enhance their efficiency, in the computer vision domain such as improvement of solvers, optimization of diffusion processes, and model distillation techniques (Cao et al. [2024]). The emergence of consistency models and its variants (Song and Dhariwal [2023], Ding and Jin [2024], Song et al. [2023]), heralds a rapid, efficient alternative capable of delivering high-quality results with significantly reduced computational demands. However, these acceleration attempts have been minimally explored in the context of 3D drug design.

In this work, we present an efficient E(3)-equivariant consistency model that combines the rapid generation capabilities of consistency models with the effectiveness of scaffold hopping (Fig. 1). To substantiate the reliability and robustness of model performance, subsequent sections will detail results and comparisons with established diffusion and consistency models. Moreover, we incorporate reinforcement learning (RL) techniques (Oertell et al. [2024]) to further refine our model based on objectives that are challenging to address through direct modeling alone, demonstrating its capability to surmount limitations encountered by previous 3D-SBDD models (Harris et al. [2023]).

Our contributions are as follows:

- We propose a target-aware equivariant consistency model tailored for scaffold-hopping, achieving speeds up to 30 times faster than traditional DDPM-based models while achieving higher drug-likeness, synthesizability, connectivity and binding affinity.

- By leveraging faster inference speeds, we apply Reinforcement Learning (RL) to enhance binding affinity scores and reduce steric clashes without redocking in 3D-SBDD—a strategy not previously explored with diffusion models. This approach enables targeted optimizations in molecule design and opens up opportunities to tackle metrics that were previously challenging to address through direct modeling.

a) Diffusion Model + De Novo generation   b) Consistency Model + Scaffold Hopping
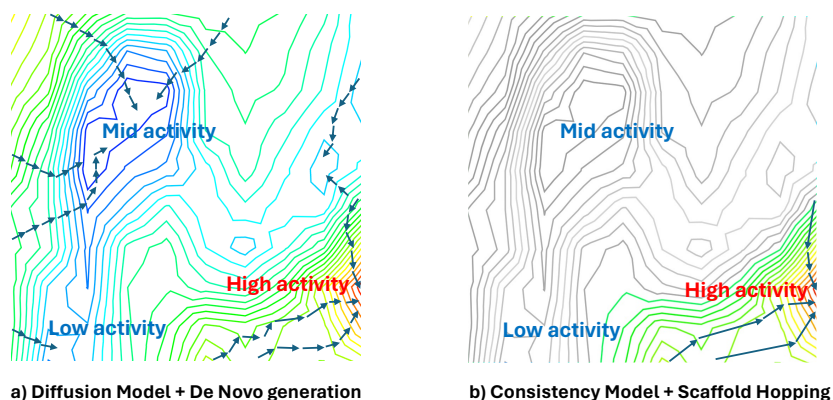
Figure 1: (a) Previous diffusion-based SBDD models methodically explore vast chemical space for pocket-active molecules, with short arrows symbolizing a gradual, step-wise inference process. (b) TurboHopp efficiently generates active ligands using a consistency model, which accelerates inference and reduces the number of steps, as illustrated by the longer arrows. Moreover, it strategically leverages the functional groups of high-activity reference molecules, shown as colored areas on the diagram, to optimize the exploration within targeted chemical space.

a) Comparison of QED over Generation Steps b) Progression of Generated Outputs over Generation Steps
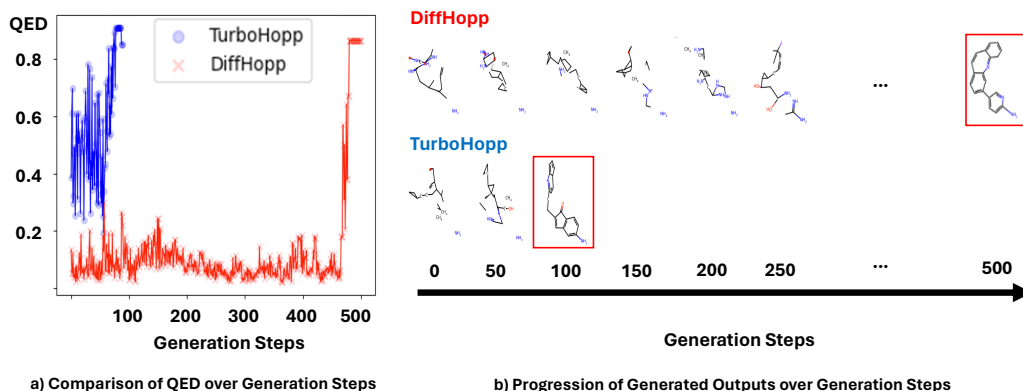
Figure 2: (a) Comparison of QED scores, with TurboHopp (blue) reaching peak values faster compared to DiffHopp (red), throughout the generation process. (b) Progression of generated outputs of both models. Final steps are highlighted with red boxes.

2 Related Works

2.1 Scaffold Hopping

Scaffold hopping is a key strategy in drug discovery aimed at identifying new compounds that share biological activities but differ structurally from known molecules. This approach is crucial for enhancing drug properties and navigating intellectual property challenges. It includes techniques like heterocycle replacements, ring modifications, peptidomimetics for stability, and topology-based hopping, which innovates on molecular shape while maintaining activity (Sun et al. [2012]). Deep-learning models, particularly in topology-based hopping, explore complex molecular structures to predict novel scaffolds, driving forward drug discovery by enabling the design of therapeutics with optimized efficacy and patentability (Torge et al. [2023]).

2.2 Diffusion-Based Molecular Generative Models and Limitations

Diffusion-based molecular generative models (DMs) apply noise processes and their reversal to generate novel molecular structures. These range from de-novo, non-conditional molecule creation to

precision drug design within protein pockets, and extend to specific tasks like scaffold hopping and linker generation (Hoogeboom et al. [2022], Vignac et al. [2023], Huang et al. [2022], Guan et al. [2023, 2024], Huang et al. [2024], Torge et al. [2023], Igashov et al. [2022]).

Despite their broad applicability, these models encounter substantial constraints. First, the iterative reversal of noise through SDEs or ODEs often requires extensive computation, typically hundreds to thousands of iterations, resulting in slow model inference. This significantly constrains the speed at which new, active molecules can be identified, diminishing the practical efficacy of these models in time-sensitive applications (Baillif et al. [2023]). Second, diffusion models often produce molecular poses that fail to comply with key biophysical constraints, as highlighted by recent evaluations (Harris et al. [2023]). Guan et al. [2023, 2024] struggle significantly in generating biophysically realistic molecular poses. Their strain energies are notably high, with median values around 1241.7 kcal/mol and 1243.1 kcal/mol, substantially exceeding the preferable baseline of 102.5 kcal/mol from the CrossDocked dataset (Francoeur et al. [2020]). Additionally, both models exhibit considerable issues with protein clashes: Schneuing et al. [2022] shows a high rate of steric clashes with a mean score of 15.33 before redocking, and Guan et al. [2023] also has a problematic level, with a mean score of 9.08 prior to redocking. These high levels of steric clashes suggest that the molecules often have conformations that are too cramped or improperly aligned for effective protein binding, highlighting significant challenges in their ability to produce physically and biologically plausible interactions.

2.3 Consistency Models

Initially utilized in the vision domain, consistency models are either developed by distilling knowledge from existing diffusion models or trained as standalone entities. Defined as a distinct class of generative models (Song et al. [2023]), they address the computational inefficiencies of traditional diffusion models that require multiple iterative steps to remove noise and generate data. By directly transforming noise into data, consistency models significantly reduce generation time and offer the flexibility of multi-step sampling to improve output quality.

2.4 RL for Consistency Models

Reinforcement Learning for Consistency Models (RLCM) is a framework that enhances the speed and efficacy of training generative models through reinforcement learning (Oertell et al. [2024]). It builds upon the fast inference speed of consistency models, enabling rapid training and inference cycles. RLCM is particularly useful as it fine-tunes models to optimize for specific reward functions, especially those hard to capture through direct modeling. In the molecule generation domain, to our knowledge there have been no attempts at fine-tuning diffusion or consistency models for task-specific purposes yet. By leveraging RLCM, we show promising evidence for mitigating issues in 3D-SBDD DMs (Harris et al. [2023]), and believe that future iteration on the reward functions with our methods will enhance their practical applicability and effectiveness.

3 Methods

3.1 Notations and Problem Definition

Notations Atoms within proteins and ligands are represented by the graph $G = \{h, x\}$, where $h \in \mathbb{R}^{N \times F}$ are node features and $x \in \mathbb{R}^{N \times 3}$ are the Cartesian coordinates of the atoms. The atomic elements are encoded as one-hot vectors within h. We consider only those protein pocket atoms that are within an 8 Ångström radius of the ligand, simplifying the protein structure to a C_α representation with node features h_P encoding residue types as one-hot vectors. Edges between atoms are defined for those within a 5 Ångström radius, incorporating both the distance d_{ij} and a normalized direction vector $\frac{x_i - x_j}{d_{ij}}$.

Problem Definition Given a context u which integrates the functional groups of a molecule denoted as g, and a protein pocket p, our focus is to generate a scaffold, a key substructure of a molecule Z. A generative model, defined as $P_\theta(Z|u)$ maps uniform Gaussian noise to conditional probability distribution Z given u. Ultimately, we aim to make sampling of scaffolds for any given molecular context u within the set as efficient as possible, without losing overall quality.

Dataset We follow the dataset preprocessing scheme regarding filtering of compounds and determining of scaffolds as done in Torge et al. [2023], filtering those above QED of 0.3, training on 19,378 protein-ligand complexes in PDBBind. We also adopt the same scaffold extraction method, using Murko-Bemis method (Bemis and Murcko [1996]).

3.2 Background

Diffusion Models In DDPMs, noise is incrementally added to the original data, represented as $\mathbf{Z}_0$, until it is completely transformed into a noise-dominated state $\mathbf{Z}_T$ after T steps. This process is governed by an SDE:

$$d\mathbf{Z} = \mathbf{f}(\mathbf{Z}, t)dt + \mathbf{u}(t)d\mathbf{w} \tag{1}$$

where $\mathbf{w}$ represents standard Brownian motion, $\mathbf{f}(\mathbf{Z}, t)$ is a drift function, and $\mathbf{u}(t)$ is a diffusion coefficient associated with the noise process. The reverse diffusion process is described by another SDE which incorporates the backward flow of time through $\tilde{\mathbf{w}}$, a Brownian motion in reverse, and modifies the forward SDE with a term involving the gradient of the log probability density.:

$$d\mathbf{Z} = [\mathbf{f}(\mathbf{Z}, t) - \mathbf{u}(t)^2 \nabla_{\mathbf{Z}} \log p_\theta(\mathbf{Z})]dt + \mathbf{u}(t)d\tilde{\mathbf{w}} \tag{2}$$

The denoising process can be mathematically framed as an ordinary differential equation (ODE), also known as the Probability Flow ODE (Song et al. [2020]), whose solution trajectories have the same marginal density at time t:

$$d\mathbf{Z} = [\mathbf{f}(\mathbf{Z}, t) - \frac{1}{2}\mathbf{u}(t)^2 \nabla_{\mathbf{Z}} \log p_\theta(\mathbf{Z})]dt \tag{3}$$

During training phase, Diffusion Models (DMs) learn to estimate the score, given by $\nabla_{\mathbf{Z}} \log p_\theta(\mathbf{Z})$ with a score model $\mathbf{s}_\theta(\mathbf{Z}, t)$. Following the modifications by Karras et al. [2022], the drift term $\mathbf{f}(\mathbf{Z}, t)$ is set to zero and the diffusion term is defined as $\mathbf{u}(t) = \sqrt{2t}$. Consequently, the empirical Probability Flow ODE becomes:

$$\frac{d\mathbf{Z}}{dt} = -t\mathbf{s}_\theta(\mathbf{Z}, t) \tag{4}$$

This form is used to solve the ODE backwards in time, starting from a noise state $\tilde{\mathbf{Z}}$ sampled from $\mathcal{N}(0, t^2\mathbf{I})$. Using numerical ODE solvers like Euler and Heun, the model computes the trajectory $\{\mathbf{Z}_t\}_{t \in [0,T]}$, resulting in $\tilde{\mathbf{Z}}_0$ as an approximate reconstruction of the original data distribution.

Consistency Function We develop an equivariant consistency function $f : (Z_t, t \mid u) \rightarrow Z_\epsilon$ for a solution trajectory Z_t of an ODE across $t \in [\epsilon, T]$, where ϵ is near zero. This function meets both self-consistency, ensuring $f(Z_t, t \mid u)$ is uniform for all (Z_t, t) on the trajectory, and boundary conditions, mandating $f(Z_\epsilon, \epsilon) = Z_\epsilon$ at the smallest time step ϵ. Successfully achieving these conditions guarantees that each solution Z_t can be traced back to its original state Z_0. The function maintains SE(3)-equivariance, adjusting centers as per methodologies outlined in Xu et al. [2022], and outputs both coordinates and node features.

3.3 Model Architecture

We wish to construct a new sample scaffold Z given a molecular context u (u is the concatenation of the pocket p and functional groups g). An equivariant consistency function $f_\theta(Z|u)$ is parameterized by a free-form deep neural network F_θ which is a (Z, t) related SE (3)-equivariant model. Theoretically, it can be any architecture that is SE(3)-equivariant. We use an adaptation of the equivariant Geometric Vector Perceptron (GVP) architecture (Jing et al. [2020]). As done in (Torge et al. [2023]), all features are embedded into a shared feature space using separate MLPs. The time factor t is embedded with a linear layer and added together as the input. As done in Torge et al. [2023], hidden node features h' and x' are obtained though message-passing layers. The outputs of the consistency function are $x'_t, h'_t = x'_z, \phi_{\text{out}}(h'_Z)$ where ϕ_{out} is an MLP to map from embedding space to original data space. As shown in Eq. (5), we separate coordinates and node features and parameterize the consistency model with a learnable function f_θ using skip connections.

$$f_\theta(x_t, t \mid u) = z_{\text{skip}}(t)x_t + z_{\text{out}}(t)x'_t \qquad f_\theta(h_t, t \mid u) = z_{\text{skip}}(t)h_t + z_{\text{out}}(t)h'_t \tag{5}$$
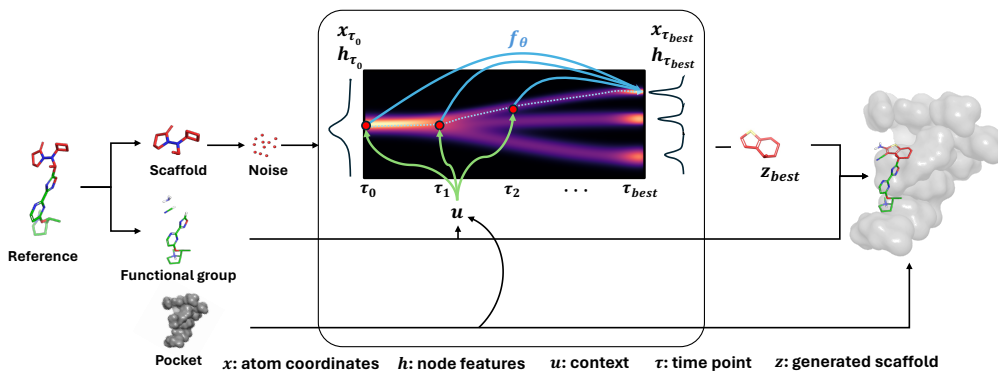
Figure 3: Model Architecture of TurboHopp. Given a reference ligand and its corresponding protein pocket, an equivariant consistency model samples scaffolds conditioned on pocket substructure and functional groups. Models are trained to map points on the same PF-ODE path to the original data given context.

$$z_{\text{skip}}(t) = \frac{\sigma_{\text{data}}^2}{(t-\epsilon)^2 + \sigma_{\text{data}}^2} \qquad z_{\text{out}}(t) = \frac{\sigma_{\text{data}}(t-\epsilon)}{\sqrt{(t-\epsilon)^2 + \sigma_{\text{data}}^2}} \qquad (6)$$

$z_{\text{skip}}(t)$ and $z_{\text{out}}(t)$ are differentiable time-dependent functions satisfying $z_{\text{skip}}(\epsilon) = 1$ and $z_{\text{out}}(\epsilon) = 0$, thus ensuring the function f_θ adheres to the boundary condition and retains differentiability. Given σ_{data} as the standard deviation of data, when $t = \epsilon$, setting $z_{\text{skip}}(\epsilon) = 1$ and $z_{\text{out}}(\epsilon) = 0$ ensures that the condition $f_\theta(Z_\epsilon) = Z_\epsilon$ is naturally satisfied. The functions z_{skip}, z_{out}, and F_θ are differentiable. As stated in Fan et al. [2023a], since z_t and F_θ are both SE-(3) equivariant, and $z_{\text{skip}}(t)$ and $z_{\text{out}}(t)$ are invariant scalars , the SE (3)-equivariance is guaranteed.

3.4 Model Training

We apply Algorithm 1 for training our consistency model. Scaffold Z extracted from the dataset is segmented into its coordinate and feature components, x and h, respectively. Next, adjustments are made to the coordinates and features along the ODE trajectory $\{x_t\}_{t \in [\epsilon, T]}$ and $\{h_t\}_{t \in [\epsilon, T]}$, respectively, by incorporating perturbations ϵ_x and ϵ_h. The loss function $L(\theta)$ reduces discrepancies between successive time steps $f_\theta(Z_{n+1}, t_{n+1}|u)$ and $f_\theta(Z_n, t_n|u)$, ensuring self-consistency.

Let $f_\theta^{n,x}$ and $f_\theta^{n,h}$ denote the coordinate and feature components of $f_\theta(Z_n, t_n|u)$, respectively. Similarly, let $f_\theta^{n+1,x}$ and $f_\theta^{n+1,h}$ denote the coordinate and feature components of $f_\theta(Z_{n+1}, t_{n+1}|u)$. The loss function $L(\theta)$ is defined as:

$$L(\theta) = \text{MSE}(f_\theta^{n+1,x}, f_\theta^{n,x}) + \text{MSE}(f_\theta^{n+1,h}, f_\theta^{n,h}) \qquad (7)$$

To enhance the stability of the training process for f_θ, a secondary function f_{θ^-}, which utilizes parameters θ^- as the exponential moving average (EMA) of θ, is employed. This methodology effectively reduces the variability between consecutive predictions by accounting for incremental perturbations, as detailed in Song and Dhariwal [2023]. The combined loss function measures the mean squared error (MSE) between adjacent states for both coordinates x and features h:

Parameter updates are conducted using stochastic gradient descent for θ, while θ^- is incrementally adjusted using an EMA protocol where μ indicates the decay rate, enhancing the model's responsiveness and accuracy.

$$L(\theta, \theta^-) = \text{MSE}(f_\theta^{n+1,x}, f_{\theta^-}^{n,x}) + \text{MSE}(f_\theta^{n+1,h}, f_{\theta^-}^{n,h}) \qquad (8)$$

$$\theta^- = \mu\theta^- + (1-\mu)\theta \qquad (9)$$

Table 1: **DiffHopp VS TurboHopp VS TurboHopp Metric Sampling** .Results on models trained on PDBBind. Mean and standard deviation of the common molecular metrics for the molecules from the baseline models as well as time-step variations of our model."metric" refers that inference was done with metric-based sampling. QVina score (kcal/mol) refers to estimated binding affinity measured by QVina2. Time refers to average time (seconds) required to generate a batch of molecules per complex. Best metrics are in bold. 2nd best are underlined.

Method	Connectivity (↑)	Diversity (↑)	Novelty (↑)	QED (↑)	SA (↑)	QVina (↓)	Steps	Time
DiffHopp	0.918±0.23	0.589±0.17	0.999±0.01	0.621±0.12	0.662±0.14	-7.923±3.29	500	107.10
DiffHopp$_{EGNN}$	0.752±0.41	**0.641±0.15**	**1.000±0.01**	0.510±0.12	0.592±0.11	-7.210±1.45	500	-
DiffHopp$_{metric}$	0.927±0.26	0.597±0.20	**1.000±0.02**	0.634±0.18	0.670±0.13	-7.783±2.78	500	442.48
TurboHopp-50	0.872±0.19	0.562±0.18	**1.000±0.00**	0.576±0.12	0.635±0.25	-7.823±1.53	50	**3.19**
TurboHopp-100	0.948±0.22	0.563±0.23	0.997±0.09	0.589±0.19	0.724±0.12	-8.272±1.21	100	5.69
TurboHopp-150	**1.000±0.00**	0.573±0.12	**1.000±0.00**	0.618±0.19	0.715±0.13	-8.277±1.87	150	15.82
TurboHopp-50$_{metric}$	0.917±0.28	0.612±0.21	**1.000±0.00**	0.583±0.20	0.631±0.15	-7.865±1.72	50	3.83
TurboHopp-100$_{metric}$	0.997±0.06	0.561±0.22	**1.000±0.00**	**0.664±0.19**	**0.737±0.11**	**-8.319±1.38**	100	7.14
TurboHopp-150$_{metric}$	**1.000±0.00**	0.569±0.22	**1.000±0.00**	0.648±0.19	0.727±0.11	-8.219±1.722	150	20.88
PDBBind Test	1.000±0.00	-	1.000±0.00	0.599±0.17	0.742±0.11	-8.712±0.18		

3.5 Reinforcement Learning Architecture

We adopt RLCM from Oertell et al. [2024], where we biject a Markov decision process to multistep consistency model inference. More formally, we adopt this mapping:

$$s_t \triangleq (\widetilde{Z}_{\tau_t}, \tau_t, \boldsymbol{u}) \quad \pi(\boldsymbol{a}_t|\boldsymbol{s}_t) \triangleq f_\theta\left(\widetilde{Z}_{\tau_t}, \tau_t|\boldsymbol{u}\right) + \varepsilon \quad P(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t) \triangleq (\delta_{\widetilde{Z}_{\tau_{t+1}}}, \delta_{\tau_{t+1}}, \delta_{\boldsymbol{u}})$$

$$\boldsymbol{a}_t \triangleq \widetilde{Z}_{\tau_{t+1}} \quad \mu \triangleq (\mathcal{N}(0, I), \delta_{\tau_0}, p(\boldsymbol{u})) \quad R_H(\boldsymbol{s}_H) = r(f_\theta(\widetilde{Z}_{\tau_H}, \tau_H, \boldsymbol{u}), \boldsymbol{u})$$

where ε represents the noise added during the noising procedure of the consistency model.

As with Oertell et al. [2024], we create a mapping from MDP states to noised states along the consistency model trajectory. Likewise, we follow a deterministic transition from the state to the predicted action. This action is a combination of the output from the consistency model convolved with random noise in accordance with the DDPM stochastic differential equation (Ho et al. [2020], Song et al. [2021]). Because this policy now becomes stochastic, we can apply Algorithm 3 (a modified version of reinforce with a clipping). The agent then receives a reward at the end of the trajectory corresponding to a combination of desirable chemical attributes. We use PPO (Schulman et al. [2017]) for the core optimization strategy but others like REBEL (Gao et al. [2024]) have been shown to work for RLCM optimization with sometimes stronger results.

3.6 Metric-based Sampling

By leveraging the inherent efficiency of consistency models, we tailor molecule designs to precise standards. Contrary to the fixed-step approach suggested by Song et al. [2023], our strategy utilizes a flexible, customizable metric that adapts to the variability in protein conditions, as evidenced by our empirical findings shown in Fig. 8. This approach allows for dynamic adjustment during generation, ensuring optimal results by selecting scaffolds with the highest evaluated scores. This not only boosts efficiency but also significantly enhances molecule properties, achieving superior docking and QED scores that surpass those found in the initial dataset.

In Algorithm 2, initial samples are drawn from random Gaussian noise $\widetilde{Z}_T \sim \mathcal{N}(0, T^2 I)$. f_θ is then applied to generate initial scaffolds from these samples: $\widetilde{Z}_t = f_\theta(Z_T, T)$. Across a sequence of time points $\{\tau_1, \tau_2 \ldots \tau_{N-1}\}$ we iteratively refine the generated scaffold. After reaching a designated step m, we score metrics to assesses the quality of each Z_t. Then we select the one with the best metric at the end of all the iterations as the final output. Our metric function is based on drug-likeness and synthesizability.

Table 2: **TurboHopp VS TurboHopp-RLCM with Metric Sampling**. We optimized TurboHopp-100 with RLCM using $r_{\text{docking score}}$. Single optimization task is based on multiple reference protein-ligand complexes of the PDBBind testset; we do not optimize one protein at a time. To fully support faster parallel multi ligand - multi protein docking, we used AutoDockGPU instead of QVina2. Binding affinity scores surpassed reference docking scores without losing drug-likeliness and synthesizability.

Method	Connectivity (↑)	Diversity (↑)	Novelty (↑)	QED (↑)	SA (↑)	Vina (↓)	Steps	Time
TurboHopp-100$_{\text{metric}}$	**0.997**±0.06	0.561±0.22	**1.000**±0.00	**0.664**±0.19	**0.737**±0.11	-8.298±1.82	100	7.14
TurboHoppRL-50$_{\text{metric}}$	0.980±0.14	**0.869**±**0.17**	0.936±0.17	0.619±0.18	0.680±0.13	**-9.804**±**2.84**	50	**3.69**
PDBBind Test	1.000±0.00	-	1.000±0.00	0.599±0.17	0.742±0.11	-8.643±2.08		

4 Experiments and Results

4.1 Performance Comparison with DDPM-based baseline

In this section, we compare 3D scaffold generation qualities between TurboHopp and DiffHopp (Torge et al. [2023]), a DDPM-based scaffold-hopping model. For TurboHopp, we also vary the number of steps in multi-step generation.

Evaluation Metrics   We evaluate sampling quality using several metrics outlined in Torge et al. [2023] . Drug-likeness and synthetic accessibility are assessed with Bickerton et al. [2012] and Ertl and Schuffenhauer [2009] respectively. Connectivity is defined as the percentage of fully connected molecules without fragments. Diversity is calculated using Tanimoto Dissimilarity, according to Alhossary et al. [2015]. Novelty is determined by the proportion of molecules that were not present in the training dataset. Additionally, we use Vina Score from QVina2 Alhossary et al. [2015] to estimate binding affinity.

Results and Discussion   Under the same context conditions, for a batch of 150 molecules, Turbo-Hopp generates on average 5-30 times faster than baseline model (Torge et al. [2023]). Furthermore, as shown in Table 1, compared to DMs, performance is superior in all metrics except diversity, which most likely falls compared to DMs due to the fact that consistency models are trained to have consistent results for data on the same trajectory. Also as total steps for multi-step generation increases, we can see an increase in diversity, due to outputs being refined less as steps decrease.

We applied the same score-based sampling method to the baseline model (DiffHopp$_{\text{Scored}}$) and compared to the model without using it (DiffHopp). All metrics rose except binding affinity. However, in DiffHopp, this method significantly reduces sampling efficiency, as it necessitates more frequent evaluations of the generated products towards the end. In contrast, TurboHopp had a major increase in all metrics with a small downside of increased generation time of only a few seconds. This is possible due to the fact that there are less steps required to generate during multi-step generation, thus less steps to evaluate afterwards. This illustrates how, with the right methods, efficient sampling can significantly enhance the search for high-quality molecules within chemical space.

4.2 Performance Optimization using RLCM

In this section, we describe the optimization of our model using two specifically designed reward functions, each aimed at achieving distinct objectives in molecular design:

- **Binding Affinity:** To enhance binding affinity, we began training with TurboHopp-100 and reduced the sampling steps to 50. The reward function is defined as:

$$r_{\text{docking score}}(s) = -2\left\{DS(s) - RS(s)\right\} + QED(s) + \frac{10 - SA(s)}{9} + C(s) \quad (10)$$

where $DS(s)$ and $RS(s)$ represent the docking scores of the generated and reference molecules, respectively, highlighting the importance of relative score differences during learning. $SA(s)$ signifies synthetic accessibility, and $C(s)$ denotes connectivity. Following the insights from Lee et al. [2023], we incorporate QED and SA to mitigate reward hacking, with connectivity included to ensure the structural validity of the molecules.

- **Protein Steric Clashes:** Steric clashes are defined as interactions where the distance between a protein and ligand atom is less than their combined van der Waals radii by a tolerance of 0.5 Å, indicating an energetically unfavorable and often physically implausible pose, crucial for SBDD performance evaluation. We refer to Harris et al. [2023] for estimating number of clashes. To robustly manage steric clashes, we retained the original 100 steps from TurboHopp-100. The corresponding reward function is:

$$r_{\text{steric clash}}(s) = C(s) + QED(s) - (SC(s) - RSC(s)) \tag{11}$$

where SC and RSC denote the steric clash scores for the generated and reference molecules, respectively, with the difference between these scores included to weigh their importance accurately.

For both models, we initially optimize QED and Connectivity for 50 and 100 steps, respectively, and subsequently finetune each task with the defined rewards metrics. Table 2 shows that though metric-based sampling and RLCM, docking scores surpass those of the training dataset.

Unlike in the reward conditioned image generation setting (Fan et al. [2023b], Black et al. [2024]), we omit any KL regularization to the base model during training. This decision stems from the nature of drug discovery where overfitting is less of a concern as long as the model maintains high scores in targeted areas. Furthermore, we empirically found that our approach does not significantly compromise diversity and novelty.

Optimization Metrics We optimize TurboHoppRL for the same test set we used for evaluating TurboHopp (refer to Table 2). Optimization encompasses QED, synthesizability, connectivity, docking score and number of protein clashes. For training stability and computational efficiency, we utilize AutodockGPU (Tang et al. [2022]) to estimate binding affinities, replacing QVina2 (Alhossary et al. [2015]) used for Section 4.1 .

Fig. 4 and Table 2 compares TurboHopp-100 and TurboHoppRL-50. During training, we achieved higher binding affinity scores of about 2-3 kcal/mol compared to TurboHopp-100. Examples can be found at Fig. 10.

As highlighted in Fig. 7, we also fine-tuned a model to minimize steric clashes, achieving promising results as shown in Fig. 11. However, optimizing the reward function further could further enhance clash minimization and improve the balance between metrics, which we propose as a direction for future research. While RL fine-tuned models are not necessarily expected to generalize to unseen molecules in drug discovery, there is potential for some generalization capabilities (Oertell et al. [2024]), a topic also reserved for future exploration.
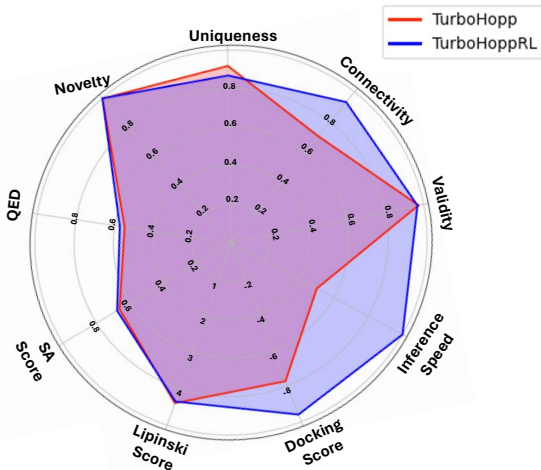

Figure 4: Compared to TurboHopp-100 (Red), TurboHoppRL-50 (Blue) has enhancing binding affinity without losing performance in other metrics.

4.3 Performance Comparison with diffusion-based 3D-SBDD inpainting models

Our model is compared with other diffusion based 3D-SBDD models, Guan et al. [2023] and Guan et al. [2024]. Inpainting was applied (Lugmayr et al. [2022], Algorithm 4) on recent conditional de-novo diffusion models to create variations suitable for fair comparison. The same Bemis-Murcko scaffold determines scaffolds and functional groups. With regards to sampling conditions, the number of atoms is fixed to the reference scaffold for all models. For Guan et al. [2024], since it additionally uses bond diffusion, a bond mask was created accordingly, and reference priors are used while

Table 3: **TurboHopp VS 3D-SBDD inpainting models**. Results on models trained on CrossDocked. "Inpainting" refers to models using inpainting method explained in Algorithm 1. "metric" refers that inference was done with metric-based sampling. QVina score (kcal/mol) refers to estimated binding affinity measured by QVina2. Time refers to average time (seconds) required to generate a batch of molecules per complex. Best metrics are in bold. 2nd best are underlined.

Method	Validity(↑)	Connectivity (↑)	Diversity (↑)	Novelty (↑)	QED (↑)	SA (↑)	QVina (↓)	Time
TargetDiff$_{inpainting}$	0.927±0.13	0.826±0.12	0.841±0.13	0.914±0.08	0.424±0.11	0.661±0.14	-5.896±1.93	740.33
DecompDiff$_{inpainting}$	0.876±0.33	0.722±0.17	**0.856±0.21**	0.895±0.18	0.420±0.07	0.648±0.15	-6.225±0.87	1263.72
TurboHopp-100	0.990±0.10	0.853±0.36	0.484±0.27	**0.936±0.24**	0.488±0.21	0.702±0.13	-7.051±2.14	**6.17**
TurboHopp-100$_{metric}$	**0.993±0.08**	**0.906±0.29**	0.486±0.28	0.935±0.25	**0.502±0.21**	**0.710±0.14**	**-7.204±2.00**	8.18
CrossDocked Test	1.000±0.00	-	1.000±0.00	0.599±0.17	0.476±0.21	0.727±0.14	-7.510±2.36	

sampling. The sampling hyperparameters for inpainting (resampling and jump length parameters) were determined by finding the ones with best validity. Also, since these models were trained on CrossDocked, our model was additionally trained on CrossDocked for fair comparison. The same train-test split suggested in Guan et al. [2024] is followed, with an additional QED minimum filter of 0.3 when constructing the dataset. Only the alpha carbon residues of the protein pocket atoms are used to reduce computational burden.

Results and Discussion Table 3 shows that despite having lower diversity, our model has much faster generation speed as well as relatively high docking score compared to inpainting versions of reference SBDD models. Our findings show that a custom scaffold-hopping model outperforms a repurposed de-novo model.

Geometric properties and ring distributions (Appendix E) were also compared. Results show that our model has closer bond length/angle/atom-atom length distributions to the reference molecules compared to Guan et al. [2023], but poorer results compared in bond/torsion angles. In all aspects, Guan et al. [2024] was outstanding largely because it learns the distribution of bonds while ours do not. For ring distributions, results show that our model is capable of generating similar ring types compared to the reference. In future research, we plan to design our model to learn bond properties as well.

5 Conclusion

We introduced TurboHopp, an equivariant conditional consistency model that significantly boosts generation efficiency and outperforms traditional DDPM-based models in various metrics. Empirical results indicate that our model achieves a 5-30 times improvement in generation speed while maintaining or enhancing traditional quality metrics. Additionally, by leveraging this increased inference speed, we demonstrated that reinforcement learning (RL) is feasible for 3D-SBDD-DMs, enabling the fine-tuning of generative models to meet specific requirements. Most importantly, our method has the potential to be adapted to numerous variations explored within 3D-SBDD-DMs, increase speed and efficiency.

Key areas for improvement and future research include enhancing the denoising model, expanding dataset size, and diversifying reward functions for drug discovery tasks. Incorporating bond diffusion (Peng et al. [2023]) and including hydrogens in proteins and molecules could significantly bolster the robustness of the denoising model. In the consistency model, techniques such as eliminating the Exponential Moving Average (EMA), adopting Pseudo-Huber loss, and refining noise and loss schedules may yield more robust outcomes (Song and Dhariwal [2023]). Furthermore, for RLCM, exploring reward functions that integrate steric energies and interaction fingerprints presents a promising avenue for future research.

Acknowledgements

References

Peter Kirkpatrick and Clare Ellis. Chemical space. *Nature*, 432(7019):823–824, 2004.

Clemens Isert, Kenneth Atz, and Gisbert Schneider. Structure-based drug design with geometric deep learning. *Current Opinion in Structural Biology*, 79:102548, 2023.

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.

Vendy Fialková, Jiaxi Zhao, Kostas Papadopoulos, Ola Engkvist, Esben Jannik Bjerrum, Thierry Kogej, and Atanas Patronov. Libinvent: reaction-based generative scaffold decoration for in silico library design. *Journal of Chemical Information and Modeling*, 62(9):2046–2063, 2021.

Hannes H Loeffler, Jiazhen He, Alessandro Tibo, Jon Paul Janet, Alexey Voronov, Lewis H Mervin, and Ola Engkvist. Reinvent 4: Modern ai–driven generative molecule design. *Journal of Cheminformatics*, 16(1):20, 2024.

Matthew Ragoza, Tomohide Masuda, and David Ryan Koes. Generating 3d molecules conditional on receptor binding sites with deep generative models. *Chemical science*, 13(9):2701–2713, 2022.

Shitong Luo, Jiaqi Guan, Jianzhu Ma, and Jian Peng. A 3d molecule generative model for structure-based drug design. *arXiv preprint arXiv:2203.10446*, 2022.

Benoit Baillif, Jason Cole, Patrick McCabe, and Andreas Bender. Deep generative models for 3d molecular structure. *Current Opinion in Structural Biology*, 80:102566, 2023.

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.

Zihan Ding and Chi Jin. Consistency models as a rich and efficient policy class for reinforcement learning, 2024.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. *arXiv preprint arXiv:2311.12908*, 2023.

Daeseok Lee and Yongjun Cho. Fine-tuning pocket-conditioned 3d molecule generation via reinforcement learning. In *ICLR 2024 Workshop on Generative and Experimental Perspectives for Biomolecular Design*, 2024.

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.

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.

Arne Schneuing, Yuanqi Du, Charles Harris, Arian Jamasb, Ilia Igashov, Weitao Du, Tom Blundell, Pietro Lió, Carla Gomes, Max Welling, et al. Structure-based drug design with equivariant diffusion models. *arXiv preprint arXiv:2210.13695*, 2022.

Jos Torge, Charles Harris, Simon V Mathis, and Pietro Lio. Diffhopp: A graph diffusion model for novel drug design via scaffold hopping. *arXiv preprint arXiv:2308.07416*, 2023.

Ilia Igashov, Hannes Stärk, Clément Vignac, Arne Schneuing, Victor Garcia Satorras, Pascal Frossard, Max Welling, Michael Bronstein, and Bruno Correia. Equivariant 3d-conditional diffusion model for molecular linker design. *Nature Machine Intelligence*, pages 1–11, 2024.

Iiris Sundin, Alexey Voronov, Haoping Xiao, Kostas Papadopoulos, Esben Jannik Bjerrum, Markus Heinonen, Atanas Patronov, Samuel Kaski, and Ola Engkvist. Human-in-the-loop assisted de novo molecular design. *Journal of Cheminformatics*, 14(1):86, 2022.

Hanqun Cao, Cheng Tan, Zhangyang Gao, Yilun Xu, Guangyong Chen, Pheng-Ann Heng, and Stan Z Li. A survey on generative diffusion models. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. *arXiv preprint arXiv:2310.14189*, 2023.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Owen Oertell, Jonathan D Chang, Yiyi Zhang, Kianté Brantley, and Wen Sun. Rl for consistency models: Faster reward guided text-to-image generation. *arXiv preprint arXiv:2404.03673*, 2024.

Charles Harris, Kieran Didi, Arian Jamasb, Chaitanya Joshi, Simon Mathis, Pietro Lio, and Tom Blundell. Posecheck: Generative models for 3d structure-based drug design produce unrealistic poses. In *NeurIPS 2023 Generative AI and Biology (GenBio) Workshop*, 2023.

Hongmao Sun, Gregory Tawa, and Anders Wallqvist. Classification of scaffold-hopping approaches. *Drug discovery today*, 17(7-8):310–324, 2012.

Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pages 8867–8887. PMLR, 2022.

Clement Vignac, Nagham Osman, Laura Toni, and Pascal Frossard. Midi: Mixed graph and 3d denoising diffusion for molecule generation, 2023.

Lei Huang, Hengtong Zhang, Tingyang Xu, and Ka-Chun Wong. Mdm: Molecular diffusion model for 3d molecule generation, 2022.

Zhilin Huang, Ling Yang, Zaixi Zhang, Xiangxin Zhou, Yu Bao, Xiawu Zheng, Yuwei Yang, Yu Wang, and Wenming Yang. Binding-adaptive diffusion models for structure-based drug design. *arXiv preprint arXiv:2402.18583*, 2024.

Ilia Igashov, Hannes Stärk, Clément Vignac, Victor Garcia Satorras, Pascal Frossard, Max Welling, Michael Bronstein, and Bruno Correia. Equivariant 3d-conditional diffusion models for molecular linker design. *arXiv preprint arXiv:2210.05274*, 2022.

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.

Guy W Bemis and Mark A Murcko. The properties of known drugs. 1. molecular frameworks. *Journal of medicinal chemistry*, 39(15):2887–2893, 1996.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2020.

Zhiguang Fan, Yuedong Yang, Mingyuan Xu, and Hongming Chen. Ec-conf: An ultra-fast diffusion model for molecular conformation generation with equivariant consistency. *arXiv preprint arXiv:2308.00237*, 2023a.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Zhaolin Gao, Jonathan D. Chang, Wenhao Zhan, Owen Oertell, Gokul Swamy, Kianté Brantley, Thorsten Joachims, J. Andrew Bagnell, Jason D. Lee, and Wen Sun. Rebel: Reinforcement learning via regressing relative rewards, 2024.

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.

Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1:1–11, 2009.

Amr Alhossary, Stephanus Daniel Handoko, Yuguang Mu, and Chee-Keong Kwoh. Fast, accurate, and reliable molecular docking with quickvina 2. *Bioinformatics*, 31(13):2214–2216, 2015.

Seul Lee, Jaehyeong Jo, and Sung Ju Hwang. Exploring chemical space with score-based out-of-distribution generation. In *International Conference on Machine Learning*, pages 18872–18892. PMLR, 2023.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *arXiv preprint arXiv:2305.16381*, 2023b.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning, 2024.

Shidi Tang, Ruiqi Chen, Mengru Lin, Qingde Lin, Yanxiang Zhu, Ji Ding, Haifeng Hu, Ming Ling, and Jiansheng Wu. Accelerating autodock vina with gpus. *Molecules*, 27(9):3041, 2022.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11461–11471, 2022.

Xingang Peng, Jiaqi Guan, Qiang Liu, and Jianzhu Ma. Moldiff: addressing the atom-bond inconsistency problem in 3d molecule diffusion generation. *arXiv preprint arXiv:2305.07508*, 2023.

A   Algorithm Pseudocode

In this section we present the pseudocode for consistency model training (Algorithm 1), consistency model sampling (Algorithm 2), RL training (Algorithm 3), and inpainting. (Algorithm 4)

Algorithm 1 Training

1: **Input:** dataset $D = \{(Z_i, u_i)\}_{i \in M}$, where M refers to the number of data points in D, Z_i is composed of x_i and h_i, initial model parameter θ, learning rate η, step schedule $\mathcal{N}(\cdot)$, EMA decay rate schedule $\mu(\cdot)$, $\theta^- \leftarrow \theta$ and $k \leftarrow 0$;
2: **repeat**
3:     Sample $z, u \sim D$, and $n \sim \mathcal{U}[1, N(k) - 1]$;
4:     Decompose z into x and h;
5:     Sample $\epsilon_v \sim \mathcal{N}(0, I)$ for $v \in \{x, h\}$;
6:     Subtract center of gravity from ϵ_x;
7:     Define $f_\theta^{n,v} \leftarrow f_\theta(v + t_n \cdot \epsilon_v, t_n | u)$ for $v \in \{x, h\}$;
8:     $L(\theta, \theta^-) \leftarrow \sum_{v \in \{x, h\}} \text{MSE}(f_\theta^{n+1,v}, f_\theta^{n,v})$;
9:     $\theta \leftarrow \theta - \eta \nabla_\theta L(\theta, \theta^-)$;
10:     $\theta^- \leftarrow \mu(k)\theta^- + (1 - \mu(k))\theta$;
11:     $k \leftarrow k + 1$;
12: **until** convergence

Algorithm 2 Sampling with Scoring and Selection

1: **Input:** Consistency model $f_\theta(\cdot, \cdot) = (f_\theta^x, f_\theta^h)$, sequence of time points $\{\tau_1, \tau_2, \ldots, \tau_{N-1}\}$ where $\tau_1 > \tau_2 > \ldots > \tau_{N-1}$, evaluation starting point m, where $1 \leq m \leq N - 1$
2: Sample $\epsilon_x \sim \mathcal{N}(0, I)$ and $\epsilon_h \sim \mathcal{N}(0, I)$
3: Subtract center of gravity from ϵ_x
4: $\epsilon \leftarrow [\epsilon_x, \epsilon_h]$
5: $\widetilde{Z}_T \leftarrow \epsilon$
6: $z \leftarrow f_\theta(\widetilde{Z}_T, T | u)$
7: Initialize max_score $\leftarrow -\infty$
8: Initialize $Z_{\text{best}} \leftarrow$ null
9: **for** $n = 1$ to $N - 1$ **do**
10:     Sample $\epsilon_x \sim \mathcal{N}(0, I)$ and $\epsilon_h \sim \mathcal{N}(0, I)$
11:     Subtract center of gravity from ϵ_x
12:     $\epsilon \leftarrow [\epsilon_x, \epsilon_h]$
13:     $\widetilde{Z}_{\tau_n} \leftarrow z + \sqrt{\tau_n - \tau_{n+1}} \cdot \epsilon$
14:     $z \leftarrow f_\theta(\widetilde{Z}_{\tau_n}, \tau_n | u)$
15:     **if** $n \geq m$ **then**
16:         score $\leftarrow$ CustomScore(z)
17:         **if** score $>$ max_score **then**
18:             max_score $\leftarrow$ score
19:             $Z_{\text{best}} \leftarrow z$
20:         **end if**
21:     **end if**
22: **end for**
23: **output:** Z_{best}

Algorithm 3 Policy Gradient Version of RLCM

1: **Input:** Consistency model policy $\pi_\theta = f_\theta(\cdot, \cdot) + \varepsilon$, finetune horizon H, context set set $\mathcal{C}$, batch size b, inference pipeline P
2: **for** $i = 1$ **to** M **do**
3:    Sample b contexts from $\mathcal{C}$, $\boldsymbol{u} \sim \mathcal{C}$.
4:    $\boldsymbol{Z} \leftarrow P(f_\theta, H, \boldsymbol{u})$ ▷ where $\boldsymbol{Z}$ is the batch of molecules
5:    Normalize rewards $r(\cdot, \cdot)$ per context
6:    Split $\boldsymbol{Z}$ into k minibatches.
7:    **for** each minibatch **do**
8:       **for** $t = 0$ to H **do**
9:          Accumulate gradients of θ using rule:

$$\nabla_\theta \left[\min \left\{ r(\boldsymbol{x}_0, \boldsymbol{c}) \cdot \frac{\pi_{\theta_{i+1}}(a_t|s_t)}{\pi_{\theta_i}(a_t|s_t)}, r(\boldsymbol{x}_0, \boldsymbol{c}) \cdot \mathtt{clip}\left(\frac{\pi_{\theta_{i+1}}(a_t|s_t)}{\pi_{\theta_i}(a_t|s_t)}, 1 - \varepsilon, 1 + \varepsilon \right) \right\} \right]$$

10:       **end for**
11:       Update parameters based on accumulated gradients.
12:    **end for**
13: **end for**
14: Output trained consistency model $f_\theta(\cdot, \cdot)$

Algorithm 4 TargetDiff/DecompDiff Inpainting Procedure

   Set number of atoms to generate based on reference scaffold. Get scaffold mask m.
   Move center of mass of protein atoms to zero.
   Sample initial molecular atom coordinates $x_T \sim \mathcal{N}(0, I)$
4: Sample initial bond types/atom types by $v_T = \text{one\_hot}(\arg \max_i g_i)$, where $g \sim \text{Gumbel}(0, 1)$
   for $t = T, \ldots, 1$ **do**
      for $u = 1, \ldots, U$ **do**
         1) Get known
8:       $x_{t-1}^{\text{known}} = \sqrt{\alpha_t} x_0 + (1 - \alpha_t)\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$
         $\log c = \log(\alpha_t v_0 + (1 - \alpha_t)/K)$
         $v_{t-1}^{\text{known}} = \text{one\_hot}(\arg \max_i [g_i + \log c_i])$, where $g \sim \text{Gumbel}(0, 1)$
         2) Get unknown
12:      $x_{t-1}^{\text{unknown}} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \alpha_t}} e_\theta(x_t, t) \right) + \sigma_t z, z \sim \mathcal{N}(0, I)$
         $v_{t-1}^{\text{unknown}} = $ Compute the posterior atom types $c(v_t, e_\theta(v_t, t))$
         3) Merge according to mask m
         $x_{t-1} = m \odot x_{t-1}^{\text{known}} + (1 - m) \odot x_{t-1}^{\text{unknown}}$
16:      $v_{t-1} = m \odot v_{t-1}^{\text{known}} + (1 - m) \odot v_{t-1}^{\text{unknown}}$
         if $u < U$ and $t > 1$ **then**
            Sample x_t from the posterior $q_\theta(x_{t-1} \mid x_t)$
            Sample v_t from the posterior $q_\theta(v_{t-1} \mid v_t)$
20:         **end if**
      end for
   end for
   return M with coordinates x_0 and atom types/ bond types v_0

B Information on Hyperparameters and Experiment Details

Parameter setting for TurboHopp

Setting	Parameters
TurboHopp	timesteps: 150, 100, 50, 25
Denoiser: GVP	batch size: 256
layers: 6	lr: 1e-4
hidden features: 256	schedule: ReduceLROnPlateau (min: 1e-6, factor:
GNN layers: 7	0.9)
Attention: True	num epochs: 5500
Embedding size: 64	σ_{min}: 0.002
Optimizers: Adam	σ_{max}: 80.0
γ: 1e-3	σ_{data}: 0.5
β: (0.9, 0.995)	ρ: 7.0
Dataset: PDBBind filtered	
Device: 4x NVIDIA A100 GPUs	

Parameter setting for RLCM

Setting	Parameters
Parameters for Docking Objective and Steric Clashes	gradient accumulation steps: 1
Dataset: PDBBind filtered test set	batch size: 215
Device: 8x NVIDIA A100 GPUs	num epochs: 200
	sample iters: 1
	buffer size: 32
	min count: 16
	train batch size per gpu: 215
	num inner epochs: 1
	lr: 1e-5
	clip range: 1e-4
	max grad norm: 10

C RLCM and Consistency Model Training Curves

For reproducibility, we present the curves from our training runs for both the consistency model and the use of RLCM.
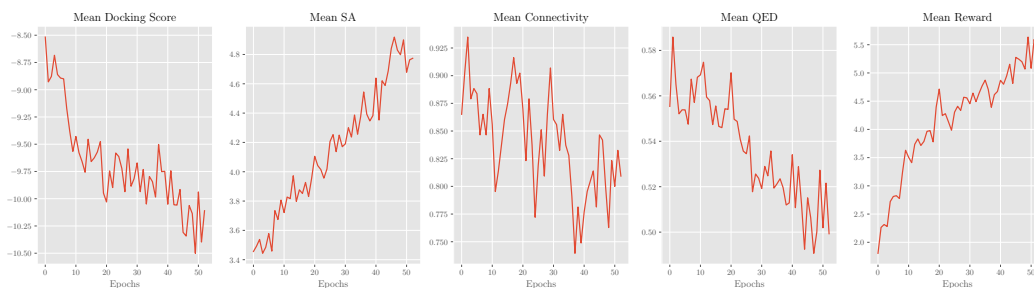

Figure 5: Training curves for the metrics which compose of the loss function. Notice that all either increase or maintain approximately the same value. Connectivity and QED score slightly decrease because we start from a previously RL finetuned checkpoint which optimizes only for connectivity, SA, and QED score.
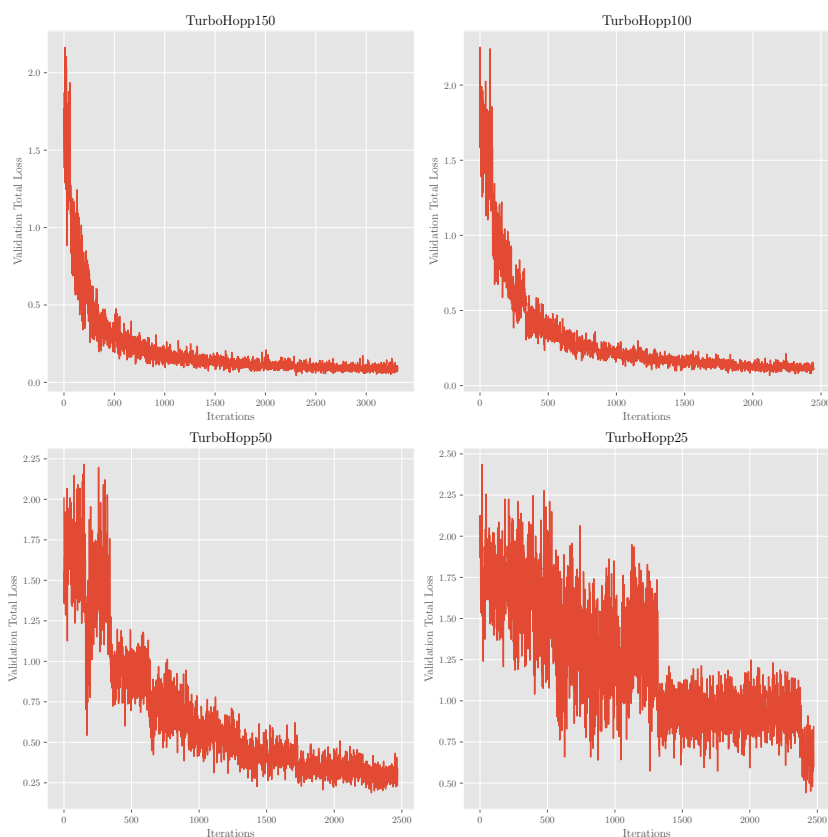

Figure 6: Training curves of total validation loss for different step size variants. We train a number of consistency models to empirically determine the optimal tradeoff between step size fidelity and speed. Turbohopp25 had low validity and proved to be too unstable for consistent generations.
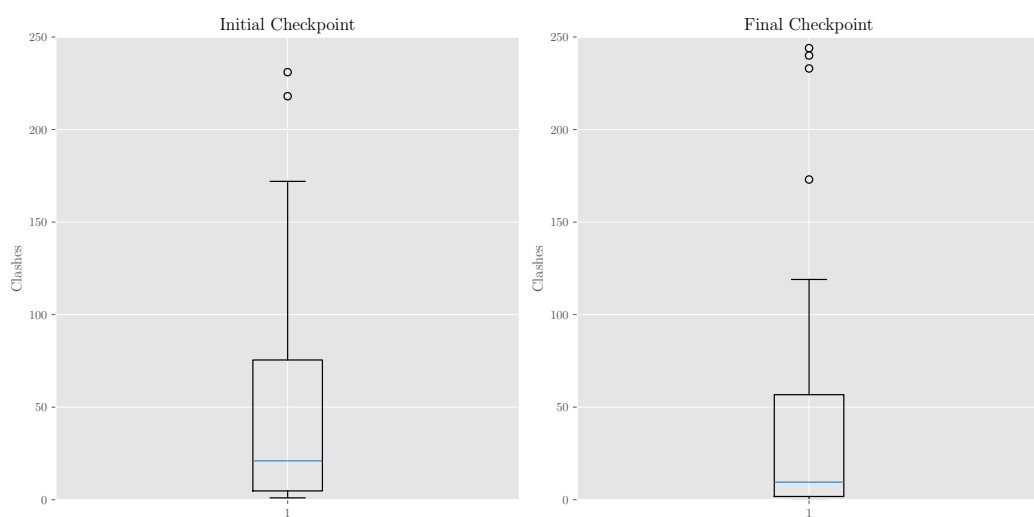
17

Figure 7: Plot of clashes before and after finetuning with a reward function mentioned in the main text. The initial Turbohopp-100 model was a RLCM finetuned model for connectivity, synthesizability, and QED score ($r_{\text{steric clash}}$. Notice the shift toward smaller number of clashes during training. However, we believe that further iteration of the reward function will lead to more effective finetuning but we leave this to future work.
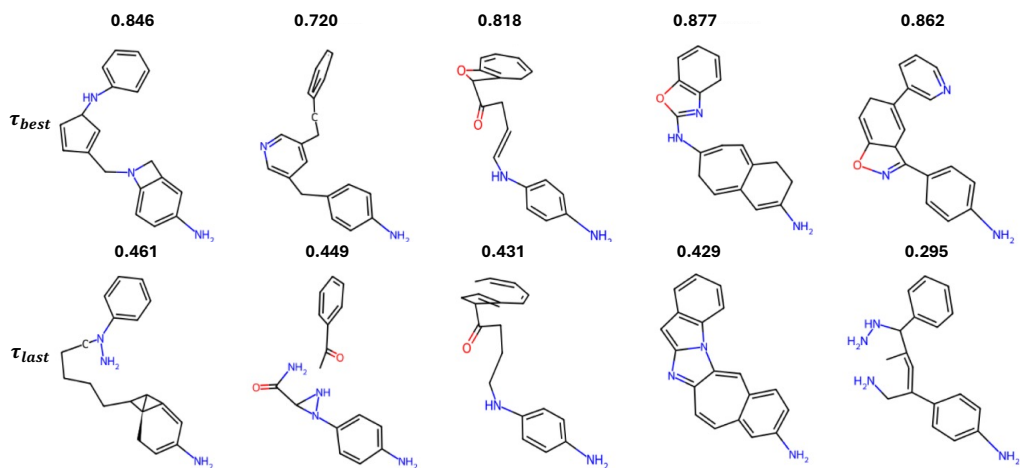
D Sample Molecules

Figure 8: Examples of samples generated for PDB 6QQW with drug-likeliness. 1st row samples collected during multi-step phase with best scores and 2nd row indicates samples from the final step. Connectivity and overall metrics increased when we adopted custom score-based sampling.
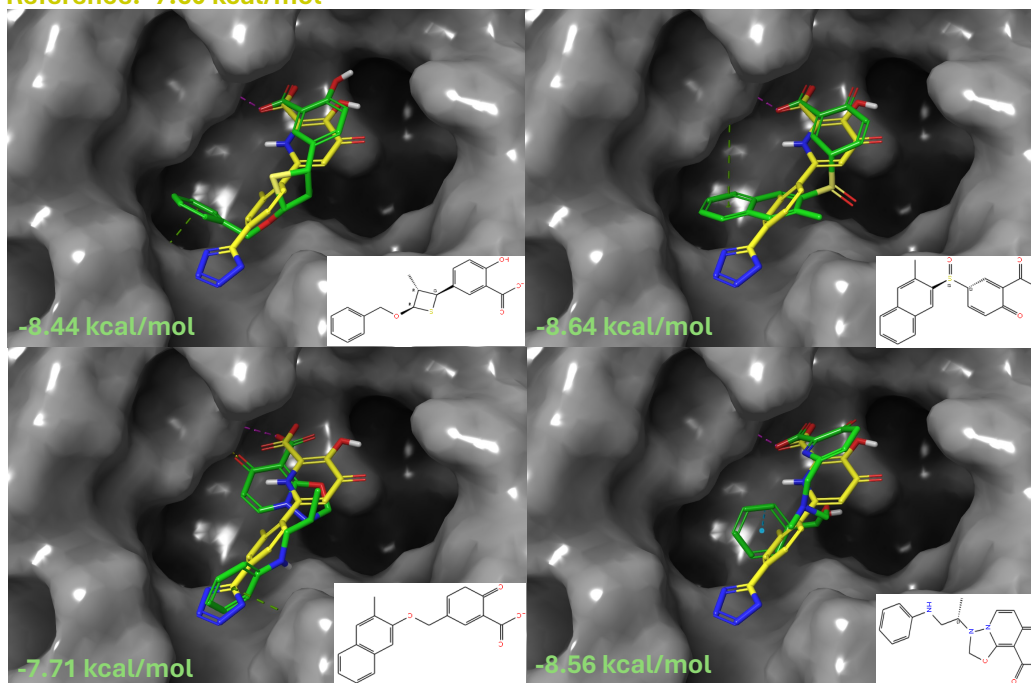

Figure 9: Scaffolds generated for PDB ID 6E6W by TurboHopp-100. Reference molecule is in yellow, while generated molecules are in green. Functional groups are the carboxylic acid and hydroxyl groups in the upper corner. Dotted lines in refer to ligand-receptor intermolecular interactions: green, blue, yellow, purple being pi-cation, pi-pi stacking, hydrogen bonds, halogen bonds respectively. Compared to reference molecule, generated molecules had new interactions and higher binding affinity, while maintaining similar binding pose.

Figure 10: Comparison of Reference molecule(Yellow), and molecules generated by Turbo-Hopp(Green) and TurboHopp-RL(Orange). Notice that Turbohopp and TurboHopp-RL generate molecules that have higher binding affinity with the protein.
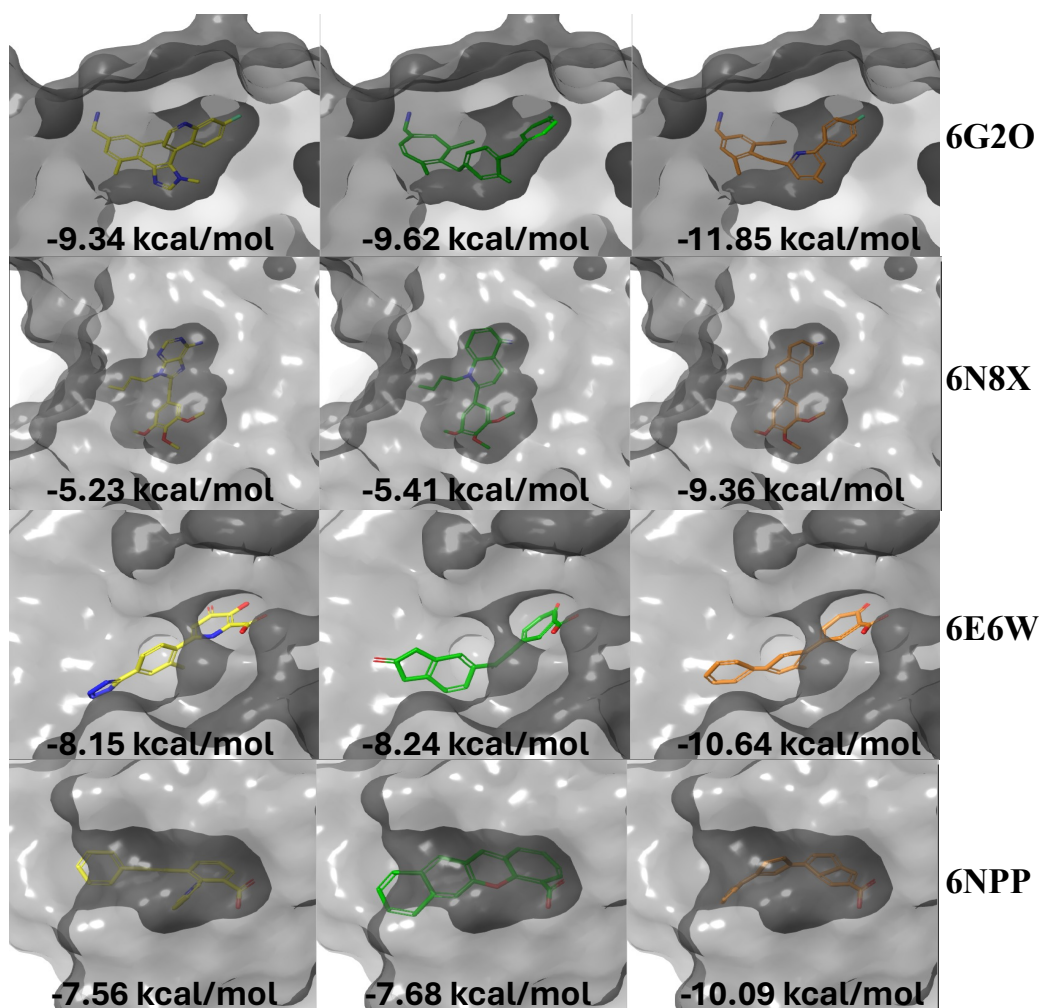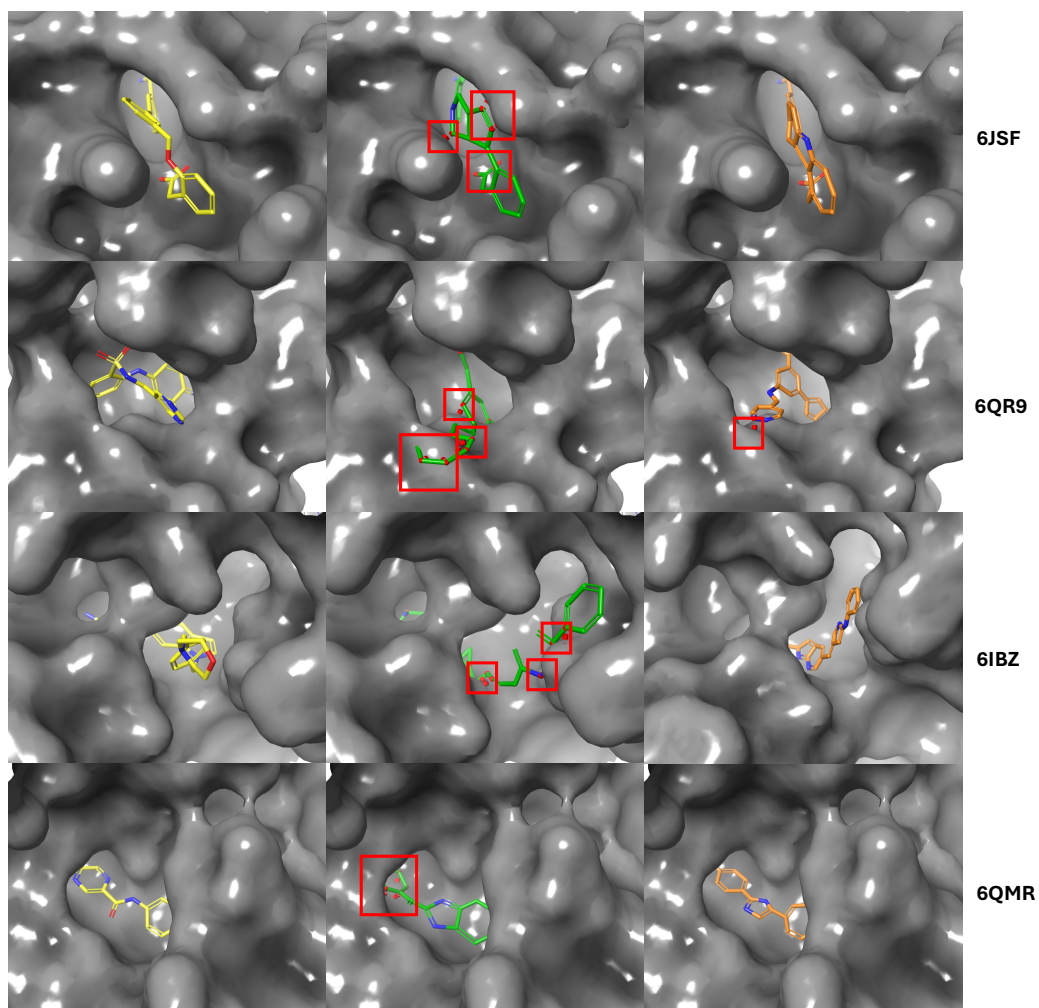
Figure 11: Comparison of Reference molecule(Yellow), and molecules generated by Turbo-Hopp(Green) and TurboHopp-RL(Orange). Red box indicates collision points with protein atoms. TurboHopp-RL generates molecules that has less clashes with the protein.

E   Geometric Properties Comparison between TurboHopp and 3D-SBDD inpainting models

Table 4: Jensen-Shannon divergence between bond distance distributions of generated VS reference(CrossDocked)

Bond	TargetDiff$_{inpainting}$	DecompDiff$_{inpainting}$	TurboHopp-100
C–C	0.4984	**0.2516**	0.4376
C=C	0.6303	**0.2962**	0.4743
C–N	0.4833	**0.2188**	0.4178
C=N	0.6300	**0.3913**	0.5339
C–O	0.3882	**0.2841**	0.3809
C=O	0.3953	**0.2395**	0.3552
C:C	0.6829	**0.2058**	0.5562
C:N	0.6705	**0.2734**	0.5743

Table 5: Jensen-Shannon divergence between atom-atom distance distributions of generated VS reference(CrossDocked)

Bond	TargetDiff$_{inpainting}$	DecompDiff$_{inpainting}$	TurboHopp-100
C-C	0.55	**0.28**	0.45
all	0.17	**0.08**	0.13

Table 6: Percentage of different ring sizes for reference(CrossDocked) and model-generated molecules.

Ring Size	Ref.	TargetDiff$_{inpainting}$	DecompDiff$_{inpainting}$	TurboHopp-100
3	1.70	4.77	0.00	7.85
4	0.00	1.12	4.00	0.81
5	30.20	30.08	37.21	21.16
6	67.40	57.91	41.70	62.51
7	0.70	5.28	14.06	6.19
8	0.00	0.81	2.54	1.12
9	0.00	0.05	0.48	0.31

Table 7: Jensen-Shannon Divergence of Top 3 Torsion Angle Distributions (CrossDocked)

Metric	DecompDiff$_{inpainting}$	TargetDiff$_{inpainting}$	TurboHopp
CCNC	0.184	0.204	0.207
CCCC	0.264	0.249	0.197
C:C:C:C	0.178	0.258	0.383

Table 8: Jensen-Shannon Divergence of Top 5 Bond Angle Distributions (CrossDocked)

Metric	DecompDiff$_{inpainting}$	TargetDiff$_{inpainting}$	TurboHopp
CNC	0.208	0.274	0.321
CCN	0.168	0.207	0.332
C:C:C	0.096	0.120	0.324
CCC	0.231	0.231	0.298
CCO	0.229	0.263	0.319

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: We have claimed our model's performance and contributions based on empirical findings and experimental results. Our model can be easily adapted to similar frameworks using diffusion models for drug discovery.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We have added limitations and possible further research studies in the conclusion section.

   Guidelines:

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

3. **Theory Assumptions and Proofs**

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

Answer: [NA]

Justification: Our focus remains on empirical findings and application, bypassing the need for formal theoretical frameworks or validations.

Guidelines:

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

4. **Experimental Result Reproducibility**

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

Answer: [Yes]

Justification: We will share our code/checkpoints upon acceptance. In the paper, we provide the algorithms for training model, RL, and inference. We also share training curves for model training and RL as well as hyperparameters required to reproduce our results.

Guidelines:

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

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

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

Answer: [Yes]

Justification: We provide the data we used to train our model, and upon code release, we will specify preprocessing code ,modifications, as well as environment for reproduction.

Guidelines:

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

6. **Experimental Setting/Details**

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

Answer: [Yes]

Justification: We specify the hyperparameters in the appendix. Further, we use a standardized dataset.

Guidelines:

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

7. **Experiment Statistical Significance**

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

Answer: [Yes]

Justification: All claims of statistical significance were justified through a number of trials. Multiple seeds are not needed for RLCM since we are applying a method previously shown to work.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments Compute Resources**

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

Answer: [Yes]

Justification: We specify the compute resources required in the appendix.

Guidelines:

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

9. **Code Of Ethics**

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

Answer: [Yes]

Justification: Yes we read the ethics guidlines and are abiding by them.

Guidelines:

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

10. **Broader Impacts**

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

Answer: [Yes]

Justification: Please check the conclusion section of the paper.

Guidelines:

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

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

11. **Safeguards**

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

Answer: [NA]

Justification: Models are not yet going to be released and we will take into account proper safeguards when releasing models with high risk for misuse.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: All terms of use and licences are respected.

Guidelines:

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

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

13. **New Assets**

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

    Answer: [NA]

    Justification: No new assets are released

    Guidelines:

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

14. **Crowdsourcing and Research with Human Subjects**

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

    Answer: [NA]

    Justification: We do not do any crowdsourcing nor research with human subjects.

    Guidelines:

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

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

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

    Answer: [NA]

    Justification: We do not do crowdsourcing nor research with human subjects.

    Guidelines:

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

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