

Straight-Line Diffusion Model for Efficient 3D Molecular Generation

Yuyan Ni^{1,4,*†} Shikun Feng^{2,*} Haohan Chi³ Bowen Zheng⁵ Huan-ang Gao³
Wei-Ying Ma³ Zhi-Ming Ma¹ Yanyan Lan^{3,6,7,‡}

¹ Academy of Mathematics and Systems Science, Chinese Academy of Sciences

² Zhongguancun Institute of Artificial Intelligence, China

³ Institute for AI Industry Research (AIR), Tsinghua University

⁴ University of Chinese Academy of Sciences

⁵ Huazhong University of Science and Technology

⁶ Beijing Frontier Research Center for Biological Structure, Tsinghua University

⁷ Beijing Academy of Artificial Intelligence

Abstract

Diffusion-based models have shown great promise in molecular generation but often require a large number of sampling steps to generate valid samples. In this paper, we introduce a novel Straight-Line Diffusion Model (SLDM) to tackle this problem, by formulating the diffusion process to follow a linear trajectory. The proposed process aligns well with the noise sensitivity characteristic of molecular structures and uniformly distributes reconstruction effort across the generative process, thus enhancing learning efficiency and efficacy. Consequently, SLDM achieves state-of-the-art performance on 3D molecule generation benchmarks, delivering a 100-fold improvement in sampling efficiency.¹

1 Introduction

3D molecular generation is an essential task in drug discovery, material science, and molecular engineering. The goal is to computationally design 3D molecular structures that not only capture intricate physical and chemical constraints but also fulfill specific properties.

Recently, diffusion models have been widely applied in this field, inspired by their remarkable success in image synthesis [Dhariwal and Nichol, 2021, Rombach et al., 2022, Peebles and Xie, 2023], and other domains [Brooks et al., 2024, Abramson et al., 2024]. Methods like EDM [Hoogeboom et al., 2022], EDM-Bridge [Wu et al., 2022] and GeoLDM [Xu et al., 2023] have demonstrated the potential of diffusion-based frameworks to generate chemically valid 3D molecular structures. However, these direct applications of diffusion methods usually require a large number of sampling steps to produce valid molecules.

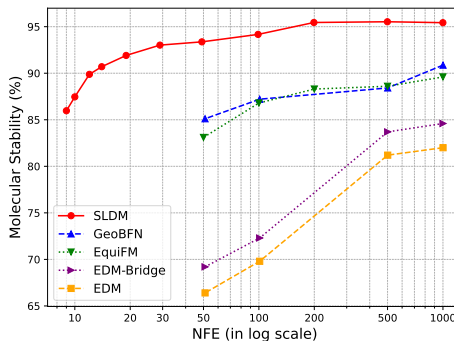


Figure 1: Comparison of molecule stability (↑) across diffusion-based molecular generation models on QM9 unconditional generation, evaluated by the number of function evaluations (NFE) during the sampling process.

*Equal contribution. † Work was done during Yuyan’s internship at AIR. ‡ Corresponding author: Yanyan Lan (lanyanyan@air.tsinghua.edu.cn).

¹The code is open-sourced at <https://github.com/fengshikun/SLDM>

Taking EDM as an example, it requires approximately 1000 steps of function evaluations to generate molecules with around 82% stability, which is a key metric for assessing sample quality by quantitatively measuring whether the molecule satisfies chemical constraints. Reducing the sampling steps significantly degrades the molecule stability.

To improve sampling efficiency, EquiFM [Song et al., 2024] and GeoBFN [Song et al., 2023a] have been proposed to utilize the Flow Matching (FM) framework [Tong et al., 2023] and Bayesian Flow Networks (BFN) [Graves et al., 2023] for molecular generation. The use of these advanced generative AI models enables a speedup of $5\times$ and $20\times$, respectively, compared to EDM. However, they still require a large number of sampling steps (e.g. 1000) to achieve high molecule stability (e.g. 90%), as shown in Figure 1.

To understand why existing methods suffer from low efficiency, we analyze the issue through the lens of truncation error in sampling. We begin by establishing a unified perspective on previous diffusion-based methods, including diffusion models, flow matching methods, and Bayesian flow networks. Specifically, their noise corrupting process can be generalized as $x_t = \mu(t)x_0 + \sigma(t)\epsilon$, $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_N)$, where x_0 represents the clean data, x_t is the noise corrupted data at time $t \in [0, T]$, $\mu(t)$ and $\sigma(t)$ define the schedule of the process. In this framework, all these processes can be equivalently framed as continuous-time Ordinary Differential Equations (ODEs), even though they may employ stochastic sampling in practice. This viewpoint allows the sampling process to be interpreted as a numerical approximation of the solution trajectory of the underlying ODE. Crucially, most existing methods rely on first-order estimation, whose truncation error is governed by the second-order term $\frac{d^2 x(t)}{dt^2}(\Delta t)^2$. We observe that $\frac{d^2 x(t)}{dt^2}$ in these approaches can be large, requiring small step sizes Δt to reduce the truncation error, which results in a large number of sampling steps.

To address this issue, we propose a novel diffusion process called Straight-Line Diffusion Model (SLDM). The key idea is to minimize truncation error by striving to achieve a linear sampling trajectory, i.e. $\frac{d^2 x(t)}{dt^2} = 0$. This approach allows the diffusion dynamics to tolerate larger step sizes without sacrificing accuracy, leading to a substantial improvement in sampling efficiency. Building on this objective, we theoretically prove that when $\mu(t) = 1 - t/T$ and σ is a small constant, the process guarantees a near-linear trajectory. Intuitively, this process features a linearly decreasing mean term and a consistently small variance term, representing a smooth linear progression from the origin point to the data.

Notably, SLDM strikes a good balance between efficiency and efficacy by using our straight-line schedule. Firstly, this strategy aligns well with the inductive bias of molecular generation, preventing the introduction of chemically implausible conformations. Unlike images, molecular structures are much more sensitive to noise, and even small perturbations can lead to unrealistic structures that violate chemical principles. This challenge requires a slower signal-to-noise ratio (SNR) decay during the noise-adding process, as suggested in Song et al. [2023a]. By naturally satisfying a slower SNR decay, our method maintains chemical information of the intermediate states, enhancing computational efficiency compared to traditional methods such as EDM. Secondly, our strategy achieves a more balanced generative process, significantly improving the model’s learning efficacy. In methods like GeoBFN, minimal perturbations are applied in the later stages, shifting most of the reconstruction burden to earlier stages. Although this reduces the computational load in the later stages, it creates an uneven distribution of effort, limiting the model’s learning capacity. In contrast, our approach evenly distributes the reconstruction effort across the entire process, enabling the model to learn effectively at each stage. This balance results in a more stable and efficient learning process, enhancing the robustness and accuracy of the generated molecular structures.

We conduct extensive experiments to demonstrate the potential of straight-line diffusion in 3D molecular generation and other domains. As shown in Figure 1, using only at most 10 or 15 sampling steps, SLDM surpasses EDM or EquiFM, GeoBFN with 1000 sampling steps, achieving up to **100- or 70-fold** improvement in sampling efficiency. In terms of generation quality, SLDM with 200 sampling steps achieves 95% molecular stability, significantly outperforming the best baseline, GeoBFN, which requires 1000 steps to reach 90% molecular stability. We also observe that similar improvements can be achieved when applying SLDM to the conditional generation task, i.e. generating molecules with a desired property, highlighting its potential to enable more practical and controllable molecular design in future applications.

2 Analysis on Sampling Efficiency

We begin by theoretically analyzing the underlying factors contributing to the sampling efficiency issue. In particular, we first present a unified framework for diffusion-based methods, including their SDE and ODE formulations. We then examine the sampling truncation error from the ODE perspective, highlighting the critical role of the process’s second-order derivative in improving sampling efficiency.

We denote the data as $\mathbf{x} \in \mathbb{R}^N$, where for molecules, it refers to a 3D point cloud comprising atomic coordinates and potentially other atomic features. According to Karras et al. [2022], Xue et al. [2024a], various diffusion-based models, including DDPM [Ho et al., 2020], DDIM [Song et al., 2021a], VE [Song et al., 2021b], FM [Lipman et al., 2023], and BFN [Graves et al., 2023], can be formulated as a unified form with the noise corrupting process defined as:

$$\mathbf{x}_t = \mu(t)\mathbf{x}_0 + \sigma(t)\epsilon, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_N), t \in [0, T] \quad (1)$$

where $\mathbf{x}_0, \mathbf{x}_t$ are clean data and noise corrupted data respectively, $\mu(t)$ and $\sigma(t)$ define the schedule of the process. Specifically, as detailed in Appendix A.2, the schedule parameters are summarized as follows: $\mu_{\text{DDPM(EDM)}} = 1 - (t/T)^2$, $\mu_{\text{VE}} = \mu_{\text{DDIM}} = 1$, $\mu_{\text{BFN}} = 1 - \sigma_{\min}^{2(1-t/T)}$, $\mu_{\text{FM}} = 1 - t/T$; $\sigma_{\text{DDPM(EDM)}} = \sqrt{1 - (1 - (t/T)^2)^2}$, $\sigma_{\text{VE}} = \sqrt{t}$, $\sigma_{\text{DDIM}} = t$, $\sigma_{\text{BFN}} = \sqrt{\mu_{\text{BFN}}(1 - \mu_{\text{BFN}})}$, $\sigma_{\text{FM}} = t/T + (1 - t/T)\sigma_{\min}$, where DDPM(EDM) uses the approximated DDPM schedule given in EDM [Hoogeboom et al., 2022]. $\sigma_{\min}$ are defined as small constants to ensure $\mu(0) \approx 1$ and $\sigma(0) \approx 0$. T is typically chosen to be sufficiently large so that $\mathbf{x}_T$ approximates a known distribution.

Extending a similar theoretical technique from Karras et al. [2022] to the unified form, we can prove that equation 1 is the solution to the following linear stochastic differential equation (SDE):

$$d\mathbf{x}_t = \frac{\dot{\mu}(t)}{\mu(t)}\mathbf{x}_t dt + \sqrt{2\sigma(t)\dot{\sigma}(t) - 2\sigma(t)^2 \frac{\dot{\mu}(t)}{\mu(t)}} dw_t, \quad (2)$$

where $\mu(t)$ and $\sigma(t)$ should adhere to the constraints $\mu(0) = 1$, $\sigma(0) = 0$, with $\mu(t) \geq 0$ being monotone non-increasing, $\sigma(t) \geq 0$, and $\sigma(t)/\mu(t)$ being monotone non-decreasing. The detailed proof is elaborated in Appendix A.1.

According to the ODE and SDE relations revealed in Song et al. [2021b], we can derive the equivalent ODE that follows the same marginal probability densities as the above SDE:

$$d\mathbf{x}_t = \left[\frac{\dot{\mu}(t)}{\mu(t)}\mathbf{x}_t - \left(\sigma(t)\dot{\sigma}(t) - \sigma(t)^2 \frac{\dot{\mu}(t)}{\mu(t)} \right) \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t) \right] dt. \quad (3)$$

Consequently, most existing diffusion-based methods widely used in 3D molecular generation, including DDPM, DDIM, VE, FM, and BFN, can be interpreted from a continuous-time perspective, where their diffusion process can equivalently be viewed as an SDE in equation 2 or an ODE in equation 3. Therefore, we can use the ODE formulation as a valuable perspective to analyze the sampling efficiency issue.

Specifically, the diffusion sampling process can be interpreted as a numerical approximation of the backward solution trajectory of the underlying ODE in equation 3². This numerical approximation inherently introduces truncation errors at each step. For example, under Euler’s method, the simplest and widely-used numerical scheme, $x(t - \Delta t)$ is approximated by $x(t) - \frac{dx(t)}{dt}\Delta t$ when solving the ODE backward in time. However, the true value can be derived from the Taylor expansion:

$$x(t - \Delta t) = x(t) - \frac{dx(t)}{dt}\Delta t + \frac{1}{2} \frac{d^2x(t)}{dt^2}(\Delta t)^2 + O((\Delta t)^3), \quad (4)$$

where Δt is a small step size. Thus, the truncation error primarily arises from the second-order term, governed by $\frac{d^2x(t)}{dt^2}$.

To minimize this truncation error, traditional numerical analysis primarily focuses on developing higher-order solvers to estimate higher-order terms, assuming the ODE structure is fixed. EquiFM

²Connections between sampling of baseline methods and first-order ODE sampling are discussed in appendix A.3.

[Song et al., 2024] applied such a technique in 3D molecular generation to speedup the sampling process. However, higher-order solvers face a trade-off between the number of function evaluations (NFE) and accuracy, limiting their ability to achieve even smaller NFEs. As a result, the issue remains partially unsolved. In contrast, we take a different approach: rather than focusing on the sampling algorithm, we reformulate the diffusion process itself to minimize the truncation error, which directly influences the sampling efficiency.

However, our unified formulation of diffusion introduces a key flexibility: the ability to modify the schedule of the process, thereby altering the ODE structure itself to reduce these errors. Therefore, we emphasize that the key to minimizing the truncation error is to reduce the second-order derivative of the process.

3 Straight-Line Diffusion Process

To reduce the truncation error of the sampling process, we aim to design a new diffusion process whose inherent ODE exhibits minimal $\frac{d^2 x(t)}{dt^2}$. By achieving this, even a basic Euler iteration can deliver low truncation error without resorting to complex solvers that require multiple function evaluations. This novel perspective advances the Pareto frontier of efficiency and accuracy in diffusion sampling.

3.1 Derivation of SLDM

Given the intractability of the score function for general data distributions, we begin by examining a simplified case where the initial distribution is a delta distribution. This setup provides a tractable backward ODE, enabling a more straightforward analysis. In this case, the only solution that ensures $\frac{dx}{dt}$ remains constant is for $\sigma(t)$ to be constant and $\mu(t)$ to be a linear function of t . Details can be found in Appendix A.4.

Given the boundary condition that x_0 approximates data distribution and $x_{t=1}$ approximates a known distribution, the only feasible choice is to set $\sigma(t)$ to a small positive constant and $\mu(t) = 1 - t/T$, which satisfies all the schedule constraints noted under equation 2, except $\sigma(0) = 0$. This is because setting $\sigma(0) = 0$ exactly would result in a trivial denoising objective. To avoid this, we instead approximate $\sigma(0)$ by a small value, which keeps the denoising task non-trivial while still satisfying the boundary condition approximately. In practice, we use a value for σ that is two orders of magnitude smaller than the data scale, yielding good results.

Rescaling time to the interval $[0, 1]$, the resulting diffusion process is:

$$\mathbf{x}_t = (1 - t)\mathbf{x}_0 + \sigma\epsilon, t \in [0, 1], \quad (5)$$

which ensures a straight-line trajectory under the delta data distribution assumption.

We then analyze the above diffusion process for general data distribution and demonstrate that a near-linear trajectory could be achieved by setting a small constant value for σ , as shown in the following theorem.

Theorem 3.1 (Near-linear Trajectory of SLDM). *For a general data distribution and the schedule in equation 5, the following inequality holds for each dimension i :*

$$P\left(\left|\frac{d\mathbf{x}_t^{(i)}}{dt} + \frac{\mathbf{x}_t^{(i)}}{1-t}\right| \geq \delta\right) \leq \frac{\sigma^2}{\delta^2(1-t)^2}. \quad (6)$$

When $\sigma \rightarrow 0$, $\frac{d\mathbf{x}_t}{dt} + \frac{\mathbf{x}_t}{1-t}$ converges to zero in probability, where the solution to equation $\frac{d\mathbf{x}_t}{dt} + \frac{\mathbf{x}_t}{1-t} = 0$ is that $\frac{\mathbf{x}_t}{1-t}$ is constant, which corresponds to a linear trajectory.

In other words, for $t \in [0, 1 - \frac{\sigma}{\delta\epsilon}]$, $\left|\frac{d\mathbf{x}_t^{(i)}}{dt} + \frac{\mathbf{x}_t^{(i)}}{1-t}\right| < \delta$ holds with probability at least $1 - \epsilon^2$, indicating that setting σ to a small value ensures that the trajectory remains close to a straight line for most timesteps. This aligns with our empirical observations shown in Figure 6: the overall SLDM trajectory is straighter compared to other diffusion processes, and the sampling trajectory deviates more from a straight line in the initial steps but becomes increasingly linear later.

To further improve the sampling quality, we aim to mitigate the impact of early-stage truncation errors. This is achieved by our stochastic sampling design, which exhibits robustness to such early-stage errors, as supported by empirical evidence provided in Appendix C.6. As a result, despite the

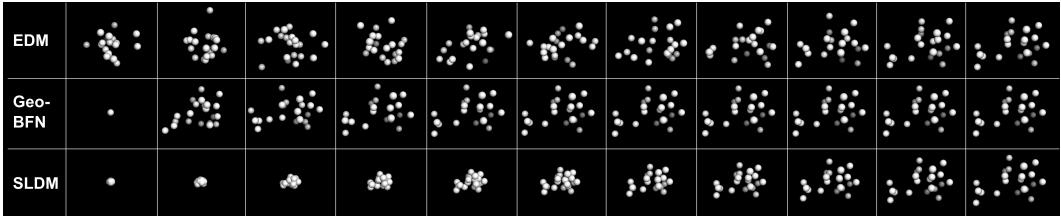


Figure 2: The diffusion process of atomic coordinates in EDM, GeoBFN and SLDM.

challenges in minimizing the early-stage truncation error, our method manage to achieve high-quality samples with few sampling steps.

3.2 Comparisons with Previous Diffusion Processes

Previous studies [Nichol and Dhariwal, 2021, Rombach et al., 2022] have underscored the importance of amortizing the generative difficulty through the diffusion process to improve sample quality. These works typically customize signal-to-noise ratio (SNR) schedules to adapt to varying data characteristics. Notably, as pointed out in Song et al. [2023a], the point cloud representation of molecular structures is far more sensitive to noise compared to images. Consequently, for molecular generation, the SNR needs to be decreased at a much slower pace than in the image generation domain.

Our proposed method aligns remarkably well with these insights. As shown in Figure 3, the SNR in our method decreases significantly slower than that of previous methods. In addition, our schedule results in a smoother and more stable generative process for molecules, as demonstrated in Figure 2. The process unfolds uniformly from the origin, preserving the relative spatial relationships of atomic coordinates in intermediate states and retaining critical chemical information throughout the generative trajectory.

In contrast, traditional diffusion models like EDM, which follows the process schedule of DDPM, also known as VP, involve a noise

variance that increases monotonically during the forward process. This increasing noise quickly disrupts the spatial structure of molecules, rendering much of the diffusion process chemically meaningless. As a result, many computational resources are wasted on steps that attempt to reconstruct signal-less states, leading to a severe reduction in overall model efficiency. Though GeoBFN adopts a low-noise regime for most timesteps, it keeps $\mu \approx 1$ and $\sigma \approx 0$ for over a half of the process, making little changes to the molecular structure. Therefore, GeoBFN shifts the majority of reconstruction difficulty to the earlier stages, creating a severely unbalanced generative process. More illustrations of generation process are provided in Appendix C.7

To sum up, compared to existing diffusion-based models, the new schedule in SLDM aligns better with the inductive biases of molecular data. The approach of distributing the reconstruction difficulty more evenly across the entire diffusion process helps facilitate more effective learning, improving the model’s efficiency and efficacy.

3.3 Sampling Strategy of SLDM

Here we introduce the specific sampling strategy of SLDM. First, we need to derive the reverse process of equation 2 or equivalently equation 3. Notably, previous work [Xue et al., 2024b] (Equation 6) provides a unified form of the reverse-time stochastic differential equation (SDE), which shares the same marginal distribution as the forward process, thereby ensuring that the sampling process has the ability to reconstruct the data distribution. We apply the parameter μ and σ of SLDM to this equation and then discretize it using the Euler-Maruyama method, yielding the following discretized

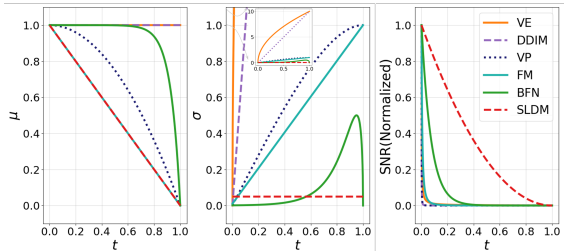


Figure 3: Comparison of schedule parameters across diffusion-based models. Here SNR refers to $(\mu/\sigma)^2$. For SLDM, we set $\sigma = 0.05$, consistent with experiments. The cutoff values $\sigma_{\max}$ and $\sigma_{\min}$ used in other baselines are provided in Appendix A.2.

Algorithm 1 SLDM Training

Input: data $\mathbf{x}_0$ with dimension N , neural network ϕ , constant σ

repeat

 Sample $t \sim U([0, 1])$

 Sample $\epsilon \sim \mathcal{N}(\mathbf{0}, I_N)$

 Compute $\mathbf{x}_t = (1 - t)\mathbf{x}_0 + \sigma\epsilon$

 Minimize $\|\epsilon - \phi(\mathbf{x}_t, t)\|^2$

until converged;

return ϕ

Algorithm 2 SLDM Sampling

Input: trained network ϕ , same N and σ as training, sampling steps T , temperature annealing rate ν

Sample $\mathbf{x} \sim \sigma \cdot \mathcal{N}(\mathbf{0}, I_N)$

for $i = T - 1$ **to** 1 **do**

$\mathbf{x} \leftarrow \frac{T-(i-1)}{T-i} \cdot (\mathbf{x} - \sigma \cdot \phi(\mathbf{x}, t))$

if $i > 1$ **then**

$\mathbf{x} \leftarrow \mathbf{x} + (\frac{i}{T})^\nu \sqrt{2} \cdot \sigma \cdot \epsilon, \epsilon \sim \mathcal{N}(\mathbf{0}, I_N)$

end

end

return $\mathbf{x}$

reverse-time SDE:

$$\mathbf{x}_{t-\Delta t} = \underbrace{\mathbf{x}_t + \frac{\Delta t}{1-t} (\mathbf{x}_t + \sigma^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t))}_{\text{ODE Sampling of equation 3}} + \underbrace{\beta(t) \frac{\Delta t}{1-t} \sigma^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t) + \sqrt{2\beta(t) \frac{\Delta t}{1-t}} \sigma \epsilon}_{\text{Langevin dynamics}}, \quad (7)$$

where $\beta(t)$ is any non-negative bounded function.

Similar to previous works [Xue et al., 2024b, Karras et al., 2022], this sampling can be interpreted as a combination of ODE sampling and Langevin dynamics. The ODE component drives the denoising process along deterministic trajectories, while the Langevin dynamics introduce stochastic corrections. Specifically, we choose $\beta(t) = (1 - t)/\Delta t$, to ensure the expectation of the RHS of equation 7 is $\mathbb{E}[\mathbf{x}_{t-\Delta t} | \mathbf{x}_t]$, as it provides the optimal estimation of $\mathbf{x}_{t-\Delta t}$ in terms of minimizing the mean squared error (MSE). Details are demonstrated in Appendix A.5.

Thus the sampling algorithm becomes:

$$\mathbf{x}_{t-\Delta t} = \frac{1 - (t - \Delta t)}{1 - t} (\mathbf{x}_t + \sigma^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)) + \sqrt{2} \sigma \epsilon. \quad (8)$$

Note that $1 - t$ appears in the denominator of the above equation. To avoid potential division by zero, we choose to skip the $t = 1$ sampling step and find that it works well. This might be because the early sampling steps are less critical to the final result, likely due to Langevin dynamics not strictly requiring a specific prior, which imparts an inherent error-correction capability to the algorithm. This property is further supported by our observation that the sampling process remains relatively robust even when initialized from an invalid distribution, as demonstrated in Appendix C.6.

The above sampling algorithm provides a principled way to approximately sample from the input data distribution, but its practical application in molecular datasets presents unique challenges. In particular, the molecular stability of the dataset is not guaranteed to be 100%. To enhance stability and mitigate the impact of data noise, low-temperature sampling can be employed to generate samples that preserve essential properties of the input data. For Langevin dynamics, the sampling temperature can be controlled by scaling the stochastic term with a constant. However, prior work has demonstrated that such temperature control in diffusion model sampling often fails to effectively balance diversity and fidelity [Dhariwal and Nichol, 2021]. Additionally, as observed in our toy dataset experiments, conventional low-temperature sampling suffers from mode collapse, failing to fully cover all high-density regions of the target distribution.

To address these limitations, we propose a time-annealing temperature schedule:

$$\mathbf{x}_{t-\Delta t} = \frac{1 - (t - \Delta t)}{1 - t} (\mathbf{x}_t + \sigma^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)) + t^\nu \sqrt{2} \sigma \epsilon, \quad (9)$$

where ν controls the decay rate of temperature over time. This approach allows for higher stochasticity during the initial stages of sampling, enabling the exploration of distant probability density maxima. As the temperature decreases in later stages, the process converges to the probability density maxima. Further theoretical explanations and toy data illustrations are provided in Appendix A.6. With the help of the temperature control, our sampling strategy can reduce the impact of noise in the data, and generate molecules with enhanced stability. The complete training and sampling procedure of

straight-line diffusion are given in algorithm 1 and 2. We further discussed the relation to relevant techniques used in general generative approaches in Related Work (Appendix D.2), including noise scheduling, flow-based methods, other straight-line trajectory models, etc.

Table 1: Unconditional molecular generation results on QM9 and GEOM-Drugs datasets. For all diffusion-based models, T denotes sampling steps. Metrics are calculated with 10000 samples generated from each model. Higher values indicate better performance.

#Metrics	QM9				GEOM-Drugs	
	Atom sta(%)	Mol sta(%)	Valid(%)	V*U(%)	Atom sta(%)	Valid(%)
Data	99.0	95.2	97.7	97.7	86.5	99.9
E-NF	85.0	4.9	40.2	39.4	-	-
G-Schnet	95.7	68.1	85.5	80.3	-	-
EDM (T=1000)	98.7	82.0	91.9	90.7	81.3	92.6
GDM (T=1000)	97.6	71.6	90.4	89.5	77.7	91.8
EDM-Bridge (T=1000)	98.8	84.6	92.0	90.7	82.4	92.8
GeoLDM (T=1000)	98.9	89.4	93.8	92.7	84.4	99.3
EquiFM (T=200)	98.9	88.3	94.7	<u>93.5</u>	84.1	98.9
GeoBFN (T=1000)	99.08	90.87	95.31	92.96	85.60	92.08
END (T=1000)	98.9	89.1	94.8	92.6	87.0	89.2
SLDM (T=1000)	99.43	95.42	97.07	90.42	<u>88.30</u>	99.95
SLDM (T=50)	<u>99.30</u>	<u>93.37</u>	<u>96.24</u>	93.63	89.03	<u>99.57</u>

4 Experiment

To validate the advantages of our method in molecular generation, we evaluate its overall performance and sampling efficiency in both unconditional and conditional generation scenarios.

4.1 Setup

Datasets We evaluate our model using two widely adopted datasets for unconditional molecular generation, with all dataset splitting strictly following baseline settings [Hooeboom et al., 2022, Song et al., 2024, 2023a]. QM9 [Ruddigkeit et al., 2012, Ramakrishnan et al., 2014] contains approximately 134,000 small organic molecules with up to nine heavy atoms. It is split into training (100K), validation (18K), and test (13K) sets. GEOM-Drugs [Axelrod and Gomez-Bombarelli, 2022] focuses on drug-like molecules, comprising around 430,000 molecules with sizes ranging up to 181 atoms and an average of 44.4 atoms per molecule. Its larger size and greater diversity make it more challenging for generative models. The dataset is randomly divided into training, validation, and test sets using an 8:1:1 ratio.

For conditional molecular generation, we adopt the QM9 dataset with the same setup as prior work [Hooeboom et al., 2022, Song et al., 2024, 2023a]. The QM9 training partition is split into two halves, each containing 50K samples. Specifically, the QM9 training set is divided into two halves of 50K samples each. The first half is used to train a classifier for ground-truth property labels, while the second half is used to train the conditional generative model.

Implementation The molecule is represented by atomic coordinates and atom types, $z = (\mathbf{x}, \mathbf{h})$, where $\mathbf{x} \in \mathbb{R}^{3M}$ denotes the atomic coordinates, M is the number of atoms, and $\mathbf{h}$ encodes the atom type information. Thus for molecule generation, the model needs to generate both coordinates and atom types. A key requirement for the molecular coordinates is ensuring the SE(3) invariance of the probability distribution, meaning that the probability of generating two molecular conformations should be identical if they only differ by translation or rotation. Following Xu et al. [2022], Hooeboom et al. [2022], we tailor the SLDM algorithm to satisfy equivariance. Specifically, to ensure translation invariance, we constrain the coordinates to a zero Center of Mass (CoM) space, while rotation invariance is preserved by employing equivariant neural networks to predict the noise. To ensure a fair comparison of generative algorithms, we use EGNN [Satorras et al., 2021] as the backbone model, consistent with the baseline methods [Garcia Satorras et al., 2021, Hooeboom et al., 2022, Wu et al., 2022, Xu et al., 2023, Song et al., 2024, 2023a]. We prove that the generated data distribution satisfies SE(3) invariance, as provided in Appendix A.7. For atom types, we follow

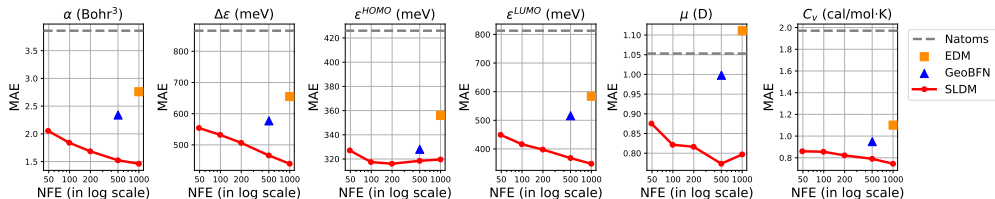


Figure 4: Comparison of performance in conditional generation ($\downarrow$), with respect to the number of function evaluations (NFE) during the sampling process.

UniGEM [Feng et al., 2024] to predict atom types based on the generated coordinates. The SLDM algorithms tailored for molecular generation are provided in Appendix B. Hyperparameters are summarized in Appendix E. An introduction to the baseline models is included in Section D.1.

Baselines To ensure a fair comparison, we select competitive baselines that also focus on generative modeling and have the same neural architectures (e.g., EGNN) and input information (e.g., coordinates and types). We also acknowledge that some studies propose strategies orthogonal to generative algorithms, as discussed in Related Work (Appendix D.1). However, these strategies are not directly comparable to our work, as they do not aim to replace or improve the generative model itself but rather focus on improving network expressivity and augmenting input information. We believe that substituting their generative algorithms with ours could lead to improvements, and exploring this will be part of the future work.

For conditional molecular generation, Two additional basic baselines: Random and N_{atoms} follow EDM [Hoogetboom et al., 2022]. The Random baseline involves shuffling property labels in the training data and evaluating the property classifier on the shuffled data. The N_{atoms} baseline uses the property classifier network to predict the property based solely on the number of atoms in the molecule.

Metrics It is important to note that evaluation protocols differ across the literature, as discussed in section D.1. To ensure consistency, we strictly adhere to the evaluation methods used in our baselines. We sample 10,000 molecules and predict the bond type (single, double, triple, or non-existent) based on the distances between each pair of atoms, as in Hoogetboom et al. [2022]. Atom stability is computed as the proportion of atoms with correct valency, and molecule stability is the fraction of generated molecules in which all atoms are stable. Validity is evaluated using RDKit by checking whether the 3D molecular structures can be successfully parsed into SMILES format. Uniqueness is determined as the ratio of distinct molecules among all valid samples, indicating the diversity of generated molecules. $V*U$ means the ratio of valid and unique molecules.

4.2 Unconditional Molecular Generation

Unconditional generation assesses the model’s ability to learn the underlying molecular data distribution, aiming to generate chemically valid and structurally diverse molecules. The results, summarized in Table 1, show that our method significantly outperforms the baselines across both quality and diversity metrics for the generated molecules. Of particular note is the substantial improvement in the stability of the generated molecules, indicating that our method better satisfies chemical constraints. This validates our hypothesis that our low-noise dynamic is well-suited for generating molecular data. A visualization of generated molecules is provided in Appendix C.7.

Additionally, our approach can achieve superior results with significantly fewer generative steps. A detailed comparison, presented in Figure 1, demonstrates that SLDM achieves 100 $\times$ faster sampling than the baseline EDM and around 70 $\times$ speedup compared to GeoBFN and EquiFM. Since the methods use the same network architecture, NFE can reflect practical sampling time. It is important to note that EquiFM utilizes a variety of advanced and efficient ODE solvers, and the results we present correspond to the best performance reported in their paper for different step sizes. Besides, EDM and EDM-Bridge results are from the EDM-Bridge paper while GeoBFN results are reproduced from the official codebase. This observation underscores the advantages of our straight-line diffusion process: while sophisticated solvers play an important role, the diffusion process design may offer more fundamental improvements. Furthermore, our handcrafted sampling strategies can be seamlessly combined with modern sampling techniques, such as optimal time discretization and advanced solvers, which we leave for future exploration.

Table 2: Conditional generation on QM9 dataset, evaluated by MAE($\downarrow$) between the property condition and the properties of generated molecules predicted by a pretrained EGNN classifier. SLDM uses $T = 1000$.

Property	α	$\Delta\epsilon$	ϵ^{HOMO}	ϵ^{LUMO}	μ	C_v
Units	Bohr ³	meV	meV	meV	D	$\frac{\text{cal}}{\text{mol K}}$
Data	0.10	64	39	36	0.043	0.040
Random	9.01	1470	645	1457	1.616	6.857
N_{atoms}	3.86	866	426	813	1.053	1.971
EDM	2.76	655	356	584	1.111	1.101
GeoLDM	2.37	587	340	522	1.108	1.025
EquiFM	2.41	591	337	530	1.106	1.033
GeoBFN	2.34	577	328	516	0.998	0.949
SLDM	1.46	440	320	348	0.797	0.745

Table 3: Comparison of generative methods within the UniGEM framework, marked by $_U$, for unconditional generation on the QM9 dataset using small sampling steps T . Higher values indicate better performance.

T	Model	Atom sta(%)	Mol sta(%)	Valid(%)	V*U(%)
50	EDM $_U$	98.55	85.73	93.29	91.78
50	GeoBFN $_U$	98.28	87.16	93.97	91.82
50	SLDM	99.30	93.37	96.24	93.63
30	EDM $_U$	97.58	78.75	89.39	87.96
30	GeoBFN $_U$	96.74	81.05	90.93	87.47
30	SLDM	99.30	93.02	96.20	92.76

4.3 Conditional Molecule Generation

Conditional generation evaluates the model’s capability to produce molecules with desired properties. Following the baseline approaches, we incorporate property values as additional inputs during training and sample them from a prior distribution during inference. Descriptions of baselines and evaluations are provided in Appendix E.

The results in Table 2 demonstrate that our method consistently outperforms baseline models across all metrics. Moreover, as illustrated in Figure 4, our approach achieves a 20-fold acceleration over previous state-of-the-art methods. This result showcases its potential for application in a wide range of controllable generation scenarios.

4.4 Ablation Study

Previous approaches differ in their methods for atom type generation. EDM represents atom types as one-hot vectors, generating them simultaneously with coordinates through diffusion. In contrast, GeoBFN treats atom types as atomic numbers and generates them by the BFN algorithm for discretized data, which incorporates a binning technique to convert continuous probabilities into discrete probabilities. UniGEM, on the other hand, generates only coordinates via diffusion and predicts atom types based on the generated coordinates. We adopt the UniGEM framework due to its superior performance.

To compare generative methods without the influence of atom type generation differences, we integrated EDM and GeoBFN coordinate generation algorithms into the UniGEM framework and compared them with our approach. The results, shown in Table 3 with sampling steps $T = 50$ and $T = 30$, demonstrate that our method exhibits a clear advantage in smaller steps, confirming the efficiency benefits of our generative algorithm. A stability comparison w.r.t various sampling steps is provided in Figure 7. Besides, we also conduct an extensive comparison of sampling efficiency of SLDM under joint diffusion framework, i.e. without UniGEM framework with GeoBFN and EDM in Appendix C.3 and an ablation study about temperature control in Appendix C.5.

4.5 Evaluation on Toy Dataset

We provide a toy dataset experiment in Appendix C.4, demonstrating the superior generative capabilities of our approach in faithfully modeling complex data distributions. These results suggest that SLDM holds promise for generalization to broader application domains.

5 Conclusion

This paper proposes the Straight-Line Diffusion Model (SLDM), a novel generative method that ensures a near-linear diffusion trajectory, effectively reducing truncation error during sampling.

The proposed process schedule naturally aligns with the characteristics of point cloud molecular data and effectively balances the generative difficulty. As a result, our method achieves significant improvements in both sampling efficiency and quality, as demonstrated in both unconditional and conditional generation settings, paving the way for large-scale, controllable molecular generation in practice.

Several challenges remain open for future investigation. On the theoretical side, refining the schedule to fully satisfy the boundary conditions, establishing principled guidelines for selecting the noise level σ , such as analyzing its impact on training stability and sampling accuracy, and deriving optimal discretization schemes could strengthen the theoretical foundations of our approach and further accelerate sampling. On the practical side, applying our method to diverse controllable molecular generation tasks and extending it to broader domains presents exciting opportunities for future advancements.

Acknowledgment

This work is supported by Beijing Academy of Artificial Intelligence and Beijing Frontier Research Center for Biological Structure Fundings.

References

- Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.
- Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators, 2024.
- Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, pages 1–3, 2024.
- Emiel Hoogeboom, Victor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pages 8867–8887. PMLR, 2022.
- Lemeng Wu, Chengyue Gong, Xingchao Liu, Mao Ye, and Qiang Liu. Diffusion-based molecule generation with informative prior bridges. *Advances in Neural Information Processing Systems*, 35:36533–36545, 2022.
- Minkai Xu, Alexander S Powers, Ron O Dror, Stefano Ermon, and Jure Leskovec. Geometric latent diffusion models for 3d molecule generation. In *International Conference on Machine Learning*, pages 38592–38610. PMLR, 2023.
- Yuxuan Song, Jingjing Gong, Minkai Xu, Ziyao Cao, Yanyan Lan, Stefano Ermon, Hao Zhou, and Wei-Ying Ma. Equivariant flow matching with hybrid probability transport for 3d molecule generation. *Advances in Neural Information Processing Systems*, 36, 2024.
- Yuxuan Song, Jingjing Gong, Hao Zhou, Mingyue Zheng, Jingjing Liu, and Wei-Ying Ma. Unified generative modeling of 3d molecules with bayesian flow networks. In *The Twelfth International Conference on Learning Representations*, 2023a.
- Alexander Tong, Nikolay Malkin, Guillaume Hugué, Yanlei Zhang, Jarrod Rector-Brooks, Kilian FATRAS, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. In *ICML Workshop on New Frontiers in Learning, Control, and Dynamical Systems*, 2023. URL <https://openreview.net/forum?id=HgDwiZrpVq>.

- Alex Graves, Rupesh Kumar Srivastava, Timothy Atkinson, and Faustino Gomez. Bayesian flow networks. *arXiv preprint arXiv:2308.07037*, 2023.
- Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.
- Kaiwen Xue, Yuhao Zhou, Shen Nie, Xu Min, Xiaolu Zhang, JUN ZHOU, and Chongxuan Li. Unifying bayesian flow networks and diffusion models through stochastic differential equations. In *Forty-first International Conference on Machine Learning*, 2024a. URL <https://openreview.net/forum?id=1jHiq640y1>.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. 2021a. URL <https://openreview.net/forum?id=St1giarCHLP>.
- Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. URL <https://openreview.net/forum?id=PXTIG12RRHS>.
- Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=PqvMRDCJT9t>.
- Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021.
- Shuchen Xue, Mingyang Yi, Weijian Luo, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhi-Ming Ma. Sa-solver: Stochastic adams solver for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024b.
- Lars Ruddigkeit, Ruud Van Deursen, Lorenz C Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *Journal of chemical information and modeling*, 52(11):2864–2875, 2012.
- Raghuathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.
- Simon Axelrod and Rafael Gomez-Bombarelli. Geom, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9(1):185, 2022.
- Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.
- Victor Garcia Satorras, Emiel Hooeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.
- Victor Garcia Satorras, Emiel Hooeboom, Fabian Fuchs, Ingmar Posner, and Max Welling. E (n) equivariant normalizing flows. *Advances in Neural Information Processing Systems*, 34:4181–4192, 2021.
- Shikun Feng, Yuyan Ni, Yan Lu, Zhi-Ming Ma, Wei-Ying Ma, and Yanyan Lan. Unigem: A unified approach to generation and property prediction for molecules. *arXiv preprint arXiv:2410.10516*, 2024.
- Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=zyLVMgsZOU_.
- Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

- Yuyan Ni, Shikun Feng, Wei-Ying Ma, Zhi-Ming Ma, and Yanyan Lan. Revisiting sampling strategies for molecular generation. In *ICML 2025 Generative AI and Biology (GenBio) Workshop*, 2025. URL <https://openreview.net/forum?id=ID6GjrJaJx>.
- Zihan Zhou, Xiaoxue Wang, and Tianshu Yu. Generating physical dynamics under priors. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=eNjXcP6COH>.
- Niklas Gebauer, Michael Gastegger, and Kristof Schütt. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. *Advances in neural information processing systems*, 32, 2019.
- Youzhi Luo and Shuiwang Ji. An autoregressive flow model for 3d molecular geometry generation from scratch. In *International conference on learning representations (ICLR)*, 2022.
- Ameya Daigavane, Song Eun Kim, Mario Geiger, and Tess Smidt. Symphony: Symmetry-equivariant point-centered spherical harmonics for 3d molecule generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=MIEnYt1Gyv>.
- Changyou Chen, Chunyuan Li, Liqun Chen, Wenlin Wang, Yunchen Pu, and Lawrence Carin Duke. Continuous-time flows for efficient inference and density estimation. In *International Conference on Machine Learning*, pages 824–833. PMLR, 2018.
- Jonas Köhler, Leon Klein, and Frank Noé. Equivariant flows: exact likelihood generative learning for symmetric densities. In *International conference on machine learning*, pages 5361–5370. PMLR, 2020.
- François Cornet, Grigory Bartosh, Mikkel Schmidt, and Christian Andersson Naesseth. Equivariant neural diffusion for molecule generation. *Advances in Neural Information Processing Systems*, 37: 49429–49460, 2024.
- Lei Huang, Hengtong Zhang, Tingyang Xu, and Ka-Chun Wong. Mdm: Molecular diffusion model for 3d molecule generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 5105–5112, 2023a.
- Zian Li, Cai Zhou, Xiyuan Wang, Xingang Peng, and Muhan Zhang. Geometric representation condition improves equivariant molecule generation. *arXiv preprint arXiv:2410.03655*, 2024.
- Chenqing Hua, Sitao Luan, Minkai Xu, Zhitao Ying, Jie Fu, Stefano Ermon, and Doina Precup. Mudiff: Unified diffusion for complete molecule generation. In *Learning on Graphs Conference*, pages 33–1. PMLR, 2024.
- Tuan Le, Julian Cremer, Frank Noé, Djork-Arné Clevert, and Kristof Schütt. Navigating the design space of equivariant diffusion-based generative models for de novo 3d molecule generation. 2024a.
- Ross Irwin, Alessandro Tibo, Jon Paul Janet, and Simon Olsson. Efficient 3d molecular generation with flow matching and scale optimal transport. In *ICML 2024 AI for Science Workshop*, 2024. URL <https://openreview.net/forum?id=CxAjGjdkqu>.
- Han Huang, Leilei Sun, Bowen Du, and Weifeng Lv. Learning joint 2d & 3d diffusion models for complete molecule generation. *arXiv preprint arXiv:2305.12347*, 2023b.
- Giangiacomo Mercatali, Yogesh Verma, Andre Freitas, and Vikas Garg. Diffusion twigs with loop guidance for conditional graph generation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 137741–137767. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/f90338be8ad676ee02cc32d239fc40e7-Paper-Conference.pdf.
- Xingang Peng, Jiaqi Guan, Qiang Liu, and Jianzhu Ma. Moldiff: Addressing the atom-bond inconsistency problem in 3d molecule diffusion generation. *International conference on machine learning*, 2023.

- Clement Vignac, Nagham Osman, Laura Toni, and Pascal Frossard. Midi: Mixed graph and 3d denoising diffusion for molecule generation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 560–576. Springer, 2023.
- Tuan Le, Julian Cremer, Frank Noé, Djork-Arné Clevert, and Kristof Schütt. Navigating the design space of equivariant diffusion-based generative models for de novo 3d molecule generation. *International Conference on Learning Representations*, 2024b.
- Shuchen Xue, Zhaoqiang Liu, Fei Chen, Shifeng Zhang, Tianyang Hu, Enze Xie, and Zhenguo Li. Accelerating diffusion sampling with optimized time steps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8292–8301, 2024c.
- Lijiang Li, Huixia Li, Xiawu Zheng, Jie Wu, Xuefeng Xiao, Rui Wang, Min Zheng, Xin Pan, Fei Chao, and Rongrong Ji. Autodiffusion: Training-free optimization of time steps and architectures for automated diffusion model acceleration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7105–7114, 2023.
- 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. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.
- Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.
- Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=TIIdIXIpzhoI>.
- Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. 2023b.
- Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.
- Nikita Maksimovich Kornilov, Petr Mokrov, Alexander Gasnikov, and Alexander Korotin. Optimal flow matching: Learning straight trajectories in just one step. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL <https://openreview.net/forum?id=kqmucDKVcU>.
- Grigory Bartosh, Dmitry P Vetrov, and Christian Andersson Naesseth. Neural flow diffusion models: Learnable forward process for improved diffusion modelling. *Advances in Neural Information Processing Systems*, 37:73952–73985, 2024.

NeurIPS Paper Checklist

1. Claims

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

Answer: [\[Yes\]](#)

Justification: The claims in the abstract and introduction are well-aligned with the paper's contributions and scope. The abstract clearly presents the main methodological innovation — the Straight-Line Diffusion Model (SLDM) — and its motivation in addressing the inefficiency of standard diffusion sampling. It also articulates the key theoretical insight (the straight-line trajectory, alignment with noise sensitivity and uniform reconstruction effort) and the empirical outcome (improved learning efficiency and efficacy). These claims are supported by detailed analyses and experimental results in the main body.

Guidelines:

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

2. Limitations

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

Answer: [\[Yes\]](#)

Justification: We discuss the limitations through future work in the conclusion session.

Guidelines:

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

3. Theory assumptions and proofs

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

Answer: [Yes]

Justification: We provide comprehensive proof and assumptions of the theoretical analysis in Appendix A.

Guidelines:

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

4. Experimental result reproducibility

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

Answer: [Yes]

Justification: We provide pseudocode and the experiment settings in algorithm 1, algorithm 2 and Appendix B. We provide the experiment settings in Section 4.1 and Appendix E.

Guidelines:

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

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

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

Answer: [Yes]

Justification: The code is open-sourced in an anonymous link, with sufficient instructions to reproduce the main experimental results. The link is in the footnote of the abstract.

Guidelines:

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

6. Experimental setting/details

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

Answer: [Yes]

Justification: As our paper propose a new generative algorithm, for fair comparison, the data splits follow previous baselines and model hyperparameters and optimizer follow UniGEM. We discuss how to choose the constant variance parameter σ in section 3.1, we discuss the theoretical requirements and provide an empirical guideline.

Guidelines:

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

7. Experiment statistical significance

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

Answer: [No]

Justification: Following the experimental protocol of the baselines (EDM, GeoBFN, MolFM, etc.), we did not perform repeated runs. However, since the evaluation metrics are computed over 10,000 generated samples for both unconditional and conditional molecule generation experiments, the results are already statistically stable and exhibit minimal variance.

Guidelines:

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

8. Experiments compute resources

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

Answer: [Yes]

Justification: We provide computer resources information in appendix E.

Guidelines:

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

9. Code of ethics

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

Answer: [Yes]

Justification: The paper conforms the Code of Ethics.

Guidelines:

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

10. Broader impacts

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

Answer: [NA]

Justification: This paper belongs to foundational research and not tied to particular deployments.

Guidelines:

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

11. Safeguards

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

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

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

12. Licenses for existing assets

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

Answer: [Yes]

Justification: All assets are cited.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.

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

13. **New assets**

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

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

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

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

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

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

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

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

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

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA 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 (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

Appendix

Table of Contents

A	Supplementary Theoretical Results	22
A.1	The Derivation of A Unified Formulation of Diffusion Models	22
A.2	A Summary of Previous Process Schedules	23
A.3	Connection Between Sampling of Baseline Methods and First-Order ODE Sampling	24
A.4	Diffusion Schedule with Straight-line Trajectory	24
A.5	Supplementary Proof for Sampling	26
A.6	Temperature Control	27
A.7	Modeling Invariant Probability Density for 3D Coordinate Generation	27
B	Molecular Generation Algorithms	29
C	Supplementary Illustrations and Results	29
C.1	Sampling Trajectory of Diffusion-based Models	29
C.2	Sampling Efficiency Comparisons under UniGEM Framework	30
C.3	Sampling Efficiency Under Joint Diffusion Framework	30
C.4	Toy Data Results	31
C.5	Ablation Study for Temperature Control	32
C.6	Robustness to Different Prior Distributions	33
C.7	Visualizations of Generated Molecules and Sampling Process	34
D	Related Work	34
D.1	An overview of 3D Molecular Generation	34
D.2	Related Diffusion-Based Studies and Key Differences with SLDM	35
E	Implementation Detail	36

A Supplementary Theoretical Results

A.1 The Derivation of A Unified Formulation of Diffusion Models

The stochastic process of the diffusion-based model can be formulated in general as a linear stochastic differential equation (SDE):

$$dX_t = f(t)X_t dt + g(t)dW_t, \quad (10)$$

where X_t characterizes the noise corrupting process of the data in a diffusion model, $f(t) \leq 0$ and $g(t) \geq 0$ are measurable functions defined on the interval $[0, \infty)$, W_t represents a standard Wiener process. The process X_t is a time rescaled Ornstein-Uhlenbeck process whose law converges exponentially fast to the standard Gaussian distribution [Chen et al., 2023].

Under the assumption that all the relevant integrals exist, the solution of the above SDE is given by:

$$X_t = X_0 \cdot \exp\left(\int_0^t f(\xi)d\xi\right) + \int_0^t \exp\left(\int_s^t f(\xi)d\xi\right)g(s)dW_s, t \in [0, \infty) \quad (11)$$

A quick proof is as follows:

Proof. By Itô's formula, and applying equation 10:

$$\begin{aligned} d(X_t \cdot \exp(-\int_0^t f(\xi)d\xi)) &= (dX_t - X_t f(t)dt) \cdot \exp(-\int_0^t f(\xi)d\xi) \\ &= \exp(-\int_0^t f(\xi)d\xi) \cdot g(t)dW_t \end{aligned} \quad (12)$$

Next, we integrate both sides and obtain:

$$X_t \cdot \exp(-\int_0^t f(\xi)d\xi) - X_0 = \int_0^t \exp(-\int_0^s f(\xi)d\xi) \cdot g(s)dW_s. \quad (13)$$

Finally, we multiply both sides of the equation by $\exp(\int_0^t f(\xi)d\xi)$ and rearrange the terms to obtain equation 11. $\square$

By the property that the stochastic integral with respect to a Wiener process, X_t is a normal random variable. The mean and variance of X_t are calculated as follows:

$$\mathbb{E}[X_t] = X_0 \cdot \exp\left(\int_0^t f(\xi)d\xi\right) \quad (14)$$

$$\mathbb{COV}[X_t] = E[(\int_0^t \exp(\int_s^t f(\xi)d\xi)g(s)dW_s)^2] \mathbf{I} = \int_0^t (\exp(\int_s^t f(\xi)d\xi)g(s))^2 ds \mathbf{I}. \quad (15)$$

The covariance computation uses the Itô isometry property.

To fit the process X_t to the general form

$$x_t = \mu(t)x_0 + \sigma(t)\epsilon, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad (16)$$

we compare it with our solution in equation 11, and identify that

$$\mu(t) = \exp\left(\int_0^t f(\xi)d\xi\right), \quad \sigma(t) = \mu(t) \sqrt{\int_0^t g(s)^2 / \mu(s)^2 ds}. \quad (17)$$

Furthermore, we can express $f(t)$ and $g(t)$ in terms of $\mu(t)$ and $\sigma(t)$:

$$\begin{aligned} f(t) &= \frac{d(\ln \mu(t))}{dt} = \frac{\dot{\mu}(t)}{\mu(t)}, \\ g(t)^2 &= 2\sigma(t)\mu(t) \frac{d(\sigma(t)/\mu(t))}{dt} = 2\sigma(t)\dot{\sigma}(t) - 2\sigma(t)^2 \frac{\dot{\mu}(t)}{\mu(t)}. \end{aligned} \quad (18)$$

$\mu(t)$ and $\sigma(t)$ satisfy the initial conditions $\mu(0) = 1$, $\sigma(0) = 0$ and additional conditions: $\mu(t) \geq 0$ is monotone non-increasing, $\sigma(t) \geq 0$, $\frac{d(\sigma(t)/\mu(t))}{dt} \geq 0$, i.e. $\sigma(t)/\mu(t)$ is monotone non-decreasing.

Substituting the expressions for $f(t)$ and $g(t)$ back into the original SDE equation 10, we obtain:

$$dX_t = \frac{\dot{\mu}(t)}{\mu(t)} X_t dt + \sqrt{2\sigma(t)\dot{\sigma}(t) - 2\sigma(t)^2 \frac{\dot{\mu}(t)}{\mu(t)}} dW_t, \quad (19)$$

This result is conceptually equivalent to that presented in Appendix B of Karras et al. [2022]. But they adopt a different definition of the schedule compared to equation 16, which leads to a different outcome compared to equation 19.

A.2 A Summary of Previous Process Schedules

This section provides a comprehensive overview of several widely adopted diffusion-based models and their corresponding process schedules.

The Denoising Diffusion Probabilistic Model (DDPM) [Ho et al., 2020] is characterized by a process schedule that exhibits the **Variance Preserving (VP)** property, which can be expressed as: $\mu(t)^2 + \sigma(t)^2 = 1$. This formulation ensures the preservation of variance across timesteps, assuming the data has unit variance. In the domain of 3D molecular unconditional generation, the Equivariant Diffusion Model (EDM) [Hooeboom et al., 2022] employs a schedule closely resembling the cosine noise schedule introduced by Nichol and Dhariwal [2021], albeit with a simplified notation:

$$\mathbf{x}_t = (1 - t^2)\mathbf{x}_0 + \sqrt{1 - (1 - t^2)^2}\epsilon, t \in [0, 1], \quad (20)$$

The **Variance Exploding (VE)** schedule, initially proposed in the context of Denoising Score Matching [Song and Ermon, 2019], can also be categorized as a denoising diffusion model with a distinct process schedule [Karras et al., 2022, Song et al., 2021b], which is given by:

$$\mathbf{x}_t = \mathbf{x}_0 + \sqrt{t}\epsilon, t \in [0, T_{max}]. \quad (21)$$

By rescaling time, we derive the following expression:

$$\mathbf{x}_t = \mathbf{x}_0 + \sqrt{t}\sigma_{max}\epsilon, t \in [0, 1], \quad (22)$$

where σ_{max} needs to be sufficiently large to ensure that $\mathbf{x}_1$ approximates a uniform distribution. For illustrative clarity, we set $\sigma_{max} = 10$ for Figure 3 and $\sigma_{max} = 20$ for Figure 6.

Denoising Diffusion Implicit Model (DDIM) offers an accelerated sampling process for diffusion models. As proved by Karras et al. [2022], DDIM employs the following schedule:

$$\mathbf{x}_t = \mathbf{x}_0 + t\epsilon, t \in [0, T_{max}]. \quad (23)$$

By rescaling time, we derive the following expression:

$$\mathbf{x}_t = \mathbf{x}_0 + t\sigma_{max}\epsilon, t \in [0, 1], \quad (24)$$

For illustrative clarity, we set $\sigma_{max} = 10$ for Figure 3 and Figure 6.

Flow Matching (FM) [Lipman et al., 2023] proposes a linear interpolation between the data distribution and a standard Gaussian, with a neural network learning the corresponding vector field. This noise-adding process can also be viewed as defining a diffusion process schedule, given by:

$$\mathbf{x}_t = (1 - t)\mathbf{x}_0 + (t + (1 - t)\sigma_{min})\epsilon, t \in [0, 1], \quad (25)$$

where σ_{min} needs to be set sufficiently small to ensure that $\mathbf{x}_t$ aligns with the data distribution at $t = 0$. Besides, smaller values of σ_{min} have been reported to yield better results for FM [Tong et al., 2023]. Accordingly, we set $\sigma_{min} = 0.001$ for both Figure 3 and Figure 6.

Bayesian Flow Networks (BFN) [Graves et al., 2023] is a generative model based on Bayesian inference that accommodates both continuous and discrete variables. For continuous variables, the data is parameterized by Gaussian distributions. In this scenario, the generative algorithms can be interpreted as a denoising diffusion model [Xue et al., 2024a] with a process schedule given by:

$$\mathbf{x}_t = (1 - \sigma_{min}^{2t})\mathbf{x}_0 + \sqrt{(1 - \sigma_{min}^{2t})\sigma_{min}^{2t}}\epsilon, t \in [0, 1], \quad (26)$$

where σ_{min} needs to be set small to satisfy $\mathbf{x}_t$ align with the data distribution when $t = 0$. Following the settings in [Song et al., 2023a], we set $\sigma_{min} = 0.001$ for Figure 3 and Figure 6.

A.3 Connection Between Sampling of Baseline Methods and First-Order ODE Sampling

In section 2, we analyzed the error of first-order ODE discretization methods and claim that the baseline sampling methods primarily rely on first-order ODE discretization. In this section, we provide further clarification on this matter: EquiFM [Song et al., 2024] directly utilizes ODE-based sampling which includes first-order ODE discretization. For methods like EDM [Hooeboom et al., 2022] and GeoBFN [Song et al., 2023a], while they resemble first-order methods due to requiring only a single function evaluation per iteration, they incorporate random sampling. This distinction necessitates an explanation of how their random sampling processes relate to ODEs. Specifically, EDM uses the same sampling method as DDPM, which is proved as a first-order discretization to the reverse-time SDE of DDPM in Appendix E of Song et al. [2021b]. Similarly, GeoBFN adopts the same sampling method as BFN [Graves et al., 2023], which is proved as a first-order discretization to the reverse-time SDE of BFN in Proposition 4.2 in Xue et al. [2024a]. Please note that both of the reverse-time SDE can be decomposed as an ODE and Langevin dynamics:

$$\begin{aligned} d\mathbf{x}_t &= [f(t)\mathbf{x}_t - g^2(t)\nabla_{\mathbf{x}} \log p(\mathbf{x}_t)]dt + g(t)d\mathbf{w}_t \\ &= \underbrace{[f(t)\mathbf{x}_t - \frac{g^2(t)}{2}\nabla_{\mathbf{x}} \log p(\mathbf{x}_t)]dt}_{\text{reverse time ODE in equation 3}} - \underbrace{\frac{g^2(t)}{2}\nabla_{\mathbf{x}} \log p(\mathbf{x}_t)dt + g(t)d\mathbf{w}_t}_{\text{Langevin dynamics}} \end{aligned} \quad (27)$$

where $f(t) = \frac{\dot{\mu}(t)}{\mu(t)}$ and $g(t)^2 = 2\sigma(t)\dot{\sigma}(t) - 2\sigma(t)^2\frac{\dot{\mu}(t)}{\mu(t)}$. Thus the sampling processes of EDM and GeoBFN can be effectively approximated as a first-order discretization of an ODE augmented by Langevin dynamics. By isolating and analyzing the discretization error of this ODE component, we gain valuable insights into the limitations of methods with first-order discretization, including baseline approaches in molecular generation such as EDM, GeoBFN, and EquiFM.

A.4 Diffusion Schedule with Straight-line Trajectory

As illustrated in section 3.1, we aim to reduce the truncation error during sampling by minimizing the second-order derivative of the trajectory. To this end, we first consider a simple case where the initial distribution is a delta distribution $\mathbf{x}_0 \sim \delta_{\mathbf{a}}(\mathbf{x})$, $\mathbf{a} \in \mathbb{R}^N$. In this scenario, $\mathbf{x}_t$ follows a normal distribution $\mathcal{N}(\mu(t)\mathbf{a}, \sigma(t)^2\mathbf{I}_N)$ according to equation 1. The score function is then tractable as $\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t) = -\frac{\mathbf{x} - \mu(t)\mathbf{a}}{\sigma(t)^2}$. Substituting this into equation 3 gives:

$$\frac{d\mathbf{x}}{dt} = \frac{\dot{\sigma}(t)}{\sigma(t)}\mathbf{x} + \dot{\mu}(t)\mathbf{a} - \frac{\dot{\sigma}(t)}{\sigma(t)}\mu(t)\mathbf{a} \quad (28)$$

We aim to keep $\frac{d\mathbf{x}}{dt}$ constant, which requires $\frac{\dot{\sigma}(t)}{\sigma(t)} = 0$ and $\dot{\mu}(t)$ to be constant. This implies that $\sigma(t)$ must be constant, and $\mu(t)$ must be a linear function of t .

Given the boundary condition that $\mathbf{x}_{t=0}$ approximates data distribution and $\mathbf{x}_{t=T}$ approximates a known distribution, the only feasible choice is to set $\sigma(t)$ to a constant $\mu(t) = 1 - t/T$. This satisfy all the schedule constraints noted under equation 2, except $\sigma(0) = 0$, which is approximately satisfied by choosing σ to be a small value. In practice, we use a value for σ that is two orders of magnitude smaller than the data scale, yielding good results. We can also rescale the time as in $[0, 1]$, leading to the following schedule for the diffusion process:

$$\mathbf{x}_t = (1 - t)\mathbf{x}_0 + \sigma\epsilon, t \in [0, 1], \quad (29)$$

which ensures a straight-line trajectory under a special data distribution assumption.

Next, we extend the result into general data distribution in the following theorem.

Theorem A.1. *For a general data distribution and our process schedule $\mathbf{x}_t = (1 - t)\mathbf{x}_0 + \sigma\epsilon$, $t \in [0, 1]$, the following inequality holds for each data dimension i :*

$$P\left(\left|\frac{d\mathbf{x}_t^{(i)}}{dt} + \frac{\mathbf{x}_t^{(i)}}{1-t}\right| \geq \delta\right) \leq \frac{\sigma^2}{\delta^2(1-t)^2}. \quad (30)$$

As $\sigma \rightarrow 0$, the term $\frac{d\mathbf{x}_t}{dt} + \frac{\mathbf{x}_t}{1-t}$ converges to zero in probability. More specifically, the solution to the equation $\frac{d\mathbf{x}_t}{dt} + \frac{\mathbf{x}_t}{1-t} = 0$ is that $\frac{\mathbf{x}_t}{1-t}$ becomes a constant, which corresponds to a linear trajectory.

In practice, we set σ to be small and when $t \ll 1 - \sigma$, the trajectory is approximately linear. The proof of the theorem needs two following lemmas.

Lemma A.2. For a general data distribution $f(\mathbf{x}_0)$ and a general diffusion process with the schedule $\mathbf{x}_t = \mu(t)\mathbf{x}_0 + \sigma(t)\epsilon$, its ODE description in equation 3 can be rewritten as

$$\frac{d\mathbf{x}_t}{dt} = \dot{\mu}(t)\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t] + \frac{\dot{\sigma}(t)}{\sigma(t)}(\mathbf{x}_t - \mu(t)\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]). \quad (31)$$

Proof of Lemma A.2. For the general schedule, we have $p_t(\mathbf{x}_t|\mathbf{x}_0) \sim \mathcal{N}(\mu(t)\mathbf{x}_0, \sigma(t)^2\mathbf{I})$. We start by considering the score function

$$\begin{aligned} \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t) &= \frac{\int f(\mathbf{x}_0) \nabla_{\mathbf{x}} p(\mathbf{x}_t|\mathbf{x}_0) d\mathbf{x}_0}{p(\mathbf{x}_t)} \\ &= \frac{\int f(\mathbf{x}_0) p(\mathbf{x}_t|\mathbf{x}_0) \left(-\frac{\mathbf{x}_t - \mu(t)\mathbf{x}_0}{\sigma(t)^2}\right) d\mathbf{x}_0}{p(\mathbf{x}_t)} \\ &= -\frac{\mathbf{x}_t}{\sigma(t)^2} + \frac{\mu(t)}{\sigma(t)^2} \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]. \end{aligned} \quad (32)$$

Substituting this into the ODE form of the diffusion process, we have:

$$\begin{aligned} \frac{d\mathbf{x}_t}{dt} &= \frac{\dot{\mu}(t)}{\mu(t)} \mathbf{x}_t - \left(\sigma(t)\dot{\sigma}(t) - \sigma(t)^2 \frac{\dot{\mu}(t)}{\mu(t)} \right) \left(-\frac{\mathbf{x}_t}{\sigma(t)^2} + \frac{\mu(t)}{\sigma(t)^2} \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t] \right) \\ &= \dot{\mu}(t)\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t] + \frac{\dot{\sigma}(t)}{\sigma(t)}(\mathbf{x}_t - \mu(t)\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]). \end{aligned} \quad (33)$$

□

Lemma A.3. Let $\mathbf{a}$ be a random variable that satisfies $p(\mathbf{a}|\mathbf{x}_0) \sim \mathcal{N}(\mathbf{x}_0, \sigma(t)^2/\mu(t)^2\mathbf{I})$. Define $\mathbf{y} = \mathbb{E}[\mathbf{x}_0|\mathbf{a}] - \mathbf{a}$. Then, the following properties hold:

1. $\mathbb{E}[\mathbf{y}] = \mathbf{0}$,
2. $\text{Var}[y^{(i)}] \leq \sigma(t)^2/\mu(t)^2$, $i = 1, \dots, N$, where $y^{(i)}$ is the i^{th} component of $\mathbf{y}$.

Proof of Lemma A.3. The expectation of $\mathbf{y}$ is given by:

$$\mathbb{E}\mathbf{y} = \mathbb{E}[\mathbb{E}[\mathbf{x}_0|\mathbf{a}] - \mathbf{a}] = \mathbb{E}_{\mathbf{a}} \mathbb{E}_{\mathbf{x}_0|\mathbf{a}}[\mathbf{x}_0 - \mathbf{a}] = \mathbb{E}_{\mathbf{x}_0} \mathbb{E}_{\mathbf{a}|\mathbf{x}_0}[\mathbf{x}_0 - \mathbf{a}] = \mathbf{0} \quad (34)$$

For the i -th component $y^{(i)}$, we compute the variance:

$$\begin{aligned} \text{Var}[y^{(i)}] &= \mathbb{E}[(y^{(i)})^2] = \mathbb{E}[(\mathbb{E}_{\mathbf{x}_0|\mathbf{a}}[x_0^{(i)}] - a^{(i)})^2] \\ &\leq \mathbb{E}[\mathbb{E}_{\mathbf{x}_0|\mathbf{a}}[(x_0^{(i)} - a^{(i)})^2]] = \mathbb{E}_{\mathbf{x}_0} \mathbb{E}_{\mathbf{a}|\mathbf{x}_0}[(x_0^{(i)} - a^{(i)})^2] = \sigma(t)^2/\mu(t)^2 \end{aligned} \quad (35)$$

The inequality follows from Jensen's inequality. □

Proof of A.1. For our specific schedule $\sigma(t) = \sigma$ and $\mu(t) = 1 - t$, equation 31 in lemma A.2 reduces to:

$$\frac{d\mathbf{x}_t}{dt} = -\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]. \quad (36)$$

For simplicity of notation, we define $\mathbf{a} = \mathbf{x}_t/\mu(t)$, satisfying $p(\mathbf{a}|\mathbf{x}_0) \sim \mathcal{N}(\mathbf{x}_0, \sigma(t)^2/\mu(t)^2\mathbf{I})$. By applying lemma A.3, the defined $\mathbf{y} = \mathbb{E}[\mathbf{x}_0|\mathbf{a}] - \mathbf{a}$ satisfies $\mathbb{E}[\mathbf{y}] = \mathbf{0}$ and $\text{Var}[y^{(i)}] \leq \sigma(t)^2/\mu(t)^2 = \frac{\sigma^2}{(1-t)^2}$, for all dimension i .

Applying Chebyshev's inequality, we get:

$$P(|y^{(i)}| \geq \delta) \leq \frac{\text{Var}[y^{(i)}]}{\delta^2} \leq \frac{\sigma^2}{\delta^2(1-t)^2}. \quad (37)$$

Thus, we have the inequality:

$$P(|\frac{d\mathbf{x}_t^{(i)}}{dt} - (-\mathbf{x}_t^{(i)}/(1-t))| \geq \delta) \leq \frac{\sigma^2}{\delta^2(1-t)^2}. \quad (38)$$

□

A.5 Supplementary Proof for Sampling

Lemma A.4 (Conditional Expectation Minimizes MSE). *The conditional expectation $\mathbb{E}[\mathbf{x}_{t-\Delta t}|\mathbf{x}_t]$ is the estimator of $\mathbf{x}_{t-\Delta t}$ given $\mathbf{x}_t$ that minimizes the mean squared error (MSE). That is, for any estimator $h(\mathbf{x}_t)$, the following inequality holds:*

$$\mathbb{E}[(\mathbf{x}_{t-\Delta t} - h(\mathbf{x}_t))^2] \geq \mathbb{E}[(\mathbf{x}_{t-\Delta t} - \mathbb{E}[\mathbf{x}_{t-\Delta t}|\mathbf{x}_t])^2]. \quad (39)$$

Proof. We begin by expanding the conditional squared error term. By the properties of conditional expectation, the cross-term vanishes:

$$\begin{aligned} \mathbb{E}[(\mathbf{x}_{t-\Delta t} - h(\mathbf{x}_t))^2|\mathbf{x}_t] &= \mathbb{E}[(\mathbf{x}_{t-\Delta t} - \mathbb{E}[\mathbf{x}_{t-\Delta t}|\mathbf{x}_t] + \mathbb{E}[\mathbf{x}_{t-\Delta t}|\mathbf{x}_t] - h(\mathbf{x}_t))^2|\mathbf{x}_t] \\ &= \mathbb{E}[(\mathbf{x}_{t-\Delta t} - \mathbb{E}[\mathbf{x}_{t-\Delta t}|\mathbf{x}_t])^2|\mathbf{x}_t] + \mathbb{E}[(\mathbf{x}_{t-\Delta t} - \mathbb{E}[\mathbf{x}_{t-\Delta t}|\mathbf{x}_t])(\mathbb{E}[\mathbf{x}_{t-\Delta t}|\mathbf{x}_t] - h(\mathbf{x}_t))|\mathbf{x}_t] \\ &\quad + \mathbb{E}[(\mathbb{E}[\mathbf{x}_{t-\Delta t}|\mathbf{x}_t] - h(\mathbf{x}_t))^2|\mathbf{x}_t] \\ &= \mathbb{E}[(\mathbf{x}_{t-\Delta t} - \mathbb{E}[\mathbf{x}_{t-\Delta t}|\mathbf{x}_t])^2|\mathbf{x}_t] + \mathbb{E}[(\mathbb{E}[\mathbf{x}_{t-\Delta t}|\mathbf{x}_t] - h(\mathbf{x}_t))^2|\mathbf{x}_t] \\ &\geq \mathbb{E}[(\mathbf{x}_{t-\Delta t} - \mathbb{E}[\mathbf{x}_{t-\Delta t}|\mathbf{x}_t])^2|\mathbf{x}_t] \end{aligned} \quad (40)$$

Taking the expectation with respect to $\mathbf{x}_t$, we get equation 39, which completes the proof. $\square$

Proposition A.5. *For the straight-line diffusion schedule defined in equation 5, we have*

$$\mathbb{E}[\mathbf{x}_{t-\Delta t}|\mathbf{x}_t] = \frac{1 - (t - \Delta t)}{1 - t} (\mathbf{x}_t + \sigma^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)) \quad (41)$$

Proof. From equation 32, the conditional expectation of $\mathbf{x}_0$ given $\mathbf{x}_t$ is:

$$\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t] = \frac{1}{\mu(t)} (\mathbf{x}_t + \sigma(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)) \quad (42)$$

For $\mathbf{x}_{t-\Delta t}$, the conditional expectation is derived as follows:

$$\begin{aligned} \mathbb{E}[\mathbf{x}_{t-\Delta t}|\mathbf{x}_t] &= \int \mathbf{x}_{t-\Delta t} p(\mathbf{x}_{t-\Delta t}|\mathbf{x}_t) d\mathbf{x}_{t-\Delta t} \\ &= \int \mathbf{x}_{t-\Delta t} \int p(\mathbf{x}_{t-\Delta t}|\mathbf{x}_0) p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 d\mathbf{x}_{t-\Delta t} \\ &= \int \left(\int \mathbf{x}_{t-\Delta t} p(\mathbf{x}_{t-\Delta t}|\mathbf{x}_0) d\mathbf{x}_{t-\Delta t} \right) p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 \\ &= \int \mu(t - \Delta t) \mathbf{x}_0 p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 \\ &= \mu(t - \Delta t) \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]. \end{aligned} \quad (43)$$

Here, we make the assumption that $\mathbf{x}_t$ and $\mathbf{x}_{t-\Delta t}$ are independent given $\mathbf{x}_0$. In fact, the specific temporal dependency among the random variables $(\mathbf{x}_t)_{t=0, \dots, 1}$ defines a particular sampling method within the broader family of valid samplers characterized by equation 7. Our subsequent work Ni et al. [2025] generalizes this framework beyond the SLDM schedule and introduces the sampling approach used in this paper as StoMax (maximal stochasticity) sampling, which corresponds to the variant with the highest degree of randomness within the family.

Substituting $\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]$ from equation 42, we have:

$$\mathbb{E}[\mathbf{x}_{t-\Delta t}|\mathbf{x}_t] = \frac{\mu(t - \Delta t)}{\mu(t)} (\mathbf{x}_t + \sigma(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)) \quad (44)$$

Adopting the straight-line diffusion schedule defined in equation 5, we produce equation 41. $\square$

As proved in [Xue et al., 2024b], there are a family of reverse processes that share the same marginal probability distributions as equation 2 and equation 3. When applied to our process schedule and apply Euler-Maruyama method discretization, we obtain the following iterative algorithm:

$$\mathbf{x}_{t-\Delta t} = \frac{1 - t + \Delta t}{1 - t} \mathbf{x}_t + (1 + \beta(t)) \frac{\Delta t}{1 - t} \sigma^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t) + \sqrt{2\beta(t) \frac{\Delta t}{1 - t}} \sigma^2 \epsilon. \quad (45)$$

equation 45 uses a Gaussian distribution to model the backward probability $p(\mathbf{x}_{t-\Delta t}|\mathbf{x}_t)$, whose expectation is $\frac{1-t+\Delta t}{1-t}\mathbf{x}_t + (1+\beta(t))\frac{\Delta t}{1-t}\sigma^2\nabla_{\mathbf{x}}\log p_t(\mathbf{x}_t)$. According to Lemma A.4, the optimal iterative step that minimizes MSE should satisfy $\frac{1-t+\Delta t}{1-t}\mathbf{x}_t + (1+\beta(t))\frac{\Delta t}{1-t}\sigma^2\nabla_{\mathbf{x}}\log p_t(\mathbf{x}_t) = \frac{1-(t-\Delta t)}{1-t}(\mathbf{x}_t + \sigma^2\nabla_{\mathbf{x}}\log p_t(\mathbf{x}_t))$. This result in $\beta(t) = \frac{1-t}{\Delta t}$.

A.6 Temperature Control

For Langevin dynamics, the sampling temperature can be controlled by scaling the stochastic term with a constant τ .

$$d\mathbf{x} = \frac{1}{2}g(t)^2\nabla_{\mathbf{x}}\log p_t(\mathbf{x})dt + \tau g(t)dw'. \quad (46)$$

where τ is the temperature parameter. Then $\pi(\mathbf{x}) \propto p(\mathbf{x})^{\frac{1}{\tau^2}}$ is the stationary distribution of the process in equation 46, as proved as follows:

Proof. The marginal probability density $\pi_t(\mathbf{x})$ evolves according to Fokker-Planck equation

$$\frac{\partial\pi_t(\mathbf{x})}{\partial t} = -\nabla \cdot \left[\frac{1}{2}g(t)^2\nabla_{\mathbf{x}}\log p_t(\mathbf{x})\pi_t(\mathbf{x}) \right] + \frac{1}{2}\nabla \cdot \nabla \cdot [\tau^2 g(t)^2\pi_t(\mathbf{x})] \quad (47)$$

For the stationary distribution, the probability density becomes time-independent, i.e., $\frac{\partial\pi_t(\mathbf{x})}{\partial t} = 0$. Thus, we solve:

$$\nabla \cdot \left[\frac{1}{2}g(t)^2\nabla_{\mathbf{x}}\log p_t(\mathbf{x})\pi(\mathbf{x}) \right] = \frac{1}{2}\nabla \cdot \nabla \cdot [\tau^2 g(t)^2\pi(\mathbf{x})] \quad (48)$$

We can easily validate that $\pi(\mathbf{x}) \propto p(\mathbf{x})^{\frac{1}{\tau^2}}$ satisfies the stationary equation. $\square$

Thus, higher temperatures ($\tau \rightarrow \infty$) increase diversity, with $\pi(\mathbf{x})$ approaching a uniform distribution. Conversely, lower temperatures ($\tau \rightarrow 0$) enhance fidelity, with $\pi(\mathbf{x})$ converging to a δ -distribution at the global maximum of $p(\mathbf{x})$.

However, prior work has shown that this low-temperature sampling in diffusion models fails to effectively balance diversity and fidelity in image generation, often resulting in blurred and overly smoothed outputs [Dhariwal and Nichol, 2021]. We also notice that this low-temperature sampling approach applied to our straight-line diffusion fail to fully cover all high-density regions of the target distribution, as shown in Figure 5a.

To address this, we propose an empirically designed time-annealing schedule that introduces higher stochasticity during the initial stages of sampling, allowing for the exploration of distant probability density maxima:

$$\mathbf{x}_{t-\Delta t} = \frac{1-t+\Delta t}{1-t}(\mathbf{x}_t + \sigma^2\nabla_{\mathbf{x}}\log p_t(\mathbf{x})) + t^\nu\sqrt{2}\sigma\epsilon, \quad (49)$$

where ν controls the decay rate of temperature over time. Larger ν enhances molecule stability and enables the model to cover all modes, as shown in Figure 5b. The low-temperature sampling helps mitigate the disturbance caused by noise in the training data, and thereby can improve the quality of the samples. The default value of ν for molecular generation is analyzed through an ablation study in Section C.5.

A.7 Modeling Invariant Probability Density for 3D Coordinate Generation

In this section, we aim to prove that the probability density of the generated atomic coordinates, as produced by Algorithms 3 and 4, is invariant to both translations and rotations. Formally, we aim to establish that for the Probability Density modeled by our model $p(\mathbf{x}_0) = p(\mathbf{x}_0 + \mathbf{t})$ and $p(\mathbf{x}_0) = p(\mathbf{R}\mathbf{x}_0)$, where $\mathbf{t}$ is a translation vector and $\mathbf{R}$ is an orthogonal matrix representing a rotation. The proof is in the same spirit of that in Hooeboom et al. [2022], Xu et al. [2022]. The result can be seen as a special case of Theorem 1 and Theorem 2 in Zhou et al. [2025].

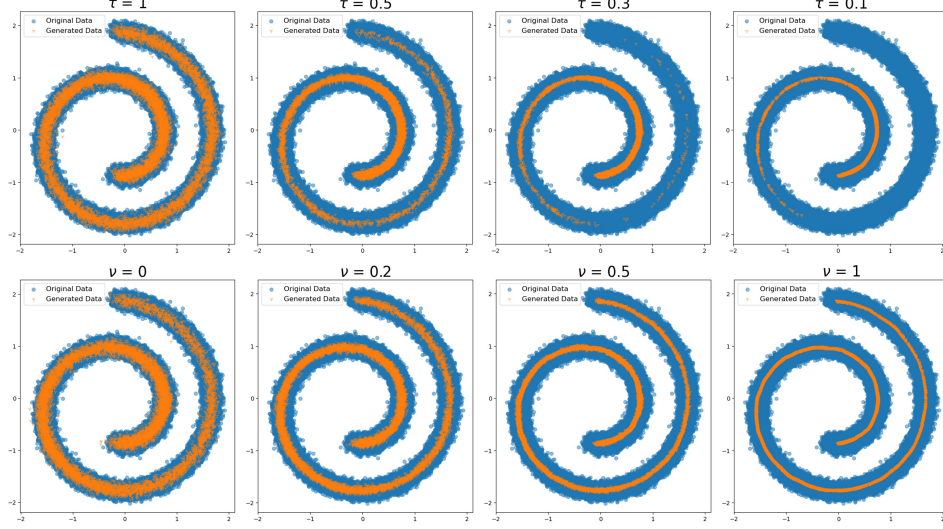


Figure 5: **a.** Generation results from straight-line diffusion with vanilla temperature control as defined in equation 46. Training data (black points) represent 2D Swiss roll coordinates with added Gaussian noise, where the highest density region lies in the interior of the spiral. Diffusion-generated samples (yellow points) exhibit reduced diversity as τ decreases but fail to cover all high-density regions, favoring those near the origin. **b.** Generation results from straight-line diffusion with our annealing temperature control as described in equation 49. Faster temperature decay (larger ν) leads to concentrated samples in high-density regions, successfully covering all local maxima.

To ensure translation invariance, the generative process is defined in the quotient space of translations, specifically the zero Center of Mass (CoM) space. This is achieved through two key operations. First, noise is sampled from a CoM-restricted Gaussian distribution $\epsilon \sim \mathcal{N}_{\text{CoM}}(\mathbf{0}, I_{3M})$, which is sampled by first sampling a standard Gaussian noise vector $\epsilon' \sim \mathcal{N}(\mathbf{0}, I_{3M})$ and then subtracting its center of mass: $\epsilon = \epsilon' - \frac{1}{3M} \sum_{i=1}^{3M} \epsilon'_i$. Second, the network’s output at each step is projected into the zero CoM space. These operations ensure that all intermediate coordinates $\mathbf{x}_t, t = 0, \dots, 1$ remain strictly within the zero CoM space throughout the generative process.

To establish rotation invariance, we rely on two key properties. First, we use an equivariant neural network satisfying $\phi^{(x)}(\mathbf{R}\mathbf{x}_t, t) = \mathbf{R}\phi^{(x)}(\mathbf{x}_t, t)$. Second, we utilize the rotational invariance of the zero mean isotropic Gaussian distributions. This property can also be extended to the CoM-restricted Gaussian distribution, whose probability density function is: $f_{\mathcal{N}_{\text{CoM}}}(\mathbf{x}) = \frac{1}{(2\pi)^{3(M-1)/2}} \exp(-\frac{1}{2}\|\mathbf{x}\|^2)$. Then, we can verify $f_{\mathcal{N}_{\text{CoM}}}(\mathbf{x}) = f_{\mathcal{N}_{\text{CoM}}}(\mathbf{R}\mathbf{x})$ for any orthogonal rotation matrix $\mathbf{R}$. Now we prove the rotation invariance of the generation probability as follows:

At each iterative step of the generative process, we have:

$$\mathbf{x}_{t-\Delta t} \sim \mathcal{N}_{\text{CoM}}\left(\frac{1-(t-\Delta t)}{1-t} \cdot (\mathbf{x}_t - \sigma \cdot \phi(\mathbf{x}_t, t)), 2t^{2\nu} \sigma^2 I_{3M}\right) \quad (50)$$

$$\begin{aligned} p(\mathbf{R}\mathbf{x}_{t-\Delta t} | \mathbf{R}\mathbf{x}_t) &= f_{\mathcal{N}_{\text{CoM}}}\left(\frac{\mathbf{R}\mathbf{x}_{t-\Delta t} - \frac{1-(t-\Delta t)}{1-t} \cdot (\mathbf{R}\mathbf{x}_t - \sigma \cdot \phi(\mathbf{R}\mathbf{x}_t, t))}{\sqrt{2}t^\nu \sigma}\right) \\ &= f_{\mathcal{N}_{\text{CoM}}}\left(\frac{\mathbf{x}_{t-\Delta t} - \left(\frac{1-(t-\Delta t)}{1-t} \cdot (\mathbf{x}_t - \sigma \cdot \phi(\mathbf{x}_t, t))\right)}{\sqrt{2}t^\nu \sigma}\right) \\ &= f_{\mathcal{N}_{\text{CoM}}}\left(\frac{\mathbf{x}_{t-\Delta t} - \left(\frac{1-(t-\Delta t)}{1-t} \cdot (\mathbf{x}_t - \sigma \cdot \phi(\mathbf{x}_t, t))\right)}{\sqrt{2}t^\nu \sigma}\right) = p(\mathbf{x}_{t-\Delta t} | \mathbf{x}_t) \end{aligned} \quad (51)$$

In the second equality, we apply the equivariance property of the neural network. The third equality follows from the rotational invariance of the isotropic Gaussian distribution.

Additionally, the initial distribution $p(\mathbf{x}_1) = \mathcal{N}_{\text{CoM}}(\mathbf{0}, \sigma^2 I_{3M})$ is rotation invariant. Combining these facts, we can propagate rotation invariance across the generative process. Thus, for the final generated distribution:

$$\begin{aligned} p(\mathbf{R}\mathbf{x}_0) &= \int \cdots \int \prod_{t=\Delta t}^1 p(\mathbf{R}\mathbf{x}_{t-\Delta t} | \mathbf{R}\mathbf{x}_t) p(\mathbf{R}\mathbf{x}_1) d\mathbf{x}_1 \cdots d\mathbf{x}_{\Delta t} \\ &= \int \cdots \int \prod_{t=\Delta t}^1 p(\mathbf{x}_{t-\Delta t} | \mathbf{x}_t) p(\mathbf{x}_1) d\mathbf{x}_1 \cdots d\mathbf{x}_{\Delta t} = p(\mathbf{x}_0). \end{aligned} \quad (52)$$

In the second equality, we use the rotational invariance of both the transition probabilities and the initial distribution. This proves that the final probability density of the generated data is rotation invariant.

B Molecular Generation Algorithms

Algorithm 3 Straight-Line Diffusion Training for Molecules (UniGEM)

Input: 3D molecular data $\mathbf{z}_0 = [\mathbf{x}_0, \mathbf{h}_0]$ with M atoms, $\mathbf{x}_0 \in \text{CoM space}$, neural network ϕ , variance constant σ , nucleation time $t_n \in [0, 1]$

repeat

Sample $t \sim \frac{1}{2}U([0, t_n]) + \frac{1}{2}U([t_n, 1])$
Sample $\epsilon \sim \mathcal{N}_{\text{CoM}}(\mathbf{0}, I_{3M})$
Compute $\mathbf{x}_t = (1-t)\mathbf{x}_0 + \sigma\epsilon$
Minimize $\|\epsilon - \phi^{(x)}(\mathbf{x}_t, t)\|^2 + 1_{t \leq t_n} \|\mathbf{h}_0 - \phi^{(h)}(\mathbf{x}_t, t)\|$

until converged;

Algorithm 4 Straight-Line Diffusion Sampling for Molecules (UniGEM)

Input: Number of atoms M , neural network ϕ , variance constant σ , nucleation time t_n , sampling steps T , temperature annealing rate ν

Sample $\mathbf{x}_1 \sim \sigma \cdot \mathcal{N}_{\text{CoM}}(\mathbf{0}, I_{3M})$

for $i = T - 1$ **to** 1 **do**

$t = i/T, \Delta t = 1/T$
 $\mathbf{x}_{t-\Delta t} = \frac{1-(t-\Delta t)}{1-t} \cdot (\mathbf{x}_t - \sigma \cdot \phi(\mathbf{x}_t, t))$
Projecting $\mathbf{x}_{t-\Delta t}$ into zero CoM space
if $i > 1$ **then**
| Sample $\epsilon \sim \mathcal{N}_{\text{CoM}}(\mathbf{0}, I_{3M})$
| $\mathbf{x}_{t-\Delta t} = \mathbf{x}_{t-\Delta t} + t^\nu \sqrt{2} \cdot \sigma \cdot \epsilon$

end

else

$\mathbf{h}_0 = \phi^{(h)}(\mathbf{x}_0, 0)$

end

end

return $\mathbf{z}_0 = [\mathbf{x}_0, \mathbf{h}_0]$

C Supplementary Illustrations and Results

C.1 Sampling Trajectory of Diffusion-based Models

To evaluate whether SLDM exhibits a near-linear trajectory under general data distributions, as suggested by Theorem 3.1, we visualize the ODE trajectory and compare it with other diffusion-based models. We consider a scenario where the data distribution is a one-dimensional Gaussian mixture, as this setup offers a tractable score function and serves as a representative example of general distributions. The resulting trajectories are shown in Figure 6. Our method maintains a trajectory that is consistently closer to a straight line compared to other approaches, leading to smaller truncation

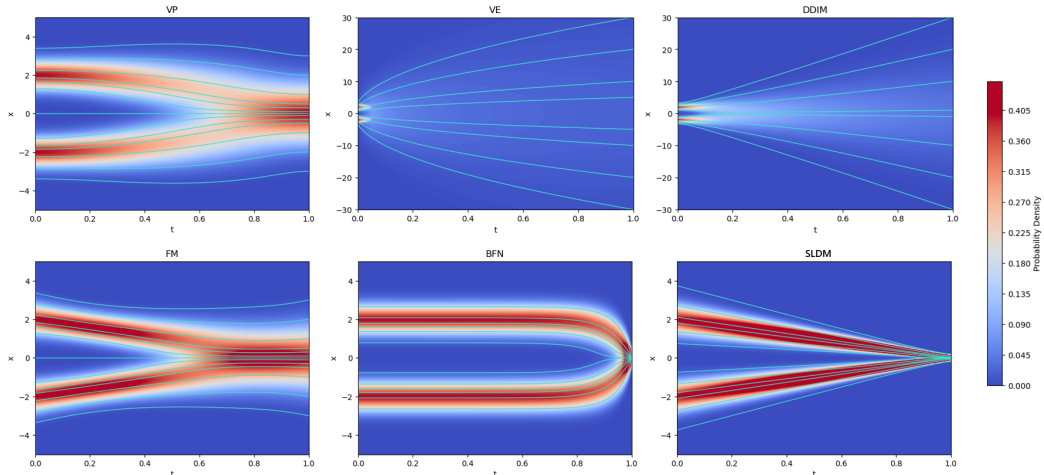


Figure 6: The trajectory of the ODE in equation 3 of various diffusion-based models. The data distribution is a mixture of Gaussians in one dimension, defined as $x_0 \sim 0.5 \cdot \mathcal{N}(2, 1/4) + 0.5 \cdot \mathcal{N}(-2, 1/4)$. The background color indicates the value of probability density $p(x_t)$. During sampling, the timestep t decreases from 1 to 0.

errors under first-order numerical discretization. As suggested in Theorem 3.1, the trajectory slightly deviates from a straight line in the early stages (i.e., as t approaches 1), which could increase the sampling error in theory. However, our empirical observations indicate that the sampling process is relatively robust to such errors during these initial steps. This robustness may be attributed to the incorporation of Langevin dynamics in the sampling algorithm, which can mitigate the impact of early-stage inaccuracies. As a result, SLDM achieves an overall lower sampling error.

In comparison, DDIM demonstrates a relatively linear trajectory during the early stages but exhibits significant curvature in later stages, where accurate sampling is more crucial. This could potentially degrade its performance. For the FM method, where the initial distribution is Gaussian and no explicit prior-data joint distribution is predefined as in Tong et al. [2023], it fails to achieve a straight-line trajectory. Other methods also demonstrate an evident curved trajectory. Besides, we can also see from the illustration that the distribution remains nearly static during the later stages of BFN sampling, which aligns with the discussion in section 3.2 and suggests potential inefficiencies of time scheduling.

C.2 Sampling Efficiency Comparisons under UniGEM Framework

We complement the ablation study in Section 4.4 by incorporating results with diverse sampling steps. The results demonstrate that while UniGEM enhances the performance of EDM and BFN when sampling step is abundant, these methods still struggle to achieve satisfactory molecular stability in the low sampling step scenario.

C.3 Sampling Efficiency Under Joint Diffusion Framework

A natural question arises: can SLDM achieve strong performance without the UniGEM framework? Specifically, can it jointly generate atom types and coordinates using diffusion models, following the approach of EDM and GeoBFN? To address this, we introduce a variant called *SLDM (joint diffusion)* as a preliminary exploration, where atom types are generated using an SLDM model tailored for discretized data, replacing UniGEM’s prediction strategy. The training and sampling procedures of SLDM (joint diffusion) for molecule generation are detailed in Algorithms 5 and 6, respectively. Unlike UniGEM, we follow GeoBFN to represent atom types by their atomic numbers (positive integers) and adopt the binning technique from BFN to handle discretized data. Here, $P_{\hat{\mu}_h, \hat{\sigma}_h}(b)$ denotes the discretized cumulative distribution function (CDF), defined as: $F(b) = \frac{1}{2}[1 + \text{erf}(\frac{b-\mu}{\sqrt{2}\sigma})]$, with $P(b) = F(b)$ for $-1 < b < 1$, $P(b) = 0$ for $b \leq -1$, and $P(b) = 1$ for $b \geq 1$. "NearestCenter" decodes the generated continuous continuous normalized atomic number into an atomic type by

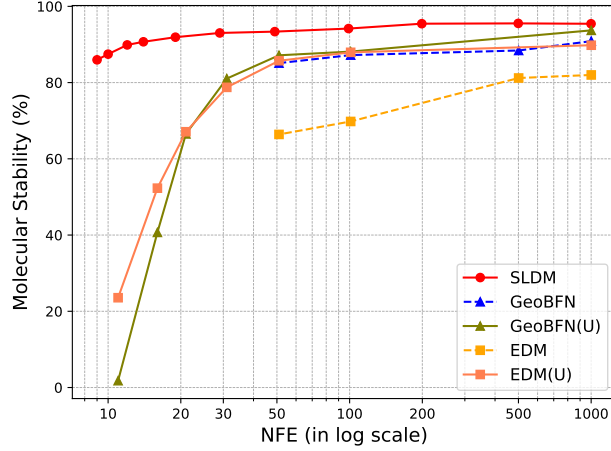


Figure 7: Comparison of molecule stability ($\uparrow$) across diffusion-based molecular generation models with UniGEM framework on QM9 unconditional generation, evaluated with respect to the Number of Function Evaluations (NFE) during the sampling process.

finding the nearest center of bins (normalized atomic numbers), then unnormalizing it and mapping it to the corresponding atomic type. Notably, there remains substantial room for further exploration regarding the handling of diffusion for discretized data, which we leave for future work.

Algorithm 5 SLDM Training (joint diffusion)

Input: Number of bins $K \in \mathbb{N}$ (equal to the largest atomic number), fixed variance σ , 3D molecular data $z_0 = [x_0, h_0]$ with M atoms, $x_0 \in \text{CoM space}$, normalized atom number $\tilde{h}_0 = (2 * h_0 - 1)/K - 1 \in [\frac{1}{K} - 1, 1 - \frac{1}{K}]^D$

repeat

Sample $t \sim \mathcal{U}(0, 1)$, $\epsilon^x \sim \mathcal{N}_{\text{CoM}}(\mathbf{0}, I_{3M})$, $\epsilon^h \sim \mathcal{N}(\mathbf{0}, I_{3M})$

Compute $x_t = (1 - t)x_0 + \sigma\epsilon^x$

Compute $h_t = (1 - t)h_0 + \sigma\epsilon^h$

Network output: $\hat{\epsilon}^x, \hat{\mu}_h^\epsilon, \ln \hat{\sigma}_h^\epsilon = \Phi(x_t, h_t, t)$

$\hat{\mu}_h = \frac{h_t - \sigma\hat{\mu}_h^\epsilon}{1 - t}$, $\hat{\sigma}_h = \frac{\sigma \exp(\ln \hat{\sigma}_h^\epsilon)}{1 - t}$

Minimize denoise loss: $\|\epsilon^x - \hat{\epsilon}^x\|^2 + \left\| (\hat{\mu}_h^\epsilon - \epsilon^h) - \frac{1-t}{\sigma} \sum_{i=1}^K P_{\hat{\mu}_h, \hat{\sigma}_h}(b_i)(b_i - \hat{\mu}_h) \right\|^2$

until converged;

Results are presented in Table 4. The proposed temperature annealing mechanism endows the model with greater flexibility, enabling a balanced trade-off between stability/validity and uniqueness/diversity. We manually tuned the temperature annealing rates for coordinates (ν_x) and atomic numbers (ν_h), and empirical observations indicate that setting $\nu_x > \nu_h$ yields better performance in achieving such balance. It should be noted that we did not conduct exhaustive hyperparameter searches, leaving room for potential further improvements. SLDM (joint diffusion) outperforms EDM across all metrics and exhibits superior stability and validity with slightly lower uniqueness compared to GeoBFN when the number of sampling steps T is limited. Notably, the overall performance of this variant is still inferior to the original SLDM which employs the UniGEM framework. This finding reaffirms that the UniGEM framework may be a more optimal approach for generating small-molecule.

C.4 Toy Data Results

We evaluate our model on several 2D toy data to test its generality. The datasets included swissroll and moons to represent continuous data distributions, as well as chessboard to simulate discretized data distributions. The dataset consists of 100,000 samples. For the moons and swissroll datasets,

Algorithm 6 SLDM Sampling (joint diffusion)

Input: Number of atoms M , neural network ϕ , variance constant σ , nucleation time t_n , sampling steps T , temperature annealing rate ν_x and ν_h

Sample $\mathbf{x}_1 \sim \sigma \mathcal{N}_{\text{CoM}}(\mathbf{0}, I_{3M})$, $\mathbf{h}_1 \sim \sigma \mathcal{N}(\mathbf{0}, I_{3M})$

for $i = T - 1$ **to** 1 **do**

$t = i/T$, $\Delta t = 1/T$

 Network output: $\hat{\epsilon}^x, \hat{\mu}_h^\epsilon, \ln \hat{\sigma}_h^\epsilon = \Phi(\mathbf{x}_t, \mathbf{h}_t, t)$

$\hat{\mu}_h = \frac{\mathbf{h}_t - \sigma \hat{\mu}_h^\epsilon}{1-t}$, $\hat{\sigma}_h = \frac{\sigma \exp(\ln \hat{\sigma}_h^\epsilon)}{1-t}$

 Update $\mathbf{x}_{t-\Delta t} = \frac{1-(t-\Delta t)}{1-t} \cdot (\mathbf{x}_t - \sigma \cdot \hat{\epsilon}^x)$

 Update $\hat{\mathbf{h}}_0 = \sum_{i=1}^K P_{\hat{\mu}_h, \hat{\sigma}_h}(b_i) b_i$, $\mathbf{h}_{t-\Delta t} = (1 - (t - \Delta t)) \hat{\mathbf{h}}_0$

 Projecting $\mathbf{x}_{t-\Delta t}$ into zero CoM space

if $i > 1$ **then**

 Sample $\epsilon^x \sim \mathcal{N}_{\text{CoM}}(\mathbf{0}, I_{3M})$, $\epsilon^h \sim \mathcal{N}(\mathbf{0}, I_{3M})$

$\mathbf{x}_{t-\Delta t} = \mathbf{x}_{t-\Delta t} + t^{\nu_x} \sqrt{2} \cdot \sigma \cdot \epsilon^x$

$\mathbf{h}_{t-\Delta t} = \mathbf{h}_{t-\Delta t} + t^{\nu_h} \sqrt{2} \cdot \sigma \cdot \epsilon^h$

end

return $\mathbf{z}_0 = [\mathbf{x}_0, \text{NearestCenter}(\mathbf{h}_0)]$

Table 4: Sampling efficiency comparison under joint diffusion framework on QM9 dataset unconditional generation task.

Method	Atom Sta (%)	Mol Sta (%)	Valid (%)	V×U (%)
EDM (T=1000)	98.70	82.00	91.90	90.70
GeoBFN (T=1000)	99.08	90.87	95.31	92.96
SLDM (joint diffusion) (T=1000)	99.01	91.57	95.20	91.57
GeoBFN (T=500)	98.78	88.42	93.35	91.78
SLDM (joint diffusion) (T=500)	99.89	90.57	94.00	91.11
GeoBFN (T=100)	98.64	87.21	93.03	91.53
SLDM (joint diffusion) (T=100)	98.92	90.31	93.84	89.36
GeoBFN (T=50)	98.28	85.11	92.27	90.72
SLDM (joint diffusion) (T=50)	98.70	88.09	92.84	89.53

we performed training with 40 diffusion steps and 100 epochs, and for the chessboard dataset, the training was extended to 600 epochs with 100 diffusion steps to ensure convergence for all models. Other settings are kept the same for all datasets: The model is a 5-layer MLP. The batch size was set to 2048, and the optimizer used was Adam with a learning rate of 0.001. No temperature control is used during sampling. These settings follow <https://github.com/albarji/toy-diffusion/>, and are kept consistent across all generative algorithms to ensure a fair comparison. The results are provided in Figure 8. The data generated by our model show the closest alignment to the original distributions. Specifically, our model produced samples with fewer outliers and achieved good coverage of the data distributions. These findings highlight the superior generative capabilities of our approach in faithfully modeling complex data distribution.

C.5 Ablation Study for Temperature Control

Theoretically, a larger temperature annealing rate v corresponds to a faster cooling scheme. The results in Table 5 demonstrate that adjusting the temperature can effectively enhance molecular stability. However, extreme values of v may significantly reduce diversity. We selected $v = 0.5$ as the default setting for our model, as it achieves the highest $U \times V$ score, indicating the optimal balance between diversity and fidelity.

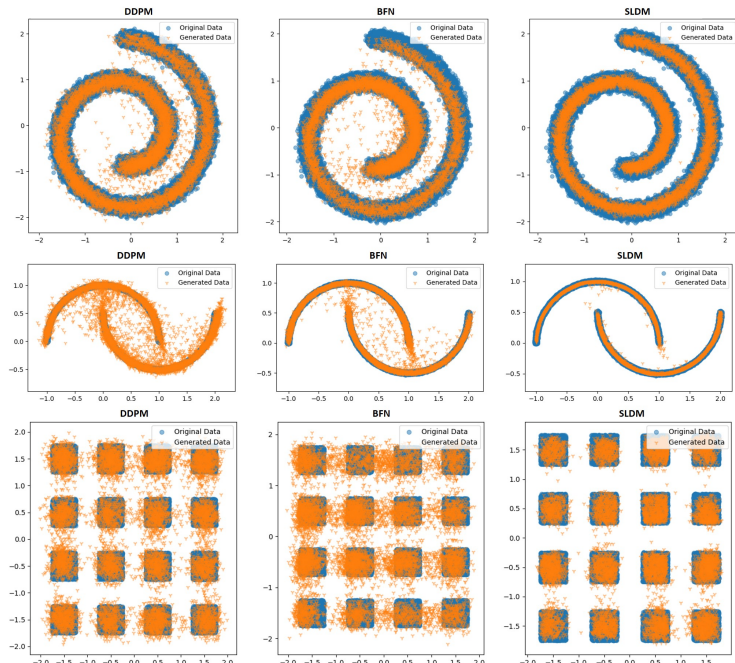


Figure 8: Generative performance comparison of Straight-Line Diffusion, DDPM, and BFN on 2D toy datasets.

Table 5: Impact of temperature annealing rate evaluated on the QM9 unconditional generation with sampling step $T = 50$. Higher values indicate better performance.

	Atom sta(%)	Mol sta(%)	Valid(%)	$U \times V(\%)$
SLDM($v=0$)	99.14	91.74	95.40	92.76
SLDM($v=0.5$)	99.28	93.03	96.20	93.28
SLDM($v=1$)	99.27	93.46	96.08	92.42
SLDM($v=3$)	99.37	94.34	96.84	90.08
SLDM($v=5$)	99.42	94.83	97.33	86.10
SLDM($v=10$)	99.53	96.06	97.98	75.13

C.6 Robustness to Different Prior Distributions

Empirically, we find that SLDM maintains high validity and stability even when the initial distribution during sampling deviates from the theoretical valid one, i.e., Gaussian distribution with standard deviation σ , as shown in Table 6. This robustness helps mitigate the impact of deviations from the ideal linear trajectory in the early sampling steps, as discussed under the Theorem 3.1.

Table 6: Robustness of SLDM to different initial sampling distributions on the QM9 unconditional generation task. Here $\epsilon \sim \mathcal{N}(0, I)$ and the sampling step $T = 50$.

Prior Distribution	Mol_stab (%)	Atom_stab (%)	Validity (%)	V*U (%)
$\sigma \cdot \epsilon$ (theoretical)	93.43	99.32	96.10	93.31
$2 \cdot \sigma \cdot \epsilon$	91.81	99.12	95.00	92.79
$0.1 \cdot \sigma \cdot \epsilon$	93.31	99.29	96.07	93.46
$0.01 \cdot \sigma \cdot \epsilon$	92.94	99.24	96.06	93.06

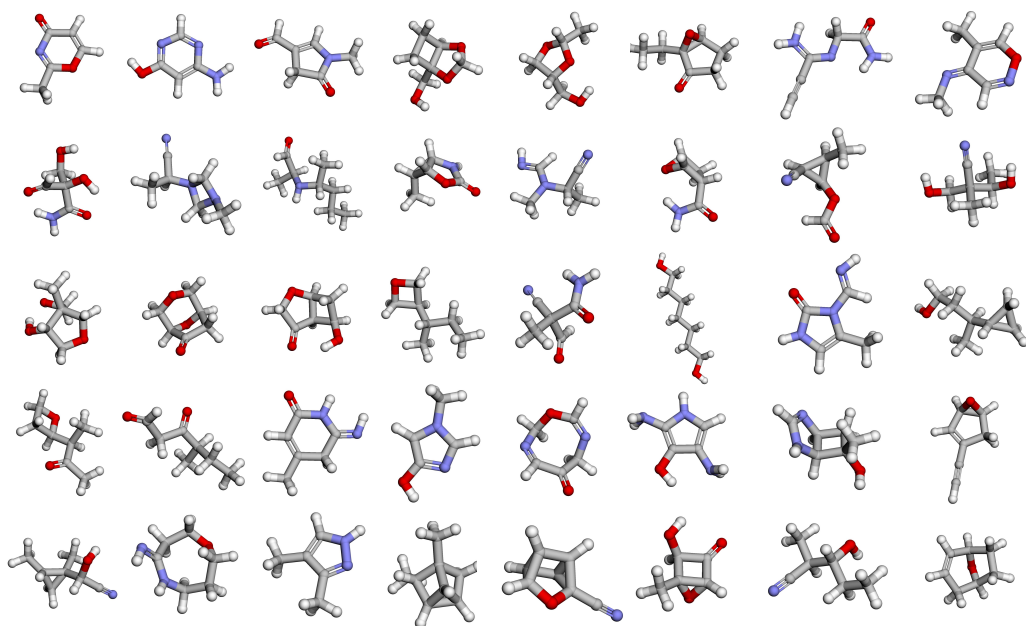


Figure 9: 3D molecules generated by the SLDM model trained on the QM9 dataset.

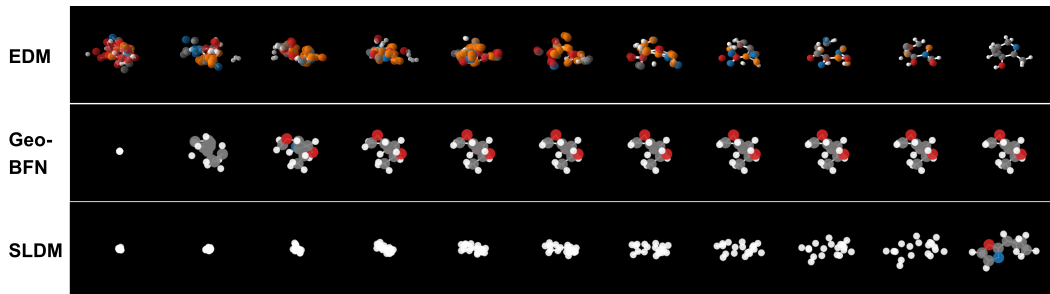


Figure 10: The generative process of EDM, GeoBFN and SLDM models.

C.7 Visualizations of Generated Molecules and Sampling Process

We have randomly generated 40 molecules using our trained models on QM9 in the unconditional setting, and visualized their 3D structures using py3Dmol, as shown in Figure 9. These examples clearly demonstrate the structural plausibility and diversity of our generated molecules.

In Figure 2, we compare the diffusion processes of different models by adding noise to the same molecule’s coordinates, enabling a straightforward visual comparison of their noise schedules. As a complement, we present a comparison of the practical sampling processes of the three models in Figure 10. Although generating the exact same molecule across models is not feasible in this case, we still arrive at conclusions analogous to those in Section 3.2 of the main text. Additionally, unlike the noise-adding process, the generation process of SLDM demonstrates more pronounced randomness.

D Related Work

D.1 An overview of 3D Molecular Generation

Recent advances in 3D molecular generation can be categorized based on the underlying **generative algorithms**: Autoregressive methods generate molecules step by step, progressively connecting atoms or molecular fragments. G-SchNet [Gebauer et al., 2019] and G-SphereNet [Luo and Ji, 2022] are early examples that use this strategy to model 3D molecular structures. Building on

these, Symphony [Daigavane et al., 2024] incorporates higher-degree $E(3)$ -equivariant features and message-passing to improve the modeling of molecular geometries. Normalizing flow [Chen et al., 2018] has also been applied to molecule generation by E-NF [Garcia Satorras et al., 2021] and Köhler et al. [2020]. Diffusion-based models have gained prominence for 3D molecular generation recently. Hoogetboom et al. [Hoogetboom et al., 2022] introduced the Equivariant Diffusion Model (EDM) to jointly generate atomic coordinates and atom types. Extensions of EDM include EDM-Bridge [Wu et al., 2022], which enhances performance through prior bridges, and GeoLDM [Xu et al., 2023], which performs diffusion in a latent space using an autoencoder. EquiFM [Song et al., 2024] employs flow matching for efficient molecule generation, and GeoBFN [Song et al., 2023a] combines Bayesian Flow Networks with distinct generative algorithms tailored for discretized charges and continuous coordinates. END [Cornet et al., 2024] proposes a learnable data- and time-dependent noise schedule of the diffusion process, and achieves improved sampling efficiency.

Several studies offer **complementary strategies** to generative algorithms. First, representation-conditioned generation has been explored by MDM [Huang et al., 2023a], which conditions on VAE representations, and GeoRCG [Li et al., 2024], which extends EDM by leveraging pretrained molecular representations. Second, some studies propose advanced network architectures to enhance molecular generation [Hua et al., 2024, Huang et al., 2023a, Le et al., 2024a, Irwin et al., 2024][Huang et al., 2023b, Mercatali et al., 2024]. Third, additional input information has been introduced to the molecular generation. For instance, MolDiff [Peng et al., 2023] explicitly predicts bonds during generation. MiDi [Vignac et al., 2023], JODO [Huang et al., 2023b], EQGAT-diff [Le et al., 2024a], and SemlaFlow [Irwin et al., 2024] further extend generation to include bonds and formal charges, enriching the molecular inputs. Twigs [Mercatali et al., 2024], on the other hand, integrates additional molecular properties into the diffusion training process, leading to enhanced conditional generation.

It is worth noting that **evaluation strategies** for 3D molecular generation can be different across methods. EDM [Hoogetboom et al., 2022] employs strict rules for bond definitions based on interatomic distances, implicitly enforcing constraints on bond lengths and steric hindrance. In this framework, the metric "stability" is rigorously defined, requiring correct valency and neutral atomic charges. Our method, along with most diffusion-based approaches [Wu et al., 2022, Xu et al., 2023, Song et al., 2023a], adheres to this evaluation standard. In contrast, another line of studies [Vignac et al., 2023, Le et al., 2024a, Irwin et al., 2024] infers bonds and formal charges directly from the model, allowing atoms to have non-zero formal charges. Therefore, the bond inference imposes no constraints on bond lengths or steric hindrance. Further, the "stability" metric is defined more loosely, permitting discrepancies between valency and the number of covalent bonds. This relaxed evaluation framework makes it challenging to directly compare these two approaches. We advocate future efforts to establish more equitable evaluation methods to fairly assess the strengths of both paradigms.

D.2 Related Diffusion-Based Studies and Key Differences with SLDM

Noise Scheduling It is important to note that our proposed process schedule differs fundamentally from previous works on noise scheduling [Karras et al., 2022, Nichol and Dhariwal, 2021, Le et al., 2024b], time discretization strategies [Xue et al., 2024c, Li et al., 2023], and adaptive step size methods [Lu et al., 2022]. These approaches can be interpreted as applying a time transformation $\mathcal{T}(\cdot)$ that jointly rescales the diffusion process schedule as $\mu(\mathcal{T}(t))$ and $\sigma(\mathcal{T}(t))$. Notably, such transformations preserve the monotonicity and endpoints of the schedule functions. In contrast, our method decouples μ and σ , and fundamentally alters the monotonicity and endpoint of $\sigma(t)$.

Linear $\mu(t)$. Although our method is derived from the diffusion perspective, its process schedule shares some similarities with flow matching (FM) algorithms, such as FM-OT [Lipman et al., 2023] and conditional flow matching (CFM) [Tong et al., 2023]. Both methods employ a linear $\mu(t)$, and CFM introduces additional low-scale noise. From this perspective, our process schedule can also be interpreted as a variant of CFM. Specifically, our approach employs a prior distribution modeled as a small-scale Gaussian distribution centered at the origin. However, FM and diffusion differ fundamentally in their perspectives: FM models the generative process as an ODE, while diffusion models a stochastic process. This core distinction leads to differences in both the learning targets and sampling methods. Therefore, unlike FM that learns the velocity field and relies on ODE-based sampling, our approaches focus on learning the noise and employ stochastic sampling.

Straight Sampling Trajectory. Moreover, similar to our approach, flow-based methods also aim at flows with straight trajectories. However, these methods rely on predefined prior-data joint distribution

to produce straighter paths. This distribution is procured by solving an optimal transport (OT) problem during the training of flow matching [Tong et al., 2023, Song et al., 2024], and solving such problem is often challenging. When the OT solution is unavailable, achieving straight-line trajectories typically requires additional distillation steps or solving optimization problems, as studied in rectified flow [Liu et al., 2022], progressive distillation [Salimans and Ho, 2022], consistency model [Song et al., 2023b, Luo et al., 2023] and optimal flow matching [Kornilov et al., 2024]. NFDM [Bartosh et al., 2024] proposes a learnable forward process to adapt to align with the reverse process and introduces penalties on the curvature of the reverse process’s trajectories. In contrast to these existing methods, our approach offers a more straightforward solution for achieving a straight trajectory, by designing a novel diffusion process that minimizes the second-order derivative of the trajectory.

E Implementation Detail

The hyperparameter settings for molecular generation are detailed in Table 7. Settings follow UniGEM [Feng et al., 2024], with two additional tunable hyperparameters introduced by our generative algorithm: the noise variance σ and the temperature annealing rate ν .

Table 7: Network and training hyperparameters.

Network Hyperparameters	Value
Embedding size	256 for unconditional generation, 192 for conditional generation
Layer number	9 for QM9, 4 for Geom-Drugs
Shared layers	1
Training Hyperparameters	Value
Batch size	64 for QM9, 128 for Geom-Drugs
Train epoch	3000 for QM9, 32 for Geom-Drugs
Learning rate	1.00×10^{-4}
Optimizer	Adam
Sample steps T	$10 \sim 1000$
Nucleation time	10
Oversampling ratio	0.5 for each branch
Loss weight	1 for each loss term
Generative Algorithm Hyperparameters	Value
Noise Variance σ	0.05 for unconditional generation, 0.1 for conditional generation
Temperature Annealing Rate ν	0.5 for unconditional generation, 3 for conditional generation
Non-uniform Discretization	False if $T > 13$

For QM9, it takes approximately 10 days on a single A100 GPU. For GEOM-drugs, it takes approximately 16 days on four A100 GPUs. While the training epochs are longer compared to the baselines, this extended duration is not mandatory. Although longer training leads to slight performance gains, we found that the mol stable and atom stable metrics remain stable around 2000 epochs on QM9 and 13 epochs on GEOM-Drugs during the training process.

For sampling steps greater than 13, the geometric straight-line diffusion use a uniform time discretization like GeoBFN and EDM. However, according to theorem 3.1, our trajectory exhibits a larger second-order derivative at the beginning of sampling. Therefore, a more efficient discretization strategy is to use fine-grained discretization for larger t values. We manually set an empirical discretization strategy that yields a 1% to 10% improvement in Mol Stable when $T \leq 13$. For sampling steps greater than 13, the impact on the results is less significant ($< 1\%$). We leave the exploration of the optimal discretization strategy for straight-line diffusion for future work.

For conditional molecular generation, we follow the baseline approaches [Hoogeboom et al., 2022] by incorporating property values as additional inputs during training and sampling them from a prior distribution during inference. The results are measured using a property classification network trained on the first half of the QM9 dataset, while the remaining portion is used for training the generative model. In addition to generative model baselines, we include two basic baselines: Random and N_{atoms} . The Random baseline involves shuffling property labels in the training data and evaluating

the property classifier on the shuffled data. The N_{atoms} baseline uses the property classifier network to predict the property based solely on the number of atoms in the molecule. The property classification network is used for evaluating the generated molecules. The metric is the MAE between predicted properties and target properties. As for the conditioning mechanism, property values are concatenated to atom types as EGNN’s invariant input, which is in line with other baselines.