Confidence-Based Diffusion Sampling with Geometric Readiness Awareness for Accelerated Structure-based Drug Design

Pinzhen Chi, Kejing Yin*, and William K. Cheung
Department of Computer Science, Hong Kong Baptist University

Abstract

Structure-based drug design with diffusion models seeks to generate 3D molecules that bind tightly to target protein pockets, but full-step sampling is too slow for high-throughput use. Extending the recent development of training-free fast sampling methods for diffusion models (e.g., DDIM and DNDM) to high-quality 3D molecule generation is non-trivial, given that both discrete variables (atom types) and continuous variables (atom coordinates) are involved. In this paper, we propose a unified fast sampling framework that allows a shared time schedule across discrete and continuous variables for strategically accelerating the 3D ligand generation. Guided by an error-equalization principle, we combine the continuous and discrete error rates of hybrid sampling into a single step-importance density and derive a training-free coupling-aware schedule. We further propose a confidence-guided sampling strategy that integrates constraints of model prediction confidence, motion stability, and steric clash to control the acceleration based on geometric readiness. Extensive experiments demonstrate that our method achieves 18–20$\times$ speedup compared to full-step diffusion models while preserving state-of-the-art generation quality across chemical properties, binding affinity, and chemical validity. The project repository is available at `https://github.com/chipinzhen/Confidence-Based-Diffusion-Sampling-for-SBDD`.

1 Introduction

Structure-based drug design (SBDD) seeks to generate novel molecules with high binding affinity to specific protein targets. Driven by progress in generative AI, diffusion models have become the leading paradigm for SBDD [Lin et al., 2025, Guan et al., 2023a, Schneuing et al., 2024]. By framing generation as the reversal of a noising process, they effectively capture the complex multi-modal distributions over atom types and 3D coordinates, substantially surpassing prior auto-regressive [Peng et al., 2022] and normalizing flow-based methods [Liu et al., 2022].

However, the generation quality comes at a steep computational price. Unlike non-iterative generative models, diffusion models typically need hundreds to thousands of denoising steps to yield a valid structure. This inefficiency is a major bottleneck for real-world drug discovery, which relies on high-throughput virtual screening across vast chemical spaces to find viable hits. While recent progress in molecular property prediction has sped up the *evaluation* stage [Tripathi et al., 2024, Pei et al., 2023], the *generation* stage remains rate-limiting. Accelerating the sampling of diffusion-based SBDD models without sacrificing quality is therefore a critical priority.

A well-established route to accelerating diffusion sampling reformulates the Markovian forward process as a non-Markovian one, allowing the reverse process to skip intermediate steps while preserving distributional equivalence via variational inference. DDIM [Song et al., 2021] applies this to continuous variables by sampling along timestep subsequences using the same DDPM-trained

*Correspondence to: Kejing Yin <cskjyin@comp.hkbu.edu.hk>.

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

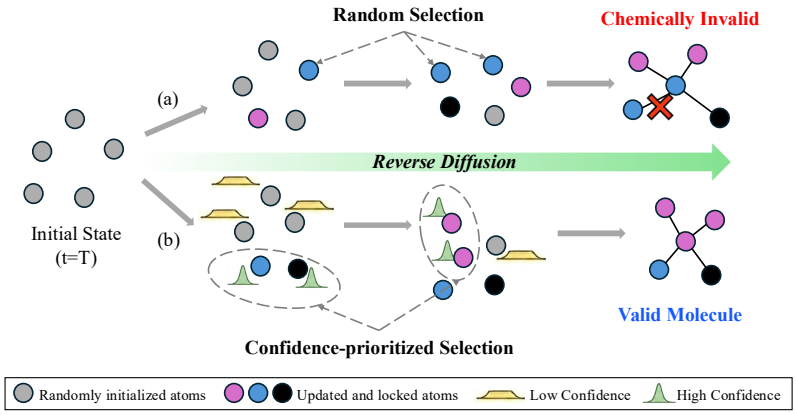


Figure 1: **Advantages of our proposed confidence-guided acceleration over naive SBDD acceleration methods.** (a) **Naive:** Lacking guidance, the standard random schedule updates atoms indiscriminately—even under uncertain predictions or geometric violations (e.g., steric clashes)—resulting in **failed generation** (invalid molecules). (b) **Ours (Confidence-Guided):** Leveraging model confidence and geometric constraints, we filter risky updates, prioritizing "confident" atoms and deferring low-confidence ones, so each transition purposefully improves the structure.

noise predictor [Ho et al., 2020], while DNDM [Chen et al., 2024] accelerates discrete sampling by giving each variable a *predetermined transition time*, sampled once and reused thereafter.

Yet, applying these acceleration methods to SBDD is fundamentally challenging due to the hybrid nature of SBDD, where molecules consist of coupled *continuous* variables (3D coordinates) and *discrete* variables (atom types). The key challenges include:

Challenge 1: Schedule mismatch between discrete and continuous variables in SBDD. DDIM uses a unified, predetermined time sequence shared across all continuous variables (typically fixed intervals), whereas DNDM uses *independent random sampling*, where each discrete variable draws its own transition time. Designing a principled, unified schedule shared by both discrete and continuous variables for accelerating SBDD sampling while preserving the aggregate noise schedule of the original diffusion process is non-trivial. Also, the two modalities have distinct error characteristics that interact through the network's joint prediction, requiring a schedule design that can consider the *coupling* effect of two types of variables.

Challenge 2: Premature transition of atom types. In original DNDM, each discrete variable independently samples its transition time from a predetermined distribution, and commits to a specific value at a randomly determined time. While effective for purely discrete domains (like text generation), this approach does not consider the atoms' attributes, like their geometric state, which is crucial and unique for SBDD. In early stages of generation, when molecular geometry is still highly unstructured and noisy, transition to specific atom types often results in physically invalid structures (e.g., steric clashes or valency violations). Methods to determine the transition time with mutual dependencies between geometry and chemistry, considered in a coordinated way, are still lacking.

To address the challenges, we propose a unified acceleration framework specifically tailored for SBDD, with the following main contributions:

- We propose a **unified fast sampling framework** with a shared time schedule across discrete and continuous variables. We propose conditional transition probabilities that synchronize discrete and continuous updates at the shared intervals, paired with a **coupling-aware schedule** motivated by an error-equalization heuristic.
- We propose a **confidence-guided sampling strategy** that combines model prediction confidence, motion stability, and steric information as readiness signals for selecting atom-type transition times.
- Across three SBDD backbones, experiments demonstrate **18–20$\times$ speedup** over full-step sampling while **retaining competitive generation quality**, with metric-dependent trade-offs in chemical properties, binding affinity, and chemical validity.

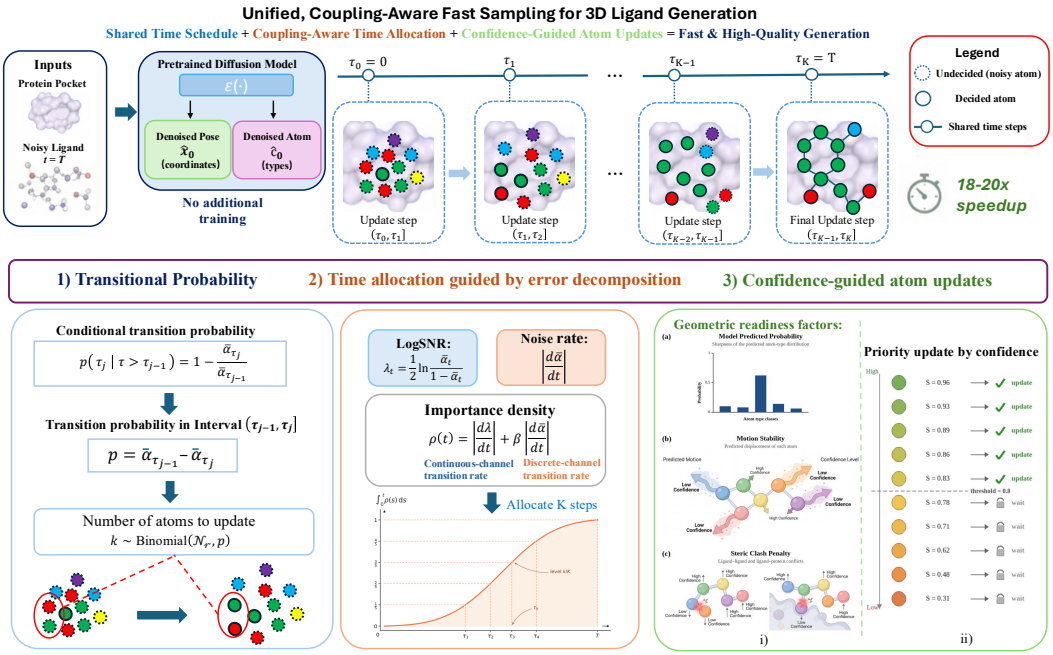


Figure 2: **Overview of the proposed unified fast sampling framework.** Given a protein pocket and a noisy ligand, the pretrained denoiser is queried on $K \ll T$ shared time steps $\{\tau_j\}$ through three components — a **conditional transition probability** synchronizing discrete and continuous updates, a **coupling-aware time allocation** motivated by joint error analysis, and a **confidence-guided atom update** driven by geometric readiness — yielding 18–20× speedup while retaining competitive generation quality on the tested backbones.

2 Related Work

2.1 Diffusion-based Generative Models for Structure-Based Drug Design

Diffusion models have become a leading paradigm for structure-based drug design (SBDD), with representative methods including DiffBP [Lin et al., 2025], TargetDiff [Guan et al., 2023a], and DiffSBDD [Schneuing et al., 2024]. These methods generate ligands conditioned on protein pockets by corrupting and denoising both continuous 3D coordinates and discrete atom types via a forward-reverse diffusion process. Follow-up work injects domain structure: D3FG [Lin et al., 2023] diffuses over curated rigid functional groups; DecompDiff [Guan et al., 2023b] splits generation into scaffold and linker design; IPDiff [Huang et al., 2024] embeds protein–ligand interaction priors into the transition dynamics; and ALIDiff [Gu et al., 2024] aligns generation with ligand preferences. Alternative paradigms such as Bayesian flow networks [Graves et al., 2023] like MolCRAFT [Qu et al., 2024]—lie outside our scope; we target training-free acceleration of the dominant DDPM/D3PM-based diffusion family in SBDD.

2.2 Fast Training-Free Diffusion Sampling Methods

Training-free sampling acceleration is well developed for continuous diffusion models, where reverse-time dynamics admit an ODE view. DDIM [Song et al., 2021] recasts sampling as a non-Markovian trajectory while preserving the DDPM training objective, enabling substantially fewer sampling steps. This perspective connects naturally to the probability flow ODE [Song et al., 2021], making sampling amenable to numerical integration. Building on this, the DPM-Solver family (DPM-Solver [Lu et al., 2022], DPM-Solver++ [Lu et al., 2025], and DPM-Solver-v3 [Zheng et al., 2023]) leverages the semi-linear structure of diffusion ODEs to achieve high-order integration with $\sim$10–20 function evaluations while maintaining competitive sample quality [Lu et al., 2022, Zheng et al., 2023].

For discrete diffusion, acceleration takes a different form because discrete state spaces generally lack a direct probability-flow ODE counterpart. Existing methods exploit the discrete Markov structure via de-randomization (e.g., selecting transition times and skipping intermediate steps) or study continuous-time discrete diffusion (e.g., DNDM) [Chen et al., 2024] to identify effective non-Markovian trajectories with fewer evaluations. In a related direction, LLaDA-style confidence

remasking [Nie et al., 2025] revises previously committed tokens by reverting low-confidence predictions back to [MASK] for further denoising. Recently, Fast-dLLM [Wu et al., 2026] targets masked diffusion language models, improving efficiency via parallel decoding and KV caching.

Confidence guidance vs. confidence remasking. Although our atom-type confidence shares the term used by the remasking literature, they are mechanistically different: remasking unbinds already-committed tokens, whereas our method withholds commitment for uncertain atoms without ever revoking finalized types. Moreover, masked language diffusion involves no continuous geometric coupling, while SBDD requires geometric readiness signals (e.g., motion stability, steric configuration).

To the best of our knowledge, there is no prior work that explicitly addresses the schedule design for hybrid discrete–continuous diffusion in SBDD.

3 Preliminaries

3.1 Diffusion-Based Generative Models for SBDD

Given a protein structure $\mathcal{P}$, SBDD aims to generate ligand molecules $M = (\mathbf{x}, \mathbf{v})$, where $\mathbf{x} \in \mathbb{R}^{N\times 3}$ refers to atoms' spatial coordinates and $\mathbf{v} \in \{1, 2, \ldots, C\}^N$ denote atom types for its N atoms. Following TargetDiff [Guan et al., 2023a] and related works [Huang et al., 2024], diffusion-based SBDD models being considered consist of forward and reverse processes. For the sake of clarity, we present the standard diffusion formulation and omit method-specific components such as the prior-shifting mechanism in IPDiff. The forward process gradually adds noise over T steps:

$$q\left(M_t \mid M_{t-1}, \mathcal{P}\right) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}\right) \cdot \mathcal{C}\left(\mathbf{v}_t \mid (1-\beta_t)\,\mathbf{v}_{t-1} + \beta_t / C\right) \tag{1}$$

where β_t is the noise schedule, $\mathcal{N}$ denotes a Gaussian distribution, and $\mathcal{C}$ denotes a categorical distribution. Denoting $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, we have:

$$q\left(M_t \mid M_0, \mathcal{P}\right) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, \left(1-\bar{\alpha}_t\right)\mathbf{I}\right) \cdot \mathcal{C}\left(\mathbf{v}_t \mid \bar{\alpha}_t \mathbf{v}_0 + \left(1-\bar{\alpha}_t\right)/C\right) \tag{2}$$

The reverse process recovers the molecule via a neural network parameterized by θ:

$$p_\theta\left(M_{t-1} \mid M_t, \mathcal{P}\right) = \mathcal{N}\left(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta\left(\left[\mathbf{x}_t, \mathbf{v}_t\right], t, \mathcal{P}\right), \sigma_t^2 \mathbf{I}\right) \cdot \mathcal{C}\left(\mathbf{v}_{t-1} \mid \boldsymbol{v}_\theta\left(\left[\mathbf{x}_t, \mathbf{v}_t\right], t, \mathcal{P}\right)\right) \tag{3}$$

where $\boldsymbol{\mu}_\theta$ and $\boldsymbol{v}_\theta$ predict distributions for coordinates and atom types, respectively. During sampling, the process starts from the prior distributions: $\mathbf{x}_T$ is sampled from a standard Gaussian prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$, and $\mathbf{v}_T$ is sampled from a uniform categorical prior. The model is trained by minimizing the variational lower bound. The training loss consists of two components: coordinate loss and atom type loss. For coordinates, the loss is typically simplified to predicting the original coordinates:

$$\mathcal{L}_x = \mathbb{E}_{t, M_0, \mathcal{P}}\left[\left\|\mathbf{x}_0 - \hat{\mathbf{x}}_0\right\|^2\right] \tag{4}$$

where $\hat{\mathbf{x}}_0$ is predicted by the network. For atom types, the loss is the cross-entropy between the true and predicted categorical distributions:

$$\mathcal{L}_v = \mathbb{E}_{t, M_0, \mathcal{P}}\left[\text{CrossEntropy}(\mathbf{v}_0, \hat{\mathbf{v}}_0)\right]. \tag{5}$$

The overall training loss is $\mathcal{L} = \mathcal{L}_x + \lambda \mathcal{L}_v$, where λ is a weighting factor.

3.2 Accelerated Sampling Methods

Fast samplers accelerate diffusion by skipping intermediate steps through variational inference for non-Markovian forward processes. For continuous variables, DDIM [Song et al., 2021] enables large timestep jumps while reusing the DDPM-trained noise predictor. The sampling process is:

$$\mathbf{x}_{\tau_{i-1}}(\eta) = \sqrt{\bar{\alpha}_{\tau_{i-1}}}\left(\frac{\mathbf{x}_{\tau_i} - \sqrt{1-\bar{\alpha}_{\tau_i}}\boldsymbol{\epsilon}_\theta\left(\mathbf{x}_{\tau_i}, \tau_i\right)}{\sqrt{\bar{\alpha}_{\tau_i}}}\right) + \sqrt{1-\bar{\alpha}_{\tau_{i-1}} - \sigma_{\tau_i}(\eta)^2} \cdot \epsilon_\theta\left(\mathbf{x}_{\tau_i}, \tau_i\right) + \sigma_{\tau_i}(\eta)\boldsymbol{\epsilon} \tag{6}$$

where

$$\sigma_{\tau_i}(\eta) = \eta \sqrt{\frac{1-\bar{\alpha}_{\tau_{i-1}}}{1-\bar{\alpha}_{\tau_i}}} \sqrt{1 - \frac{\bar{\alpha}_{\tau_i}}{\bar{\alpha}_{\tau_{i-1}}}}. \tag{7}$$

Here, $\boldsymbol{\epsilon}_\theta$ is derived from $\boldsymbol{\mu}_\theta$, η controls the deterministic degree of the sampling process, $\boldsymbol{\tau} = \{\tau_1, \tau_2, \ldots, \tau_n\}$ is a subset of sampling timesteps with $\tau_1 < \tau_2 < \cdots < \tau_n = T$, and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is random noise.

For discrete variables, fast sampling methods leverage the equivalence between the Markovian diffusion process (with multiple noise injections) and a non-Markovian process (with a single marginal noise injection), enabling accelerated sampling through variational inference. DNDM [Chen et al., 2024] accelerates discrete sampling by introducing a non-Markovian construction with a *predetermined transition time* for each discrete variable. Specifically, each variable n is associated with a transition time τ_n (sampled once) and is *only updated* when the reverse process reaches $t = \tau_n$; otherwise, its value is only copied across timesteps. The sampling process is thus defined as:

$$\mathbf{v}_{t-1,n} = \mathbf{1}(\tau_n = t)\,\mathbf{v}_{0,n} + \mathbf{1}(\tau_n \neq t)\,\mathbf{v}_{t,n}, \quad \forall n \in [N] \tag{8}$$

where $\mathbf{1}(\cdot)$ is the indicator function, and the transition time τ_n follows $\mathbb{P}(\tau_n = t) = \bar{\alpha}_{t-1} - \bar{\alpha}_t$.

4 Methodology

In this section, we introduce the details of proposed unified acceleration method that can be applied to existing diffusion-based SBDD models. In addition, we also propose a confidence-based sampling strategy to generate molecules with higher quality.

4.1 Unified Fast Sampling via Conditional Transition Probabilities

To address the limitations of independent time sampling in DNDM [Chen et al., 2024], we propose a unified sampling strategy that facilitates efficient batch processing while strictly adhering to the original noise schedule. Instead of sampling a transition time τ_n for each variable independently, we start with a time subsequence $\boldsymbol{\tau} = \{\tau_0, \tau_1, \ldots, \tau_K\}$ with $\tau_0 = 0$ and $\tau_K = T$, shared across all variables in a batch.

For each discrete variable, we perform a sequential sampling process over $\boldsymbol{\tau}$. At each step τ_j (given that the variable has not transitioned at any $\tau_k < \tau_j$), we determine the transition event based on the transition probability derived from the noise schedule. We formalize this process and its distributional properties in the following proposition:

Proposition 1. *The marginal probability that the variable has **not** transitioned by time τ_k is exactly $\bar{\alpha}_{\tau_k}$, and the marginal probability of transitioning within the interval $(\tau_{j-1}, \tau_j]$ is $\bar{\alpha}_{\tau_{j-1}} - \bar{\alpha}_{\tau_j}$ if:*

$$p(\tau_j | \tau > \tau_{j-1}) = 1 - \frac{\bar{\alpha}_{\tau_j}}{\bar{\alpha}_{\tau_{j-1}}}. \tag{9}$$

Based on Proposition 1, with the transition probability applied independently to each discrete variable, it can be proved that the cumulative probability of any single variable remaining at the noisy state matches the marginal of the DNDM process, and thus the original D3PM [Austin et al., 2021].[2] Confidence-guided selection (Sec. 4.3) reorders and filters transitions using readiness scores; the independent-transition result does not establish distributional equivalence or an error bound for this guided sampler.

Unified Sampling Algorithm We propose a unified generation framework to synchronize the denoising of both continuous (coordinates) and discrete (atom types) variables. For the coordinates $\mathbf{x}$, we employ the standard DDIM update rule [Song et al., 2021] for TargetDiff/DecompDiff; for IPDiff we use a generalized DDIM that accounts for the prior shift, derived in Appendix E. For the atom types $\mathbf{v}$, according to Proposition 1, we calculate the transition probability mass for the current interval $(\tau_{j-1}, \tau_j]$ as $p_{\text{trans}} = \bar{\alpha}_{\tau_{j-1}} - \bar{\alpha}_{\tau_j}$. Then, let N_{rem} be the set of atoms that are still in the noisy state at the beginning of the interval. The number of atoms k that will transition in this interval is sampled from a binomial distribution given by: $k \sim \text{Binomial}(N_{rem}, p_{\text{trans}})$.

4.2 Coupling-Aware Time Schedule

Sec. 4.1 leaves the placement of the K steps along t unspecified. We use an *error-equalization heuristic*: the step density is set proportional to an analytic importance rate $r(t)$, aiming to distribute estimated error across steps. For hybrid continuous and discrete sampling, we combine two rate surrogates into a single importance density $\rho(t)$.

(i) Continuous error rate. When denoising continuous variables (3D coordinates), how rapidly the noise predictor $\boldsymbol{\epsilon}_\theta$ varies between adjacent timesteps will cause different truncation errors. As the rate of change of $\boldsymbol{\epsilon}_\theta$ is proportional to its smoothness in the logSNR coordinate (as shown in [Lu et al., 2022]), it can be estimated by computing $|d\lambda/dt|$ where

$$\lambda_t \;=\; \tfrac{1}{2} \ln \tfrac{\bar{\alpha}_t}{1-\bar{\alpha}_t}. \tag{10}$$

[2] A formal proof can be found in Appendix C.

Equalizing this rate across steps allocates equal $\Delta\lambda$ per step, giving the *logSNR-uniform* schedule, which allocates steps according to the continuous rate alone.

(ii) Discrete error rate. For discrete variables (atom types), the denoising is based on the transition probability $p_{\text{trans}} = \bar{\alpha}_{t-h} - \bar{\alpha}_t \approx |d\bar{\alpha}/dt| \cdot h$, where h is the time interval between two adjacent timesteps. The rate $|d\bar{\alpha}/dt|$ measures transition activity and serves as a surrogate for the discrete contribution to scheduling. Again, equalizing this rate allocates equal $\Delta\bar{\alpha}$ per step, giving the $\bar{\alpha}$-*uniform* schedule, which allocates steps according to transition activity alone.

Importance density. We combine the two rate surrogates into a practical importance density:

$$\rho(t) \;=\; \left|\frac{d\lambda}{dt}\right| + \beta \cdot \left|\frac{d\bar{\alpha}}{dt}\right|, \qquad \beta \geq 0. \tag{11}$$

As the two terms take different scales, where $|d\lambda/dt|$ diverges at the endpoints of $[0, T]$ while $|d\bar{\alpha}/dt|$ is bounded on $[0, 1]$, we set the value of β empirically. By construction, $\beta = 0$ recovers the logSNR-uniform schedule and $\beta \to \infty$ recovers the $\bar{\alpha}$-uniform schedule, with intermediate values placing finer steps in regions where atom-type transitions concentrate relative to the continuous-error rate. We select β per backbone by a Pareto-coverage grid search on a held-out validation set; the chosen value is reported in Sec. 5.4 and the full sweep is provided in Appendix I.4.

Schedule from ρ. Applying the error-equalization principle to the joint rate $\rho(t)$, we allocate $K+1$ grid points so that each adjacent pair spans an equal share of the total ρ-mass:

$$\int_0^{\tau_k} \rho(s)\, ds \;=\; \frac{k}{K} \int_0^{T} \rho(s)\, ds, \qquad k = 0, 1, \ldots, K. \tag{12}$$

This places more grid points where $\rho(t)$ is large (high scheduling importance) and fewer where it is small; by construction $\tau_0 = 0$ and $\tau_K = T$, so the resulting $\boldsymbol{\tau}$ is a strictly increasing subsequence and the independent-transition statement of Proposition 1 continues to hold on this grid. The practical density is motivated by the heuristic analysis in Appendix B.

4.3 Confidence-Based Sampling for Atoms

As discussed in Sec. 4.1, the denoising time for updating the atom type of each atom is determined independently according to the diffusion schedule. Yet, we notice that not all atoms are equally ready for updating during the denoising process. And as they are connected as molecular structures, premature updates of their atom types may result in generation of invalid molecules or disconnected fragments. To mitigate this issue, we introduce several confidence factors to define a confidence-based prioritized sampling strategy for guiding the sampling process adaptively.

Model Confidence For discrete diffusion models (Eq. 5), at each timestep t the model predicts a distribution over $\hat{\mathbf{v}}_0$. Its sharpness (e.g., max probability or entropy) naturally reflects the model's confidence in the denoised atom-type prediction. We denote the maximum probability for atom i at time t as $c_{\text{prob}}^{(i,t)}$, and adopt it as one of our confidence factors.

Predicted Motion Stability We use the magnitude of predicted residual motion as a proxy for geometric readiness, prioritizing atoms with smaller values for atom-type commitment.

We estimate the residual-motion magnitude $\|\Delta\mathbf{x}^{(i,t)}\|_2$ from the predicted noise $\boldsymbol{\epsilon}_\theta^{(i,t)}$ as:

$$\|\Delta\mathbf{x}^{(i,t)}\|_2 = \sqrt{1-\bar{\alpha}_t} \cdot \|\boldsymbol{\epsilon}_\theta(\mathbf{x}_t^{(i)}, t)\|_2 \tag{13}$$

where $\bar{\alpha}_t$ denotes the cumulative product of the noise schedule. The motion stability confidence is then defined to decay exponentially with the displacement:

$$c_{\text{motion}}^{(i,t)} = \exp\left(-\frac{\|\Delta\mathbf{x}^{(i,t)}\|_2}{\tau_{\text{motion}}}\right) \tag{14}$$

where τ_{motion} is a temperature hyperparameter that controls the sensitivity to atomic movement. We fix τ_{motion}=1.0 throughout, and the performance remains stable over $\tau_{\text{motion}} \in [0.5, 4.0]$ (Appendix I).

Steric Clash Penalty To promote the validity of the generated ligands, we incorporate physical constraints as a penalty term to modulate the confidence scores. Specifically, we delay the determination of atom types for atoms exhibiting spatial conflicts. This soft penalty targets atoms violating steric

Table 1: Performance comparison across three SBDD backbones (TargetDiff, IPDiff, DecompDiff) at 50 sampling steps. **Bold** rows are our **Confidence-Guided Sampling (CGS)** variants; all experiments use our coupling-aware schedule; the shared confidence-threshold mechanism is omitted from row labels for brevity. "Full (1000 steps)" is each backbone's full-sampling reference. Among accelerated methods, green marks the best and light green the second-best.

Model	Acceleration	QED ↑	SA ↑	Vina Score ↓		Vina Min ↓		Vina Dock ↓		Div. ↑	Rec. % ↑	Comp. % ↑	Speed
				Avg.	Med.	Avg.	Med.	Avg.	Med.				
TargetDiff	Full (1000 steps)	0.48	0.58	-5.47	-6.30	-6.64	-6.83	-7.80	-7.91	0.72	98.97	92.20	1×
TargetDiff	Vanilla	0.45	0.57	-5.17	-5.84	-6.07	-6.20	-7.27	-7.49	0.75	78.65	72.81	19.7×
TargetDiff	**Prob**	0.49	0.62	-5.30	-6.10	-6.41	-6.51	-7.60	-7.69	0.74	96.27	87.36	19.4×
TargetDiff	**Prob+Motion**	0.49	0.64	-5.25	-6.12	-6.45	-6.55	-7.61	-7.63	0.74	97.38	88.11	19.2×
TargetDiff	**Prob+Clash**	0.49	0.63	-5.32	-6.08	-6.46	-6.53	-7.59	-7.70	0.74	97.13	88.16	19.1×
TargetDiff	**Prob+Motion+Clash**	0.49	0.64	-5.18	-6.09	-6.41	-6.54	-7.62	-7.73	0.74	97.35	87.95	18.9×
IPDiff	Full (1000 steps)	0.52	0.61	-6.42	-7.01	-7.45	-7.48	-8.57	-8.51	0.74	98.25	93.80	1×
IPDiff	Vanilla	0.48	0.56	-6.02	-6.84	-6.88	-6.90	-7.86	-7.86	0.76	92.55	85.72	19.7 ×
IPDiff	**Prob**	0.49	0.57	-6.30	-7.07	-7.23	-7.23	-8.11	-8.05	0.75	96.97	90.53	19.2×
IPDiff	**Prob+Motion**	0.49	0.61	-6.11	-6.96	-7.19	-7.22	-8.02	-7.97	0.77	97.75	89.28	19.0×
IPDiff	**Prob+Clash**	0.50	0.59	-6.36	-7.13	-7.27	-7.25	-8.19	-8.18	0.75	97.45	90.80	19.1×
IPDiff	**Prob+Motion+Clash**	0.49	0.62	-6.01	-7.01	-7.12	-7.18	-8.04	-8.01	0.77	98.35	89.32	19.0×
DecompDiff	Full (1000 steps)	0.49	0.64	-6.11	-6.16	-6.68	-6.58	-7.53	-7.59	0.71	99.60	93.59	1×
DecompDiff	Vanilla	0.46	0.61	-5.27	-5.26	-5.87	-5.80	-7.02	-6.99	0.73	82.34	78.41	19.8×
DecompDiff	**Prob**	0.48	0.59	-6.25	-6.24	-6.56	-6.41	-7.48	-7.42	0.73	89.80	86.78	19.5 ×
DecompDiff	**Prob+Motion**	0.49	0.62	-6.43	-6.49	-6.75	-6.68	-7.63	-7.64	0.72	94.73	91.47	19.2×
DecompDiff	**Prob+Clash**	0.48	0.62	-6.28	-6.33	-6.60	-6.53	-7.46	-7.51	0.72	91.55	88.58	19.0×
DecompDiff	**Prob+Motion+Clash**	0.50	0.62	-6.45	-6.46	-6.77	-6.68	-7.56	-7.59	0.73	92.12	92.80	18.9×

constraints, such as overlapping with the protein pocket or colliding with other ligand atoms. We first define the minimum distances for atom i to the protein surface and other ligand atoms as:

$$d_{LP}^{(i)} = \min_{j \in \mathcal{P}} \|\mathbf{x}_i - \mathbf{x}_j\|_2, \quad d_{LL}^{(i)} = \min_{k \in \mathcal{L} \setminus \{i\}} \|\mathbf{x}_i - \mathbf{x}_k\|_2 \tag{15}$$

where $\mathcal{P}$ and $\mathcal{L}$ denote the sets of protein and ligand atoms, respectively. Based on these distances, we compute the steric clash discount factor $w_{\text{clash}}^{(i)}$:

$$w_{\text{clash}}^{(i)} = \min\left(\sigma\big(\lambda(d_{LL}^{(i)} - \delta_{LL})\big),\ \sigma\big(\lambda(d_{LP}^{(i)} - \delta_{LP})\big)\right). \tag{16}$$

Here, δ_{LL} and δ_{LP} are distance thresholds (defaults: 1.0 Å and 2.5 Å), and λ is the sigmoid sharpness (100 throughout); all three are robust across the ranges tested in Appendix I. The sigmoid $\sigma(\cdot)$ smooths the transition, reducing the priority of atoms with close steric contacts.

Summary of the Sampling Algorithm: Given a unified time schedule $\boldsymbol{\tau}$ produced by the coupling-aware procedure (Sec. 4.2), at each step j we first estimate the denoised ligand pose $\mathbf{x}_0$ and atom type probabilities $\hat{\mathbf{p}}_0$. We then compute the composite confidence score $S^{(i)}$ for each atom by aggregating model confidence and motion stability via the geometric mean, further modulated by the steric clash penalty (Sec. 4.3). The number of atoms to update is determined by sampling from a binomial distribution governed by the transition probability p_{trans} (Sec. 4.1). We prioritize updating atoms with the highest confidence scores $S^{(i)}$, provided they exceed the threshold γ (we use γ=0.8 throughout; sensitivity over $\gamma \in [0.6, 0.9]$ is reported in Appendix I). Once an atom's type is determined, it remains fixed for the rest of the generation process, while its position continues to be refined via DDIM. This rule prioritizes atoms with high composite readiness scores for type commitment. Algorithm 1 (referred to Appendix A) summarizes the complete generation process.

5 Experiments

We evaluate our proposed fast sampling method across several state-of-the-art SBDD diffusion models. We measure the speedup of the proposed acceleration framework and evaluate the chemical properties of the generated molecules to assess to what extent the performance can be retained with reference to the full-timestep sampling process.

5.1 Experimental Setup and Baselines

To demonstrate the versatility of our approach, we employ three state-of-the-art SBDD diffusion models as backbones: **TargetDiff** [Guan et al., 2023a], a 3D equivariant model that uses DDPM [Ho et al., 2020] and D3PM [Austin et al., 2021] to model atom coordinates and atom types respectively;

Table 2: Comparison with flow-matching baselines at 50 sampling steps. MolFORM is evaluated without RL and PAFlow without guidance. CGS uses Prob+Motion+Clash for TargetDiff and DecompDiff, and Prob+Clash for IPDiff. CGS entries reproduce the mean results from Table 1. Vina values are in kcal/mol; Comp. is reported as a proportion rounded to two decimal places for all methods. Best values among the evaluated configurations are in **bold**.

Model	Vina Score ↓	Vina Min ↓	Vina Dock ↓	QED ↑	SA ↑	Comp. ↑	Div. ↑
MolFORM (no RL)	−5.20	−6.03	−7.08	0.41	0.60	0.80	**0.76**
PAFlow (no guidance)	−5.16	−6.99	**−8.37**	**0.52**	0.57	0.92	0.70
TargetDiff + CGS	−5.18	−6.41	−7.62	0.49	**0.64**	0.88	0.74
IPDiff + CGS	−6.36	**−7.27**	−8.19	0.50	0.59	0.91	0.75
DecompDiff + CGS	**−6.45**	−6.77	−7.56	0.50	0.62	**0.93**	0.73

IPDiff [Huang et al., 2024], which incorporates protein–ligand interaction priors into both the forward and reverse diffusion trajectories; and **DecompDiff** [Guan et al., 2023b], which decomposes the generation process into scaffold and linker design. Beyond the original sampling methods of these backbones, we further include a **Vanilla Integration** baseline that applies our framework without confidence-guided atom prioritization.

5.2 Evaluation Protocols and Metrics

For a fair comparison, we adopt the same experimental setup as TargetDiff and IPDiff, sampling ligand sizes from the ligand-pocket dataset distribution. Further details are provided in Appendix G. We evaluate performance along four dimensions: 1) **Chemical Properties**, namely Drug-likeness (QED) and Synthetic Accessibility (SA); 2) **Binding Affinity**, reported via Vina Score, Vina Min, and Vina Dock, all computed with AutoDock Vina [Eberhardt et al., 2021, Trott and Olson, 2010]; 3) **Geometric Fidelity**, captured by Reconstruction Rate (Rec.) and Completeness (Comp.); and 4) **Molecular Diversity** (Div.), where, following TargetDiff [Guan et al., 2023a], we report the mean pairwise Tanimoto dissimilarity ($1 - \overline{\text{Tanimoto}}$) of Morgan fingerprints among ligands generated for each pocket, averaged over all test pockets (the higher the better). A molecule is considered valid only if it satisfies all valency constraints.

5.3 Experiment Results

Vanilla integration fails to preserve generation quality. Without confidence-guided updates, the unified sampler degrades all three backbones relative to the full-step reference (Table 1). Chemical descriptors remain close (QED drops by 0.03–0.04, SA by at most 0.05), but geometric validity and binding affinity degrade noticeably on every backbone: Reconstruction and Completeness drop by 15–20 pp on TargetDiff and DecompDiff (and $\sim$6–8 pp on IPDiff), while Vina Dock worsens uniformly by 0.5–0.7 kcal/mol. This consistent pattern indicates that hybrid discrete–continuous diffusion cannot be trivially accelerated by step-skipping alone.

Motion and Clash factors are complementary, with a backbone-specific caveat. The two physically-motivated factors address distinct failure modes of accelerated sampling. **Motion Stability** prioritizes atom-type commitment for atoms with smaller predicted residual motion and improves Rec./SA in several tested configurations (e.g., adding Motion to Prob lifts DecompDiff Rec. from 89.8 to 94.7). **Steric Clash** downweights atoms with local steric conflicts when selecting types to commit; its effect on Vina Dock varies across backbones. Their combination, **Prob+Motion+Clash**, gives the most balanced profile on TargetDiff and DecompDiff, where the two factors complement each other. IPDiff is the exception: adding Motion still improves geometry (Rec. $96.97 \rightarrow 97.75$, SA $0.57 \rightarrow 0.61$) but *slightly hurts binding* (Vina Dock $-8.11 \rightarrow -8.02$, Vina Score $-6.30 \rightarrow -6.11$). One possible explanation is that IPDiff's interaction-prior-shifted trajectory affects the usefulness of Motion as a readiness signal. This explanation has not been established causally. In the tested IPDiff configuration, Prob+Clash achieves the best mean Vina Dock among accelerated variants; **Prob+Clash** is thus recommended on IPDiff. We also include the results of per-pocket wall-clock times in Appendix H. At the distribution level, both vanilla integration and CGS stay within JS ≤ 0.01 of the full-step DDPM on all three backbones (Appendix F.1), indicating small differences in aggregate atom-pair distance distributions.

Direct validation of Motion and Clash and final-sample geometry are reported in Appendices F.2–F.4; additional results across sampling budgets are in Appendix J.

Table 3: Equal-time-budget evaluation on PDB 2Z3H chain A and PDB 4AAW chain A. Under the same wall-clock budget, the 1000-step baseline samples 100 molecules per pocket while our **CGS** (Prob+Clash variant under our coupling-aware schedule from Table 1) can sample 1,800. Each cell reports *baseline* $\rightarrow$ *CGS*. Top-50 Vina scores in kcal/mol; PB-Valid and Strain energy $<$50 kcal/mol are sample counts. Full Top-10/20/50 breakdown with multipliers can be found in Appendix L.

Backbone	Pocket	Top-50 Vina Min ↓	Top-50 Vina Score ↓	PB-Valid	Strain<50
TargetDiff	2Z3H_A	$-8.18 \rightarrow \mathbf{-10.60}$	$-7.48 \rightarrow \mathbf{-9.95}$	$32 \rightarrow \mathbf{776}$	$1 \rightarrow \mathbf{12}$
	4AAW_A	$-7.55 \rightarrow \mathbf{-10.05}$	$-7.11 \rightarrow \mathbf{-9.67}$	$64 \rightarrow \mathbf{1110}$	$4 \rightarrow \mathbf{50}$
IPDiff	2Z3H_A	$-8.94 \rightarrow \mathbf{-11.60}$	$-8.94 \rightarrow \mathbf{-11.39}$	$8 \rightarrow \mathbf{246}$	$0 \rightarrow 0$
	4AAW_A	$-8.90 \rightarrow \mathbf{-10.85}$	$-8.69 \rightarrow \mathbf{-10.54}$	$20 \rightarrow \mathbf{391}$	$1 \rightarrow \mathbf{8}$
DecompDiff	2Z3H_A	$-8.47 \rightarrow \mathbf{-11.70}$	$-7.70 \rightarrow \mathbf{-11.49}$	$31 \rightarrow \mathbf{700}$	$0 \rightarrow 0$
	4AAW_A	$-7.30 \rightarrow \mathbf{-10.39}$	$-6.63 \rightarrow \mathbf{-10.01}$	$41 \rightarrow \mathbf{485}$	$0 \rightarrow 0$

Table 4: Effect of the sampling schedule under a fixed acceleration backbone with probability as confidence at 50 steps. "Uniform" is the default $\tau_j = j \cdot T/K$; "logSNR" is the β=0 limit of Eq. (11); "Ours" uses the per-model β selected on a held-out pocket set (Appendix I.4). For DecompDiff the selected β=0 coincides with logSNR. Best per row group in **bold**.

Model	Schedule	QED ↑	SA ↑	Vina Score ↓	Vina Min ↓	Vina Dock ↓	Rec. % ↑	Comp. % ↑
TargetDiff	Uniform	0.48	**0.62**	-6.08	-6.49	-7.63	95.41	85.98
	logSNR (β=0)	**0.49**	**0.62**	**-6.14**	**-6.52**	-7.62	96.09	86.78
	Ours (β=0.25)	**0.49**	**0.62**	-6.10	-6.51	**-7.69**	**96.27**	**87.36**
IPDiff	Uniform	**0.50**	**0.57**	-6.81	-6.96	-7.81	95.35	**90.60**
	logSNR (β=0)	**0.50**	**0.57**	-7.05	**-7.23**	**-8.10**	96.88	90.48
	Ours (β=0.5)	**0.50**	**0.57**	**-7.07**	**-7.23**	-8.05	**96.97**	90.53
DecompDiff	Uniform	0.45	**0.59**	-5.92	-6.14	-7.31	81.40	77.35
	Ours (β=0, $\equiv$ logSNR)	**0.48**	**0.59**	**-6.24**	**-6.41**	**-7.48**	**89.80**	**86.78**

Accelerated diffusion models remain competitive. We compare with MolFORM [Huang and Zhang, 2025] without RL and PAFlow [Zhou et al., 2025] without guidance, using 50 sampling steps for all methods (Table 2). The results show metric-dependent trade-offs: IPDiff+CGS achieves the best Vina Min, DecompDiff+CGS the best Vina Score and Completeness, and TargetDiff+CGS the best SA, while PAFlow leads on Vina Dock and QED. This comparison uses a matched sampling-step budget; wall-clock costs may differ across architectures.

Equal-Time-Budget Evaluation. Under a fixed wall-clock budget, CGS (**Prob+Clash** variant from Table 1) samples 1800 molecules per pocket against the 1000-step baseline's 100, on PDB 2Z3H chain A and PDB 4AAW chain A. Across all 6 cells (Table 3), Top-50 Vina-Min improves by 1.95–3.23 kcal/mol; PB-Valid yield scales 12–30$\times$, exceeding the 18$\times$ sample multiplier. Per-method breakdown is in Appendix L.

5.4 Coupling-Aware Schedule Ablation

We isolate the contribution of the coupling-aware schedule (Sec. 4.2) by comparing three variants under a fixed acceleration backbone with *probability-only* confidence at 50 steps, so that the schedule remains the sole variable: (i) the default **Uniform** schedule ($\tau_j = j \cdot T/K$); (ii) the **logSNR**-uniform schedule (the β=0 limit of Eq. (11)); and (iii) our coupling-aware schedule **Ours**, with β chosen per model on a held-out set of 30 randomly sampled training pockets, by Pareto-front frequency across five quality metrics (full protocol in Appendix I.4). The selected values are β=0.25 (TargetDiff), β=0.5 (IPDiff), and β=0 (DecompDiff)—where β=0 reduces **Ours** exactly to logSNR, showing the parameterization adds no deviation when the baseline schedule is already near-optimal.

Across all three backbones, *Ours* matches or surpasses both alternatives on most metrics. On TargetDiff, *Ours* (β=0.25) improves Vina Dock from -7.63 (Uniform) and -7.62 (logSNR) to -7.69, and boosts geometric fidelity (Reconstruction $95.41 \rightarrow 96.27$, Completeness $85.98 \rightarrow 87.36$) while preserving QED/SA. On IPDiff, the gain over Uniform is most pronounced on docking-related metrics (Vina Score $-6.81 \rightarrow -7.07$, Vina Min $-6.96 \rightarrow -7.23$, Vina Dock $-7.81 \rightarrow -8.05$), and *Ours* (β=0.5) marginally exceeds logSNR on Vina Score and Reconstruction. On DecompDiff, the per-model selection picks β=0, reducing to logSNR by construction; this variant alone closes the largest gap to Uniform in the table—Reconstruction $81.40 \rightarrow 89.80$ and Completeness $77.35 \rightarrow 86.78$—confirming that schedule shape, not the specific β value, is the dominant driver.

6 Limitations and Broader Impacts

Limitations. We evaluate our method on three SBDD base models (TargetDiff, IPDiff, DecompDiff). Extending it to fragment-based generators such as D3FG [Lin et al., 2023] would require lifting the readiness criterion from atoms to fragments, while accelerating recent strong models built on different generative paradigms, e.g., the Bayesian-flow-network-based MolCRAFT [Qu et al., 2024], would require adapting the schedule to their non-DDPM forward process. We leave both directions for future work. Also, our method uses the predicted class probability as a proxy for confidence, which is only meaningful when the underlying model produces well-calibrated probabilities. If the network is consistently over-confident on frequent atom types (typically carbon), the delay rule fires too rarely; a post-hoc calibration step, such as temperature scaling, should mitigate this.

Higher-order continuous solvers such as DPM-Solver++ [Lu et al., 2025] and discrete continuous-time Markov chain solvers [Ren et al., 2025] are promising extensions. Applying them to hybrid SBDD requires coupling coordinate and atom-type updates under a shared schedule and incorporating CGS's confidence-based commitment rule. For continuous-time discrete solvers, a compatible transition-rate formulation is also needed for the pretrained discrete-time backbones. These considerations motivate coupling-aware higher-order sampling as future work.

Our current sampler keeps atom types fixed after commitment. Limited confidence-triggered remasking is a promising complement to CGS: readiness guidance controls when to commit a type, while remasking could revise that decision if confidence later drops. The denoiser's subsequent predictions could guide these bounded revisions.

Broader Impacts. Our method cuts the sampling cost of diffusion-based SBDD by roughly an order of magnitude without any retraining, putting large-scale virtual screening within reach of smaller research groups. Faster generative chemistry raises the usual misuse concern. But our work only changes how efficiently the underlying model samples, not what it can sample, so the incremental risk over the base model is small; the toxicity and synthesizability filters standard in SBDD pipelines apply unchanged.

7 Conclusion

This paper presents a unified training-free fast-sampling framework for structure-based drug design that couples DDIM and DNDM under a shared time schedule. Motivated by a heuristic error analysis, we construct a coupling-aware schedule from continuous and discrete rate surrogates without additional training. We complement this with confidence-guided sampling that defers low-confidence atoms via model probability, motion stability, and steric clash. Across three SBDD backbones, the resulting sampler achieves 18–$20\times$ speedup while retaining competitive generation quality, with metric-dependent trade-offs relative to full-step sampling. Coupling-aware higher-order solvers and limited confidence-triggered remasking offer promising, complementary directions for extending this framework.

Acknowledgments and Disclosure of Funding

We thank the authors of TargetDiff, IPDiff, and DecompDiff for making their code publicly available. We also thank the broader diffusion-model research community for advancing generative modeling through openly shared ideas, methods, and software.

The authors declare that they have no competing interests related to this work.

References

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pages 17981–17993, 2021.

Martin Buttenschoen, Garrett M. Morris, and Charlotte M. Deane. PoseBusters: AI-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical Science*, 15(9):3130–3139, 2024. doi: 10.1039/D3SC04185A.

Zixiang Chen, Huizhuo Yuan, Yongqian Li, Yiwen Kou, Junkai Zhang, and Quanquan Gu. Fast sampling via discrete Non-Markov diffusion models with predetermined transition time. In *Advances in Neural Information Processing Systems*, volume 37, pages 106870–106905, 2024. doi: 10.52202/079017-3393.

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, 2021.

Paul Francoeur, Tomohide Masuda, Jocelyn Sunseri, Andrew Jia, Richard B Iovanisci, Ian Snyder, and David Ryan Koes. 3D convolutional neural networks and a crossdocked dataset for structure-based drug design. *Journal of Chemical Information and Modeling*, 2020.

Alex Graves, Rupesh Kumar Srivastava, Timothy Atkinson, and Faustino Gomez. Bayesian flow networks. *arXiv preprint arXiv:2308.07037*, 2023.

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. In *Advances in Neural Information Processing Systems*, volume 37, pages 44040–44063, 2024. doi: 10.52202/079017-1398.

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. In *Proceedings of International Conference on Learning Representations*, 2023a.

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. In *Proceedings of International Conference on Machine Learning*, 2023b.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.

Jie Huang and Daiheng Zhang. MolFORM: Multi-modal flow matching for structure-based drug design. In *ICML 2025 Generative AI and Biology (GenBio) Workshop*, 2025.

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 *Proceedings of International Conference on Learning Representations*, 2024.

Haitao Lin, Yufei Huang, Odin Zhang, Yunfan Liu, Lirong Wu, Siyuan Li, Zhiyuan Chen, and Stan Z Li. Functional-group-based diffusion for pocket-specific molecule generation and elaboration. In *Advances in Neural Information Processing Systems*, volume 36, pages 34603–34626, 2023. doi: 10.52202/075280-1504.

Haitao Lin, Yufei Huang, Odin Zhang, Siqi Ma, Meng Liu, Xuanjing Li, Lirong Wu, Jishui Wang, Tingjun Hou, and Stan Z Li. DiffBP: Generative diffusion of 3D molecules for target protein binding. *Chemical Science*, 16(3):1417–1431, 2025.

Meng Liu, Youzhi Luo, Kanji Uchino, Koji Maruhashi, and Shuiwang Ji. Generating 3D molecules for target protein binding. In *Proceedings of International Conference on Machine Learning*, 2022.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems*, volume 35, pages 5775–5787, 2022. doi: 10.52202/068431-0418.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22 (4):730–751, 2025. doi: 10.1007/s11633-025-1562-4.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *Advances in Neural Information Processing Systems*, volume 38, pages 56354–56392, 2025. doi: 10.52202/085713-1689.

Qizhi Pei, Kaiyuan Gao, Lijun Wu, Jinhua Zhu, Yingce Xia, Shufang Xie, Tao Qin, Kun He, Tie-Yan Liu, and Rui Yan. FABind: Fast and accurate protein-ligand binding. In *Advances in Neural Information Processing Systems*, volume 36, pages 55963–55980, 2023. doi: 10.52202/075280-2440.

Xingang Peng, Shitong Luo, Jiaqi Guan, Qi Xie, Jian Peng, and Jianzhu Ma. Pocket2mol: Efficient molecular sampling based on 3D protein pockets. In *Proceedings of International Conference on Machine Learning*, 2022.

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741, 2023. doi: 10.52202/075280-2338.

Yinuo Ren, Haoxuan Chen, Yuchen Zhu, Wei Guo, Yongxin Chen, Grant Rotskoff, Molei Tao, and Lexing Ying. Fast solvers for discrete diffusion models: Theory and applications of high-order algorithms. In *Advances in Neural Information Processing Systems*, volume 38, pages 167228–167282, 2025. doi: 10.52202/085713-5575.

Arne Schneuing, Charles Harris, Yuanqi Du, Kieran Didi, Arian Jamasb, Ilia Igashov, Weitao Du, Carla Gomes, Tom L Blundell, Pietro Lio, Max Welling, Michael Bronstein, and Bruno Correia. Structure-based drug design with equivariant diffusion models. *Nature Computational Science*, 4 (12):899–909, Dec 2024. ISSN 2662-8457.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *Proceedings of International Conference on Learning Representations*, 2021.

Gaurav Tripathi, Amit Agarwal, Rohit Singh Yadav, Sakshi Tyagi, and Amit Kumar. High throughput ai-driven drug discovery pipeline. https://developer.nvidia.com/blog/high-throughput-ai-driven-drug-discovery-pipeline/, Oct 2024. NVIDIA Developer Blog.

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

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8228–8238, 2024.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Fast-dLLM: Training-free acceleration of diffusion LLM by enabling KV cache and parallel decoding. In *Proceedings of International Conference on Learning Representations*, 2026.

Kaiwen Zheng, Cheng Lu, Jianfei Chen, and Jun Zhu. DPM-Solver-v3: Improved diffusion ODE solver with empirical model statistics. In *Advances in Neural Information Processing Systems*, volume 36, pages 55502–55542, 2023. doi: 10.52202/075280-2423.

Jingyuan Zhou, Hao Qian, Shikui Tu, and Lei Xu. Prior-guided flow matching for target-aware molecule design with learnable atom number. In *Advances in Neural Information Processing Systems*, volume 38, pages 121513–121543, 2025. doi: 10.52202/085713-4052.

Appendix

A Confidence-Guided DNDM Sampling Algorithm

The complete generation procedure summarized in Sec. 4.3 and Sec. 4.2 is detailed in Algorithm 1 below. It instantiates the unified fast sampling framework (Sec. 4.1) together with the coupling-aware schedule of Sec. 4.2 and the composite confidence score $S^{(i)}$ used throughout the analysis in Appendix B. Note that the confidence factor motion and clash factor is optional. If they are not chosen as factors, we treat these values as 1.0.

Algorithm 1 Confidence-Guided DNDM Sampling

Input: Pocket $\mathcal{P}$, Total Steps K, Confidence Threshold γ, Noise Schedule $\bar{\alpha}$, Coupling weight β
Output: Ligand pose $\mathbf{x}_0$, atom types $\mathbf{v}_0$
Generate schedule $\boldsymbol{\tau} = \{\tau_0, \ldots, \tau_N\}$ via coupling-aware procedure (Sec. 4.2, Eq. 11)
Initialize $\mathbf{x}_{\tau_N} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{v}_{\text{init}} \sim \mathcal{U}(K)$
Initialize state $\mathbf{v} \leftarrow \mathbf{v}_{\text{init}}$, mask $\mathbf{m} \leftarrow \mathbf{0}$
Initialize prev. prob $\mathbf{p}_{\text{prev}} \leftarrow \text{Uniform}(K)$
for $j = N$ **to** 1 **do**
  $\mathbf{v}_{\text{in}} \leftarrow \mathbf{v}$
  Predict $\hat{\mathbf{x}}_0, \hat{\mathbf{p}}_0, \boldsymbol{\epsilon} \leftarrow \Phi_\theta(\mathbf{x}_{\tau_j}, \mathbf{v}_{\text{in}}, \mathcal{P}, \tau_j)$
  $\mathbf{x}_{\tau_{j-1}} \leftarrow \text{DDIM}(\mathbf{x}_{\tau_j}, \hat{\mathbf{x}}_0, \boldsymbol{\epsilon}, \tau_j, \tau_{j-1})$
  // Compute Multiple Confidence for each atom i
  Compute confidence factors $c^{(i)}_{\text{prob}}$, $c^{(i)}_{\text{motion}}$
  $S^{(i)} \leftarrow \text{GeoMean}(c^{(i)}_{\text{prob}}, c^{(i)}_{\text{motion}})$
  // Physics-Guided Refinement
  Compute steric clash factor $w^{(i)}_{\text{clash}}$
  $S^{(i)} \leftarrow S^{(i)} \cdot w^{(i)}_{\text{clash}}$
  // Transition Selection
  $N_{\text{rem}} \leftarrow \sum_i (1 - m^{(i)})$
  $p_{\text{trans}} \leftarrow \bar{\alpha}_{\tau_{j-1}} - \bar{\alpha}_{\tau_j}$
  $k \sim \text{Binomial}(N_{\text{rem}}, p_{\text{trans}})$
  $\mathcal{I}_{\text{cand}} \leftarrow \text{TopK}(\{S^{(i)} \cdot (1 - m^{(i)})\}_i, k)$
  // Threshold Filtering & Update
  if $j = 1$ **then**
    $\mathcal{I}_{\text{upd}} \leftarrow \{i \mid m^{(i)} = 0\}$ *// Force update for remaining atoms*
  else
    $\mathcal{I}_{\text{upd}} \leftarrow \{i \in \mathcal{I}_{\text{cand}} \mid S^{(i)} \geq \gamma\}$
  end if
  $\mathbf{v}[\mathcal{I}_{\text{upd}}] \leftarrow \text{GumbelArgmax}(\hat{\mathbf{p}}_0[\mathcal{I}_{\text{upd}}])$
  $\mathbf{m}[\mathcal{I}_{\text{upd}}] \leftarrow 1$ *// Recording transitioned atoms*
  $\mathbf{p}_{\text{prev}} \leftarrow \hat{\mathbf{p}}_0$
  $\mathbf{x}_{\tau_j} \leftarrow \mathbf{x}_{\tau_{j-1}}$
end for
$\mathbf{v}_0 \leftarrow \mathbf{v}$
Return $\mathbf{x}_0, \mathbf{v}_0$

B Heuristic Error Analysis for the Coupling-Aware Schedule (Corollary 1)

This section provides a heuristic first-order analysis that motivates the coupling-aware schedule of Sec. 4.2. The continuous-error component (drawn from DPM-Solver [Lu et al., 2022]) and the Lagrangian optimization (Corollary 1) are formal; the coupling-error component is a first-order Taylor expansion under bounded perturbation of the TopK ranking, and the additive form of the practical importance density $\rho(t)$ in Eq. (11) is a design choice that approximates the optimum of Eq. (23) rather than a strict consequence of it. We make these caveats explicit at each step.

B.1 Assumptions

(A1) Smoothness of the noise predictor along each solver subinterval. Within any solver step $[\tau_{j-1}, \tau_j]$ over which the atom-type configuration $\mathbf{v}$ is held fixed by operator splitting (Sec. 4.1), the noise prediction network $\boldsymbol{\epsilon}_\theta(\mathbf{x}_\lambda, \lambda; \mathbf{v})$, evaluated along the probability-flow ODE trajectory of $\mathbf{x}$ at fixed $\mathbf{v}$, has bounded $(p+1)$-th derivative in λ: there exists $M_{p+1} < \infty$ such that $\|d^{p+1}\boldsymbol{\epsilon}_\theta/d\lambda^{p+1}\| \leq M_{p+1}$. Locally on the j-th interval $[\lambda_{\tau_j}, \lambda_{\tau_{j-1}}]$ we denote this bound by L_j. This is the standard DPM-Solver smoothness assumption [Lu et al., 2022] applied conditionally on $\mathbf{v}$; the impact of $\mathbf{v}$-jumps at step boundaries $\{\tau_j\}$ is captured separately by the coupling-error term in Theorem B.2.

(A2) Lipschitz dependence of confidence on coordinates. We assume the composite confidence score $S^{(i)}$ used in Algorithm 1 is Lipschitz in the coordinate state $\mathbf{x}$ on the working domain visited along the sampling trajectory, with effective constant $L_c < \infty$. This is a standard working assumption analogous to (A1); only the existence of a finite L_c enters the first-order coupling analysis below.

B.2 Hybrid Sampling Error Decomposition Analysis

Indexing convention and informal error semantics. Throughout this appendix we use the indexing $\tau_0 = 0$ (clean end) and $\tau_K = T$ (noisy end), with sampling proceeding from $j = K$ down to $j = 1$; the j-th step takes the state from τ_j to τ_{j-1}, so $h_j = \lambda_{\tau_{j-1}} - \lambda_{\tau_j} > 0$ is the per-step logSNR increment. The quantity E_{total} refers to the end-to-end accelerated-sampling error of the hybrid sampler in an informal sense, decomposed into a continuous-side component E_{cont} (state error in the coordinate ODE) and a discrete-side component E_{coup} (expected impact of TopK selection perturbations on atom-type assignments). The additive split $E_{\text{total}} = E_{\text{cont}} + E_{\text{coup}}$ is a notational convention for combining these heterogeneous error contributions; the relative weighting between them is absorbed into the constants C_1, C_2. Local truncation errors are aggregated to global state errors under the standard Lipschitz stability of the probability-flow ODE in λ [Lu et al., 2022], with the stability factor absorbed into C_p.

Hybrid sampling error decomposition. Let $\lambda_t = \frac{1}{2}\ln\frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}$ denote the logSNR, $h_j = \lambda_{\tau_{j-1}} - \lambda_{\tau_j}$ the step size in logSNR space, L_j a smoothness constant bounding the $(p+1)$-th derivative of $\boldsymbol{\epsilon}_\theta$ in λ on $[\tau_j, \tau_{j-1}]$, and n_j the (expected) number of atoms transitioned at step j. Under Assumptions (A1)–(A2), the total accelerated-sampling error of a p-th order solver admits the heuristic upper bound

$$E_{\text{total}} \;\lesssim\; C_p \sum_{j=1}^{K} W_j \cdot L_j \cdot h_j^{p+1}, \quad W_j = C_1 + C_2 \cdot n_j, \tag{17}$$

where C_p, C_1, C_2 are constants independent of $\boldsymbol{\tau}$. The first term captures continuous solver truncation; the second captures the first-order coupling effect of transitioning n_j atoms on imperfect coordinates. We use $\lesssim$ to emphasize that the coupling-error contribution is a Taylor approximation, not a strict worst-case inequality.

Proof. The total error of the accelerated K-step sampling decomposes into a continuous solver error E_{cont} and a coupling error E_{coup} induced by performing atom-type transitions on imperfect coordinate states.

Continuous error. By DPM-Solver Theorem 1 [Lu et al., 2022], under (A1) the local truncation error of a p-th order solver on the j-th interval is bounded by

$$\delta_j \;\leq\; C_p \cdot L_j \cdot h_j^{p+1}, \tag{18}$$

where C_p depends only on the solver. Summing over all steps,

$$E_{\text{cont}} \;\leq\; C_p \sum_{j=1}^{K} L_j \cdot h_j^{p+1}. \tag{19}$$

Coupling error (first-order analysis). On the j-th interval, the solver delivers $\mathbf{x}_{\tau_{j-1}}$ with error of order δ_j. To first order in δ_j, this perturbs the composite confidence score $S^{(i)}$ for each atom by an amount of order $L_c \cdot \delta_j$ (by A2). The TopK transition operator is discrete, but treating each affected

ranking decision as receiving an additive perturbation of this magnitude and aggregating across the n_j atoms transitioning at step j, the resulting deviation from the noise-free transition outcome is on the order of

$$E_{\text{coup}}^{(j)} \lesssim L_c \cdot n_j \cdot \delta_j \lesssim L_c \cdot C_p \cdot n_j \cdot L_j \cdot h_j^{p+1}. \tag{20}$$

Caveat. This is a first-order, expectation-level approximation rather than a worst-case bound; a rigorous treatment of the discrete TopK sensitivity would require additional assumptions on the score gap distribution between selected and non-selected atoms. We therefore use $\lesssim$ rather than $\leq$ throughout the coupling-error analysis. Summing across steps,

$$E_{\text{coup}} \lesssim L_c \cdot C_p \sum_{j=1}^{K} n_j \cdot L_j \cdot h_j^{p+1}. \tag{21}$$

Combination. Letting $C_1 = 1$, $C_2 = L_c$, and $W_j = C_1 + C_2 \cdot n_j$,

$$E_{\text{total}} = E_{\text{cont}} + E_{\text{coup}} \lesssim C_p \sum_{j=1}^{K} W_j \cdot L_j \cdot h_j^{p+1}, \tag{22}$$

which gives the heuristic bound stated in analysis B.2. □

B.3 Step Allocation Minimizing the Heuristic Bound (Corollary 1)

Given the heuristic bound of analysis B.2, we treat its right-hand side as the objective to minimize over the choice of step sizes $\{h_j\}$. The resulting optimization is convex and admits a closed-form solution.

Corollary 1 (Coupling-Aware Step Allocation under the Heuristic Bound)**.** *The step sizes minimizing* $\sum_{j=1}^{K} W_j L_j h_j^{p+1}$ *subject to* $\sum_{j=1}^{K} h_j = H$ *(the total logSNR range* $H = \lambda_{\max} - \lambda_{\min}$*) satisfy*

$$h_j^* \propto (W_j \cdot L_j)^{-1/p}. \tag{23}$$

When n_j is constant across steps, this reduces to $h_j^* \propto L_j^{-1/p}$, recovering the logSNR-uniform schedule of pure continuous diffusion [Lu et al., 2022] under additionally uniform L_j*; when L_j is constant, it reduces to $h_j^* \propto n_j^{-1/p}$, allocating finer steps where transitions concentrate.*

Proof. The Lagrangian construction below is fully rigorous; only the underlying objective from analysis B.2 is heuristic. Minimize $\sum_{j=1}^{K} W_j L_j h_j^{p+1}$ subject to $\sum_{j=1}^{K} h_j = H$ via the Lagrangian

$$\mathcal{L}(\{h_j\}, \mu) = \sum_{j=1}^{K} W_j L_j h_j^{p+1} - \mu \left(\sum_{j=1}^{K} h_j - H \right). \tag{24}$$

The first-order condition $\partial \mathcal{L} / \partial h_j = 0$ gives $(p+1) W_j L_j h_j^p = \mu$, hence

$$h_j^* = \left(\frac{\mu}{(p+1) W_j L_j} \right)^{1/p} \propto (W_j L_j)^{-1/p}. \tag{25}$$

The Hessian $\partial^2 \mathcal{L} / \partial h_j^2 = p(p+1) W_j L_j h_j^{p-1} > 0$ confirms strict convexity, so this critical point is the global minimum. The two limiting cases stated above (n_j constant $\Rightarrow$ logSNR-uniform under additionally uniform L_j; L_j constant $\Rightarrow$ a schedule concentrating steps in dense-transition regions, collapsing to $\bar{\alpha}$-uniform when $p = 1$) follow by direct substitution. □

B.4 Practical Surrogates and the Design of $\rho(t)$

The step allocation derived from the heuristic bound (Corollary 1) depends on the smoothness constant L_j and the per-step transition count n_j, neither of which is directly computable. We replace them by analytic surrogates derivable from the noise schedule alone:

- $L_j \approx |d\lambda / dt|$: the noise predictor $\boldsymbol{\epsilon}_\theta$ is approximately smooth in the log-SNR coordinate [Lu et al., 2022], so its variation rate in t is proportional to the rate of logSNR change.

- $n_j \approx |d\bar{\alpha}/dt|$: under DNDM, the per-step transition probability is $p_{\text{trans}}^{(j)} = \bar{\alpha}_{\tau_{j-1}} - \bar{\alpha}_{\tau_j} \approx |d\bar{\alpha}/dt| \cdot h_j$, so the expected number of transitions per unit time is proportional to $|d\bar{\alpha}/dt|$.

Both surrogates are obtained from the noise schedule $\bar{\alpha}_t$; no model evaluation or profiling experiment is required.

Why an additive $\rho(t)$. The bound in analysis B.2 couples L_j and n_j multiplicatively through $W_j L_j$, so a strict surrogate substitution would yield an importance density in which $|d\lambda/dt|$ and $|d\bar{\alpha}/dt|$ multiply rather than add. We instead use the additive density of Eq. (11) for three reasons:

(a) Robustness near schedule boundaries. As $t \to 0$ or $t \to T$, the logSNR λ_t saturates, so $|d\lambda/dt|$ approaches zero while $|d\bar{\alpha}/dt|$ can remain non-trivial—precisely where atom-type transitions tend to concentrate in our hybrid setting. A multiplicative surrogate would assign near-zero step density to such regions regardless of the discrete transition rate, leaving high-coupling boundary regions under-resolved. The additive form correctly elevates step density wherever *either* error source is large, independent of the magnitude of the other.

(b) Limiting cases recovered exactly. Setting $\beta = 0$ in Eq. (11) reproduces the logSNR-uniform schedule of pure continuous diffusion, and $\beta \to \infty$ reproduces the $\bar{\alpha}$-uniform schedule. The corresponding limits of a multiplicative surrogate (large C_2/C_1) instead yield a density proportional to $|d\lambda/dt| \cdot |d\bar{\alpha}/dt|$, which coincides with neither limiting schedule.

(c) Decoupled tuning and interpretation. Each term in the additive form retains an independent physical meaning—continuous-error rate vs. transition-rate—and the relative weight is captured by a single scalar β rather than by nested products. In our experiments we select β by a small grid search; we empirically verify that the resulting schedule is effective in Sec. 5.

We therefore treat the multiplicative coupling in analysis B.2 as a heuristic upper bound that motivates including *both* error sources in the schedule, and the additive $\rho(t)$ of Eq. (11) as the engineered density that practical step allocation actually uses.

C Proof of Per-Variable Marginal Equivalence (Proposition 1)

In this section, we formally prove Proposition 1: under the conditional transition rule of Eq. (9), applied *independently* to each discrete variable, the per-variable marginal probability of remaining at the noisy state at any τ_k matches the original DNDM (and hence D3PM) noise schedule.

Scope of the proof. The argument below establishes *per-variable* marginal preservation under the assumption of independent transitions across variables. The Vanilla Integration baseline (Sec. 5) instantiates this independence directly. Confidence-guided selection (Sec. 4.3) breaks per-variable independence by reordering transitions according to the composite confidence score $S^{(i)}$; in that regime, individual variables' marginals are reweighted by their confidence trajectory and no longer coincide with $\bar{\alpha}_{\tau_k}$. For a variant that commits exactly the sampled k atoms, the binomial draw $k \sim \text{Binomial}(N_{\text{rem}}, p_{\text{trans}})$ determines the expected transition count through $\mathbb{E}[k \mid N_{\text{rem}}] = N_{\text{rem}}\, p_{\text{trans}}$. This count identity does not establish per-variable equivalence; threshold filtering can also reduce the actual number of committed atoms.

C.1 Proof

Proof. Let E_j denote the event that the variable transitions at time step τ_j, and let $\bar{E}_j$ denote the event that it does not transition at τ_j. The event that a variable remains in the noise state (has not transitioned) up to step τ_k, denoted as S_k, is the intersection of non-transition events for all steps $i \leq k$:

$$S_k = \bigcap_{i=1}^{k} \bar{E}_i \tag{26}$$

Using the chain rule of probability, we can express the probability of S_k as:

$$\mathbb{P}(S_k) = \mathbb{P}(\bar{E}_1) \cdot \mathbb{P}(\bar{E}_2|\bar{E}_1) \cdot \ldots \cdot \mathbb{P}(\bar{E}_k|\bar{E}_{k-1}, \ldots, \bar{E}_1) \tag{27}$$

Note that our sampling process is Markovian in the sense that the decision at τ_i depends only on the state at τ_{i-1}. The term $\mathbb{P}(\bar{E}_i| \cap_{m=1}^{i-1} \bar{E}_m)$ corresponds to $1 - p(\tau_i|\tau > \tau_{i-1})$. Substituting the

definition from Eq. (9):

$$\mathbb{P}(\bar{E}_i|\text{survival up to } i-1) = 1 - \left(1 - \frac{\bar{\alpha}_{\tau_i}}{\bar{\alpha}_{\tau_{i-1}}}\right) = \frac{\bar{\alpha}_{\tau_i}}{\bar{\alpha}_{\tau_{i-1}}} \tag{28}$$

The cumulative survival probability $\mathbb{P}(S_k)$ thus forms a telescoping product:

$$\mathbb{P}(S_k) = \prod_{i=1}^{k} \frac{\bar{\alpha}_{\tau_i}}{\bar{\alpha}_{\tau_{i-1}}} = \frac{\bar{\alpha}_{\tau_1}}{\bar{\alpha}_{\tau_0}} \cdot \frac{\bar{\alpha}_{\tau_2}}{\bar{\alpha}_{\tau_1}} \cdot \ldots \cdot \frac{\bar{\alpha}_{\tau_k}}{\bar{\alpha}_{\tau_{k-1}}} \tag{29}$$

By canceling adjacent terms and noting that $\bar{\alpha}_{\tau_0} = \bar{\alpha}_0 = 1$, we obtain:

$$\mathbb{P}(S_k) = \frac{\bar{\alpha}_{\tau_k}}{\bar{\alpha}_{\tau_0}} = \bar{\alpha}_{\tau_k} \tag{30}$$

This confirms that the probability of the variable remaining in the noise state at time τ_k matches the noise schedule $\bar{\alpha}_{\tau_k}$ exactly.

Consequently, the marginal probability of transitioning exactly at step τ_j is the probability of surviving up to τ_{j-1} and then failing to survive at τ_j:

$$\begin{aligned} \mathbb{P}(\text{transition at } \tau_j) &= \mathbb{P}(S_{j-1}) \cdot \mathbb{P}(E_j|S_{j-1}) \\ &= \bar{\alpha}_{\tau_{j-1}} \cdot \left(1 - \frac{\bar{\alpha}_{\tau_j}}{\bar{\alpha}_{\tau_{j-1}}}\right) \\ &= \bar{\alpha}_{\tau_{j-1}} - \bar{\alpha}_{\tau_j} \end{aligned} \tag{31}$$

This matches the probability of each specific transition time, $P(\tau_n = t) = \alpha_{t-1} - \alpha_t$, in the original DNDM formulation. Thus, under the independent-transition assumption stated in the scope note above, the per-variable transition-time distribution of our unified sequential sampling matches the original DNDM process on the discretized time grid. □

C.2 Consistency with D3PM and Pre-trained Models

It is important to note that the original DNDM framework is designed to match the marginal distribution of the Discrete Denoising Diffusion Probabilistic Models (D3PM) [Austin et al., 2021]. Since we have proven (under the independent-transition assumption) that our unified sequential sampling strategy reproduces the per-variable marginal of DNDM, it follows transitively that, in the same regime, our method also reproduces the per-variable marginal of the underlying D3PM process. This per-variable equivalence does not extend to confidence-guided selection, which reorders and filters transitions as discussed in the scope note above.

Formally, let $q(v_t|v_0)$ denote the forward marginal distribution of the discrete variable v at time t given the clean data v_0. Our proof ensures that this forward marginal remains consistent. According to Bayes' rule, the posterior distribution of the clean data given the noisy state is:

$$q(v_0|v_t) = \frac{q(v_t|v_0)q(v_0)}{q(v_t)} \propto q(v_t|v_0)q(v_0) \tag{32}$$

Since the forward marginal $q(v_t|v_0)$ is preserved, the target posterior $q(v_0|v_t)$ induced by our non-Markovian process remains consistent with the original Markovian diffusion process.

This consistency has a crucial practical implication: **Training-free Compatibility**. Because the target posterior is unchanged, the neural network predictor $p_\theta(v_0|v_t)$, which was trained to approximate the denoising posterior of the original D3PM/DNDM process, can be directly employed in our unified sampling framework.

D Interaction Prior-Guided Diffusion (IPDIFF) Details

Standard diffusion models typically assume a forward process that gradually adds Gaussian noise to the data, converging to a standard normal distribution. IPDIFF [Huang et al., 2024], however, introduces a novel mechanism called *Prior-Shifting* to incorporate protein-ligand interaction information explicitly into the diffusion trajectory.

D.1 The Biased Forward Process

Unlike standard DDPMs where the mean of the forward marginal distribution solely decays towards zero, IPDIFF shifts the mean of the ligand atom coordinates X_t based on the interaction potential with the target protein pocket P. Formally, given the ground truth ligand X_0 and the interaction prior representation F_0 extracted by a pre-trained IPNet, the marginal distribution at time step t is defined as:

$$q(X_t|X_0, P, F_0) = \mathcal{N}\left(X_t; \sqrt{\bar{\alpha}_t}X_0 + S_t, (1-\bar{\alpha}_t)\mathbf{I}\right), \tag{33}$$

where $\bar{\alpha}_t$ follows the standard variance schedule, and $S_t \in \mathbb{R}^{N\times 3}$ represents the cumulative position shift derived from the interaction prior.

D.2 Interaction Prior Shift S_t

The shift term S_t is designed to adapt the diffusion trajectory towards binding-favorable regions. It is computed via a learnable adapter network $\psi_{\theta 2}$ conditioned on the interaction features F and the time step t:

$$S_t = \eta \cdot k_t \cdot \psi_{\theta 2}(F, t), \tag{34}$$

where η is a scaling factor, and $k_t = \sqrt{\bar{\alpha}_t}(1-\sqrt{\bar{\alpha}_t})$ is a time-dependent coefficient that ensures the shift vanishes at $t = 0$ (data) and $t = T$ (noise). This modified forward process implies that the reverse generative process must also account for this time-dependent bias, necessitating the generalized sampling derivation provided in Section E.

E Coordinate DDIM Sampling with Interaction-Guided Shift Terms

To accelerate the sampling process of IPDIFF, we derive a generalized non-Markovian update rule consistent with the interaction-guided diffusion process. Recall that the core of IPDIFF is the biased forward process defined by the protein-ligand interaction. According to the definition of IPDIFF, the marginal distribution of the ligand coordinates X_t at any time step t conditioned on X_0 is given by:

$$q(X_t|X_0, P) = \mathcal{N}(X_t; \sqrt{\bar{\alpha}_t}X_0 + S_t(X_0), (1-\bar{\alpha}_t)\mathbf{I}), \tag{35}$$

where $S_t(X_0)$ denotes the interaction prior shift at time t. Using the reparameterization trick, a sample X_{τ_i} at time step τ_i can be expressed as:

$$X_{\tau_i} = \sqrt{\bar{\alpha}_{\tau_i}}X_0 + S_{\tau_i}(X_0) + \sqrt{1-\bar{\alpha}_{\tau_i}} \cdot \epsilon_{\tau_i}, \tag{36}$$

where $\epsilon_{\tau_i} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the standard Gaussian noise.

E.1 Noise Estimation via Denoising

In the reverse generative process, the ground truth X_0 is unknown. We employ the denoising network ϕ_θ to predict the clean ligand structure $\hat{X}_0$ from the current state X_{τ_i}:

$$\hat{X}_0 = \phi_\theta(X_{\tau_i}, \tau_i, P). \tag{37}$$

Conditioned on this predicted $\hat{X}_0$, we can invert Eq. (36) to estimate the noise vector $\hat{\epsilon}_{\tau_i}$ that implicitly drove the process to the current state X_{τ_i}. Note that the shift term $S_{\tau_i}(\hat{X}_0)$ must be subtracted to isolate the noise:

$$\hat{\epsilon}_{\tau_i} = \frac{X_{\tau_i} - \sqrt{\bar{\alpha}_{\tau_i}}\hat{X}_0 - S_{\tau_i}(\hat{X}_0)}{\sqrt{1-\bar{\alpha}_{\tau_i}}}. \tag{38}$$

E.2 Reconstruction at Previous Step τ_{i-1}

We seek to generate a sample $X_{\tau_{i-1}}$ at the previous time step τ_{i-1} such that it is consistent with the marginal distribution defined in Eq. (35). Specifically, $X_{\tau_{i-1}}$ must satisfy:

$$X_{\tau_{i-1}} = \underbrace{\sqrt{\bar{\alpha}_{\tau_{i-1}}}\hat{X}_0}_{\text{Signal Scaling}} + \underbrace{S_{\tau_{i-1}}(\hat{X}_0)}_{\text{Interaction Prior Shift}} + \underbrace{\sqrt{1-\bar{\alpha}_{\tau_{i-1}}} \cdot \hat{\epsilon}_{\tau_{i-1}}}_{\text{Noise Component}}, \tag{39}$$

where $S_{\tau_{i-1}}(\hat{X}_0)$ is the explicit prior shift required by the IPDIFF forward process definition at time τ_{i-1}.

Following the DDIM assumption, the generative process is deterministic given X_{τ_i} and $\hat{X}_0$ (excluding the optional stochastic term σ_t). This implies that the "direction" of the noise remains constant across steps, i.e., $\hat{\epsilon}_{\tau_{i-1}} = \hat{\epsilon}_{\tau_i}$. For the generalized case with stochasticity σ_{τ_i}, we decompose the noise term in Eq. (39) into a deterministic direction pointing to X_{τ_i} and an independent random noise:

$$\sqrt{1-\bar{\alpha}_{\tau_{i-1}}} \cdot \hat{\epsilon}_{\tau_{i-1}} = \sqrt{1-\bar{\alpha}_{\tau_{i-1}}-\sigma_{\tau_i}^2} \cdot \hat{\epsilon}_{\tau_i} + \sigma_{\tau_i}\epsilon, \tag{40}$$

where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

E.3 Final Update Rule

Substituting the estimated noise $\hat{\epsilon}_{\tau_i}$ from Eq. (38) and the noise decomposition from Eq. (40) into the target distribution Eq. (39), we obtain the final update rule:

$$\begin{aligned} X_{\tau_{i-1}} &= \sqrt{\bar{\alpha}_{\tau_{i-1}}}\hat{X}_0 + S_{\tau_{i-1}}(\hat{X}_0) + \sqrt{1-\bar{\alpha}_{\tau_{i-1}}-\sigma_{\tau_i}^2} \cdot \hat{\epsilon}_{\tau_i} + \sigma_{\tau_i}\epsilon \\ &= \sqrt{\bar{\alpha}_{\tau_{i-1}}}\hat{X}_0 + S_{\tau_{i-1}}(\hat{X}_0) + \sqrt{1-\bar{\alpha}_{\tau_{i-1}}-\sigma_{\tau_i}^2} \cdot \left(\frac{X_{\tau_i} - \sqrt{\bar{\alpha}_{\tau_i}}\hat{X}_0 - S_{\tau_i}(\hat{X}_0)}{\sqrt{1-\bar{\alpha}_{\tau_i}}} \right) + \sigma_{\tau_i}\epsilon. \end{aligned} \tag{41}$$

Here, the term $S_{\tau_{i-1}}(\hat{X}_0)$ retains IPDiff's interaction-prior shift in the accelerated coordinate update; its effect on binding affinity is evaluated empirically.

F Geometric Quality and Readiness Analysis

F.1 Atom-Pair and Bond Distance Distributions

To assess the geometric fidelity of the generated molecules, we compare the distribution of pairwise distances over all atom pairs (bonded and non-bonded) between the reference data, the unaccelerated 1,000-step DDPM sampler, and our accelerated samplers. Figure 3 reports this comparison on three backbones — TargetDiff, IPDiff, and DecompDiff . For each method, the inset table reports two Jensen–Shannon divergences: between the method's atom-pair distance distribution and the reference data (column *JS vs Ref*), and between the method's distribution and the same-backbone 1,000-step DDPM sampler (column *JS vs DDPM*). The first column measures how close the method is to the empirical CrossDocked distribution; the second isolates the effect of the acceleration itself by treating the full-step DDPM as the unaccelerated anchor (so the DDPM row in that column is undefined and shown as "–"). Across all three backbones, both vanilla integration and our confidence-guided sampling sit within 0.01 JS of the full-step DDPM, indicating a small difference in this distribution under acceleration. The JS-vs-Ref values are also close to those of the full-step DDPM. These distribution-level comparisons complement the direct clash assessments in Appendices F.3 and F.4.

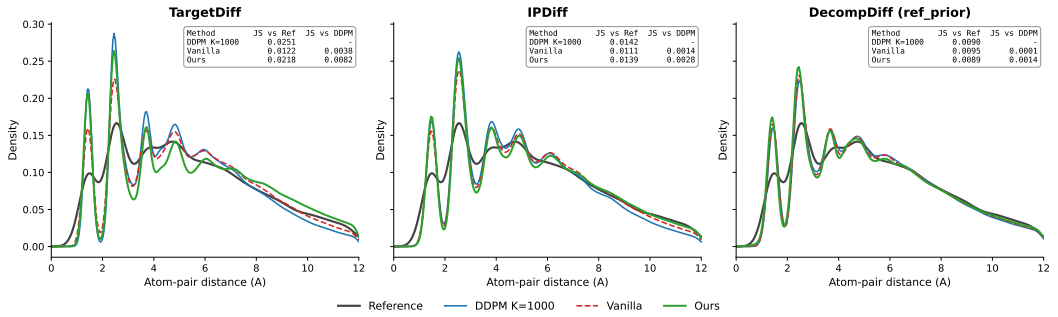


Figure 3: Atom-pair distance distributions on TargetDiff, IPDiff, and DecompDiff (ref_prior). Each panel overlays the CrossDocked reference distribution (gray), the unaccelerated 1,000-step DDPM sampler (blue), the vanilla integration baseline (red, dashed), and our confidence-guided sampling (green). For each method (one per row), the inset table reports two Jensen–Shannon divergences: between that method's distribution and the reference (column *JS vs Ref*), and between that method's distribution and the same-backbone 1,000-step DDPM (column *JS vs DDPM*); the DDPM row in the second column is undefined ("–"). Lower is better. Across all three backbones, the accelerated samplers (Vanilla and Ours) stay within 0.01 JS of the unaccelerated DDPM, indicating small differences in the aggregate atom-pair distance distributions.

To complement the all-atom-pair view above, we further provide a per-bond-type comparison of the bond distance distributions between the original **TargetDiff** and our proposed accelerated versions. Table 5 shows bond-dependent differences in Jensen–Shannon divergence (JSD): acceleration increases JSD for several bond types, whereas confidence-guided variants reduce it for aromatic C:C bonds. These comparisons characterize bond-length distributions and complement the clash analyses below.

Table 5: Jensen-Shannon divergence between the distributions of bond distance for **TargetDiff** vs. **Accelerated Versions**. Note that "-", "=", and ":" represent single, double, and aromatic bonds, respectively. Lower values are better.

Method	**C-C**	**C=C**	**C-N**	**C=N**	**C-O**	**C=O**	**C:C**	**C:N**
TargetDiff	0.369	0.505	0.363	0.550	0.421	0.461	0.263	0.235
Vanilla	0.413	0.509	0.373	0.557	0.425	0.463	0.266	0.277
Probability	0.412	0.514	0.373	0.568	0.445	0.460	0.250	0.296
Prob. & Motion	0.440	0.520	0.377	0.575	0.435	0.475	0.236	0.312
Prob. & Clash	0.436	0.524	0.371	0.575	0.436	0.463	0.243	0.295
Prob. & Motion & Clash	0.448	0.530	0.375	0.589	0.446	0.474	0.239	0.305

F.2 Direct Validation of Motion

Motion and Clash guide atom-type commitment; they do not directly modify the coordinate update rule, and coordinates continue to evolve after commitment. To test whether Motion selects atoms that are ready for commitment, we use two test pockets, 2Z3H-A and 4AAW-A, with 100 samples per pocket and matched random seeds across variants. For each sample, we first record the number of atoms committed by Prob-only at every reverse step, then require Prob+Motion to commit exactly the same number at each step. This comparison isolates the choice of atoms from the number committed.

Table 6 reports next-step displacement for four atom groups. Adding Motion reduces displacement among both newly committed and all currently committed atoms, while displacement over all ligand atoms remains nearly unchanged. These results support Motion as a signal for selecting more stable atoms for type commitment, rather than as a mechanism that globally suppresses coordinate motion.

Table 6: Next-step displacement (Å) with matched per-step commitment counts. Each cell reports Prob → Prob+Motion.

Atom group	2Z3H-A	4AAW-A
Newly committed	$0.0390 \rightarrow 0.0258$	$0.0355 \rightarrow 0.0252$
All currently committed	$0.0550 \rightarrow 0.0330$	$0.0471 \rightarrow 0.0322$
Uncommitted	$0.1326 \rightarrow 0.1344$	$0.1319 \rightarrow 0.1339$
All ligand atoms	$0.1324 \rightarrow 0.1328$	$0.1314 \rightarrow 0.1318$

F.3 Direct Validation of Clash

We compare Prob-only and Prob+Clash using the same two pockets, 100 samples per pocket, matched seeds, and matched per-step commitment counts. For newly committed atoms, a distance violation occurs when an atom lies within 1.0 Å of another ligand heavy atom or within 2.5 Å of a protein heavy atom. We check these atoms at commitment and at subsequent steps, and also use PoseBusters [Buttenschoen et al., 2024] to evaluate clashes in the next predicted clean structure $\hat{x}_0$.

As shown in Table 7, Clash reduces distance violations among newly committed atoms, with lower violation rates persisting two and five steps later. The zero current-step rates reflect the geometric criterion used for selection; the later-step results provide additional evidence that the selected atoms remain less clash-prone. This benefit is concentrated in the committed subset: when all ligand atoms, including uncommitted atoms, are evaluated at intermediate steps, the whole-ligand PoseBusters clash rate changes only from 0.564% to 0.554%.

Table 7: Clash checks for newly committed atoms with matched commitment counts. Entries are percentages and report Prob → Prob+Clash. LL and LP denote ligand–ligand and ligand–protein distance violations, respectively. The PoseBusters row reports the rate of at least one clash in the next $\hat{x}_0$ prediction.

Check	2Z3H-A	4AAW-A
Current-step distance violation	$31.91 \rightarrow 0.00$	$28.34 \rightarrow 0.00$
Next-prediction PoseBusters clash	$0.590 \rightarrow 0.143$	$0.00 \rightarrow 0.00$
LL violation, 2 steps later	$27.08 \rightarrow 0.08$	$24.49 \rightarrow 0.52$
LP violation, 2 steps later	$0.48 \rightarrow 0.08$	$0.10 \rightarrow 0.00$
LL violation, 5 steps later	$21.63 \rightarrow 0.80$	$18.54 \rightarrow 1.69$

F.4 Final-Sample Geometry

To assess whether these signals also benefit final samples, we compare TargetDiff variants with matched random seeds. We measure protein–ligand clash counts with PoseCheck [Harris et al., 2023] and intramolecular clash failures with PoseBusters [Buttenschoen et al., 2024]. Table 8 shows that Prob+Motion+Clash has the lowest values on both metrics among the evaluated configurations: relative to Prob-only, the mean protein–ligand clash count decreases from 9.154 to 8.836, and the intramolecular clash rate decreases from 8.91% to 5.84%. Together with the matched-commitment analyses, these results provide empirical support for the geometric readiness signals in the tested settings; they do not establish general optimality or robustness guarantees.

Table 8: Final-sample geometry on TargetDiff. Protein–ligand clashes are reported as mean PoseCheck counts; intramolecular clashes are reported as PoseBusters intramolecular-clash failure rates. Lower is better for both metrics.

Setting	Protein–ligand clashes ↓	Intramolecular clashes (%) ↓
Full TargetDiff (1,000 steps)	11.975	7.00
Prob	9.154	8.91
Prob+Clash	9.099	7.19
Prob+Motion	8.976	6.64
Prob+Motion+Clash	**8.836**	**5.84**

G Detailed Experiment settings

Dataset. In alignment with TargetDiff and IPDiff, we utilized the CrossDocked2020 dataset [Francoeur et al., 2020] for all experiments and strictly followed their data splitting protocols. Baselines and Configuration. For TargetDiff [Guan et al., 2023a], the reported metrics for the non-accelerated version were obtained directly from the original paper and its official pre-trained checkpoints. To ensure a fair comparison, we sampled 100 ligands for each target pocket with a sampling batch size of 50, strictly following the metric calculation methods described in the original work. For IPDiff [Huang et al., 2024] and DecompDiff [Guan et al., 2023b], we re-executed the full sampling process as a reference baseline using their official pre-trained checkpoints, also generating 100 ligands per pocket so that the evaluation budget matches TargetDiff. All accelerated variants likewise generate 100 ligands per pocket on each of the 100 test pockets, so that every cell of Table 1 aggregates 10,000 samples under a uniform protocol across the three backbones. Implementation Details. Throughout the experiments, we use $\gamma = 0.8$, $\delta_{LL} = 1.0\,\text{Å}$, $\delta_{LP} = 2.5\,\text{Å}$, $\lambda = 100$, and $\tau_{\text{motion}} = 1.0$ as default hyperparameters; the robustness of these choices is verified in Appendix I. All experiments were conducted on NVIDIA A100 GPUs.

H Wall-Clock Sampling Time

To complement the speedup factors reported in Table 1, we report absolute per-pocket sampling time for each backbone and acceleration strategy under our coupling-aware schedule. All measurements are taken on a single NVIDIA A100 GPU, averaged over 100 ligands per pocket. The full-step (1,000 steps) row is the per-backbone baseline; speedups are taken from Table 1 and the corresponding wall-clock time is computed as baseline/speedup.

Table 9: Per-pocket wall-clock sampling time across backbones and acceleration strategies, on a single NVIDIA A100 GPU. Speedups are taken from Table 1.

Backbone	**Acceleration**	**Steps**	**Time (s)** $\downarrow$	**Speedup** $\uparrow$
TargetDiff	Full (1000 steps)	1000	831.1	1×
TargetDiff	Vanilla	50	42.2	19.7×
TargetDiff	Prob	50	42.8	19.4×
TargetDiff	Prob+Motion	50	43.3	19.2×
TargetDiff	Prob+Clash	50	43.5	19.1×
TargetDiff	Prob+Motion+Clash	50	44.0	18.9×
IPDiff	Full (1000 steps)	1000	2094.1	1×
IPDiff	Vanilla	50	106.3	19.7×
IPDiff	Prob	50	109.1	19.2×
IPDiff	Prob+Motion	50	110.2	19.0×
IPDiff	Prob+Clash	50	109.6	19.1×
IPDiff	Prob+Motion+Clash	50	110.2	19.0×
DecompDiff	Full (1000 steps)	1000	1075.4	1×
DecompDiff	Vanilla	50	54.3	19.8×
DecompDiff	Prob	50	55.1	19.5×
DecompDiff	Prob+Motion	50	56.0	19.2×
DecompDiff	Prob+Clash	50	56.6	19.0×
DecompDiff	Prob+Motion+Clash	50	56.9	18.9×

I Hyperparameter Sensitivity Analysis

To complement the defaults reported in Appendix G, this section assesses the robustness of our framework with respect to its key hyperparameters. We vary one hyperparameter at a time from the default configuration, holding all others fixed at $\delta_{LL} = 1.0\,\text{Å}$, $\delta_{LP} = 2.5\,\text{Å}$, $\lambda = 100$, $\tau_{\text{motion}} = 1.0$, and $\gamma = 0.8$. All experiments are conducted on TargetDiff with $K = 50$ sampling steps. We focus on a single backbone here for clarity and because each hyperparameter sweep already requires a non-trivial number of generation runs; trends on the other two backbones (IPDiff, DecompDiff) are

consistent with the TargetDiff results reported in Tables 1 and 14. Each parameter is swept on the configuration that exercises the relevant module: *Prob+Clash* for the clash-related hyperparameters $\delta_{LL}, \delta_{LP}, \lambda$ and the confidence threshold γ, and *Prob+Motion+Clash* for the motion-related hyperparameter τ_{motion}. The full-sampling reference (1,000 steps, $1\times$ speed) is reproduced from Table 1 for context.

Note on the time schedule. For computational efficiency during this initial hyperparameter screening, the sweeps in Sections I.1–I.3 below are run with the *uniform* time schedule ($\tau_j = j \cdot T/K$), rather than the coupling-aware schedule of Sec. 4.2. This is a deliberately conservative choice: the uniform schedule is the simpler baseline, so observing stable metrics across the full hyperparameter ranges under this setting is itself evidence that sensitivity to $(\gamma, \delta_{LL}, \delta_{LP}, \lambda, \tau_{\text{motion}})$ is small. The coupling-aware schedule (whose own hyperparameter β is examined separately in Sec. I.4) further improves absolute metrics on top of these defaults, as shown in Sec. 5.4; we view the schedule and these confidence-related hyperparameters as orthogonal axes of the framework. We note that the "uniform-time" schedule used here ($\tau_j = j \cdot T/K$) is a baseline *outside* the ρ-family of Sec. 4.2, and is distinct from the "logSNR-uniform" ($\beta{=}0$) and "$\bar{\alpha}$-uniform" ($\beta \to \infty$) schedules that arise as limits *within* that family.

I.1 Confidence Threshold γ

The confidence threshold γ governs how many low-confidence atoms are deferred at each step: only atoms with composite score $S^{(i)} \geq \gamma$ are eligible to transition. Setting γ too low approaches an unfiltered Top-K behavior, while setting it too high stalls transitions and reduces speed. We vary $\gamma \in \{0.6, 0.7, 0.8, 0.9\}$ on the Prob+Clash variant.

Table 10: Sensitivity of the confidence threshold γ on TargetDiff (50 steps, Prob+Clash). The default ($\gamma{=}0.8$) is shown in **bold**; the full-sampling reference is shown for comparison.

γ	QED ↑	SA ↑	Vina Score ↓	Vina Min ↓	Vina Dock ↓	Comp. % ↑	Rec. % ↑
0.6	0.45	0.58	−5.42	−6.52	−7.70	83.30	91.60
0.7	0.46	0.60	−5.50	−6.54	−7.75	84.20	93.50
0.8	**0.48**	**0.62**	**−5.39**	**−6.54**	**−7.62**	**86.20**	**97.24**
0.9	0.46	0.63	−5.04	−6.58	−7.81	86.00	98.50
Full (1000 steps)	0.48	0.58	−5.47	−6.64	−7.80	92.20	98.97

Across the entire range $\gamma \in [0.6, 0.9]$, chemical properties (QED, SA) and binding affinities (Vina Min, Vina Dock) remain within 0.04 and 0.19 of the default, respectively. Reconstruction rate increases monotonically with γ, as a stricter filter only admits the most confident updates, while completeness saturates around $\gamma{=}0.8$. The default $\gamma{=}0.8$ thus offers the best joint trade-off between Vina Score, completeness, and reconstruction.

I.2 Clash Hyperparameters: δ_{LL}, δ_{LP}, and λ

The steric-clash discount factor $w^{(i)}_{\text{clash}} = \min\big(\sigma(\lambda(d^{(i)}_{LL} - \delta_{LL})),\, \sigma(\lambda(d^{(i)}_{LP} - \delta_{LP}))\big)$ involves three hyperparameters: the ligand–ligand and ligand–protein distance thresholds δ_{LL}, δ_{LP} and the sigmoid sharpness λ. We sweep each independently with the others held at their defaults.

Distance thresholds. For δ_{LL}, performance is stable in a tight band around the default (0.75–1.25 Å) but degrades at $\delta_{LL}{=}1.5$ Å (-0.04 in QED, -3.3 pp in Comp.), since this value is large enough to mark physically reasonable bond distances as clashes. The default 1.0 Å is just below typical covalent bond lengths and provides a principled lower bound for "too close". For δ_{LP}, all four tested values (1.5 – 3.5 Å) yield essentially identical metrics, indicating that ligand–protein clashes are inherently a sharp signal once δ_{LP} exceeds the typical van der Waals contact range.

Sigmoid sharpness. The default $\lambda{=}100$ already places the sigmoid in a near-step-function regime, and metrics remain comparable for $\lambda \in \{100, 200, 500\}$. The smaller $\lambda{=}30$ slightly weakens Vina Dock (-0.50) because the softer transition allows the clash discount to partially down-weight atoms whose true distance only marginally violates the threshold.

Table 11: Sensitivity of clash hyperparameters δ_{LL}, δ_{LP}, and λ on TargetDiff (50 steps, Prob+Clash). The default row is shown in **bold**.

Configuration	QED ↑	SA ↑	Vina Score ↓	Vina Min ↓	Vina Dock ↓	Comp. % ↑	Rec. % ↑
Default	**0.48**	**0.62**	$\mathbf{-5.39}$	$\mathbf{-6.54}$	$\mathbf{-7.62}$	**86.20**	**97.24**
$\delta_{LL} = 0.75$ Å	0.47	0.61	-5.05	-6.47	-7.13	88.00	97.20
$\delta_{LL} = 1.25$ Å	0.46	0.62	-5.23	-6.55	-7.71	86.40	97.20
$\delta_{LL} = 1.50$ Å	0.44	0.61	-4.75	-6.32	-7.73	82.90	97.40
$\delta_{LP} = 1.5$ Å	0.46	0.62	-5.19	-6.58	-7.81	87.30	96.50
$\delta_{LP} = 2.0$ Å	0.46	0.62	-5.09	-6.59	-7.81	87.60	96.60
$\delta_{LP} = 3.0$ Å	0.46	0.61	-5.08	-6.61	-7.80	87.80	97.20
$\delta_{LP} = 3.5$ Å	0.46	0.62	-5.02	-6.51	-7.82	87.00	97.60
$\lambda = 30$	0.47	0.62	-5.08	-6.50	-7.12	87.30	98.20
$\lambda = 200$	0.45	0.62	-5.09	-6.37	-7.75	87.40	97.80
$\lambda = 500$	0.46	0.62	-5.39	-6.55	-7.71	87.40	97.10

I.3 Motion Temperature τ_{motion}

The motion temperature τ_{motion} in $c_{\text{motion}}^{(i,t)} = \exp(-\|\Delta\mathbf{x}^{(i,t)}\|/\tau_{\text{motion}})$ controls how strongly we penalize predicted atomic displacement: small τ_{motion} makes motion confidence drop sharply with displacement, while large τ_{motion} flattens the factor toward 1 (effectively disabling the motion module). We sweep $\tau_{\text{motion}} \in \{0.5, 1.0, 2.0, 4.0\}$ on the Prob+Motion+Clash variant.

Table 12: Sensitivity of the motion temperature τ_{motion} on TargetDiff (50 steps, Prob+Motion+Clash). The default (τ_{motion}=1.0) is shown in **bold**; the default row corresponds to the Prob+Motion+Clash configuration under the uniform-time schedule, i.e., the same screening setting as the rest of this sweep.

τ_{motion}	QED ↑	SA ↑	Vina Score ↓	Vina Min ↓	Vina Dock ↓	Comp. % ↑	Rec. % ↑
0.5	0.46	0.62	-4.94	-6.59	-7.24	86.70	97.50
1.0	**0.47**	**0.63**	$\mathbf{-5.23}$	$\mathbf{-6.41}$	$\mathbf{-7.71}$	**85.54**	**97.98**
2.0	0.47	0.61	-5.42	-6.65	-7.71	84.50	95.30
4.0	0.46	0.60	-5.25	-6.44	-7.60	84.70	94.30

Vina Min varies by less than 0.24 across the entire range and SA by at most 0.03. Reconstruction rate is the most sensitive metric: at τ_{motion}=4.0 it drops to 94.30%, but this still substantially exceeds the Vanilla Integration baseline (78.65% on the same backbone, cf. Table 1). The motion module is therefore beneficial across a wide range of temperatures, with $\tau_{\text{motion}} \in [0.5, 2.0]$ forming a near-flat plateau around the chosen default.

I.4 Coupling Weight β

The coupling weight β in Eq. (11) interpolates between two well-known limiting schedules: β=0 recovers the logSNR-uniform schedule of pure continuous diffusion (DPM-Solver), while $\beta \to \infty$ recovers the $\bar{\alpha}$-uniform schedule used by DNDM. The value reported in Sec. 5.4 is selected per model by a Pareto-coverage criterion on a held-out validation pocket set, described next.

Selection protocol. For each model we randomly sample 30 pockets from the training set (disjoint from the test pockets used in Table 1). For each pocket we generate 10 ligands per candidate $\beta \in \{0,\ 0.25,\ 0.5,\ 1,\ 2,\ 10^6\}$ and compute five per-pocket averaged quality scores: Completeness ratio, −Vina Score, −Vina Min, QED, and SA (all higher is better, signs flipped where appropriate). For each pocket, a configuration is on the Pareto front if no other β dominates it across all five scores. We then count, for each β, the number of pockets in which it appears on the Pareto front. The β with the highest Pareto-coverage count is selected. Ties are broken by preferring the β closer to 0 (i.e., closer to the logSNR limit), favouring the more conservative schedule.

Table 13: Per-model selection of the coupling weight β via Pareto coverage over {Completeness, $-$Vina Score, $-$Vina Min, QED, SA} on held-out training pockets (n pockets, 10 ligands each). Each cell reports the number of pockets in which that β is on the Pareto front. The selected β per model is marked with $\star$ and shown in **bold**. $\beta{=}10^6$ is a numerical surrogate for the $\bar{\alpha}$-uniform limit ($\beta \to \infty$).

β	TargetDiff (n=30)	IPDiff (n=30)	DecompDiff (n=30)
0 (logSNR-uniform)	22	23	**25** $\star$
0.25	**23** $\star$	22	18
0.5	21	**25** $\star$	23
1	21	22	20
2	22	18	23
10^6 ($\bar{\alpha}$-uniform)	16	17	13

Discussion. Two observations motivate the design of $\rho(t)$ in Eq. (11). First, performance is broadly *robust* to β within the moderate range $\beta \in [0, 2]$: the top score and the runner-up are typically separated by only 1–3 pockets out of ~ 30, and all values in this range yield Pareto coverage well above the $\beta{=}10^6$ extreme. This indicates that the proposed schedule provides a controllable but well-behaved knob rather than a brittle hyperparameter. Second, the heavily skewed $\bar{\alpha}$-uniform endpoint ($\beta{=}10^6$) is substantially worse on every model (16/17/13 vs. ≥ 22 at the optimum), confirming that schedule shape genuinely matters and that the selected β's are not artefacts of noise. The selected values exhibit a model-dependent pattern: TargetDiff prefers a small but non-zero $\beta{=}0.25$, IPDiff prefers a moderate $\beta{=}0.5$, and DecompDiff self-selects $\beta{=}0$, recovering the logSNR-uniform schedule. We adopt these per-model defaults in Sec. 5.4 (Table 4).

I.5 Summary

Across all six hyperparameters, no single sweep value pushes any metric outside a ± 0.04 band on QED, ± 0.04 on SA, or below the Vanilla Integration baseline on Reconstruction Rate. The only parameter exhibiting a clearly visible regime change is δ_{LL} at large values (where physically meaningful bond distances begin to be flagged as clashes); all other parameters are flat over their natural ranges. Importantly, the sweeps for $(\gamma, \delta_{LL}, \delta_{LP}, \lambda, \tau_{\text{motion}})$ were carried out under the simpler *uniform* time schedule rather than our optimized coupling-aware schedule; the fact that metrics remain stable even under this weaker schedule indicates that the framework's robustness is intrinsic to the confidence-guided design and not an artefact of a particular schedule choice. We therefore conclude that the reported gains are not the artefact of carefully tuned hyperparameters, and the default configuration $(\delta_{LL}, \delta_{LP}, \lambda, \tau_{\text{motion}}, \gamma, \beta) = (1.0\,\text{Å}, 2.5\,\text{Å}, 100, 1.0, 0.8,$ model-dependent: $0.25/0.5/0$ for TargetDiff/IPDiff/DecompDiff) generalizes across the SBDD task.

J Additional Results Across Sampling Budgets

J.1 Results at 100 Sampling Steps

To complement the 50-step comparison reported in Table 1 of the main text, we additionally evaluate all confidence-guided variants at **100 sampling steps**, which corresponds to roughly $9.5\times$ speedup over the full 1,000-step reference. The complete results are reported in Table 14.

100-step vs. 50-step trends. Three observations refine the main-text claim under the larger 100-step budget. **(i) Gaps to the 1,000-step reference shrink with step count.** On TargetDiff, the Completeness gap of our best confidence-guided variant narrows from 4.04 pp at 50 steps ($88.16 \to 92.20$, **Prob+Clash**) to 3.72 pp at 100 steps ($88.48 \to 92.20$, **Prob+Motion**); on IPDiff the gap closes from 3.00 pp to 2.53 pp under **Prob+Clash**. On DecompDiff the gap is already sub-percentage-point at 50 steps (0.79 pp under **Prob+Motion+Clash**) and stays in that band at 100 steps (0.86 pp), indicating that additional steps recover quality predictably without altering the qualitative picture. Vina Min displays the same direction: TargetDiff's best-variant Vina Min Avg. gap shrinks from 0.18 to 0.11 kcal/mol between the two budgets. **(ii) The relative ordering among confidence-guided variants is preserved across step budgets.** On every backbone, the geometric-readiness variants

Table 14: Performance comparison across three SBDD backbones (TargetDiff, IPDiff, DecompDiff) at **100 sampling steps**. Same conventions as Table 1: **bold** rows are confidence-guided variants; all experiments use our optimized non-uniform schedule; the shared confidence-threshold mechanism is omitted from row labels for brevity. "Full (1000 steps)" is each backbone's full-sampling reference. Among accelerated methods, green marks the best and light green the second-best.

Model	Acceleration	QED ↑	SA ↑	Vina Score ↓		Vina Min ↓		Vina Dock ↓		Div. ↑	Rec. % ↑	Comp. % ↑	Speed
				Avg.	Med.	Avg.	Med.	Avg.	Med.				
TargetDiff	Full (1000 steps)	0.48	0.58	-5.47	-6.30	-6.64	-6.83	-7.80	-7.91	0.72	98.97	92.20	1×
TargetDiff	Vanilla	0.46	0.57	-5.33	-5.91	-6.24	-6.21	-7.33	-7.49	0.75	84.05	77.71	9.7×
TargetDiff	**Prob**	0.49	0.62	-5.42	-6.06	-6.53	-6.49	-7.46	-7.68	0.74	95.86	86.93	9.6×
TargetDiff	**Prob+Motion**	0.49	0.63	-5.35	-6.09	-6.43	-6.55	-7.51	-7.72	0.74	97.15	88.48	9.5×
TargetDiff	**Prob+Clash**	0.49	0.61	-5.39	-6.12	-6.50	-6.56	-7.63	-7.71	0.74	96.67	88.11	9.4×
TargetDiff	**Prob+Motion+Clash**	0.49	0.64	-5.25	-6.10	-6.40	-6.50	-7.60	-7.72	0.74	97.78	88.21	9.4×
IPDiff	Full (1000 steps)	0.52	0.61	-6.42	-7.01	-7.45	-7.48	-8.57	-8.51	0.74	98.25	93.80	1×
IPDiff	Vanilla	0.49	0.57	-6.27	-7.00	-7.06	-7.10	-8.03	-8.04	0.76	95.15	88.60	9.8 ×
IPDiff	**Prob**	0.50	0.56	-6.39	-7.10	-7.25	-7.24	-8.04	-8.09	0.75	96.28	89.88	9.5×
IPDiff	**Prob+Motion**	0.49	0.60	-6.19	-7.02	-7.16	-7.22	-8.16	-8.12	0.77	97.65	90.77	9.4×
IPDiff	**Prob+Clash**	0.50	0.58	-6.46	-7.12	-7.27	-7.28	-8.17	-8.21	0.76	97.25	91.27	9.4×
IPDiff	**Prob+Motion+Clash**	0.49	0.61	-6.01	-7.04	-7.09	-7.19	-8.09	-8.06	0.77	98.20	90.12	9.2×
DecompDiff	Full (1000 steps)	0.49	0.64	-6.11	-6.16	-6.68	-6.58	-7.53	-7.59	0.71	99.60	93.59	1×
DecompDiff	Vanilla	0.46	0.61	-5.10	-5.16	-5.75	-5.70	-6.90	-6.96	0.73	84.95	81.95	9.8×
DecompDiff	**Prob**	0.48	0.61	-6.15	-6.15	-6.52	-6.40	-7.47	-7.48	0.72	90.22	85.17	9.5 ×
DecompDiff	**Prob+Motion**	0.50	0.62	-6.33	-6.32	-6.67	-6.52	-7.60	-7.57	0.72	94.25	91.60	9.5×
DecompDiff	**Prob+Clash**	0.50	0.62	-6.22	-6.17	-6.56	-6.41	-7.33	-7.47	0.72	92.35	89.12	9.4×
DecompDiff	**Prob+Motion+Clash**	0.49	0.62	-6.38	-6.33	-6.74	-6.59	-7.66	-7.61	0.73	95.05	92.73	9.4×

(**Prob+Motion**, **Prob+Clash**, **Prob+Motion+Clash**) lead on Completeness at both 50 and 100 steps, while pure **Prob** (probability alone) consistently trails—by under 2 pp on TargetDiff and IPDiff and by 6–8 pp on DecompDiff. This shows the geometric readiness design is not tuned to the 50-step setting and generalizes across step-budget regimes. **(iii) Vanilla integration retains a large gap to the geometric-readiness variants even at 100 steps.** On TargetDiff its Completeness is only 77.71%, lagging the geometric-readiness variants by 10–11 pp; on DecompDiff it is 81.95%, lagging by 7–11 pp. On IPDiff the gap narrows from 5.1 pp at 50 steps to 2.7 pp at 100 steps but does not vanish. Doubling the step budget therefore does not let the unprioritised baseline catch up, confirming that the bottleneck under acceleration is the absence of a coupling-aware update rule, not the step count itself.

J.2 Results at 40, 20, and 10 Sampling Steps

We evaluate more aggressive step reduction on TargetDiff using Prob+Motion+Clash. Table 15 includes the full 1,000-step reference and the 50-step configuration for context. At 40 and 20 steps, Vina Dock and chemical properties remain close to the 50-step results, while completeness decreases from 87.95% to 85.82% at 20 steps. Reducing the budget from 20 to 10 steps leads to a larger deterioration: Vina Dock changes from -7.58 to -7.21, QED from 0.47 to 0.43, and completeness falls by 10.22 percentage points to 75.60%.

Table 15: Mean performance of TargetDiff across sampling budgets. The 1,000-step row is the full-sampling reference; all reduced-step rows use CGS with Prob+Motion+Clash. The 1,000- and 50-step results are reproduced from Table 1. Vina values are in kcal/mol; Comp. is reported as a percentage.

Steps	Vina Score ↓	Vina Min ↓	Vina Dock ↓	QED ↑	SA ↑	Comp. % ↑	Div. ↑
1000	−5.47	−6.64	−7.80	0.48	0.58	92.20	0.72
50	−5.18	−6.41	−7.62	0.49	0.64	87.95	0.74
40	−5.18	−6.37	−7.59	0.48	0.64	87.25	0.77
20	−5.19	−6.36	−7.58	0.47	0.63	85.82	0.78
10	−5.29	−6.23	−7.21	0.43	0.62	75.60	0.80

K Acceleration of DPO-Aligned IPDiff

Inspired by the preference-alignment framework of AliDiff [Gu et al., 2024], we fine-tune IPDiff using direct preference optimization (DPO) [Rafailov et al., 2023, Wallace et al., 2024], with 50 generated samples for each of 200 pockets. Table 16 compares the original IPDiff and our DPO-aligned checkpoint under full 1,000-step sampling and 50-step CGS with Prob+Clash.

CGS also accelerates preference-aligned IPDiff. Applying CGS reduces sampling from 1,000 to 50 steps, with Vina Min and Vina Dock changing from -7.75 and -8.68 to -7.41 and -8.21, respectively. At 50 steps, the aligned model retains improvements in Vina Min, QED, and SA over unaligned IPDiff+CGS.

Table 16: Acceleration of DPO-aligned IPDiff. Results for unaligned IPDiff and its Prob+Clash CGS variant are reproduced from Table 1. All values are means; Vina values are in kcal/mol, and Comp. is reported as a proportion rounded to two decimal places.

Model	Steps	Vina Score ↓	Vina Min ↓	Vina Dock ↓	QED ↑	SA ↑	Comp. ↑	Div. ↑
IPDiff	1000	−6.42	−7.45	−8.57	0.52	0.61	0.94	0.74
IPDiff + CGS	50	−6.36	−7.27	−8.19	0.50	0.59	0.91	0.75
DPO-aligned IPDiff	1000	−6.67	−7.75	−8.68	0.53	0.62	0.91	0.71
DPO-aligned IPDiff + CGS	50	−6.41	−7.41	−8.21	0.51	0.63	0.87	0.74

L Equal-Time-Budget Evaluation: Full Breakdown

Table 17 reports the complete Top-10/20/50 breakdown of the equal-time-budget experiment summarized in Table 3, together with mean Vina Min and explicit CGS/baseline multipliers. Three observations refine the main-text claim.

(i) The Top-k advantage is monotonic in k. Top-10 Vina Min gaps reach up to 2.92 kcal/mol (DecompDiff/2Z3H-A) and shrink as the filter becomes less selective; the main text reports Top-50 to present the conservative end of this trend rather than the most striking one.

(ii) Mean Vina Min is an unreliable summary in the equal-time-budget setting. On IPDiff/4AAW-A the baseline mean is even slightly better (-7.67 vs -6.93), yet CGS still dominates at every Top-k by 1.65–1.95 kcal/mol. The $18\times$ larger CGS pool contains both stronger top candidates and a longer low-quality tail, so the practical sample-then-filter workflow — not the raw mean — captures what the speedup actually buys.

Table 17: Full Top-10/20/50 breakdown for the equal-time-budget experiment of Table 3. Baseline = 1000-step DDPM sampling at 100 molecules per pocket; CGS = our Confidence-Guided Sampling with the Prob+Clash under our coupling-aware schedule (50 steps) at 1800 molecules per pocket. Multipliers in parentheses denote CGS / Baseline ratios. All Vina values are in kcal/mol. Bold entries mark the better of the two methods within each (backbone, pocket, metric) cell.

Backbone	Pocket	Method	n	Valid %	Mean Vmin	Top-k Vina Min ↓			Top-k Vina Score ↓			PB-Valid	Strain<50
						T10	T20	T50	T10	T20	T50		
TargetDiff	2Z3H_A	Baseline	100	94%	−6.76	−9.87	−9.30	−8.18	−9.12	−8.58	−7.48	32	1
		CGS	1800	91%	**−7.23**	**−11.32**	**−11.04**	**−10.60**	**−10.67**	**−10.40**	**−9.95**	**776** (24×)	**12** (12×)
	4AAW_A	Baseline	100	97%	−6.40	−9.32	−8.62	−7.55	−8.77	−8.10	−7.11	64	4
		CGS	1800	94%	**−6.61**	**−11.31**	**−10.75**	**−10.05**	**−11.02**	**−10.41**	**−9.67**	**1110** (17×)	**50** (12.5×)
IPDiff	2Z3H_A	Baseline	100	83%	−7.65	−10.89	−10.16	−8.94	−10.77	−10.06	−8.94	8	0
		CGS	1800	86%	**−7.88**	**−12.41**	**−12.06**	**−11.60**	**−12.25**	**−11.92**	**−11.39**	**246** (30×)	0
	4AAW_A	Baseline	100	97%	**−7.67**	−10.26	−9.86	−8.90	−10.07	−9.57	−8.69	20	1
		CGS	1800	94%	−6.93	**−12.06**	**−11.55**	**−10.85**	**−11.61**	**−11.17**	**−10.54**	**391** (19×)	**8** (8×)
DecompDiff	2Z3H_A	Baseline	100	97%	−7.64	−9.44	−9.11	−8.47	−8.58	−8.33	−7.70	31	0
		CGS	1800	93%	**−9.27**	**−12.36**	**−12.04**	**−11.70**	**−12.10**	**−11.83**	**−11.49**	**700** (22×)	0
	4AAW_A	Baseline	100	91%	−6.56	−8.42	−8.02	−7.30	−7.73	−7.34	−6.63	41	0
		CGS	1800	87%	**−7.76**	**−10.85**	**−10.65**	**−10.39**	**−10.51**	**−10.34**	**−10.01**	**485** (12×)	0

(iii) Strain<50 zeros on IPDiff and DecompDiff are backbone-level, not method-level. Both backbones tend to generate ligands in higher-strain regions regardless of the sampling procedure, so the absolute count of low-strain molecules is small for both columns. On backbones where this metric is informative (TargetDiff and IPDiff/4AAW-A), CGS yields 8–12.5× more low-strain molecules under the same wall-clock budget.

M Sample Examples

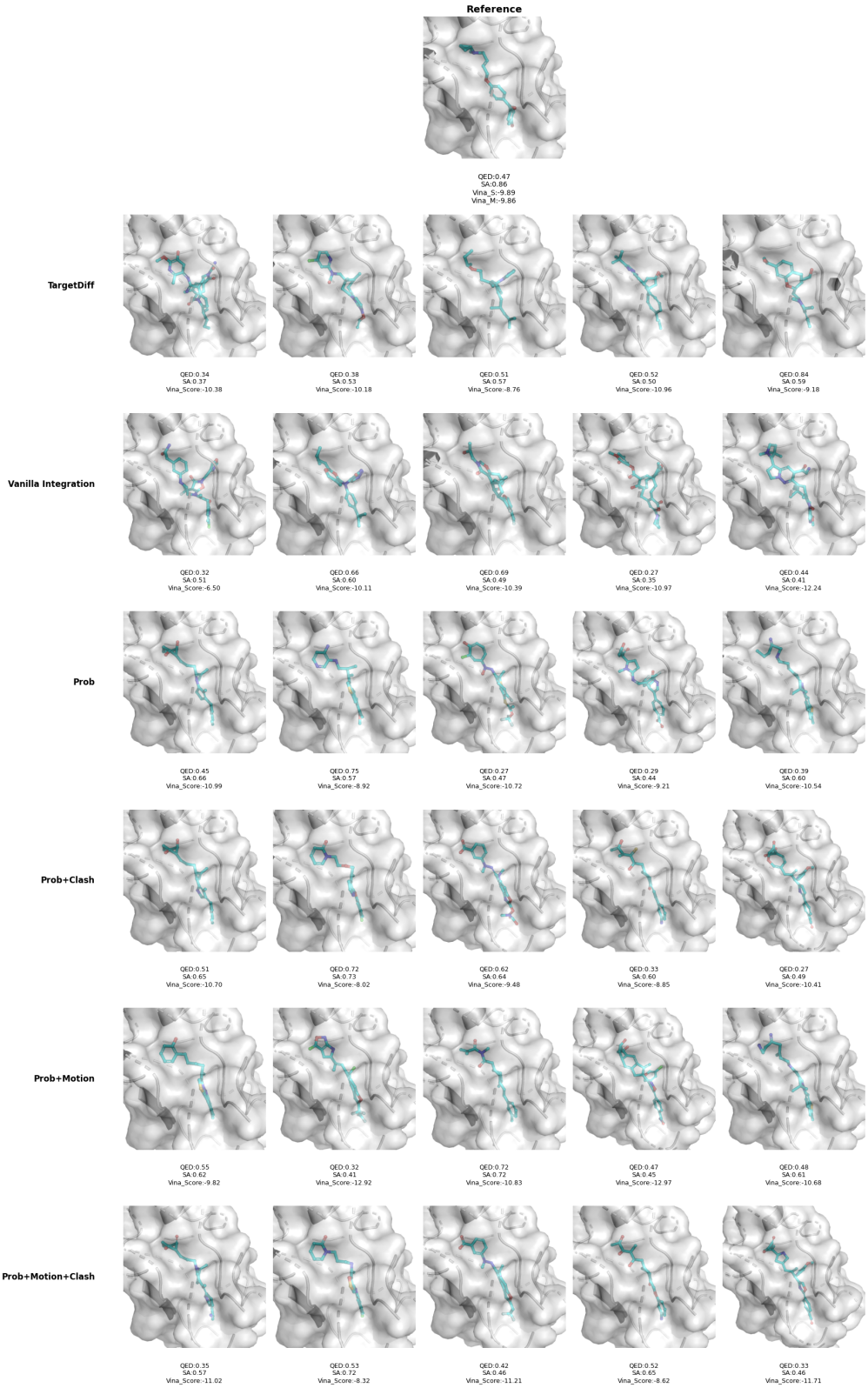


Figure 4: Sampling results using the original TargetDiff and several acceleration methods for receptor 1H36_A.

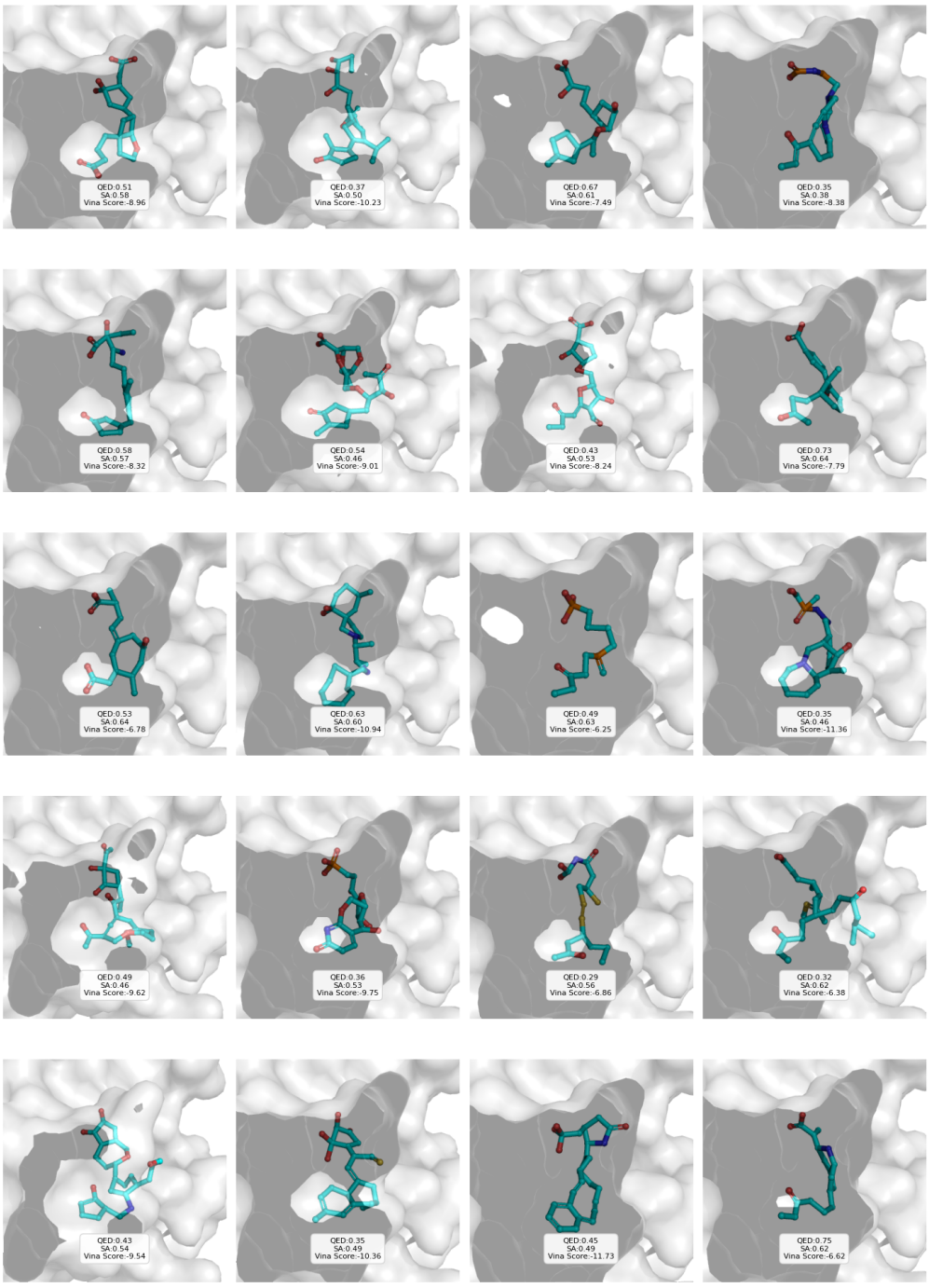


Figure 5: Accelerated Sampling results using IPDiff with Probability + Motion Confidence for receptor 2xtz_A.

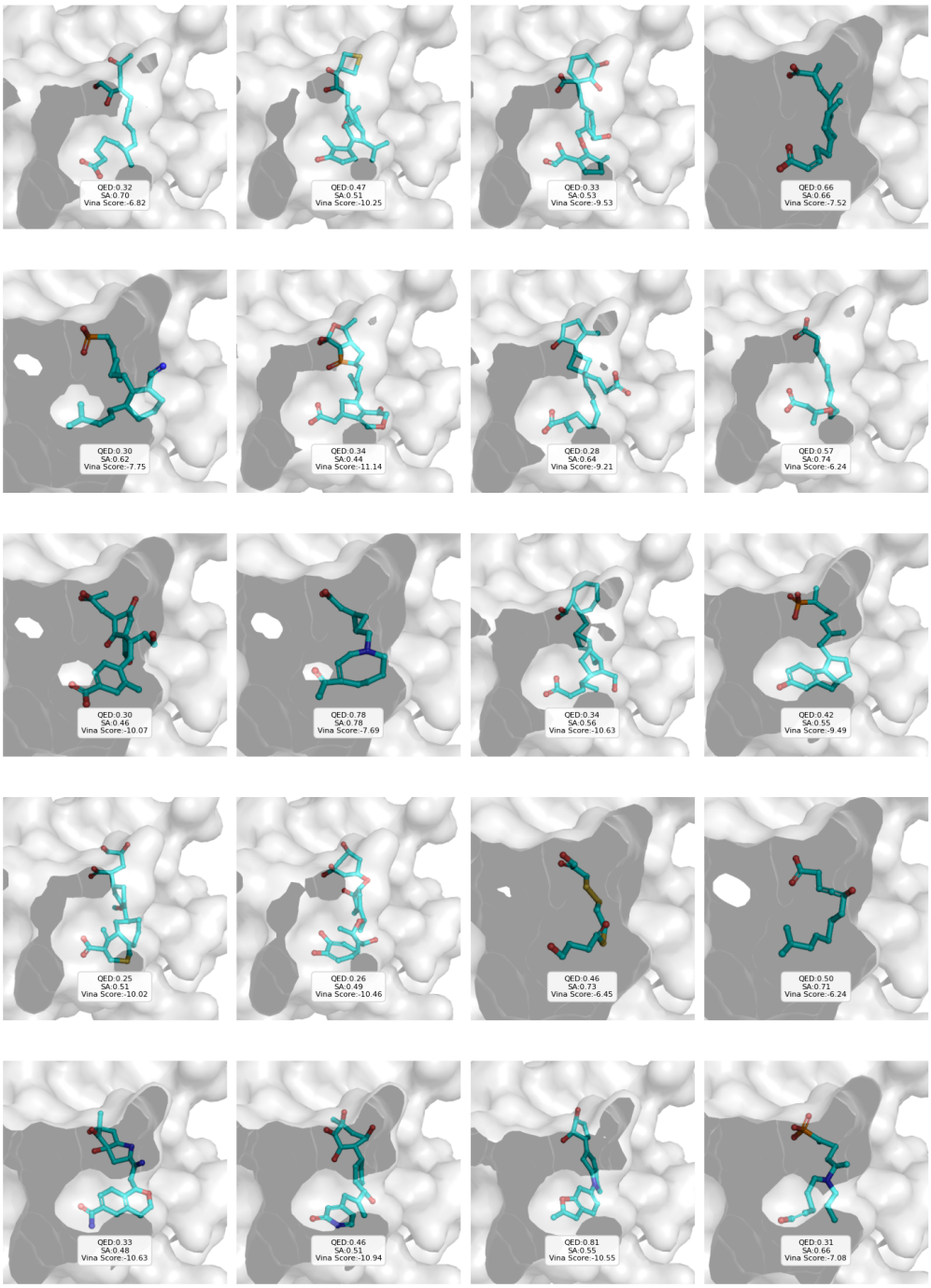


Figure 6: Accelerated Sampling results using IPDiff with Probability + Clash Discount Confidence for receptor 2xtz_A.

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: abstract and Section 1 state our three contributions: (i) a unified DDIM–DNDM sampler under a shared schedule (Section 4.1, Prop. 1); (ii) a coupling-aware schedule derived from an error-equalization principle (Section 4.2); and (iii) confidence-based atom-level sampling (Section 4.3). All claims are validated empirically in Section 5.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: A dedicated Section 6 discusses (i) the limited *backbone scope* of our evaluation (TargetDiff, IPDiff, DecompDiff) and outlines what is needed to extend the framework to more backbones and consider the acceleration of other types models; and (ii) the *calibration dependence* of the confidence-based atom-delay rule, together with temperature scaling as a practical mitigation.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: All assumptions are stated explicitly in the theorem statements (Prop. 1; Cor. 1); full proofs are in Appendix B, with assumptions summarized in Appendix B.1 and practical surrogate justifications in Appendix B.4.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The unified algorithm is given in Algorithm 1 (Appendix A), the coupling-aware schedule and its surrogates in Section 4.2 and Appendix B.4, and the confidence rules with all thresholds in Section 4.3; we use the public TargetDiff/IPDiff/DecompDiff checkpoints on the standard CrossDocked2020 split (Section 5) and require no retraining.

   Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: Codes will be released upon acceptance; the paper provides the full pseudocode (Algorithm 1, Appendix A) and all hyperparameters (Section 5, Appendix I) needed to reimplement the method on top of the public backbone checkpoints.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Dataset, split, sampling protocol, and evaluation metrics are described in Section 5; full hyperparameter values and sensitivity studies are in Appendix I (Tables 10, 11, 12, 13); we use the released checkpoints of TargetDiff, IPDiff, and DecompDiff with no retraining.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

Justification: Following the standard SBDD evaluation protocol of TargetDiff, IPDiff, and DecompDiff, each cell of Table 1 aggregates 100 generated molecules per pocket across 100 test pockets (10,000 samples per configuration). For affinity-related metrics we report both the mean (sensitive to tails) and the median (robust); their consistency, together with the large evaluation set, characterizes the distribution and supports the comparisons we draw.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: All sampling experiments are run on a single NVIDIA A100 80GB GPU; per-pocket wall-clock measurements and full-vs-accelerated runtimes are reported in Appendix H (Table 9), consistent with the $\sim 19\times$ speedup in Table 1.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics; the work uses only public protein–ligand benchmarks (CrossDocked2020) and pretrained checkpoints, involves no human subjects.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Section 6 discusses both sides: the method lowers SBDD sampling cost (positive), while inheriting standard dual-use concerns of generative chemistry; since we change only sampling efficiency rather than the generative distribution, the incremental risk over the underlying base models is small and standard toxicity/synthesizability filters apply unchanged.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This paper does not pose such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We use four publicly released assets, all of which are properly cited in Section 5: CrossDocked2020 [Francoeur et al., 2020] (`https://github.com/gnina/models/tree/master/data/CrossDocked2020`; released publicly by the authors, no explicit repository license), TargetDiff [Guan et al., 2023a] (`https://github.com/guanjq/targetdiff`; MIT), IPDiff [Huang et al., 2024] (`https://github.com/YangLing0818/IPDiff`; MIT), and DecompDiff [Guan et al., 2023b] (`https://github.com/bytedance/DecompDiff`; CC BY-NC 4.0). We comply with each license; in particular, our use of DecompDiff is non-commercial academic research consistent with the CC BY-NC 4.0 terms, and we use the official checkpoints without modification.

    Guidelines:

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

13. **New assets**

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

    Answer: [N/A]

    Justification: The paper does not release new datasets or pretrained models; the contribution is a training-free sampler operating on existing public checkpoints.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human-subjects research.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification:The paper does not involve human-subjects research, so IRB review is not applicable.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs are not part of the methodology; any use was limited to incidental writing or editing assistance, which the policy states does not require declaration.

    Guidelines:

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