RIDE: REFERENCE-ANCHORED INFERENCE-TIME DIFFUSION EDITING FOR SCAFFOLD HOPPING

Anonymous authors
Paper under double-blind review

ABSTRACT

Scaffold hopping is a critical task in drug discovery, which seeks to discover new, structurally distinct molecules that share key functional groups and similar 3D shape with a reference binding ligand. Existing diffusion-based scaffold hopping methods formulate the problem as conditional generation of scaffolds given the functional groups. However, they lack a principled mechanism to jointly enforce 2D structural novelty and preserve the 3D shape of the reference ligand. Here, we introduce RIDE, a Reference-anchored Inference-time Diffusion Editing framework for scaffold hopping. RIDE recovers the reference diffusion noise trajectory conditioned on the binding pocket and functional groups, selects an optimal trajectory segment for editing via noise perturbation, and conducts a value-guided scaffold sampling to generate new scaffolds. Extensive experimental results demonstrate that, compared to baselines, RIDE consistently generates scaffolds with lower 2D similarity and higher 3D similarity to the reference, with an average improvements of 11.7% and 7.3%, respectively. Further analysis reveals that RIDE can accommodate various reward functions, and can preserve 3D similarity even when this is not explicitly included in the reward. Two case studies illustrate RIDE's ability to generate distinct scaffolds with different structures and properties, and its ability to introduce substantial 2D variation while maintaining very high 3D similarity. RIDE is publicly available at `https://anonymous.4open.science/r/RIDE-C8A0`.

1 INTRODUCTION

Scaffold hopping is a critical task in drug discovery, which seeks to discover new, structurally distinct molecules that share key functional groups and similar 3D shape with a reference binding ligand (Böhm et al., 2004). Effective scaffold hopping increases the success rate of drug discovery by exploring a variety of molecule structures with similar binding but different properties, synthesizability, and patentability (Böhm et al., 2004). Recently, generative AI methods have opened new avenues for scaffold hopping (Zheng et al., 2021; Torge et al., 2023; Yoo et al., 2024; Yang et al., 2026). These methods provide the ability to sample new molecular structures from distributions learned from vast datasets, enabling the exploration of chemical spaces beyond those familiar to domain experts and expanding opportunities to discover novel molecular scaffolds.

Among generative AI methods, diffusion models are particularly promising for scaffold hopping, as their iterative denoising process enables flexible and fine-grained control over generation. Several efforts (Torge et al., 2023; Adams et al., 2025; Yang et al., 2026) have applied diffusion to scaffold hopping, formulating the problem as the conditional generation of scaffolds given the functional groups. While these methods can generate new scaffolds, they rely on sampling stochasticity to diversify their generated results, and lack a principled mechanism to promote structural novelty, leaving the extent of structural divergence from the original scaffold largely to chance. While scaffold hopping based on diffusion models for structure-based drug design (SBDD) (Guan et al., 2023; 2024; Huang et al., 2024) can leverage protein-ligand binding information, it lacks explicit guarantees of substantial 2D structural novelty relative to known ligands. Conversely, approaches based on diffusion for ligand-based drug design (LBDD) (Chen et al., 2025; Adams et al., 2025) can explore structurally distinct scaffolds but lack explicit guidance from protein-ligand binding.

We introduce RIDE, a Reference-anchored Inference-time Diffusion Editing framework for scaffold hopping. RIDE reformulates scaffold hopping as a diffusion editing process that transforms the reference scaffold into new scaffolds. As shown in Figure 1, it operates in three steps: **(1)** recovery of the reference diffusion noise trajectory conditioned on the binding pocket and functional groups, **(2)** selection of an optimal trajectory segment for editing via noise perturbation, and **(3)** value-guided scaffold sampling to generate new scaffolds. RIDE leverages a reward function measuring 2D structural novelty and 3D shape similarity to guide noise perturbation and scaffold sampling, jointly enforcing 2D structural novelty and preserving the 3D shape of the reference ligand. As a framework for inference-

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

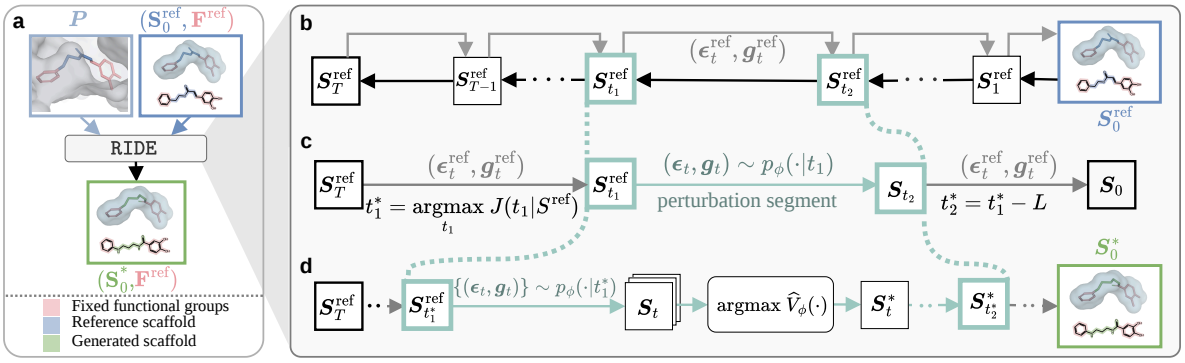


Figure 1: Overview of RIDE. **a**, RIDE generates new scaffolds based on a reference scaffold, fixed functional groups, and protein pocket; **b**, RIDE recovers the noise trajectory for the reference ligand from a diffusion-based molecule generation model adapted to scaffold hopping; **c**, RIDE identifies a reference-optimal noise trajectory segment for editing; and **d**, RIDE performs a value-guided transition within the reference-optimal trajectory segment for a controlled editing of the noise trajectory, thus sampling a set of new scaffolds.

time editing, RIDE can adapt pre-trained diffusion-based molecular generative models for controllable scaffold hopping without retraining.

We rigorously evaluate RIDE with state-of-the-art baseline models on 60 scaffold hopping tasks for protein-ligand complexes related to major human diseases. Compared to baselines, RIDE consistently generates scaffolds with lower 2D similarity and higher 3D similarity to the reference, with an average improvements of 11.7% and 7.3%, respectively. Further analysis reveals that RIDE can accommodate various reward functions, and can preserve 3D similarity even when this is not explicitly included in the reward. RIDE can be applied iteratively to explore even more structurally distinct scaffolds, where a RIDE-generated molecule is used as a new "reference" to RIDE for further scaffold hopping. Experiments reveal a trade-off in multiple hops between progressively improving the similarity profile and preserving binding. Additionally, two case studies illustrate RIDE's ability to generate distinct scaffolds with different structures and properties, and its ability to introduce substantial 2D variation while maintaining very high 3D similarity. RIDE is publicly available at `https://anonymous.4open.science/r/RIDE-C8A0`.

2 Related Work

Generative models for Scaffold Hopping Recently, generative models have shown substantial promise in SBDD, LBDD, and scaffold hopping (Zheng et al., 2021; Guan et al., 2023; Torge et al., 2023). SBDD methods generate molecules tailored to specific protein pockets without relying on a reference ligand (Guan et al., 2023; 2024; Huang et al., 2024; Gu et al., 2024; Yang et al.; Gao et al., 2026). For example, IPDiff (Huang et al., 2024) incorporates learned pocket-ligand interaction embeddings into the diffusion processes to generate molecules with strong binding affinity. Similarly, conDitar-dev (Gao et al., 2026) trains its diffusion model to leverage pretrained pocket representations and supports inference-time trajectory optimization to optimize various molecule properties. In contrast to SBDD, LBDD methods leverage known binding ligands as references to generate molecules with high 3D shape similarity to the reference (Chen et al., 2025; Adams et al., 2025; Li et al., 2026). For example, ShEPhERD (Adams et al., 2025) jointly learns molecular structures with 3D shape, electrostatics, and pharmacophore to generate molecules that resemble the reference ligand with respect to these features. Generative models for scaffold hopping can be formulated under either SBDD or LBDD through inpainting (Schneuing et al., 2024; Li et al., 2026) or additional conditioning on functional groups (Torge et al., 2023; Yoo et al., 2024). However, structure-based scaffold hopping and ligand-based scaffold hopping emphasize either binding affinity or the 3D similarity to the reference, respectively. Moreover, current formulations do not explicitly optimize 2D similarity to the reference scaffold. Our work instead incorporates the reference scaffold into the generation process of structure-based models, providing a unified framework for achieving low 2D similarity and high 3D similarity while maintaining strong binding affinity.

3 Preliminaries

3.1 Inversion-Based Diffusion Methods

Inversion-based diffusion methods aim to invert a reference sample into its latent diffusion trajectory and edit the recovered trajectory for controlled generation (Kim et al., 2022; Mokady et al., 2023; Hertz et al., 2022). By editing

the recovered trajectory, these methods can generate samples that preserve reference-specific features and introduce desired changes. These methods build upon the reverse sampling process of diffusion models. A common approach to recovering the reference state trajectory is DDIM inversion (Song et al., 2020; Wallace et al., 2023). Starting from a reference sample $\boldsymbol{s}_0^{\text{ref}}$, DDIM inversion maps it back to the corresponding intermediate states $\{\boldsymbol{s}_t^{\text{ref}}\}_{t=1}^T$, where T denotes the total number of diffusion timesteps. The inversion process is represented as

$$\boldsymbol{s}_t^{\text{ref}} = \sqrt{\bar{\alpha}_t}((\boldsymbol{s}_{t-1}^{\text{ref}} - \sqrt{1-\bar{\alpha}_{t-1}}\boldsymbol{\epsilon}_\theta(\boldsymbol{s}_{t-1}^{\text{ref}}, t-1))/\sqrt{\bar{\alpha}_{t-1}}) + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\boldsymbol{s}_{t-1}^{\text{ref}}, t-1), \tag{1}$$

where $\boldsymbol{\epsilon}_\theta(\boldsymbol{s}_t, t)$ is the noise predictor learned during diffusion training, $\bar{\alpha}_t \in (0,1)$ is a time-dependent cumulative coefficient determined by the predefined noise schedule. Under DDPM sampling (Ho et al., 2020), each transition $\boldsymbol{s}_t^{\text{ref}} \to \boldsymbol{s}_{t-1}^{\text{ref}}$ along the recovered reference state trajectory is associated with a specific sampling noise $\boldsymbol{\epsilon}_t^{\text{ref}}$. Therefore, the reference state trajectory can be equivalently represented by a corresponding noise sequence $\{\boldsymbol{\epsilon}_t^{\text{ref}}\}_{t=1}^T$, with each $\boldsymbol{\epsilon}_t^{\text{ref}}$ recovered as follows (Huberman-Spiegelglas et al., 2024),

$$\boldsymbol{\epsilon}_t^{\text{ref}} = (\boldsymbol{s}_{t-1}^{\text{ref}} - \boldsymbol{\mu}_\theta(\boldsymbol{s}_t^{\text{ref}}, t))/\sigma_t, \tag{2}$$

where $\boldsymbol{\mu}_\theta(\boldsymbol{s}_t, t) = \frac{1}{\sqrt{\alpha_t}}(\boldsymbol{s}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{s}_t, t))$ is the parameterized mean of the reverse transition distribution, β_t is the noise variance schedule at timestep t, and σ_t is the predefined standard deviation of the transition distribution.

Although the inversion-based methods were originally proposed for Gaussian diffusion, the inversion can also be extended to categorical diffusion, where each state s_t^{ref} in the reference trajectory is categorical. The inversion process can be constructed as follows (He et al., 2026),

$$s_t^{\text{ref}} \sim q(s_t \mid s_{t-1}^{\text{ref}}), \tag{3}$$

where $q(s_t \mid s_{t-1}) = \text{Cat}(\alpha_t^c \boldsymbol{e}_{s_{t-1}}) + (1-\alpha_t^c)\frac{1}{d}\mathbf{1})$ denotes the categorical forward transition (Hoogeboom et al., 2021), $\boldsymbol{e}_s$ is the one-hot vector corresponding to category s, d is the number of discrete categories, and $\alpha_t^c \in (0,1)$ is predefined noise schedule at timestep t. Given the recovered categorical reference trajectory $\{s_t^{\text{ref}}\}_{t=1}^T$, the corresponding noise sequence $\{\boldsymbol{g}_t^{\text{ref}}\}_{t=1}^T$ can be determined as follows,

$$g_{t,k}^{\text{ref}} = [\max_{j \neq s_{t-1}^{\text{ref}}} \log p_\theta(s_{t-1} = j \mid s_t^{\text{ref}}) - \log p_\theta(s_{t-1} = s_{t-1}^{\text{ref}} \mid s_t^{\text{ref}}) + \delta]_+, \text{if } k = s_{t-1}^{\text{ref}}; \text{otherwise, } 0, \tag{4}$$

where $p_\theta(s_{t-1} \mid s_t)$ denotes the one-step reverse categorical transition, δ is a small positive margin to ensure that the category can be stricly selected, and $[\cdot]_+ = \max(\cdot, 0)$ denotes the positive-part operator.

3.2 Value-Based Methods for Inference-Time Optimization in Diffusion

Inference-time optimization methods for diffusion models focus on modifying their reverse process to improve generated samples with respect to specific objectives (Chung et al., 2022; Tang et al., 2024; Kim et al., 2025). However, most existing approaches rely on gradients (Guo et al., 2026; Yang et al., 2024), which are not often available for objectives in molecular generation (Shen et al., 2025; Tan et al., 2025). Value-based methods provide an alternative for non-differentiable objectives by estimating the expected future reward of an intermediate state $\boldsymbol{s}_t$ with the value function defined as $V(\boldsymbol{s}_t) = \mathbb{E}_{\boldsymbol{s}_0 \sim p_\theta(\cdot|\boldsymbol{s}_t)}[\mathcal{R}(\boldsymbol{s}_0)]$ (Li et al., 2024; Uehara et al., 2025), where $\mathcal{R}(\cdot)$ is the reward function defined on the final state $\boldsymbol{s}_0$.

Recent efforts (Kim et al., 2026) reformulate the value estimates using the samples generated from $p_\theta(\boldsymbol{s_0})$ as follows to bypass potential computational limitations in using $p_\theta(\cdot \mid \boldsymbol{s}_t)$ (Li et al., 2024; Jain et al., 2025):

$$V(\boldsymbol{s}_t) = \mathbb{E}_{p_\theta(\boldsymbol{s}_0)}[q(\boldsymbol{s}_t \mid \boldsymbol{s}_0)\mathcal{R}(\boldsymbol{s}_0)] / \mathbb{E}_{p_\theta(\boldsymbol{s}_0)}[q(\boldsymbol{s}_t \mid \boldsymbol{s}_0)]. \tag{5}$$

This reformulation enables $V(\boldsymbol{s}_t)$ to be estimated in two steps: (1) perform $\boldsymbol{s}_t$-independent lookahead sampling from the noise prior $q(\boldsymbol{s}_T)$ to obtain $\boldsymbol{s}_0$ and $\mathcal{R}(\boldsymbol{s}_0)$, and (2) compute $V(\boldsymbol{s}_t)$ in closed form by connecting $\boldsymbol{s}_t$ to the lookahead samples $\boldsymbol{s}_0$ through the forward transition $q(\boldsymbol{s}_t \mid \boldsymbol{s}_0)$.

4 Methods

4.1 Problem Definition and Notations

A ligand can be decomposed into a scaffold and a set of functional groups. The functional groups correspond to atoms involved in key interactions with the ligand's binding pocket, and the scaffold atoms define the molecular backbone connecting them. Scaffold hopping seeks to identify ligands that preserve the 3D shape and binding pattern of a known reference ligand (or "*reference*") and that are also structurally distinct from it in 2D. Accordingly, we represent a ligand as $\boldsymbol{M} = (\boldsymbol{S}, \boldsymbol{F})$, where $\boldsymbol{S}$ denotes the scaffold and $\boldsymbol{F}$ denotes the functional groups, and represent the protein pocket

as $\boldsymbol{P}$. $\boldsymbol{S}$, $\boldsymbol{F}$, and $\boldsymbol{P}$ are all represented as sets of atoms involved, with each atom $\boldsymbol{a}$ characterized by a continuous position $\boldsymbol{x} \in \mathbb{R}^3$ and a categorical atom type $v \in \{1, \ldots, d\}$.

Given a pocket $\boldsymbol{P}$ and a reference ligand $\boldsymbol{M}^{\text{ref}} = (\boldsymbol{S}^{\text{ref}}, \boldsymbol{F}^{\text{ref}})$ bound to $\boldsymbol{P}$, scaffold hopping can be formulated as generating a new ligand $\boldsymbol{M} = (\boldsymbol{S}, \boldsymbol{F}^{\text{ref}})$ by replacing the reference scaffold $\boldsymbol{S}^{\text{ref}}$ with a different, generated scaffold $\boldsymbol{S}$. Existing methods (Torge et al., 2023; Yoo et al., 2024) typically formulate this problem as conditional scaffold generation using a diffusion model, that is, $\boldsymbol{S} \sim p_\theta(\cdot \mid \boldsymbol{P}, \boldsymbol{F}^{\text{ref}})$, parameterized by θ, where the reference scaffold $\boldsymbol{S}^{\text{ref}}$ is not involved in the generation process. As a result, the diffusion model cannot explicitly exploit the structural features of the reference scaffold to tailor generation to the desired 2D and 3D similarity profile.

Here, we propose RIDE (Figure 1), a novel reference-anchored editing framework that reformulates scaffold hopping as a principled transformation from the reference scaffold $\boldsymbol{S}^{\text{ref}}$ to a set of new scaffolds $\{\boldsymbol{S}\}$. This formulation enables critical reference-specific features, such as the 3D shape and binding pose, to be preserved while introducing desired 2D changes to the scaffolds. To implement this formulation, RIDE first adapts state-of-the-art SBDD models to scaffold hopping to construct the base model $p_\theta(\boldsymbol{S} \mid \boldsymbol{P}, \boldsymbol{F}^{\text{ref}})$, with details described in Appendix A.1, and propose an inference-time method that modifies its sampling process to obtain a new distribution $\tilde{p}_\theta(\boldsymbol{S} \mid \boldsymbol{P}, \boldsymbol{F}^{\text{ref}}; \boldsymbol{S}^{\text{ref}})$ anchored to $\boldsymbol{S}^{\text{ref}}$ (Section 4.2). Thus, the new distribution can be identified by solving the following new optimization problem:

$$\max_{\tilde{p}_\theta} \mathbb{E}_{\boldsymbol{S} \sim \tilde{p}_\theta(\cdot \mid \boldsymbol{P}, \boldsymbol{F}^{\text{ref}}; \boldsymbol{S}^{\text{ref}})} \left[\mathcal{R}(\boldsymbol{S}, \boldsymbol{S}^{\text{ref}}) \right], \tag{6}$$

where $\mathcal{R}()$ is the reward function defined as follows:

$$\mathcal{R}(\cdot, \boldsymbol{S}^{\text{ref}}) = \lambda(1 - \text{Sim}_{\text{2D}}(\cdot, \boldsymbol{S}^{\text{ref}})) + (1 - \lambda)\, \text{Sim}_{\text{3D}}(\cdot, \boldsymbol{S}^{\text{ref}}), \tag{7}$$

which considers 2D similarity Sim_{2D} and 3D similarity Sim_{3D} to $\boldsymbol{S}^{\text{ref}}$; and $\lambda \in [0, 1]$ balances the two similarities. The reward is computed only for structurally valid scaffolds and is set to 0 for invalid ones.

4.2 Reference-Anchored Scaffold Generation via Noise Editing

Our goal is to construct the generation distribution $\tilde{p}_\theta(\boldsymbol{S} \mid \boldsymbol{P}, \boldsymbol{F}^{\text{ref}}; \boldsymbol{S}^{\text{ref}})$ from the base scaffold generation distribution $p_\theta(\boldsymbol{S} \mid \boldsymbol{P}, \boldsymbol{F}^{\text{ref}})$. To address the competing objectives of low 2D similarity and high 3D similarity with $\boldsymbol{S}^{\text{ref}}$ desired by the generated scaffolds, we anchor scaffold generation to $\boldsymbol{S}^{\text{ref}}$ retain 3D shape, and apply a new value-guided sampling to encourage simultaneous 2D deviation. The algorithm is summarized in Appendix A.2.

4.2.1 Reference Noise Trajectory Recovery

To establish a reference anchor for controlled scaffold generation during inference time, RIDE first recovers the state and noise trajectory reproducing $\boldsymbol{S}^{\text{ref}}$ under the base diffusion model $p_\theta(\boldsymbol{S} \mid \boldsymbol{P}, \boldsymbol{F}^{\text{ref}})$, as illustrated in Figure 1b. We represent $\boldsymbol{S}^{\text{ref}}$ as $(\boldsymbol{X}^{\text{ref}}, \boldsymbol{V}^{\text{ref}})$, where $\boldsymbol{X}^{\text{ref}}$ and $\boldsymbol{V}^{\text{ref}}$ denote the scaffold atom positions and atom types, respectively. As the base model employs Gaussian diffusion for atom positions and categorical diffusion for atom types, we recover the reference position trajectory $\{\boldsymbol{X}_t^{\text{ref}}\}_{t=0}^T$ and reference atom-type trajectory $\{\boldsymbol{V}_t^{\text{ref}}\}_{t=0}^T$ using Eq. 1 and Eq. 3, respectively, with $\boldsymbol{X}_0^{\text{ref}} = \boldsymbol{X}^{\text{ref}}$ and $\boldsymbol{V}_0^{\text{ref}} = \boldsymbol{V}^{\text{ref}}$. The corresponding position noise trajectory $\{\boldsymbol{\epsilon}_t^{\text{ref}}\}_{t=1}^T$ and atom-type noise trajectory $\{\boldsymbol{g}_t^{\text{ref}}\}_{t=1}^T$ are then derived from the reference state trajectories using Eq. 2 and Eq. 4, respectively.

Here, we denote the reverse transition with a specific noise sequence from timestep t_1 to t_2 ($T \geq t_1 > t_2 \geq 0$) as

$$\mathcal{T}_{\theta, t_1:t_2}(\boldsymbol{S}_{t_1}, \{(\boldsymbol{\epsilon}_\tau, \boldsymbol{g}_\tau)\}_{\tau=t_2+1}^{t_1}; \boldsymbol{P}, \boldsymbol{F}^{\text{ref}}), \tag{8}$$

and the transition using the reference noise as

$$\mathcal{T}_{\theta, t_1:t_2}^{\text{ref}}(\boldsymbol{S}_{t_1}; \boldsymbol{P}, \boldsymbol{F}^{\text{ref}}) = \mathcal{T}_{\theta, t_1:t_2}(\boldsymbol{S}_{t_1}, \{(\boldsymbol{\epsilon}_\tau^{\text{ref}}, \boldsymbol{g}_\tau^{\text{ref}})\}_{\tau=t_2+1}^{t_1}; \boldsymbol{P}, \boldsymbol{F}^{\text{ref}}). \tag{9}$$

Thus, we have

$$\boldsymbol{S}_t^{\text{ref}} = \mathcal{T}_{\theta, T:t}^{\text{ref}}(\boldsymbol{S}_T^{\text{ref}}; \boldsymbol{P}, \boldsymbol{F}^{\text{ref}}). \tag{10}$$

For notational simplicity, we omit $\boldsymbol{P}$ and $\boldsymbol{F}^{\text{ref}}$ in $\mathcal{T}_\theta$ below.

4.2.2 Reference Trajectory Editing via Noise Perturbation

RIDE explores alternative scaffolds similar to the reference via reference trajectory editing, and implement this as noise perturbation over a segment of the reference noise trajectory between t_1 and t_2, with $t_2 = \max(t_1 - L, 0)$, where $L > 0$ is the segment length. Specifically, we design a perturbation distribution $p_\phi(\boldsymbol{\epsilon}_t, \boldsymbol{g}_t \mid t_1; \boldsymbol{P}, \boldsymbol{F}^{\text{ref}}, \{(\boldsymbol{\epsilon}_t^{\text{ref}}, \boldsymbol{g}_t^{\text{ref}})\}_{t=t_2+1}^{t_1}, L)$ over the atom position noise $\boldsymbol{\epsilon}_t$ and type noise $\boldsymbol{g}_t$ as follows ($\boldsymbol{P}$, $\boldsymbol{F}^{\text{ref}}$, $\{(\boldsymbol{\epsilon}_t^{\text{ref}}, \boldsymbol{g}_t^{\text{ref}})\}_{t=t_2+1}^{t_1}$ and L are omitted below for simplicity):

$$p_\phi(\boldsymbol{\epsilon}_t, \boldsymbol{g}_t \mid t_1) = p_{\phi_\epsilon}(\boldsymbol{\epsilon}_t \mid t_1) p_{\phi_g}(\boldsymbol{g}_t \mid t_1), \tag{11}$$

where $$p_{\phi_\epsilon}(\boldsymbol{\epsilon}_t \mid t_1) = \begin{cases}\mathcal{N}(\boldsymbol{\epsilon}_t; \mathbf{0}, \boldsymbol{I}), & t_1 \geq t > t_2 \\ \delta(\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_t^{\text{ref}}), & \text{otherwise,}\end{cases} \qquad p_{\phi_g}(\boldsymbol{g}_t \mid t_1) = \begin{cases}\text{Gumbel}(\boldsymbol{g}_t; 0, 1), & t_1 \geq t > t_2, \\ \delta(\boldsymbol{g}_t - \boldsymbol{g}_t^{\text{ref}}), & \text{otherwise,}\end{cases} \tag{12}$$

$\delta(\cdot)$ is the Dirac delta function. That is, RIDE explores different trajectories by editing the reference noise $(\boldsymbol{\epsilon}_t^{\text{ref}}, \boldsymbol{g}_t^{\text{ref}})$ between t_1 and t_2, replacing it with sampled noise from $(\mathcal{N}(\mathbf{0}, \boldsymbol{I}), \text{Gumbel}(0, 1))$, with the remaining noise intact. Unlike existing inversion-based methods that directly modify recovered intermediate states, we perturb the trajectories by resampling the diffusion noise at selected timesteps. This avoids the difficulty of designing appropriate state-space edits and reduces the risk of moving the generation too far from the diffusion manifold. Meanwhile, retaining a majority of the original reference noise trajectories outside t_1 and t_2 provides a strong mechanism for preserving 3D shape and binding pose enabled by p_θ.

Reference-Optimal Perturbation Segment Selection The optimal trajectory segment to edit depends on the reference ligand. Thus, RIDE adopts a reference-specific strategy to optimally select a noise trajectory segment, illustrated in Figure 1c. RIDE first discretizes the search space of t_1 as $t_1 \in \{\tau_n = \frac{nT}{N} : n = n_1, \ldots, n_2\}$, where $N > 0$ denotes the number of uniformly sampled timesteps along the full diffusion trajectory, and n_1 and n_2 specify the allowed range for t_1. For each candidate t_1, we generate K Monte Carlo samples following Eq. 11 and define a score J to quantify t_1 over the K samples as follows:

$$J(t_1 \mid \boldsymbol{S}^{\text{ref}}) = \frac{1}{K} \sum\nolimits_{k=1}^{K} \left[\mathcal{R}(\boldsymbol{S}^{(k)}(t_1), \boldsymbol{S}^{\text{ref}})\right], \tag{13}$$

where $$\boldsymbol{S}^{(k)}(t_1) = \mathcal{T}_{\theta, t_1:0}(\boldsymbol{S}_{t_1}^{\text{ref}}, \{(\boldsymbol{\epsilon}_t^{(k)}, \boldsymbol{g}_t^{(k)})\}_{t=1}^{t_1}), \quad \{(\boldsymbol{\epsilon}_t^{(k)}, \boldsymbol{g}_t^{(k)})\} \sim p_\phi(\cdot \mid t_1), \tag{14}$$

and $\mathcal{R}()$ is the reward function (Eq. 7). Given that the recovered reference trajectory already provides strong preservation of the reference 3D shape, we prioritize reducing 2D similarity when identifying the optimal t_1. Thus, we select the optimal t_1 as follows,

$$t_1^*(\boldsymbol{S}^{\text{ref}}) = \arg\max_{t_1 \in \{\tau_n\}} J(t_1 \mid \boldsymbol{S}^{\text{ref}}), \text{ and thus, } t_2^* = \max(t_1^* - L, 0), \tag{15}$$

where L is a hyperparameter determining the length of the perturbation segment. That is, the optimal t_1^* ($\boldsymbol{S}^{\text{ref}}$ is omitted for simplicity) identifies a segment that produces promising scaffolds under p_ϕ, providing a favorable starting point for subsequent optimization.

4.2.3 Value-Guided Scaffold Sampling

With the optimal perturbation segment starting from t_1^* , RIDE estimates the value of each intermediate scaffold state $\boldsymbol{S}_t$ ($t_1^* > t > t_2^*$) sampled under $p_\phi(\cdot \mid t_1^*)$ (Eq. 11) to guide state selection within the segment, as illustrated in Figure 1d. RIDE reformulates the value estimator in Eq. 5 to adapt it to a new reference-anchored sampling, using $\boldsymbol{S}_{t_1^*}^{\text{ref}}$ and samples from the segment endpoint t_2^*.

Theorem 1. *Given a selected perturbation starting timestep t_1^* and the corresponding reference state $\boldsymbol{S}_{t_1^*}^{\text{ref}}$, the value of any intermediate state $\boldsymbol{S}_t$, where $t_1^* > t > t_2^*$, can be expressed as*

$$V_\phi(\boldsymbol{S}_t \mid \boldsymbol{S}_{t_1^*}^{\text{ref}}) = \frac{\mathbb{E}_{p_{\theta,\phi}(\boldsymbol{S}_{t_2^*} \mid \boldsymbol{S}_{t_1^*}^{\text{ref}})}\left[w(\boldsymbol{S}_{t_2^*}; \boldsymbol{S}_t, \boldsymbol{S}_{t_1^*}^{\text{ref}}) \mathcal{R}(\mathcal{T}_{\theta, t_2^*:0}^{\text{ref}}(\boldsymbol{S}_{t_2^*}), \boldsymbol{S}^{\text{ref}})\right]}{\mathbb{E}_{p_{\theta,\phi}(\boldsymbol{S}_{t_2^*} \mid \boldsymbol{S}_{t_1^*}^{\text{ref}})}\left[w(\boldsymbol{S}_{t_2^*}; \boldsymbol{S}_t, \boldsymbol{S}_{t_1^*}^{\text{ref}})\right]}, \tag{16}$$

where $$w(\boldsymbol{S}_{t_2^*}; \boldsymbol{S}_t, \boldsymbol{S}_{t_1^*}^{\text{ref}}) = q(\boldsymbol{S}_t \mid \boldsymbol{S}_{t_2^*}) / q(\boldsymbol{S}_{t_1^*}^{\text{ref}} \mid \boldsymbol{S}_{t_2^*}), \tag{17}$$

$\mathcal{R}(\cdot, \boldsymbol{S}^{\text{ref}})$ is the reward function (Eq. 7), and $p_{\theta,\phi}(\boldsymbol{S}_{t_2^*} \mid \boldsymbol{S}_{t_1^*}^{\text{ref}})$ (conditioning on $\boldsymbol{P}$ and $\boldsymbol{F}^{\text{ref}}$ is omitted for simplicity) denotes the transition distribution from timestep t_1^* to t_2^* under $p_\phi(\cdot \mid t_1^*)$ and diffusion parameters θ.

The conditioning on $\boldsymbol{P}$, $\boldsymbol{F}^{\text{ref}}$ and $\boldsymbol{S}^{\text{ref}}$ is also omitted in $V_\phi(\boldsymbol{S}_t \mid \boldsymbol{S}_{t_1^*}^{\text{ref}})$ to align with $p_{\theta,\phi}(\boldsymbol{S}_{t_2^*} \mid \boldsymbol{S}_{t_1^*}^{\text{ref}})$. We provide the proof in Appendix A.3.1. In practice, we approximate the distribution $p_{\theta,\phi}(\boldsymbol{S}_{t_2^*} \mid \boldsymbol{S}_{t_1^*}^{\text{ref}})$ using samples obtained from it, which yields a Monte Carlo estimator $\widehat{V}_\phi(\boldsymbol{S}_t \mid \boldsymbol{S}_{t_1^*}^{\text{ref}})$, described in Appendix A.3.2.

Value-Guided Transition Within the perturbation segment between t_1^* and t_2^*, RIDE performs value-guided greedy search for optimal scaffold state candidates. Specifically, at selected timesteps, given the current state $\boldsymbol{S}_t$ ($t_1^* > t-1 > t_2^*$), RIDE samples noises from $p_\phi(\cdot \mid t_1^*)$ (Eq. 11), and identifies the optimal noise based on the estimated value (Eq. 21) as follows,

$$(\boldsymbol{\epsilon}_t^*, \boldsymbol{g}_t^*) = \arg\max_{(\boldsymbol{\epsilon}_t, \boldsymbol{g}_t) \sim p_\phi(\cdot \mid t_1^*)} \widehat{V}_\phi(\mathcal{T}_{\theta, t:t-1}(\boldsymbol{S}_t, (\boldsymbol{\epsilon}_t, \boldsymbol{g}_t) \mid \boldsymbol{S}_{t_1^*}^{\text{ref}}). \tag{18}$$

and thus, the optimal next scaffold state as

$$\boldsymbol{S}^*_{t-1} = \mathcal{T}_{\theta,t:t-1}(\boldsymbol{S}_t, (\boldsymbol{\epsilon}^*_t, \boldsymbol{g}^*_t)). \tag{19}$$

By iteratively selecting the optimal scaffold states along the edited noise trajectories, RIDE provides a new transition distribution to address scaffold hopping (Eq. 6).

5 EXPERIMENTS

5.1 EXPERIMENTAL SETTING

Datasets We adapt the SBDD models using the training sets of the original models, which are all derived from CrossDocked2020 training data (Francoeur et al., 2020). We evaluate RIDE on the test set introduced by Gao et al. (2026), excluding pocket-ligand complexes for which either no or all ligand atoms are identified as functional-group atoms, resulting in 60 complexes with 58 unique ligands. We use this test set because it focuses exclusively on human protein targets related to major diseases, avoiding the substantial proportion (67%) of non-human targets in the CrossDocked2020 test set and providing a more human-disease-relevant evaluation setting. This test set has no overlap with the CrossDocked2020 training set. The identification of functional-group atoms follows the protocol introduced by Guan et al. (2024). Across the 60 test complexes, ligands have an average of 28.7 heavy atoms, including 13.8 scaffold atoms and 14.9 functional-group atoms.

Baselines We evaluate three types of baselines on scaffold hopping: **(1)** conDitar (Gao et al., 2026) and IPDiff (Huang et al., 2024), adapted to scaffold hopping (Appendix A.1), referred to as conDitar-a and IPDiff-a, respectively. **(2)** DiffHopp (Torge et al., 2023); and **(3)** ShEPhERD (Adams et al., 2025), due to their strong generation performance and the availability of public checkpoints for adaptation. These three models, conDitar-a, IPDiff-a, and DiffHopp, serve two roles in our experiments: they are evaluated directly as scaffold hopping baselines, and they are also used as the base models on which RIDE performs inference-time editing. Details on datasets are available in Appendix A.4.

Evaluation Metrics We evaluate the generated scaffolds from two perspectives: **(1)** similarities with respect to the reference ligands (Sim_{2D}, Sim_{3D}), and **(2)** general physicochemical properties for drug development (Vina S/M/D, QED, SA, Conn.(%)). These metrics are described in detail in Appendix A.5.

Iterative Hopping To explore more structurally distinct scaffolds, we use a generated scaffold $\boldsymbol{S}^{\text{gen}}$ from RIDE as a new "reference" ligand and apply RIDE over $\boldsymbol{S}^{\text{gen}}$ for a second "hop." The details on $\boldsymbol{S}^{\text{gen}}$ selection are provided in Appendix A.6. We denote RIDE with two hops as $\text{RIDE}^{(2)}$, and the original RIDE with one hop as $\text{RIDE}^{(1)}$. Note that in $\text{RIDE}^{(2)}$, the reward (Eq. 7) in the second hop is evaluated against the original reference $\boldsymbol{S}^{\text{ref}}$.

Experimental Protocol In RIDE, we use the same hyperparameters as the corresponding base models. Because RIDE performs atom-level trajectory recovery of the reference scaffold, the edited scaffold is constrained to have the same number of atoms as the reference. To ensure fair comparison, we use the same scaffold size for all baselines. A potential pitfall of encouraging greater 2D structural novelty is that the generated molecules may deviate from the favorable chemical characteristics of the reference, leading to less desirable drug-like properties. Therefore, we introduce a second setting for $\text{RIDE}^{(1)}$ and $\text{RIDE}^{(2)}$ where Sim_{3D} is replaced with QED in the reward function (Eq. 7). Results under this reward function are indicated by "(QED)". The implementation details are provided in A.7.

5.2 EXPERIMENTAL RESULTS

5.2.1 OVERALL COMPARISON

Comparison of RIDE with Baselines Table 1 presents a comprehensive comparison between RIDE and all the baselines. Overall, $\text{RIDE}^{(1)}$ consistently achieves lower Sim_{2D} and higher Sim_{3D} than those from corresponding base models (as baselines). In terms of improvement of RIDE with a base model (e.g., $\text{RIDE}^{(1)}$ with conDitar-a(QED)) over its corresponding base model (e.g., conDitar-a), $\text{RIDE}^{(1)}$ achieves 11.7% decrease in Sim_{2D} and 7.3% increase in Sim_{3D} on average across all base models (ShEPhERD is excluded in this calculation). Compared to the baselines, the fundamental difference of RIDE is in its value-guided editing over the reference noise trajectories in the base models – the reference noise trajectories help preserve the reference 3D shape during editing, whereas value-guided search directs the generation toward low Sim_{2D}. The superior performance of $\text{RIDE}^{(1)}$ demonstrates that such editing is able to effectively steer the scaffold generation toward the desired chemical subspace beyond what is explored by the baselines. Meanwhile, on Vina S, $\text{RIDE}^{(1)}$ achieves an average improvement 39.1% over conDitar-a and DiffHopp,

Table 1: Comparison of RIDE with baselines on the test set with $L = 100$. Best and second-best results are shown in **bold** and underlined, respectively.

	Model	Similarity		Vina			QED ↑	SA ↑	Conn. (%) ↑
		Sim_{2D} ↓	Sim_{3D} ↑	Vina S ↓	Vina M ↓	Vina D ↓			
Baselines	conDitar-a	0.398	0.795	−7.192	−7.753	−8.564	0.455	0.622	85.2
	IPDiff-a	0.420	0.856	**−7.706**	**−8.176**	**−8.897**	0.456	0.619	94.7
	DiffHopp	0.426	0.812	−2.223	−6.209	−8.642	**0.535**	0.667	89.5
	ShEPhERD	0.417	0.881	−6.471	−7.402	−8.390	0.404	0.576	29.4
RIDE$^{(1)}$	conDitar-a	0.354	0.884	−7.508	−8.049	−8.833	0.427	0.601	91.5
	conDitar-a(QED)	0.354	0.859	−7.636	−8.163	−8.828	0.457	0.610	95.8
	IPDiff-a	0.363	**0.903**	−7.422	−7.975	−8.688	0.408	0.584	93.1
	IPDiff-a(QED)	0.356	0.897	−7.368	−7.975	−8.683	0.437	0.581	93.8
	DiffHopp	0.395	0.867	−4.593	−6.983	−8.820	0.511	0.696	96.4
RIDE$^{(2)}$	conDitar-a	0.325	0.893	−7.405	−7.988	−8.783	0.426	0.607	93.7
	conDitar-a(QED)	0.334	0.844	−7.488	−8.064	−8.801	0.470	0.604	93.0
	IPDiff-a	**0.322**	0.892	−7.088	−7.604	−8.477	0.403	0.587	95.3
	IPDiff-a(QED)	0.333	0.874	−7.046	−7.648	−8.550	0.471	0.595	94.4
	DiffHopp	0.372	0.863	−4.759	−7.109	−8.802	0.498	**0.708**	**98.2**

while maintaining similar Vina M and Vina D scores compared to those of all the base models. In terms of Conn.(%), RIDE$^{(1)}$ also introduces a noticeable average improvement 5.0% over the base models. Although RIDE$^{(1)}$ exhibits slight degradation in QED and SA scores, the resulting values remain within acceptable ranges. Such slight degradation also reflects RIDE$^{(1)}$'s exploration of novel chemical structures essential for scaffold hopping.

Base model-specific RIDE performance With Sim_{2D} and Sim_{3D} in the optimization objectives (Eqs. 6 and 7), RIDE$^{(1)}$ exhibits different improvement patterns across the two adapted SBDD models, conDitar-a and IPDiff-a. On conDitar-a, RIDE$^{(1)}$ improves both Sim_{2D} and Sim_{3D} substantially, reducing Sim_{2D} by 11.1% while increasing Sim_{3D} by 11.2%. In contrast, on IPDiff-a, which starts with worse Sim_{2D} and better Sim_{3D} as a baseline compared to conDitar-a, RIDE$^{(1)}$ achieves a larger gain in Sim_{2D} (13.6%) but a more moderate improvement in Sim_{3D} (5.5%). These results suggest that RIDE$^{(1)}$ can adapt to different SBDD base models by adjusting its optimization focus based on base models's generation behaviors. On DiffHopp, RIDE$^{(1)}$ also consistently improves both Sim_{2D} and Sim_{3D}, with improvements of 7.3% and 6.8%, respectively. Note that DiffHopp has a much worse Vina S (-2.223) than the other baselines, and therefore, the noise trajectory of the reference ligand, which has high binding affinity and thus good Vina S, could be out of distribution for DiffHopp. The fact that RIDE$^{(1)}$ substantially improves Vina S to -4.593 over DiffHopp suggests that its in-distribution perturbation and value-guided editing could partially compensate for the distribution mismatch. Overall, these improvements across different base models demonstrate that RIDE$^{(1)}$ can serve as a plug-and-play inference-time editing framework that generalizes across different molecular diffusion models.

In addition to the three structure-based models, we compare RIDE$^{(1)}$ with the ligand-based model ShEPhERD, which is specifically trained to generate molecules with shapes similar to the reference. RIDE$^{(1)}$ can achieve Sim_{3D} even higher than that of ShEPhERD, demonstrating its strong capability of retaining preferred 3D shapes.

Comparison of Different Reward Functions in RIDE We further investigate whether RIDE remains effective under various reward functions. Under RIDE$^{(1)}$, with a reward function combining Sim_{2D} and QED of generated molecules, conDitar-a(QED) and IPDiff-a(QED) achieve 7.0% and 7.1% higher QED than conDitar-a and IPDiff-a, respectively. The corresponding gains increase to 10.3% and 16.9% under RIDE$^{(2)}$. This suggests that RIDE can accommodate different reward functions, and could be extended to other molecular properties through appropriate reward design. Note that although not explicitly optimized, Sim_{3D} after Sim_{2D} and QED optimization in RIDE$^{(1)}$ still remains higher than that of the corresponding baseline, with improvements of 8.1% and 4.8% for conDitar-a(QED) and IPDiff-a(QED), respectively. This suggests that the reference noise trajectory helps retain favorable 3D shape even when Sim_{3D} is not explicitly included in the reward.

Comparison of RIDE$^{(1)}$ and RIDE$^{(2)}$ Table 1 also shows the performance changes between RIDE$^{(1)}$ and RIDE$^{(2)}$. Compared with RIDE$^{(1)}$, RIDE$^{(2)}$ generally further reduces Sim_{2D} while maintaining high Sim_{3D}. For example, Sim_{2D} decreases from 0.354 to 0.325 on conDitar-a and from 0.363 to 0.322 on IPDiff-a, while Sim_{3D} remains above 0.890 for both models. However, this increased 2D diversification is accompanied by weaker binding affinity: when averaged

over conDitar-a and IPDiff-a, with and without QED optimization, Vina S changes from -7.484 under RIDE$^{(1)}$ to -7.257 under RIDE$^{(2)}$. This might be due to the fact that the second hop uses a generated molecule as its anchor. This may cause deviations to accumulate across hops in properties not constrained by the similarity profile, potentially degrading binding affinity. Overall, these results highlight a trade-off in RIDE, and more generally, in scaffold hopping, between progressively improving the similarity profile and preserving binding affinity over multiple hops.

5.2.2 Comparison of Random Perturbation and Value-Guided Sampling

Table 2: Comparison of random perturbation and value-guided sampling on $L = T/10$ and $L = T$ (perturbation segment starting from t_1^* to the end of the diffusion trajectory), using conDitar-a as the base model. Best and second-best results are shown in **bold** and underlined, respectively.

	L		Similarity		Vina			QED ↑	SA ↑	Conn. (%) ↑
			Sim_{2D} ↓	Sim_{3D} ↑	Vina S ↓	Vina M ↓	Vina D ↓			
random perturbation (Eq. 11)	$T/10$	RIDE$^{(1)}$	0.387	**0.898**	−7.742	−8.213	−8.840	0.442	0.614	86.6
		RIDE$^{(2)}$	0.343	0.874	−7.581	−8.091	−8.813	0.425	0.604	90.0
	T	RIDE$^{(1)}$	0.391	0.866	**−7.837**	**−8.308**	−9.028	**0.462**	0.629	92.3
		RIDE$^{(2)}$	0.356	0.854	−7.763	−8.232	−8.915	0.450	0.620	95.2
value-guided sampling (Eq. 18)	$T/10$	RIDE$^{(1)}$	0.354	0.884	−7.508	−8.049	−8.833	0.427	0.601	91.5
		RIDE$^{(2)}$	**0.325**	0.893	−7.405	−7.988	−8.783	0.426	0.607	93.7
	T	RIDE$^{(1)}$	0.340	0.882	−7.612	−8.157	−8.904	0.440	0.628	95.1
		RIDE$^{(2)}$	0.334	0.887	−7.539	−8.054	−8.806	0.432	0.621	93.8

Table 2 compares results with random perturbation (Eq. 11) and value-guided sampling (Eq. 18) applied in RIDE. We choose conDitar-a for this study because RIDE applied to conDitar-a achieves strong performance and maintaining stable Vina S across hops. Overall, pairwise comparisons across the 4 settings ($T/10$ vs T, RIDE$^{(1)}$ vs RIDE$^{(2)}$) show that value-guided sampling reduces average Sim_{2D} by 8.3% and improves Sim_{3D} by 1.6% relative to random perturbation. This suggests that the performance gains arise from guided search toward desired properties rather than from perturbation-induced changes alone.

Table 2 also compares the performance of RIDE under two perturbation segment lengths: a shorter segment covering one-tenth of the diffusion trajectory ($L = T/10$) and a longer segment extending from t_1^* to the end of the diffusion trajectory (denoted as $L = T$). We use $T/10$ to provide a moderate discretization granularity over the diffusion trajectory, balancing exploration of distinct molecules and preservation of reference shapes. With $L = T/10$, when averaged over RIDE$^{(1)}$ and RIDE$^{(2)}$, value-guided sampling improves Sim_{2D} by 6.9% over random perturbation, with a minor improvement in Sim_{3D} of 0.31%. This trend becomes more significant with $L = T$, where value-guided sampling reduces Sim_{2D} by 9.6% and increases Sim_{3D} by 2.9%. The larger gains with $L = T$ may be explained by the longer perturbation segment, which preserves less information from the reference structure and provides greater room for value-guided sampling to adjust the similarity profile.

5.2.3 Comparison of Reference-Optimal t_1^* and Fixed t_1

Figures 2a and 2b illustrate Sim_{2D} and Sim_{3D} across candidate t_1 values and at the reference-optimal t_1^*, using conDitar-a and IPDiff-a as the base model, respectively, under random perturbation (Eq. 11). On both hops, compared with a fixed t_1, the reference-optimal t_1^* achieves the best Sim_{2D} while with competitive Sim_{3D}, indicating that the reference-optimal t_1^* identifies a strong starting point for subsequent value-guided sampling. Meanwhile, RIDE$^{(1)}$ and RIDE$^{(2)}$ exhibit different trends in Sim_{2D} and Sim_{3D} across candidate t_1 values. With RIDE$^{(1)}$, a later perturbation improves Sim_{3D} at the cost of degrading Sim_{2D}, which is consistent with the intuition that later perturbation preserves more of the reference shapes. A similar pattern in Sim_{2D} and Sim_{3D} is observed with RIDE$^{(2)}$, but with lower Sim_{2D} and Sim_{3D} than those in RIDE$^{(1)}$ and smaller variations across candidate t_1 values, indicating that the RIDE$^{(2)}$ is less sensitive to the choice of perturbation segment. This may be due to the fact that the RIDE$^{(2)}$ is anchored to a generated molecule that is already structurally distant from the original reference. Therefore, further perturbations can induce only limited additional changes in the similarity profile with respect to the reference molecule.

5.2.4 Case Studies

Figures 3b and 3c show two generated scaffolds by RIDE for AK1BA, a protein target involved in gastrointestinal cancers (Wang et al., 2026). Both generated scaffolds have low Sim_{2D} and high Sim_{3D} with the reference scaffold,

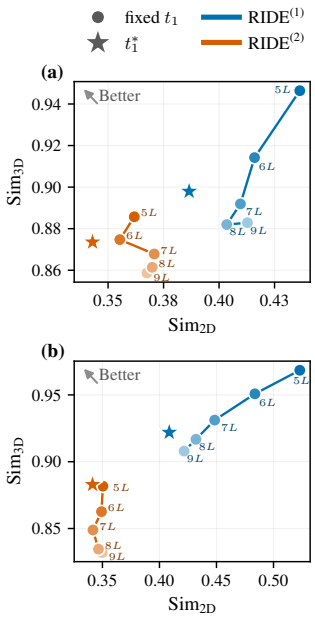


Figure 2: Comparison of Sim_{2D} vs. Sim_{3D} across $t_1 \in \{nL\}_{n=5}^{9}$ and t_1^* with $L = T/10$. **(a)** conDitar-a; **(b)** IPDiff-a.

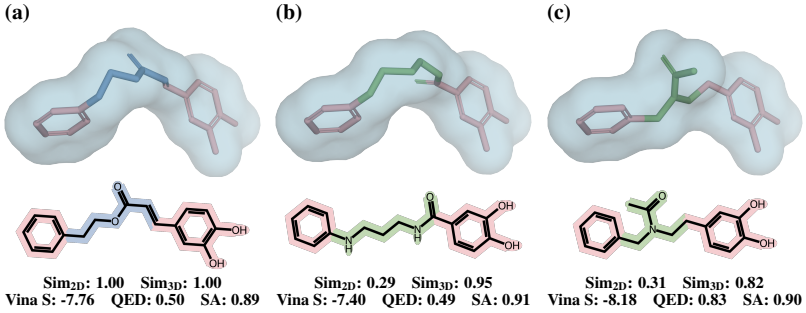


Figure 3: Example of generated scaffolds for AK1BA (PDB ID: 5liu).

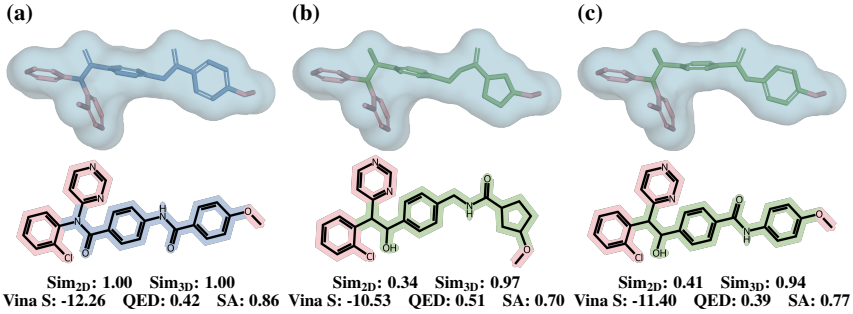


Figure 4: Example of generated scaffolds for TNKS2 (PDB ID: 5aeh).

as in Figure 3a. However, these scaffolds differ greatly in flexibility. The scaffold in Figure 3b is more flexible, with 6 rotatable bonds in the linker compared to 5 in that of the reference scaffold. Flexible molecules can better adapt to protein conformation shifts or resistance mutations in cancer (Fang et al., 2014). However, such flexibility often incurs an entropic penalty to the binding energies; this aligns with the slightly lower Vina S observed here. In contrast, the scaffold in Figure 3c has a shorter, more rigid linker than the reference, with only 4 rotatable linker bonds. More rigid molecules tend to have lower entropic binding penalties and higher selectivity, improving effectiveness while reducing off-target effects (Fang et al., 2014), Together, these examples show RIDE can generate scaffolds with different structures and properties.

Figures 4 presents another example from RIDE for TNKS2, a target implicated in many cancers (Huang et al., 2009). Both generated scaffolds (Figures 4b-c) have moderately low Sim_{2D} and very high Sim_{3D} with the reference scaffold in Figure 4a. The generated scaffolds share some common deviations from the reference: both use carbon to bind the two cyclic functional groups and replace one ketone with an alcohol. These changes alter physicochemical properties (e.g., pKa, lipophilicity) that influence absorption, distribution, metabolism, excretion, and toxicity properties of molecules. The scaffold in Figure 4b also uses cyclopentane in place of one benzene and adds an additional rotatable bond. These changes generally increase flexibility. Meanwhile, the generated scaffolds retain some key reference scaffold features – a benzene ring connected to another ring by a linker with a ketone. This helps the generated scaffolds attain very high Sim_{3D}, while still introducing substantial 2D variation via atom-type, bond-type, ring-size, and linker changes.

6 Conclusion

RIDE is a novel inference-time diffusion editing framework for scaffold hopping. Our experimental results demonstrate that RIDE is effective, consistently generating scaffolds with low 2D similarity and high 3D similarity to the reference scaffold. Furthermore, RIDE has demonstrated its effectiveness under a variety of different settings, including different base models, reward functions, perturbation segment lengths, and number of scaffold hopping iterations. These results motivate future work to extend the framework to optimize additional molecular properties and further improve generated molecule quality. Future work may also investigate the development of more adaptive perturbation strategies for even more effective optimization.

AI USE STATEMENT

During the preparation of this work, the authors used AI tools to assist with language editing, improving the clarity of the manuscript, and identifying relevant literature. All AI-assisted content was reviewed and verified by the authors.

ETHICS STATEMENT

RIDE is a novel inference-time diffusion editing framework for scaffold hopping, which is an important problem in chemistry. While we have not designed RIDE for any unsafe or harmful purpose, we acknowledge that not all generated scaffolds and molecules are safe, and that RIDE could generate harmful content. Therefore, we strongly encourage responsible human expert supervision for any use of RIDE or its generated scaffolds and molecules. Specifically, human expert chemists should verify the safety of any generated scaffolds and molecules before any attempt is made to synthesize or test them in the laboratory. Furthermore, we exhort all users of RIDE to follow all applicable safety regulations, ethical guidelines, laws, and professional best practices.

REPRODUCIBILITY STATEMENT

All the source code has been made available at `https://anonymous.4open.science/r/RIDE-C8A0`. All the datasets are public. We reported all the experimental protocal in Section 5.1, and all the hyperparameters in Appendix A.6.

REFERENCES

Keir Adams, Kento Abeywardane, Jenna Fromer, and Connor Coley. Shepherd: Diffusing shape, electrostatics, and pharmacophores for bioisosteric drug design. In International Conference on Learning Representations, volume 2025, pp. 32280–32342, 2025.

Hans-Joachim Böhm, Alexander Flohr, and Martin Stahl. Scaffold hopping. Drug discovery today: Technologies, 1 (3):217–224, 2004.

Ziqi Chen, Bo Peng, Tianhua Zhai, Daniel Adu-Ampratwum, and Xia Ning. Generating 3d small binding molecules using shape-conditioned diffusion models with guidance. Nature machine intelligence, 7(5):758–770, 2025.

Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. arXiv preprint arXiv:2209.14687, 2022.

Jerome Eberhardt, Diogo Santos-Martins, Andreas F. Tillack, and Stefano Forli. Autodock vina 1.2.0: New docking methods, expanded force field, and python bindings. Journal of Chemical Information and Modeling, 61(8):3891–3898, 08 2021. ISSN 1549-9596. doi: 10.1021/acs.jcim.1c00203. URL https://doi.org/10.1021/acs.jcim.1c00203.

Zengjun Fang, Yu'ning Song, Peng Zhan, Qingzhu Zhang, and Xinyong Liu. Conformational restriction: an effective tactic in 'follow-on'-based drug discovery. Future medicinal chemistry, 6(8):885–901, 2014.

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, 09 2020. ISSN 1549-9596. doi: 10.1021/acs.jcim.0c00411. URL https://doi.org/10.1021/acs.jcim.0c00411.

Ruoxi Gao, Jiangweizhi Peng, Ziqi Chen, Frazier N Baker, David C Kombo, John L Kane Jr, Andrew A Scholte, Yi Li, Matthew J LaMarche, Luigi I Iconaru, et al. Generating developable 3d molecules via pocket-conditioned diffusion and property-aware optimization. arXiv preprint arXiv:2607.12349, 2026.

Siyi Gu, Minkai Xu, Alexander Powers, Weili Nie, Tomas Geffner, Karsten Kreis, Jure Leskovec, Arash Vahdat, and Stefano Ermon. Aligning target-aware molecule diffusion models with exact energy optimization. Advances in neural information processing systems, 37:44040–44063, 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.

Yingqing Guo, Yukang Yang, Hui Yuan, and Mengdi Wang. Training-free guidance beyond differentiability: Scalable path steering with tree search in diffusion and flow models. Advances in Neural Information Processing Systems, 38:73343–73384, 2026.

Paul CD Hawkins, A Geoffrey Skillman, and Anthony Nicholls. Comparison of shape-matching and docking as virtual screening tools. Journal of medicinal chemistry, 50(1):74–82, 2007.

Xiaoxiao He, Quan Dao, Ligong Han, Song Wen, Minhao Bai, Di Liu, Han Zhang, Felix Juefei-Xu, Chaowei Tan, Bo Liu, et al. Dice: Discrete inversion enabling controllable editing for masked generative models. In 2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 762–772. IEEE, 2026.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control.(2022). URL https://arxiv. org/abs/2208.01626, 3(3), 2022.

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

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. Advances in neural information processing systems, 34:12454–12465, 2021.

Shih-Min A Huang, Yuji M Mishina, Shanming Liu, Atwood Cheung, Frank Stegmeier, Gregory A Michaud, Olga Charlat, Elizabeth Wiellette, Yue Zhang, Stephanie Wiessner, et al. Tankyrase inhibition stabilizes axin and antagonizes wnt signalling. Nature, 461(7264):614–620, 2009.

Zhilin Huang, Ling Yang, Xiangxin Zhou, Zhilong Zhang, Wentao Zhang, Xiawu Zheng, Jie Chen, Yu Wang, Bin CUI, and Wenming Yang. Protein-ligand interaction prior for binding-aware 3d molecule diffusion models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), International Conference on Learning Representations, volume 2024, pp. 18453–18474, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/50ca96a1a9ebe0b5e5688a504feb6107-Paper-Conference.pdf.

Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An edit friendly ddpm noise space: Inversion and manipulations. In 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12469–12478. IEEE, 2024.

Vineet Jain, Kusha Sareen, Mohammad Pedramfar, and Siamak Ravanbakhsh. Diffusion tree sampling: Scalable inference-time alignment of diffusion models. In The Thirty-ninth Annual Conference on Neural Information Processing Systems, 2025. URL https://openreview.net/forum?id=3D88hCO0Gd.

Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2416–2425. IEEE, 2022.

Sunwoo Kim, Minkyu Kim, and Dongmin Park. Test-time alignment of diffusion models without reward overoptimization. In International Conference on Learning Representations, volume 2025, pp. 87238–87272, 2025.

Yeongmin Kim, Donghyeok Shin, Byeonghu Na, Minsang Park, Richard Lee Kim, and Il chul Moon. Lookahead sample reward guidance for test-time scaling of diffusion models. In Forty-third International Conference on Machine Learning, 2026. URL https://openreview.net/forum?id=IbRm6gwmew.

Bohao Li, Xinyu Wu, Yu Cao, Jie Lin, Jingpeng Zhong, Hua Chen, Yongzhi Lu, Miru Tang, Jinping Lei, Ting Ran, et al. De novo molecular design via shape-constrained diffusion models. Journal of Chemical Information and Modeling, 66(8):4592–4606, 2026.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, et al. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. arXiv preprint arXiv:2408.08252, 2024.

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.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6038–6047. IEEE, 2023.

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.

Yuchen Shen, Chenhao Zhang, Sijie Fu, Chenghui Zhou, Newell Washburn, and Barnabas Poczos. Chemistry-inspired diffusion with non-differentiable guidance. In International Conference on Learning Representations, volume 2025, pp. 10211–10245, 2025.

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

Kim Yong Tan, Yueming Lyu, Ivor Tsang, and Yew-Soon Ong. Fast direct: Query-efficient online black-box guidance for diffusion-model target generation. In International Conference on Learning Representations, volume 2025, pp. 19385–19422, 2025.

Zhiwei Tang, Jiangweizhi Peng, Jiasheng Tang, Mingyi Hong, Fan Wang, and Tsung-Hui Chang. Inference-time alignment of diffusion models with direct noise optimization. arXiv preprint arXiv:2405.18881, 2024.

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.

Masatoshi Uehara, Yulai Zhao, Chenyu Wang, Xiner Li, Aviv Regev, Sergey Levine, and Tommaso Biancalani. Inference-time alignment in diffusion models with reward-guided generation: Tutorial and review. arXiv preprint arXiv:2501.09685, 2025.

Bram Wallace, Akash Gokul, and Nikhil Naik. Edict: Exact diffusion inversion via coupled transformations. In 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 22532–22541. IEEE, 2023.

Chen Wang, Zhuang Xiong, Xiang Wang, Jing Liang, Haoyang Li, Siqi He, Chengkun Wang, and Yang Zhang. The pathway network of aldo-keto reductase 1b10: a new perspective on gene-targeted therapy. npj Gut and Liver, 3(1): 20, 2026.

Brian Yang, Huangyuan Su, Nikolaos Gkanatsios, Tsung-Wei Ke, Ayush Jain, Jeff Schneider, and Katerina Fragkiadaki. Diffusion-es: Gradient-free planning with diffusion for autonomous and instruction-guided driving. In 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 15342–15353. IEEE, 2024.

Julong Yang, Wen Huang, Junhui Chen, and Jian Peng. Decode: Decoupling binding position and molecular conformation in 3d ligand diffusion for structure-based drug design. In Forty-third International Conference on Machine Learning.

Yuwei Yang, Xiaoqing Gong, Shukai Gu, Jing Li, Bo Liu, Yanan Tian, Qianqian Zhang, Xiaojun Yao, and Huanxiang Liu. Diffusion-based generative model with scaffold-hopping strategy yields highly potent bioactive molecules. Advanced Science, pp. e75674, 2026.

Kiwoong Yoo, Owen Oertell, Junhyun Lee, Sanghoon Lee, and Jaewoo Kang. Turbohopp: accelerated molecule scaffold hopping with consistency models. Advances in neural information processing systems, 37:41157–41185, 2024.

Shuangjia Zheng, Zengrong Lei, Haitao Ai, Hongming Chen, Daiguo Deng, and Yuedong Yang. Deep scaffold hopping with multimodal transformer neural networks. Journal of cheminformatics, 13(1):87, 2021.

A APPENDIX

A.1 ADAPTING DIFFUSION-BASED SBDD MODELS TO SCAFFOLD HOPPING

To adapt a pretrained diffusion-based SBDD model, denoted as $p_\theta((\boldsymbol{S}, \boldsymbol{F}) \mid \boldsymbol{P})$, to a base model $p_\theta(\boldsymbol{S} \mid \boldsymbol{P}, \boldsymbol{F}^{\text{ref}})$ for scaffold hopping, with $\boldsymbol{F}^{\text{ref}}$ as additional conditioning, we fix $\boldsymbol{F}^{\text{ref}}$ and restrict the diffusion loss to scaffold atoms and update the diffusion parameters as follows,

$$\min_\theta \mathbb{E}_{t,\boldsymbol{M}_t}\left[\sum\nolimits_{i=1}^{n} \mathbb{I}(\boldsymbol{a}_i \in \boldsymbol{S}_t)\mathcal{L}_\theta^i(\boldsymbol{M}_t; \boldsymbol{P}, t)\right], \tag{20}$$

where $\boldsymbol{M}_t = (\boldsymbol{S}_t, \boldsymbol{F}^{\text{ref}})$ denotes the partially noised ligand with fixed functional groups at t, $\mathbb{I}(\boldsymbol{a}_i \in \boldsymbol{S}_t)$ indicates whether the i-th atom belongs to the noised scaffold $\boldsymbol{S}_t$, and $\mathcal{L}_\theta^i$ denotes the SBDD diffusion loss on the i-th atom.

Although the additional conditioning can also be imposed through inpainting during inference time of SBDD models, inpainting can create a mismatch with the distribution $p_\theta((\boldsymbol{S}, \boldsymbol{F}) \mid \boldsymbol{P})$ learned during training, potentially leading to invalid molecular structures and unfavorable conformations. The distribution mismatch may also make the model more vulnerable to modifications on the diffusion trajectory, thereby reducing control over the generated structures. Compared to training a conditional generative model from scratch or inpainting, adapting a pretrained SBDD model offers multiple benefits: Adaptation leverages the pretrained SBDD model's capacity as a strong prior for generating chemically valid molecules with favorable binding affinity and 3D shapes; it also optimizes the model parameters to accommodate the additional conditioning ($\boldsymbol{F}^{\text{ref}}$).

A.2 ALGORITHMS

A.2.1 RIDE

Algorithm 1 RIDE

Input: Protein pocket $\boldsymbol{P}$, reference scaffold and functional groups $(\boldsymbol{S}^{\text{ref}}, \boldsymbol{F}^{\text{ref}})$, perturbation length L, diffusion model θ
Output: Generated scaffold $\boldsymbol{S}_0$

Reference noise trajectory recovery
$\{(\boldsymbol{X}_t^{\text{ref}}, \boldsymbol{V}_t^{\text{ref}})\}_{t=0}^T, \{(\boldsymbol{\epsilon}_t^{\text{ref}}, \boldsymbol{g}_t^{\text{ref}})\}_{t=1}^T = \text{INVERSION}(\boldsymbol{S}^{\text{ref}}, \boldsymbol{P}, \boldsymbol{F}^{\text{ref}})$ ▷ Details in Algorithm 2

Reference-optimal perturbation segment selection
$t_1^* = \text{SEGMENTSELECTION}(\{(\boldsymbol{X}_t^{\text{ref}}, \boldsymbol{V}_t^{\text{ref}})\}, \{(\boldsymbol{\epsilon}_t^{\text{ref}}, \boldsymbol{g}_t^{\text{ref}})\}, \boldsymbol{P}, \boldsymbol{F}^{\text{ref}}, L)$ ▷ Details in Algorithm 3
$t_2^* \leftarrow \max(t_1^* - L, 0)$

Value-guided scaffold sampling
$\boldsymbol{S}_{t_2^*} = \text{VALUEGUIDEDSAMPLING}(\{(\boldsymbol{X}_t^{\text{ref}}, \boldsymbol{V}_t^{\text{ref}})\}, \{(\boldsymbol{\epsilon}_t^{\text{ref}}, \boldsymbol{g}_t^{\text{ref}})\}, \boldsymbol{P}, \boldsymbol{F}^{\text{ref}}, t_1^*, t_2^*)$ ▷ Details in Algorithm 4
$\boldsymbol{S}_0 = \mathcal{T}_{\theta, t_2^*:0}^{\text{ref}}(\boldsymbol{S}_{t_2^*})$ ▷ Reuse reference noise
return $\boldsymbol{S}_0$

RIDE is summarized in Algorithm 1. The following sections describe each component in detail.

A.2.2 REFERENCE TRAJECTORY INVERSION

The inversion of reference scaffold is summarized in Algorithm 2.

A.2.3 OPTIMAL t_1 SELECTION

The procedure for selecting t_1^* is summarized in Algorithm 3.

A.2.4 VALUE-GUIDED SAMPLING

The procedure for value estimation and value-guided sampling is summarized in Algorithm 4.

Algorithm 2 Trajectory Inversion of $\boldsymbol{S}^{\mathrm{ref}}$

function INVERSION($\boldsymbol{S}^{\mathrm{ref}}, \boldsymbol{P}, \boldsymbol{F}^{\mathrm{ref}}$)
  $(\boldsymbol{X}_0^{\mathrm{ref}}, \boldsymbol{V}_0^{\mathrm{ref}}) \leftarrow (\boldsymbol{X}^{\mathrm{ref}}, \boldsymbol{V}^{\mathrm{ref}})$
  for $t = 1, \dots, T$ **do**
    $\hat{\boldsymbol{\epsilon}} \leftarrow \boldsymbol{\epsilon}_\theta(\boldsymbol{X}_{t-1}^{\mathrm{ref}}, t-1; \boldsymbol{V}_{t-1}^{\mathrm{ref}}, \boldsymbol{P}, \boldsymbol{F}^{\mathrm{ref}})$
    $\boldsymbol{X}_t^{\mathrm{ref}} \leftarrow \sqrt{\bar{\alpha}_t}\left(\frac{\boldsymbol{X}_{t-1}^{\mathrm{ref}} - \sqrt{1-\bar{\alpha}_{t-1}}\,\hat{\boldsymbol{\epsilon}}}{\sqrt{\bar{\alpha}_{t-1}}}\right) + \sqrt{1-\bar{\alpha}_t}\,\hat{\boldsymbol{\epsilon}}$
    $\boldsymbol{V}_t^{\mathrm{ref}} \sim q\big(\boldsymbol{V}_t \mid \boldsymbol{V}_{t-1}^{\mathrm{ref}}\big)$
  for $t = 1, \dots, T$ **do**
    $\boldsymbol{\epsilon}_t^{\mathrm{ref}} \leftarrow \dfrac{\boldsymbol{X}_{t-1}^{\mathrm{ref}} - \boldsymbol{\mu}_\theta(\boldsymbol{X}_t^{\mathrm{ref}}, t; \boldsymbol{V}_t^{\mathrm{ref}}, \boldsymbol{P}, \boldsymbol{F}^{\mathrm{ref}})}{\sigma_t}$
    $\hat{\boldsymbol{p}}_t \leftarrow p_\theta(\boldsymbol{V}_{t-1} \mid \boldsymbol{X}_t^{\mathrm{ref}}, \boldsymbol{V}_t^{\mathrm{ref}}; \boldsymbol{P}, \boldsymbol{F}^{\mathrm{ref}})$
    $c_i^{\mathrm{ref}} \leftarrow V_{t-1,i}^{\mathrm{ref}}, \quad i = 1, \dots, n$
    $$g_{t,i,k}^{\mathrm{ref}} \leftarrow \begin{cases} \left[\max\limits_{j \neq c_i^{\mathrm{ref}}} \log \hat{p}_{t,i,j} - \log \hat{p}_{t,i,c_i^{\mathrm{ref}}} + \delta\right]_+, & k = c_i^{\mathrm{ref}}, \\ 0, & k \neq c_i^{\mathrm{ref}}, \end{cases} \qquad i = 1, \dots, n, \quad k = 1, \dots, d$$
  return $\{(\boldsymbol{X}_t^{\mathrm{ref}}, \boldsymbol{V}_t^{\mathrm{ref}})\}_{t=0}^T, \{(\boldsymbol{\epsilon}_t^{\mathrm{ref}}, \boldsymbol{g}_t^{\mathrm{ref}})\}_{t=1}^T$
end function

Algorithm 3 Selection of t_1

function SEGMENTSELECTION($\{(\boldsymbol{X}_t^{\mathrm{ref}}, \boldsymbol{V}_t^{\mathrm{ref}})\}, \{(\boldsymbol{\epsilon}_t^{\mathrm{ref}}, \boldsymbol{g}_t^{\mathrm{ref}})\}, \boldsymbol{P}, \boldsymbol{F}^{\mathrm{ref}}, L$)
  for $t_1 \in \{\tau_n = \frac{nT}{N} : n = n_1, \dots, n_2\}$ **do**
    for $k = 1, \dots, K$ **do**
      $\{(\boldsymbol{\epsilon}_t^{(k)}, \boldsymbol{g}_t^{(k)})\} \sim p_\phi(\cdot \mid t_1)$
      $\boldsymbol{S}^{(k)}(t_1) \leftarrow \mathcal{T}_{\theta,t_1:0}(\boldsymbol{S}_{t_1}^{\mathrm{ref}}, \{(\boldsymbol{\epsilon}_t^{(k)}, \boldsymbol{g}_t^{(k)})\})$
    $J(t_1 \mid \boldsymbol{P}, \boldsymbol{F}^{\mathrm{ref}}, \boldsymbol{S}^{\mathrm{ref}}, L) \leftarrow \frac{1}{K}\sum_{k=1}^K \Big[\eta(1 - \mathrm{Sim}_{\mathrm{2D}}(\boldsymbol{S}^{(k)}(t_1), \boldsymbol{S}^{\mathrm{ref}})) + (1-\eta)\,\mathrm{Sim}_{\mathrm{3D}}(\boldsymbol{S}^{(k)}(t_1), \boldsymbol{S}^{\mathrm{ref}})\Big]$
  $t_1^* \leftarrow \arg\max_{t_1} J(t_1 \mid \boldsymbol{P}, \boldsymbol{F}^{\mathrm{ref}}, \boldsymbol{S}^{\mathrm{ref}}, L)$
  return t_1^*
end function

Algorithm 4 Value-Guided Sampling

function VALUEGUIDEDSAMPLING($\{(\boldsymbol{X}_t^{\mathrm{ref}}, \boldsymbol{V}_t^{\mathrm{ref}})\}, \{(\boldsymbol{\epsilon}_t^{\mathrm{ref}}, \boldsymbol{g}_t^{\mathrm{ref}})\}, \boldsymbol{P}, \boldsymbol{F}^{\mathrm{ref}}, t_1^*, t_2^*$)
  $\boldsymbol{S}_{t_1^*} \leftarrow \boldsymbol{S}_{t_1^*}^{\mathrm{ref}}$
  for $m = 1, \dots, M$ **do**
    $\boldsymbol{S}_{t_2^*}^{(m)} \sim p_\theta(\boldsymbol{S}_{t_2^*} \mid \boldsymbol{S}_{t_1^*}^{\mathrm{ref}}), \quad R^{(m)} = \mathcal{R}(\mathcal{T}_{\theta,t_2^*:0}^{\mathrm{ref}}(\boldsymbol{S}_{t_2^*}^{(m)}), \boldsymbol{S}^{\mathrm{ref}})$ ▷ Prepare lookahead samples
  for $t = t_1^*, \dots, t_2^* + 1$ **do**
    if t is a guided timestep **then**
      for $b = 1, \dots, B$ **do**
        $(\boldsymbol{\epsilon}_t^{(b)}, \boldsymbol{g}_t^{(b)}) \sim p_\phi(\cdot \mid t_1^*), \quad \boldsymbol{S}_{t-1}^{(b)} = \mathcal{T}_{\theta,t:t-1}(\boldsymbol{S}_t, (\boldsymbol{\epsilon}_t^{(b)}, \boldsymbol{g}_t^{(b)}))$ ▷ Propose a candidate for selection
        $\widehat{V}^{(b)} = \widehat{V}_\phi(\boldsymbol{S}_{t-1}^{(b)} \mid \boldsymbol{S}_{t_1^*}^{\mathrm{ref}})$ ▷ Estimate value using Eq. 21
      $b^* = \arg\max_b \widehat{V}^{(b)}, \quad \boldsymbol{S}_{t-1} \leftarrow \boldsymbol{S}_{t-1}^{(b^*)}$ ▷ Value-guided selection
    else
      $(\boldsymbol{\epsilon}_t, \boldsymbol{g}_t) \sim p_\phi(\cdot \mid t_1^*), \quad \boldsymbol{S}_{t-1} = \mathcal{T}_{\theta,t:t-1}(\boldsymbol{S}_t, (\boldsymbol{\epsilon}_t, \boldsymbol{g}_t))$ ▷ Reverse transition without selection
  return $\boldsymbol{S}_{t_2^*}$
end function

A.3 Additional theories

A.3.1 Proof for Theorem 1

Proof. By definition, the value of an intermediate state $\boldsymbol{S}_t$ is the expected terminal reward obtained by continuing the reverse process from $\boldsymbol{S}_t$.

Then,

$$
\begin{aligned}
V_\phi(\boldsymbol{S}_t \mid \boldsymbol{S}^{\mathrm{ref}}_{t_1^*}) &= \mathbb{E}_{p_{\theta,\phi}(\boldsymbol{S}_0 \mid \boldsymbol{S}_t, \boldsymbol{S}^{\mathrm{ref}}_{t_1^*})}\left[\mathcal{R}(\boldsymbol{S}_0, \boldsymbol{S}^{\mathrm{ref}})\right] \\
&= \int\int \mathcal{R}(\boldsymbol{S}_0, \boldsymbol{S}^{\mathrm{ref}}) p_{\theta,\phi}(\boldsymbol{S}_0 \mid \boldsymbol{S}_{t_2^*}, \boldsymbol{S}^{\mathrm{ref}}_{t_1^*}) p_{\theta,\phi}(\boldsymbol{S}_{t_2^*} \mid \boldsymbol{S}_t, \boldsymbol{S}^{\mathrm{ref}}_{t_1^*}) d\boldsymbol{S}_0 d\boldsymbol{S}_{t_2^*} \\
&= \int\int \mathcal{R}(\boldsymbol{S}_0, \boldsymbol{S}^{\mathrm{ref}}) \delta(\boldsymbol{S}_0 - \mathcal{T}^{\mathrm{ref}}_{\theta,t_2^*:0}(\boldsymbol{S}_{t_2^*})) p_{\theta,\phi}(\boldsymbol{S}_{t_2^*} \mid \boldsymbol{S}_t, \boldsymbol{S}^{\mathrm{ref}}_{t_1^*}) d\boldsymbol{S}_0 d\boldsymbol{S}_{t_2^*} \\
&= \int \mathcal{R}(\mathcal{T}^{\mathrm{ref}}_{\theta,t_2^*:0}(\boldsymbol{S}_{t_2^*}), \boldsymbol{S}^{\mathrm{ref}}) p_{\theta,\phi}(\boldsymbol{S}_{t_2^*} \mid \boldsymbol{S}_t, \boldsymbol{S}^{\mathrm{ref}}_{t_1^*}) d\boldsymbol{S}_{t_2^*} \\
&= \int \mathcal{R}(\mathcal{T}^{\mathrm{ref}}_{\theta,t_2^*:0}(\boldsymbol{S}_{t_2^*}), \boldsymbol{S}^{\mathrm{ref}}) \frac{p_{\theta,\phi}(\boldsymbol{S}_{t_2^*}, \boldsymbol{S}_t \mid \boldsymbol{S}^{\mathrm{ref}}_{t_1^*})}{p_{\theta,\phi}(\boldsymbol{S}_t \mid \boldsymbol{S}^{\mathrm{ref}}_{t_1^*})} d\boldsymbol{S}_{t_2^*} \\
&= \int \mathcal{R}(\mathcal{T}^{\mathrm{ref}}_{\theta,t_2^*:0}(\boldsymbol{S}_{t_2^*}), \boldsymbol{S}^{\mathrm{ref}}) \frac{p_{\theta,\phi}(\boldsymbol{S}_t \mid \boldsymbol{S}_{t_2^*}, \boldsymbol{S}^{\mathrm{ref}}_{t_1^*}) p_{\theta,\phi}(\boldsymbol{S}_{t_2^*} \mid \boldsymbol{S}^{\mathrm{ref}}_{t_1^*})}{p_{\theta,\phi}(\boldsymbol{S}_t \mid \boldsymbol{S}^{\mathrm{ref}}_{t_1^*})} dS_{t_2^*} \\
&= \frac{\int \mathcal{R}(\mathcal{T}^{\mathrm{ref}}_{\theta,t_2^*:0}(\boldsymbol{S}_{t_2^*}), \boldsymbol{S}^{\mathrm{ref}}) p_{\theta,\phi}(\boldsymbol{S}_t \mid \boldsymbol{S}_{t_2^*}, \boldsymbol{S}^{\mathrm{ref}}_{t_1^*}) p_{\theta,\phi}(\boldsymbol{S}_{t_2^*} \mid \boldsymbol{S}^{\mathrm{ref}}_{t_1^*}) d\boldsymbol{S}_{t_2^*}}{\int p_{\theta,\phi}(\boldsymbol{S}_t \mid \boldsymbol{S}_{t_2^*}, \boldsymbol{S}^{\mathrm{ref}}_{t_1^*}) p_{\theta,\phi}(\boldsymbol{S}_{t_2^*} \mid \boldsymbol{S}^{\mathrm{ref}}_{t_1^*}) d\boldsymbol{S}_{t_2^*}} \\
&= \frac{\int \mathcal{R}(\mathcal{T}^{\mathrm{ref}}_{\theta,t_2^*:0}(\boldsymbol{S}_{t_2^*}), \boldsymbol{S}^{\mathrm{ref}}) q(\boldsymbol{S}_t \mid \boldsymbol{S}_{t_2^*}, \boldsymbol{S}^{\mathrm{ref}}_{t_1^*}) p_{\theta,\phi}(\boldsymbol{S}_{t_2^*} \mid \boldsymbol{S}^{\mathrm{ref}}_{t_1^*}) d\boldsymbol{S}_{t_2^*}}{\int q(\boldsymbol{S}_t \mid \boldsymbol{S}_{t_2^*}, \boldsymbol{S}^{\mathrm{ref}}_{t_1^*}) p_{\theta,\phi}(\boldsymbol{S}_{t_2^*} \mid \boldsymbol{S}^{\mathrm{ref}}_{t_1^*}) d\boldsymbol{S}_{t_2^*}} \\
&= \frac{\int \mathcal{R}(\mathcal{T}^{\mathrm{ref}}_{\theta,t_2^*:0}(\boldsymbol{S}_{t_2^*}), \boldsymbol{S}^{\mathrm{ref}}) \cancel{q(\boldsymbol{S}^{\mathrm{ref}}_{t_1^*} \mid \boldsymbol{S}_t)} \frac{q(\boldsymbol{S}_t \mid \boldsymbol{S}_{t_2^*})}{q(\boldsymbol{S}^{\mathrm{ref}}_{t_1^*} \mid \boldsymbol{S}_{t_2^*})} p_{\theta,\phi}(\boldsymbol{S}_{t_2^*} \mid \boldsymbol{S}^{\mathrm{ref}}_{t_1^*}) d\boldsymbol{S}_{t_2^*}}{\int \cancel{q(\boldsymbol{S}^{\mathrm{ref}}_{t_1^*} \mid \boldsymbol{S}_t)} \frac{q(\boldsymbol{S}_t \mid \boldsymbol{S}_{t_2^*})}{q(\boldsymbol{S}^{\mathrm{ref}}_{t_1^*} \mid \boldsymbol{S}_{t_2^*})} p_{\theta,\phi}(\boldsymbol{S}_{t_2^*} \mid \boldsymbol{S}^{\mathrm{ref}}_{t_1^*}) d\boldsymbol{S}_{t_2^*}} \\
&= \frac{\mathbb{E}_{p_{\theta,\phi}(\boldsymbol{S}_{t_2^*} \mid \boldsymbol{S}^{\mathrm{ref}}_{t_1^*})}\left[\frac{q(\boldsymbol{S}_t \mid \boldsymbol{S}_{t_2^*})}{q(\boldsymbol{S}^{\mathrm{ref}}_{t_1^*} \mid \boldsymbol{S}_{t_2^*})} \mathcal{R}(\mathcal{T}^{\mathrm{ref}}_{\theta,t_2^*:0}(\boldsymbol{S}_{t_2^*}), \boldsymbol{S}^{\mathrm{ref}})\right]}{\mathbb{E}_{p_{\theta,\phi}(\boldsymbol{S}_{t_2^*} \mid \boldsymbol{S}^{\mathrm{ref}}_{t_1^*})}\left[\frac{q(\boldsymbol{S}_t \mid \boldsymbol{S}_{t_2^*})}{q(\boldsymbol{S}^{\mathrm{ref}}_{t_1^*} \mid \boldsymbol{S}_{t_2^*})}\right]}
\end{aligned}
$$

□

A.3.2 Theorem 2

Theorem 2. *Given M samples $\{\boldsymbol{S}^{(m)}_{t_2^*}\}_{m=1}^M$ from $p_{\theta,\phi}(\boldsymbol{S}_{t_2^*} \mid \boldsymbol{S}^{\mathrm{ref}}_{t_1^*})$, the value of an intermediate state $\boldsymbol{S}_t$ in Eq. 16 can be estimated as*

$$
\widehat{V}_\phi(\boldsymbol{S}_t \mid \boldsymbol{S}^{\mathrm{ref}}_{t_1^*}) = \sum_{m=1}^{M} \mathrm{softmax}_m(\{\ell_t^{(j)}\}_{j=1}^M) \mathcal{R}(\mathcal{T}^{\mathrm{ref}}_{\theta,t_2^*:0}(\boldsymbol{S}^{(m)}_{t_2^*}), \boldsymbol{S}^{\mathrm{ref}}), \tag{21}
$$

where

$$
\ell_t^{(m)} = \log q(\boldsymbol{S}_t \mid \boldsymbol{S}^{(m)}_{t_2^*}) - \log q(\boldsymbol{S}^{\mathrm{ref}}_{t_1^*} \mid \boldsymbol{S}^{(m)}_{t_2^*}). \tag{22}
$$

The two log-transition terms in $\ell_t^{(m)}$ are calculated in closed form using the Gaussian forward kernel for atom positions and the categorical forward kernel for atom types. We provide the proof and detailed derivation of the log-transition terms below.

Proof for theorem 2

Proof. By applying a Monte Carlo approximation to the expectations in Eq. 16, we obtain

$$\widehat{V}_\phi(\boldsymbol{S}_t \mid \boldsymbol{S}^{\text{ref}}_{t_1^*}) = \frac{\sum_{m=1}^{M} \frac{q(\boldsymbol{S}_t|\boldsymbol{S}^{(m)}_{t_2^*})}{q(\boldsymbol{S}^{\text{ref}}_{t_1^*}|\boldsymbol{S}^{(m)}_{t_2^*})} \mathcal{R}(\mathcal{T}^{\text{ref}}_{\theta,t_2^*:0}(\boldsymbol{S}^{(m)}_{t_2^*}), \boldsymbol{S}^{\text{ref}})}{\sum_{m=1}^{M} \frac{q(\boldsymbol{S}_t|\boldsymbol{S}^{(m)}_{t_2^*})}{q(\boldsymbol{S}^{\text{ref}}_{t_1^*}|\boldsymbol{S}^{(m)}_{t_2^*})}},$$

with $w_t^{(m)} = \frac{q(\boldsymbol{S}_t|\boldsymbol{S}^{(m)}_{t_2^*})}{q(\boldsymbol{S}^{\text{ref}}_{t_1^*}|\boldsymbol{S}^{(m)}_{t_2^*})}$ and $\widetilde{w}_t^{(m)} = \frac{w_t^{(m)}}{\sum_{j=1}^{M} w_t^{(j)}}$,

$$\widehat{V}_\phi(\boldsymbol{S}_t \mid \boldsymbol{S}^{\text{ref}}_{t_1^*}) = \sum_{m=1}^{M} \widetilde{w}_t^{(m)} \mathcal{R}(\mathcal{T}^{\text{ref}}_{\theta,t_2^*:0}(\boldsymbol{S}^{(m)}_{t_2^*}), \boldsymbol{S}^{\text{ref}}).$$

Define

$$\ell_t^{(m)} = \log q(\boldsymbol{S}_t \mid \boldsymbol{S}^{(m)}_{t_2^*}) - \log q(\boldsymbol{S}^{\text{ref}}_{t_1^*} \mid \boldsymbol{S}^{(m)}_{t_2^*}),$$

then,

$$\widetilde{w}_t^{(m)} = \frac{\exp(\ell_t^{(m)})}{\sum_{j=1}^{M} \exp(\ell_t^{(j)})} = \text{softmax}_m(\{\ell_t^{(j)}\}_{j=1}^{M}).$$

Therefore,

$$\widehat{V}_\phi(\boldsymbol{S}_t \mid \boldsymbol{S}^{\text{ref}}_{t_1^*}) = \sum_{m=1}^{M} \text{softmax}_m(\{\ell_t^{(j)}\}_{j=1}^{M}) \mathcal{R}(\mathcal{T}^{\text{ref}}_{\theta,t_2^*:0}(\boldsymbol{S}^{(m)}_{t_2^*}), \boldsymbol{S}^{\text{ref}}).$$

□

Detailed derivations of log transitions

Proof. The forward transition factorizes over atom positions and atom types,

$$q(\boldsymbol{S}_t \mid \boldsymbol{S}_{t_2^*}) = q(\boldsymbol{X}_t \mid \boldsymbol{X}_{t_2^*}) q(\boldsymbol{V}_t \mid \boldsymbol{V}_{t_2^*}),$$

then,

$$\log q(\boldsymbol{S}_t \mid \boldsymbol{S}_{t_2^*}) = \log q(\boldsymbol{X}_t \mid \boldsymbol{X}_{t_2^*}) + \log q(\boldsymbol{V}_t \mid \boldsymbol{V}_{t_2^*}).$$

Similarly,

$$\log q(\boldsymbol{S}^{\text{ref}}_{t_1^*} \mid \boldsymbol{S}_{t_2^*}) = \log q(\boldsymbol{X}^{\text{ref}}_{t_1^*} \mid \boldsymbol{X}_{t_2^*}) + \log q(\boldsymbol{V}^{\text{ref}}_{t_1^*} \mid \boldsymbol{V}_{t_2^*}).$$

Therefore,

$$\ell_t^{(m)} = \ell_{t,\boldsymbol{X}}^{(m)} + \ell_{t,\boldsymbol{V}}^{(m)},$$

where

$$\ell_{t,\boldsymbol{X}}^{(m)} = \log q(\boldsymbol{X}_t \mid \boldsymbol{X}^{(m)}_{t_2^*}) - \log q(\boldsymbol{X}^{\text{ref}}_{t_1^*} \mid \boldsymbol{X}^{(m)}_{t_2^*}),$$

and

$$\ell_{t,\boldsymbol{V}}^{(m)} = \log q(\boldsymbol{V}_t \mid \boldsymbol{V}^{(m)}_{t_2^*}) - \log q(\boldsymbol{V}^{\text{ref}}_{t_1^*} \mid \boldsymbol{V}^{(m)}_{t_2^*}).$$

For atom positions, the DDPM forward process admits the closed-form transition

$$q(\boldsymbol{X}_t \mid \boldsymbol{X}_{t_2^*}) = \mathcal{N}(\boldsymbol{X}_t; \sqrt{\frac{\bar{\alpha}_t^x}{\bar{\alpha}_{t_2^*}^x}}\, \boldsymbol{X}_{t_2^*}, (1 - \frac{\bar{\alpha}_t^x}{\bar{\alpha}_{t_2^*}^x})\boldsymbol{I}).$$

Let

$$\alpha^x_{t|t_2^*} = \frac{\bar{\alpha}_t^x}{\bar{\alpha}_{t_2^*}^x}, \qquad \sigma^2_{t|t_2^*} = 1 - \alpha^x_{t|t_2^*}.$$

Then, for a scaffold with n atoms,

$$\log q(\boldsymbol{X}_t \mid \boldsymbol{X}^{(m)}_{t_2^*}) = -\frac{3n}{2} \log(2\pi\sigma^2_{t|t_2^*}) - \frac{\left\| \boldsymbol{X}_t - \sqrt{\alpha^x_{t|t_2^*}} \boldsymbol{X}^{(m)}_{t_2^*} \right\|_2^2}{2\sigma^2_{t|t_2^*}}.$$

Similarly,

$$\log q(\boldsymbol{X}_{t_1^*}^{\text{ref}} \mid \boldsymbol{X}_{t_2^*}^{(m)}) = -\frac{3n}{2}\log(2\pi\sigma_{t_1^*|t_2^*}^2) - \frac{\left\|\boldsymbol{X}_{t_1^*}^{\text{ref}} - \sqrt{\alpha_{t_1^*|t_2^*}^x}\boldsymbol{X}_{t_2^*}^{(m)}\right\|_2^2}{2\sigma_{t_1^*|t_2^*}^2}.$$

Therefore,

$$\begin{aligned}\ell_{t,\boldsymbol{X}}^{(m)} = &-\frac{3n}{2}\log\frac{\sigma_{t|t_2^*}^2}{\sigma_{t_1^*|t_2^*}^2} - \frac{\left\|\boldsymbol{X}_t - \sqrt{\alpha_{t|t_2^*}^x}\boldsymbol{X}_{t_2^*}^{(m)}\right\|_2^2}{2\sigma_{t|t_2^*}^2} \\ &+ \frac{\left\|\boldsymbol{X}_{t_1^*}^{\text{ref}} - \sqrt{\alpha_{t_1^*|t_2^*}^x}\boldsymbol{X}_{t_2^*}^{(m)}\right\|_2^2}{2\sigma_{t_1^*|t_2^*}^2}.\end{aligned}$$

For atom types, the forward process is

$$q(\boldsymbol{V}_t \mid \boldsymbol{V}_{t_2^*}) = \text{Cat}(\alpha_{t|t_2^*}^c\,\text{onehot}(\boldsymbol{V}_{t_2^*}) + (1-\alpha_{t|t_2^*}^c)\frac{1}{d}\mathbf{1}),$$

where

$$\alpha_{t|t_2^*}^c = \frac{\bar{\alpha}_t^c}{\bar{\alpha}_{t_2^*}^c}.$$

For a n-atom scaffold,

$$\log q(\boldsymbol{V}_t \mid \boldsymbol{V}_{t_2^*}^{(m)}) = \sum_{i=1}^n \log\left[\alpha_{t|t_2^*}^c\,\text{onehot}(V_{t_2^*,i}^{(m)}) + (1-\alpha_{t|t_2^*}^c)\frac{1}{d}\mathbf{1}\right]_{V_{t,i}},$$

and

$$\log q(\boldsymbol{V}_{t_1^*}^{\text{ref}} \mid \boldsymbol{V}_{t_2^*}^{(m)}) = \sum_{i=1}^n \log\left[\alpha_{t_1^*|t_2^*}^c\,\text{onehot}(V_{t_2^*,i}^{(m)}) + (1-\alpha_{t_1^*|t_2^*}^c)\frac{1}{d}\mathbf{1}\right]_{V_{t_1^*,i}^{\text{ref}}}.$$

Therefore,

$$\begin{aligned}\ell_{t,\boldsymbol{V}}^{(m)} = &\sum_{i=1}^n \log(\left[\alpha_{t|t_2^*}^c\,\text{onehot}(V_{t_2^*,i}^{(m)}) + (1-\alpha_{t|t_2^*}^c)\frac{1}{d}\mathbf{1}\right]_{V_{t,i}}) \\ &- \sum_{i=1}^n \log(\left[\alpha_{t_1^*|t_2^*}^c\,\text{onehot}(V_{t_2^*,i}^{(m)}) + (1-\alpha_{t_1^*|t_2^*}^c)\frac{1}{d}\mathbf{1}\right]_{V_{t_1^*,i}^{\text{ref}}}).\end{aligned}$$

□

A.4 Baselines for Experiments

We evaluate three types of baselines on scaffold hopping: **(1)** SBDD models adapted to scaffold hopping (Appendix A.1), **(2)** a structure-based scaffold hopping model, DiffHopp (Torge et al., 2023) and **(3)** a ligand-based scaffold hopping model, ShEPhERD (Adams et al., 2025). For SBDD models, we choose two recent state-of-the-art diffusion-based models, conDitar (Gao et al., 2026) and IPDiff (Huang et al., 2024), due to their strong generation performance and the availability of public checkpoints for adaptation. To enable stable trajectory recovery and better preservation of reference shape and interaction with the binding pockets, we slightly modify IPDiff's generation process, with details provided in Appendix A.8. We denote the adapted models of conDitar and IPDiff as conDitar-a and IPDiff-a, respectively. We also include DiffHopp, the only structure-based diffusion model for scaffold hopping with publicly available checkpoints. These three models, conDitar-a, IPDiff-a, and DiffHopp, serve two roles in our experiments: they are evaluated directly as scaffold hopping baselines, and they are also used as the base models on which `RIDE` performs inference-time editing. In addition to these models, we use a ligand-based model, ShEPhERD, as a baseline. ShEPhERD performs scaffold hopping via ligand-based inpainting given 3D shape, functional group, and/or electrostatic conditions. We do not use ShEPhERD as a base model for `RIDE` because it generates molecules through inpainting, making both `RIDE` and its adaptation strategy not directly applicable.

A.5 Evaluation Metrics

We evaluate the generated scaffolds from two perspectives: **(1)** similarities with respect to the reference ligands, and **(2)** general physicochemical properties for drug development. Please note, we use similarities between entire molecules as a proxy for scaffold similarity, as functional groups already remain fixed for a given reference. For similarities with the reference ligands, we use Sim_{2D}, which is measured by the Tanimoto distance between molecular fingerprints, and Sim_{3D}, which is calculated using ROCS tool following Hawkins et al. (2007). To ensure that reference functional-group atoms remain at the same positions in the generated and reference molecules, we skip the pre-alignment step in ROCS before Sim_{3D} computation, ensuring Sim_{3D} reflects the generated pose. For general physicochemical properties, we report three binding affinity metrics predicted by AutoDock Vina (Eberhardt et al., 2021): Vina Score (Vina S) on the original generated pose, Vina Minimization (Vina M) after local energy minimization, and Vina Dock (Vina D) after docking. We also use quantitative estimates of drug-likeness (QED) and synthetic accessibility (SA) following Luo et al. (2021) as metrics to assess the overall physicochemical profile and synthesizability of generated molecules. In addition, following Torge et al. (2023), we evaluate Conn.(%), defined as the percentage of generated molecules in which the generated scaffold is successfully connected to reference functional groups. For each reference ligand, we generate 100 samples and consider only the resulting valid and connected molecules for evaluation. Each metric is first averaged over the generated molecules on a per-reference basis and then across all test complexes.

A.6 Anchor molecule selection for $\texttt{RIDE}^{(2)}$

For $\texttt{RIDE}^{(2)}$, we select the anchor molecule from the molecules generated in one hop using a simple hierarchical filtering procedure. For each reference, we first retain the top 30% of molecules according to 3D shape similarity to the reference, and then select the candidate with the lowest 2D similarity as the anchor for two hops.

A.7 Implementation Details and Reproducibility

$\texttt{RIDE}^{(1)}$ uses $\lambda = 1$ in its reward for reference-optimal perturbation segment selection for all models. $\texttt{RIDE}^{(2)}$ follows the same settings, except for IPDiff-a, where $\lambda = 0.7$ segment selection. During value-guided scaffold sampling, $\texttt{RIDE}^{(1)}$ uses $\lambda = 0.8$ for its reward, while $\texttt{RIDE}^{(2)}$ uses $\lambda = 0.5$ to better encourage retaining the reference 3D shape over multiple hops. Both conDitar-a and IPDiff-a use $T = 1000$, following their original papers (Gao et al., 2026; Huang et al., 2024). For these models, $\texttt{RIDE}^{(1)}$ and $\texttt{RIDE}^{(2)}$ use $N = 10$, $n_1 = 5$ and $n_2 = 9$. DiffHopp uses $T = 500$, following its original paper (Torge et al., 2023). Therefore, $\texttt{RIDE}^{(1)}$ and $\texttt{RIDE}^{(2)}$ use $N = 5$, $n_1 = 3$ and $n_2 = 5$ for this model to maintain the same segment length as other models. For selecting optimal segment in 13, we use $K = 100$. For estimating the value in 21, we use $M = 1000$.

A.8 Modification to IPDiff's Generation Process

IPDiff extends the standard SBDD diffusion process by incorporating protein-ligand interaction embeddings into both the forward and reverse diffusion processes. During generation, IPDiff dynamically extracts the interaction embeddings at each timestep using the predicted clean ligand from the previous timestep, leading to transition distributions evolving over the denoising trajectory. To enable a consistent and stable trajectory recovery and preserve the reference 3D shape and pocket-ligand interaction pattern, we modify IPDiff's original inference process by fixing the interaction embeddings to those extracted from the anchor ligand during generation in both hopping iterations.

A.9 Additional Experimental Results

Comparison of Value-Guided Sampling and Random Perturbation under QED optimization We observe a similar behavior to that in Figure 2 when the downstream objective is changed from Sim_{3D} to QED, as shown in Figure 5. Across candidate t_1 values, lower Sim_{2D} is accompanied by changes in QED, showing that the perturbation starting timestep also controls the trade-off between the Sim_{2D} and QED objectives. By comparing the QED curves with the Sim_{3D} curves, we observe that QED and Sim_{3D} show similar responses to the perturbation starting timestep, with higher QED generally corresponding to higher Sim_{3D} under random perturbation. This suggests that 3D preservation is partially aligned with good QED. Our similarity-based dynamic timestep selection also identifies a reasonable starting point balancing Sim_{2D}, QED and Sim_{3D} for subsequent optimization.

Comparison of Inversion and Non-Inversion By comparing results with and without inversion in Table 3, we observe that removing inversion leads to worse Sim_{3D} and Sim_{2D}, highlighting the importance of trajectory inversion in balancing 2D novelty and 3D shape preservation. With $\texttt{RIDE}^{(1)}$, trajectory inversion consistently improves both

Table 3: Comparison of generation results with and without trajectory inversion using conDitar-a as the base model. Random perturbation corresponds to generation under $p_\phi(\cdot \mid t_1^*)$, while value-guided sampling further performs search from t_1^* to $t = 0$. Best and second-best results are shown in **bold** and underlined, respectively.

Inversion	Sampling		Similarity		Vina			QED ↑	SA ↑	Conn. (%) ↑
			Sim_{2D} ↓	Sim_{3D} ↑	Vina S ↓	Vina M ↓	Vina D ↓			
✓	random perturbation	RIDE$^{(1)}$	0.391	0.866	**−7.837**	**−8.308**	−9.028	**0.462**	0.629	92.3
		RIDE$^{(2)}$	0.356	0.854	−7.763	−8.232	−8.915	0.450	0.620	95.2
	value-guided sampling	RIDE$^{(1)}$	0.340	0.882	−7.612	−8.157	−8.904	0.440	0.628	95.1
		RIDE$^{(2)}$	0.334	0.887	−7.539	−8.054	−8.806	0.432	0.621	93.8
✗	random perturbation	RIDE$^{(1)}$	0.401	0.833	−7.668	−8.144	−8.794	0.460	0.623	91.7
		RIDE$^{(2)}$	0.354	0.845	−7.825	−8.275	−8.853	0.450	0.622	**97.2**
	value-guided sampling	RIDE$^{(1)}$	0.353	0.852	−7.497	−7.985	−8.757	0.426	**0.634**	95.9
		RIDE$^{(2)}$	0.341	0.884	−7.644	−8.220	**−9.053**	0.443	0.624	94.7

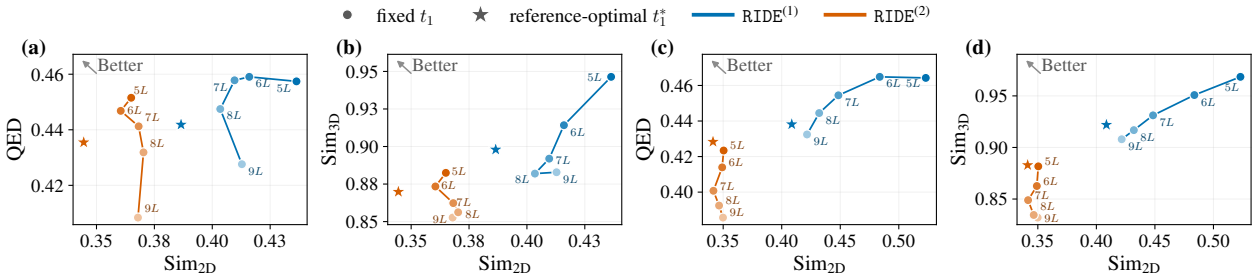


Figure 5: Comparison of Sim_{2D} vs. QED and Sim_{2D} vs. Sim_{3D} results across t_1 values ($\{nL\}_{n=5}^{9}$) and t_1^* with $L = T/10$, under reward function of Sim_{2D} and QED. **(a)-(b)**: conDitar-a, **(c)-(d)**: IPDiff-a.

objectives, increasing Sim_{3D} from 0.833 to 0.866 at t_1^* and from 0.852 to 0.882 after search, while also reducing Sim_{2D}. With RIDE$^{(2)}$, the improvement is more modest but remains consistent, with Sim_{3D} increasing from 0.845 to 0.854 at t_1^* and from 0.884 to 0.887 after search, together with a further improvement in Sim_{2D} after value-guided sampling.

Sensitivity to λ Figure 6 reports the results of two hops with λ varied from 0 to 1 in increments of 0.1. Across the tested values, increasing λ is associated with lower Sim_{2D} and a gradual decrease in Sim_{3D}. This trend is consistent with the role of λ in the reward, where larger values prioritize 2D dissimilarity over 3D similarity. The other molecular properties, including QED and SA, remain relatively stable across different λ values. Vina S shows moderate variation with better values generally observed at higher Sim_{3D}. Among the Pareto-optimal subset of the tested λ values, $\lambda = 0.5$ provides a favorable trade-off between low Sim_{2D} and high Sim_{3D}.

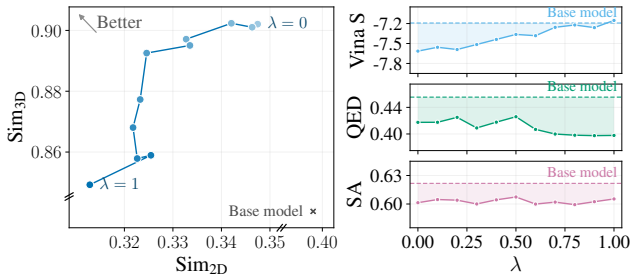


Figure 6: Sensitivity analysis of the reward weight λ in the second hop.