Straight-Line Diffusion Model for Efficient 3D Molecular Generation

Yuyan Ni [*1 2 3] **Shikun Feng** [*3] **Haohan Chi** [3] **Bowen Zheng** [4] **Huan-ang Gao** [3]
Wei-Ying Ma [3] **Zhi-Ming Ma** [1] **Yanyan Lan** [3 5]

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]

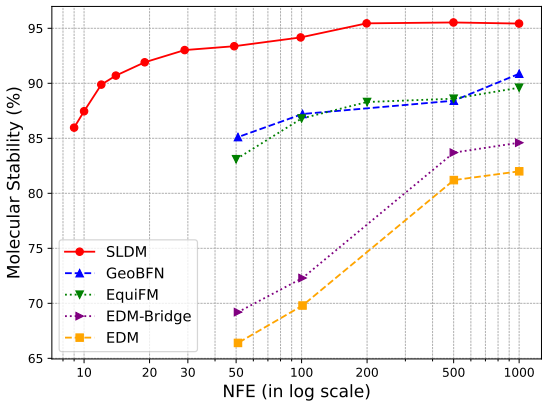

Figure 1. Comparison of molecule stability (↑) across diffusion-based molecular generation models on QM9 unconditional generation, evaluated with respect to the number of function evaluations (NFE) during the sampling process.

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 & Nichol, 2021; Rombach et al., 2022; Peebles & 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. 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., 2023b) 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× and 20×, 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

[*]Equal contribution [1]Academy of Mathematics and Systems Science, Chinese Academy of Sciences [2]University of Chinese Academy of Sciences [3]Institute for AI Industry Research (AIR), Tsinghua University (Work was done during Yuyan's internship at AIR.) [4]Huazhong University of Science and Technology [5]Beijing Academy of Artificial Intelligence. Correspondence to: Yanyan Lan <lanyanyan@air.tsinghua.edu.cn>.

Proceedings of the Workshop on Generative AI for Biology at the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

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

diffusion models, flow matching methods, and Bayesian flow networks. Specifically, their noise corrupting process can be generalized as $\boldsymbol{x}_t = \mu(t)\boldsymbol{x}_0 + \sigma(t)\boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_N)$, where $\boldsymbol{x}_0$ represents the clean data, $\boldsymbol{x}_t$ 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 $\frac{\mathrm{d}^2 x(t)}{\mathrm{d}t^2}(\Delta t)^2$. We observe that $\frac{\mathrm{d}^2 x(t)}{\mathrm{d}t^2}$ 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. $\frac{\mathrm{d}^2 x(t)}{\mathrm{d}t^2} = 0$. 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 distribution.

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. (2023b). 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 $\boldsymbol{x} \in \mathbb{R}^N$, 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:

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

where $\boldsymbol{x}_0, \boldsymbol{x}_t$ 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). σ_{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 $\boldsymbol{x}_T$ 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):

$$\mathrm{d}\boldsymbol{x}_t = \frac{\dot{\mu}(t)}{\mu(t)}\boldsymbol{x}_t\mathrm{d}t + \sqrt{2\sigma(t)\dot{\sigma}(t) - 2\sigma(t)^2\frac{\dot{\mu}(t)}{\mu(t)}}\mathrm{d}w_t, \quad (2)$$

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:

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

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 3 [2]. 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 $x(t) - \frac{\mathrm{d}x(t)}{\mathrm{d}t}\Delta t$ when solving the ODE backward in time. However, the true value can be derived from the Taylor expansion:

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

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

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 (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 $\frac{\mathrm{d}^2x(t)}{\mathrm{d}t^2}$. 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 $\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}$ 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 constraints of the design space 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 constant and $\mu(t) = 1 - t/T$. This satisfies 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. Rescaling time to the interval $[0, 1]$, the resulting diffusion process is:

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

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:*

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

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

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

In other words, for $t \in [0, 1 - \frac{\sigma}{\delta \cdot \epsilon}]$, $|\frac{\mathrm{d}\boldsymbol{x}_t^{(i)}}{\mathrm{d}t} + \frac{\boldsymbol{x}_t^{(i)}}{1-t}| < \delta$ 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. This initial deviation introduces sampling error, but empirically, we find that this error can be mitigated by the Langevin dynamics component in our stochastic sampler introduced in Section 3.3.

3.2. Comparisons with Previous Diffusion Processes

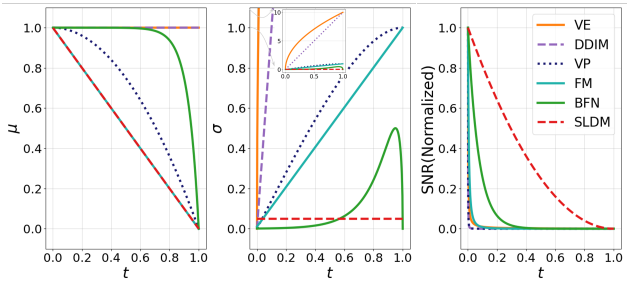

Figure 2. Comparison of schedule parameters across different diffusion-based models[4]. Here SNR refers to $(\mu/\sigma)^2$. We normalize the SNR values to the range of 0 to 1 for easier visualization comparison.

Previous studies (Nichol & 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. (2023b), 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 2, the SNR in our method decreases significantly slower than that of previous methods. Specifically, unlike traditional methods, where noise variance varies substantially during the forward process, our approach maintains a constant low noise level throughout. These designs result in a smoother and more stable generative process for molecules, as demonstrated in Figure 3. 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.

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

Now, 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., 2024c) (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 reverse-time SDE:

$$\begin{aligned}\boldsymbol{x}_{t-\Delta t} = &\underbrace{\boldsymbol{x}_t + \frac{\Delta t}{1-t}\left(\boldsymbol{x}_t + \sigma^2 \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t)\right)}_{\text{ODE Sampling of equation 3}} \\ &\underbrace{+\beta(t)\frac{\Delta t}{1-t}\sigma^2 \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t) + \sqrt{2\beta(t)\frac{\Delta t}{1-t}}\sigma\boldsymbol{\epsilon}}_{\text{Langevin dynamics}},\end{aligned} \tag{7}$$

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

Similar to previous works (Xue et al., 2024c; 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 tra-

[4] $\sigma = 0.05$ for SLDM. The cutoff values σ_{max} and σ_{min} for other models are provided in appendix A.2

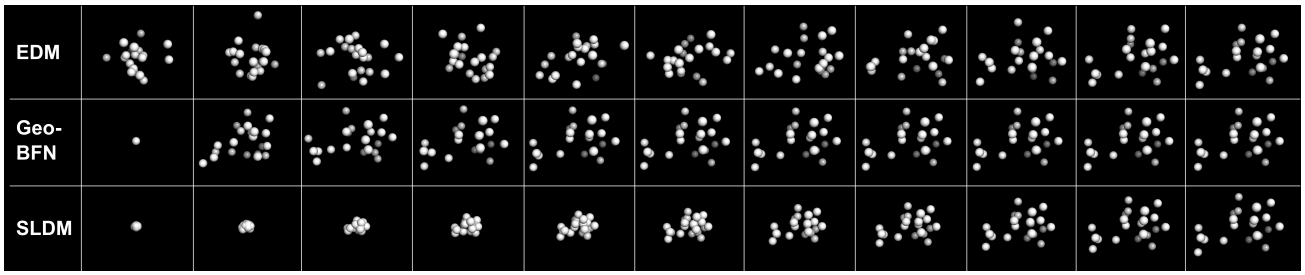

Figure 3. The diffusion process of atomic coordinates in EDM, GeoBFN and SLDM.

jectories, 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 $\mathbb{E}[\boldsymbol{x}_{t-\Delta t}|\boldsymbol{x}_t]$, as it provides the optimal estimation of $\boldsymbol{x}_{t-\Delta t}$ in terms of minimizing the mean squared error (MSE). Details are demonstrated in Appendix A.5.

Thus the sampling algorithm becomes:

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

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 evidenced by our observation that the sampling results are relatively robust to the initial distribution's variance.

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 & 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:

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

Algorithm 1 Straight-line Diffusion Training

Input: data $\boldsymbol{x}_0$ with dimension N, neural network ϕ, variance constant σ
repeat
  Sample $t \sim U([0,1])$
  Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, I_N)$
  Compute $\boldsymbol{x}_t = (1-t)\boldsymbol{x}_0 + \sigma\boldsymbol{\epsilon}$
  Minimize $||\boldsymbol{\epsilon} - \phi(\boldsymbol{x}_t, t)||^2$
until *converged*

Algorithm 2 Straight-line Diffusion Sampling

Input: trained network ϕ, same N and σ as training, sampling steps T, temperature annealing rate ν
Sample $\boldsymbol{x}_1 \sim \sigma \cdot \mathcal{N}(\mathbf{0}, I_N)$
for $i = T-1$ **to** 1 **do**
  $t = i/T$, $\Delta t = 1/T$
  $\boldsymbol{x}_{t-\Delta t} = \frac{1-(t-\Delta t)}{1-t} \cdot (\boldsymbol{x}_t - \sigma \cdot \phi(\boldsymbol{x}_t, t))$
  if $i > 1$ **then**
    Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, I_N)$
    $\boldsymbol{x}_{t-\Delta t} = \boldsymbol{x}_{t-\Delta t} + t^{\nu}\sqrt{2} \cdot \sigma \cdot \boldsymbol{\epsilon}$
  end if
end for
Return: $\boldsymbol{x}_0$

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.

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.

	QM9				GEOM-Drugs	
#Metrics	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	93.5	84.1	98.9
GeoBFN (T=1000)	99.08	90.87	95.31	92.96	85.60	92.08
GCDM (T=1000)	98.7	85.7	94.8	93.3	88.1	95.5
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	88.30	**99.95**
SLDM (T=50)	99.30	93.37	96.24	**93.63**	**89.03**	99.57

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 (Hoogeboom et al., 2022; Song et al., 2024; 2023b). 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 & 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 (Hoogeboom et al., 2022; Song et al., 2024; 2023b). 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, $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{h})$, where $\boldsymbol{x} \in \mathbb{R}^{3M}$ denotes the atomic coordinates, M is the number of atoms, and $\boldsymbol{h}$ 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); Hoogeboom 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; Hoogeboom et al., 2022; Wu et al., 2022; Xu et al., 2023; Song et al., 2024; 2023b). We prove that the generated data distribution satisfies SE(3) invariance, as provided in Appendix A.7. For atom types, we follow 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 (Hoogeboom 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 Hoogeboom 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.

Additionally, our approach can achieve superior results with significantly fewer generative steps. A detailed comparison, presented in Figure 1, demonstrates that SLDM achieves 100× faster sampling than the baseline EDM and around 70× 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.

4.3. Conditional Molecular 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.

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.

Table 2. Conditional generation result on QM9 dataset, evaluated by mean absolute error (MAE) between the property condition and the properties of generated molecules predicted by a pretrained EGNN classifier. Lower values indicate better performance. Our SLDM uses sampling steps $T = 1000$.

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
GeoBFN	2.34	577	328	516	0.998	0.949
GCDM	1.97	602	344	479	0.844	**0.689**
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. Higher values indicate better performance.

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

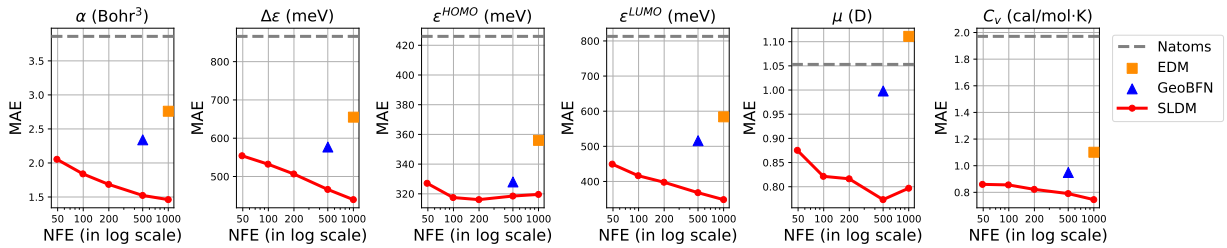

Figure 4. Comparison of performance in conditional generation (↓), with respect to the number of function evaluations (NFE) during the sampling process.

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 ablation study about temperature control in Appendix C.4.

4.5. Evaluation on Toy Dataset

We provide a toy dataset experiment in Appendix C.3, 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.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgements

This work is supported by Beijing Academy of Artificial Intelligence (BAAI).

References

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, pp. 1–3, 2024.

Axelrod, S. and Gomez-Bombarelli, R. Geom, energy-annotated molecular conformations for property predic-

tion and molecular generation. *Scientific Data*, 9(1):185, 2022.

Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., et al. Video generation models as world simulators, 2024.

Chen, C., Li, C., Chen, L., Wang, W., Pu, Y., and Duke, L. C. Continuous-time flows for efficient inference and density estimation. In *International Conference on Machine Learning*, pp. 824–833. PMLR, 2018.

Chen, S., Chewi, S., Li, J., Li, Y., Salim, A., and Zhang, A. 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=zyLVMgsZ0U_.

Cornet, F., Bartosh, G., Schmidt, M., and Andersson Naesseth, C. Equivariant neural diffusion for molecule generation. *Advances in Neural Information Processing Systems*, 37:49429–49460, 2024.

Daigavane, A., Kim, S. E., Geiger, M., and Smidt, T. 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=MIEnYtlGyv.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Feng, S., Ni, Y., Lu, Y., Ma, Z.-M., Ma, W.-Y., and Lan, Y. Unigem: A unified approach to generation and property prediction for molecules. *arXiv preprint arXiv:2410.10516*, 2024.

Garcia Satorras, V., Hoogeboom, E., Fuchs, F., Posner, I., and Welling, M. E (n) equivariant normalizing flows. *Advances in Neural Information Processing Systems*, 34: 4181–4192, 2021.

Gebauer, N., Gastegger, M., and Schütt, K. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. *Advances in neural information processing systems*, 32, 2019.

Graves, A., Srivastava, R. K., Atkinson, T., and Gomez, F. Bayesian flow networks. *arXiv preprint arXiv:2308.07037*, 2023.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pp. 8867–8887. PMLR, 2022.

Hua, C., Luan, S., Xu, M., Ying, Z., Fu, J., Ermon, S., and Precup, D. Mudiff: Unified diffusion for complete molecule generation. In *Learning on Graphs Conference*, pp. 33–1. PMLR, 2024.

Huang, H., Sun, L., Du, B., and Lv, W. Learning joint 2d & 3d diffusion models for complete molecule generation. *arXiv preprint arXiv:2305.12347*, 2023a.

Huang, L., Zhang, H., Xu, T., and Wong, K.-C. Mdm: Molecular diffusion model for 3d molecule generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 5105–5112, 2023b.

Irwin, R., Tibo, A., Janet, J. P., and Olsson, S. 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.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35: 26565–26577, 2022.

Köhler, J., Klein, L., and Noé, F. Equivariant flows: exact likelihood generative learning for symmetric densities. In *International conference on machine learning*, pp. 5361–5370. PMLR, 2020.

Kornilov, N. M., Mokrov, P., Gasnikov, A., and Korotin, A. 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.

Le, T., Cremer, J., Noé, F., Clevert, D.-A., and Schütt, K. Navigating the design space of equivariant diffusion-based generative models for de novo 3d molecule generation. *International Conference on Learning Representations*, 2024a.

Le, T., Cremer, J., Noé, F., Clevert, D.-A., and Schütt, K. Navigating the design space of equivariant diffusion-based generative models for de novo 3d molecule generation. *International Conference on Learning Representations (ICLR)*, 2024b.

Li, L., Li, H., Zheng, X., Wu, J., Xiao, X., Wang, R., Zheng, M., Pan, X., Chao, F., and Ji, R. 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, pp. 7105–7114, 2023.

Li, Z., Zhou, C., Wang, X., Peng, X., and Zhang, M. Geometric representation condition improves equivariant molecule generation. *arXiv preprint arXiv:2410.03655*, 2024.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Liu, X., Gong, C., and Liu, Q. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. 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.

Luo, S., Tan, Y., Huang, L., Li, J., and Zhao, H. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

Luo, Y. and Ji, S. An autoregressive flow model for 3d molecular geometry generation from scratch. In *International conference on learning representations (ICLR)*, 2022.

Mercatali, G., Verma, Y., Freitas, A., and Garg, V. Diffusion twigs with loop guidance for conditional graph generation. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 137741–137767. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/f90338be8ad676ee02cc32d239fc40e7-Paper-Conference.pdf.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

Peebles, W. and Xie, S. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Peng, X., Guan, J., Liu, Q., and Ma, J. Moldiff: Addressing the atom-bond inconsistency problem in 3d molecule diffusion generation. *International conference on machine learning*, 2023.

Ramakrishnan, R., Dral, P. O., Rupp, M., and Von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Ruddigkeit, L., Van Deursen, R., Blum, L. C., and Reymond, J.-L. 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.

Salimans, T. and Ho, J. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

Satorras, V. G., Hoogeboom, E., and Welling, M. E (n) equivariant graph neural networks. In *International conference on machine learning*, pp. 9323–9332. PMLR, 2021.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

Song, Y., Dhariwal, P., Chen, M., and Sutskever, I. Consistency models. *International Conference on Learning Representations*, 2023a.

Song, Y., Gong, J., Zhou, H., Zheng, M., Liu, J., and Ma, W.-Y. Unified generative modeling of 3d molecules with bayesian flow networks. In *The Twelfth International Conference on Learning Representations*, 2023b.

Song, Y., Gong, J., Xu, M., Cao, Z., Lan, Y., Ermon, S., Zhou, H., and Ma, W.-Y. Equivariant flow matching with hybrid probability transport for 3d molecule generation. *Advances in Neural Information Processing Systems*, 36, 2024.

Tong, A., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., FATRAS, K., Wolf, G., and Bengio, Y. 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.

Vignac, C., Osman, N., Toni, L., and Frossard, P. Midi: Mixed graph and 3d denoising diffusion for molecule generation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 560–576. Springer, 2023.

Wu, L., Gong, C., Liu, X., Ye, M., and Liu, Q. Diffusion-based molecule generation with informative prior bridges. *Advances in Neural Information Processing Systems*, 35: 36533–36545, 2022.

Xu, M., Yu, L., Song, Y., Shi, C., Ermon, S., and Tang, J. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.

Xu, M., Powers, A. S., Dror, R. O., Ermon, S., and Leskovec, J. Geometric latent diffusion models for 3d molecule generation. In *International Conference on Machine Learning*, pp. 38592–38610. PMLR, 2023.

Xue, K., Zhou, Y., Nie, S., Min, X., Zhang, X., ZHOU, J., and Li, C. 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.

Xue, S., Liu, Z., Chen, F., Zhang, S., Hu, T., Xie, E., and Li, Z. Accelerating diffusion sampling with optimized time steps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8292–8301, 2024b.

Xue, S., Yi, M., Luo, W., Zhang, S., Sun, J., Li, Z., and Ma, Z.-M. Sa-solver: Stochastic adams solver for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024c.

Zhou, Z., Wang, X., and Yu, T. Generating physical dynamics under priors. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=eNjXcP6C0H.

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):

$$\mathrm{d}X_t = f(t)X_t\mathrm{d}t + g(t)\mathrm{d}W_t, \tag{10}$$

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:

$$X_t = X_0 \cdot \exp(\int_0^t f(\xi)\mathrm{d}\xi) + \int_0^t \exp(\int_s^t f(\xi)\mathrm{d}\xi)g(s)\mathrm{d}W_s, t \in [0, \infty) \tag{11}$$

A quick proof is as follows:

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

$$\begin{aligned} \mathrm{d}(X_t \cdot \exp(-\int_0^t f(\xi)\mathrm{d}\xi)) &= (\mathrm{d}X_t - X_t f(t)\mathrm{d}t) \cdot \exp(-\int_0^t f(\xi)\mathrm{d}\xi) \\ &= \exp(-\int_0^t f(\xi)\mathrm{d}\xi) \cdot g(t)\mathrm{d}W_t \end{aligned} \tag{12}$$

Next, we integrate both sides and obtain:

$$X_t \cdot \exp(-\int_0^t f(\xi)\mathrm{d}\xi)) - X_0 = \int_0^t \exp(-\int_0^s f(\xi)\mathrm{d}\xi) \cdot g(s)\mathrm{d}W_s. \tag{13}$$

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

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:

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

$$\mathbb{COV}[X_t] = E[(\int_0^t exp(\int_s^t f(\xi)\mathrm{d}\xi)g(s)\mathrm{d}W_s)^2]\boldsymbol{I} = \int_0^t (exp(\int_s^t f(\xi)d\xi)g(s))^2\mathrm{d}s\boldsymbol{I}. \tag{15}$$

The covariance computation uses the Itô isometry property.

To fit the process X_t to the general form

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

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

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

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

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

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

$$\mathrm{d}X_t = \frac{\dot{\mu}(t)}{\mu(t)} X_t \mathrm{d}t + \sqrt{2\sigma(t)\dot{\sigma}(t) - 2\sigma(t)^2 \frac{\dot{\mu}(t)}{\mu(t)}} \mathrm{d}W_t, \tag{19}$$

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) (Hoogeboom et al., 2022) employs a schedule closely resembling the cosine noise schedule introduced by Nichol & Dhariwal (2021), albeit with a simplified notation:

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

The **Variance Exploding (VE)** schedule, initially proposed in the context of Denoising Score Matching (Song & 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:

$$\boldsymbol{x}_t = \boldsymbol{x}_0 + \sqrt{t}\boldsymbol{\epsilon}, t \in [0, T_{max}]. \tag{21}$$

By rescaling time, we derive the following expression:

$$\boldsymbol{x}_t = \boldsymbol{x}_0 + \sqrt{t}\sigma_{max}\boldsymbol{\epsilon}, t \in [0, 1], \tag{22}$$

where σ_{max} needs to be sufficiently large to ensure that x_1 approximates a uniform distribution. For illustrative clarity, we set $\sigma_{max} = 10$ for Figure 2 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:

$$\boldsymbol{x}_t = \boldsymbol{x}_0 + t\boldsymbol{\epsilon}, t \in [0, T_{max}]. \tag{23}$$

By rescaling time, we derive the following expression:

$$\boldsymbol{x}_t = \boldsymbol{x}_0 + t\sigma_{max}\boldsymbol{\epsilon}, t \in [0, 1], \tag{24}$$

For illustrative clarity, we set $\sigma_{max} = 10$ for Figure 2 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:

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

where σ_{min} needs to be set sufficiently small to ensure that $\boldsymbol{x}_t$ 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 $\sigma_{\text{min}} = 0.001$ for both Figure 2 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:

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

where σ_{min} needs to be set small to satisfy x_t align with the data distribution when $t = 0$. Following the settings in (Song et al., 2023b), we set $\sigma_{min} = 0.001$ for Figure 2 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 (Hoogeboom et al., 2022) and GeoBFN (Song et al., 2023b), 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:

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

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 $\boldsymbol{x}_0 \sim \delta_{\boldsymbol{a}}(\boldsymbol{x})$, $\boldsymbol{a} \in \mathbb{R}^N$. In this scenario, $\boldsymbol{x}_t$ follows a normal distribution $\mathcal{N}(\mu(t)\boldsymbol{a}, \sigma(t)^2\boldsymbol{I}_N)$ according to equation 1. The score function is then tractable as $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t) = -\frac{\boldsymbol{x}-\mu(t)\boldsymbol{a}}{\sigma(t)^2}$. Substituting this into equation 3 gives:

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

We aim to keep $\frac{d\boldsymbol{x}}{dt}$ 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 $x_{t=0}$ approximates data distribution and $x_{t=T}$ 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:

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

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 $\boldsymbol{x}_t = (1-t)\boldsymbol{x}_0 + \sigma\boldsymbol{\epsilon}$, $t \in [0,1]$, the following inequality holds for each data dimension i:*

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

As $\sigma \to 0$, the term $\frac{d\boldsymbol{x}_t}{dt} + \frac{\boldsymbol{x}_t}{1-t}$ converges to zero in probability. More specifically, the solution to the equation $\frac{d\boldsymbol{x}_t}{dt} + \frac{\boldsymbol{x}_t}{1-t} = 0$ is that $\frac{\boldsymbol{x}_t}{1-t}$ 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 $f(\boldsymbol{x}_0)$ and a general diffusion process with the schedule $\boldsymbol{x}_t = \mu(t)\boldsymbol{x}_0 + \sigma(t)\boldsymbol{\epsilon}$, its ODE description in equation 3 can be rewritten as*

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

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

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

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

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

□

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

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

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

$$\mathbb{E}\boldsymbol{y} = \mathbb{E}[\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{a}] - \boldsymbol{a}] = \mathbb{E}_a\mathbb{E}_{\boldsymbol{x}_0|\boldsymbol{a}}[\boldsymbol{x}_0 - \boldsymbol{a}] = \mathbb{E}_{\boldsymbol{x}_0}\mathbb{E}_{\boldsymbol{a}|\boldsymbol{x}_0}[\boldsymbol{x}_0 - \boldsymbol{a}] = 0 \tag{34}$$

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

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

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:

$$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = -\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]. \tag{36}$$

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 $\boldsymbol{y} = \mathbb{E}[\boldsymbol{x}_0|\boldsymbol{a}] - \boldsymbol{a}$ satisfies $\mathbb{E}[\boldsymbol{y}] = \mathbf{0}$ and $\mathrm{Var}[y^{(i)}] \leq \sigma(t)^2/\mu(t)^2 = \frac{\sigma^2}{(1-t)^2}$, for all dimension i.

Applying Chebyshev's inequality, we get:

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

Thus, we have the inequality:

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

□

A.5. Supplementary Proof for Sampling

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

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

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

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

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

□

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

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

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

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

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

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

where we use x_t and $x_t - \Delta t$ are independent given x_0.

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

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

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

As proved in (Xue et al., 2024c), 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:

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

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

$$d\boldsymbol{x} = \frac{1}{2}g(t)^2\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x})dt + \tau g(t)\mathrm{d}w'. \tag{46}$$

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

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

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

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

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

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

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

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 & 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:

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

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.4.

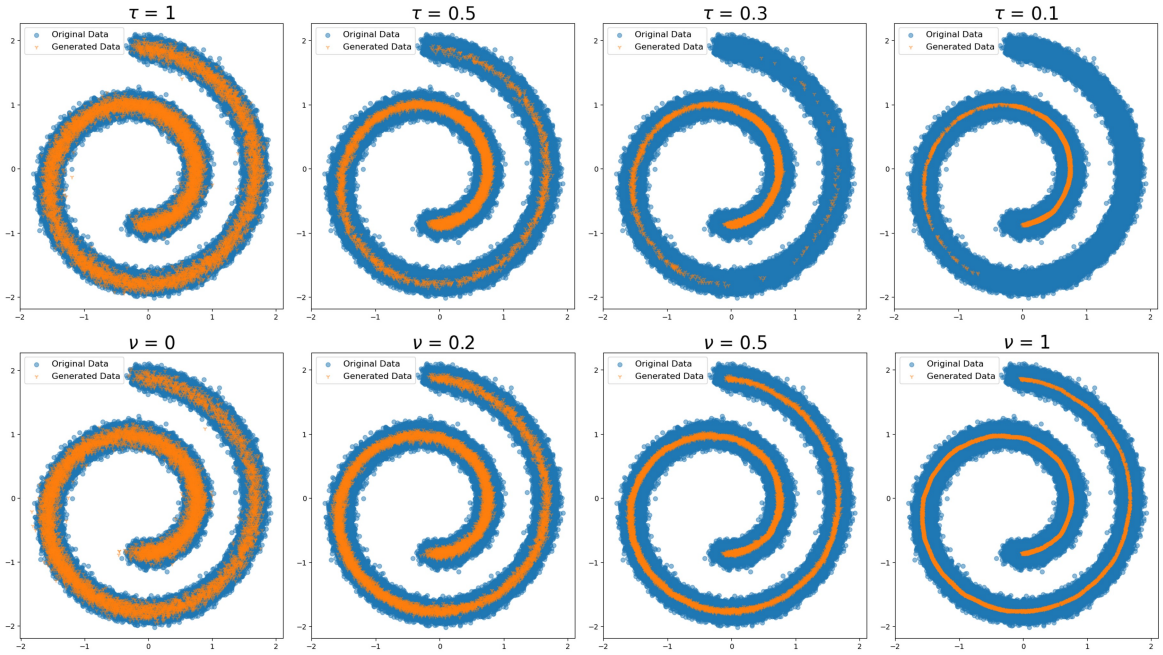

Figure 5. **a.** Generation results from straight-line diffusion with vanilla temperature control as defined in equation 46. Training data (blue points) represent 2D Swiss roll coordinates with added Gaussian noise, where the highest density region lies in the interior of the spiral. Diffusion-generated 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.

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 $p(\boldsymbol{x}_0) = p(\boldsymbol{x}_0 + \boldsymbol{t})$ and $p(\boldsymbol{x}_0) = p(\boldsymbol{R}\boldsymbol{x}_0)$, where $\boldsymbol{t}$ is a translation vector and $\boldsymbol{R}$ is an orthogonal matrix representing a rotation. The proof is in the same spirit of that in Hoogeboom 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).

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 $\boldsymbol{\epsilon} \sim \mathcal{N}_{\text{CoM}}(\boldsymbol{0}, I_{3M})$, which is sampled by first sampling a standard Gaussian noise vector $\boldsymbol{\epsilon}' \sim \mathcal{N}(\boldsymbol{0}, I_{3M})$ and then subtracting its center of mass: $\boldsymbol{\epsilon} = \boldsymbol{\epsilon}' - \frac{1}{3M}\sum_{i=1}^{3M} \epsilon'_i$. Second, the network's output at each step is projected into the zero CoM space. These operations ensure that all intermediate coordinates $\boldsymbol{x}_t, t = 0, \cdots, 1$ 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 $\phi^{(x)}(\boldsymbol{R}\boldsymbol{x}_t, t) = \boldsymbol{R}\phi^{(x)}(\boldsymbol{x}_t, t)$. 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: $f_{\mathcal{N}_{\text{CoM}}}(\boldsymbol{x}) = \frac{1}{(2\pi)^{3(M-1)/2}} exp(\frac{1}{2}||\boldsymbol{x}||^2)$. Then, we can verify $f_{\mathcal{N}_{\text{CoM}}}(\boldsymbol{x}) = f_{\mathcal{N}_{\text{CoM}}}(\boldsymbol{R}\boldsymbol{x})$ for any orthogonal rotation matrix $\boldsymbol{R}$. Now we prove the rotation invariance of the generation probability as follows:

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

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

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

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

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

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 $\boldsymbol{z}_0 = [\boldsymbol{x}_0, \boldsymbol{h}_0]$ with M atoms, 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 $\boldsymbol{\epsilon} \sim \mathcal{N}_{\text{CoM}}(\boldsymbol{0}, I_{3M})$
  Compute $\boldsymbol{x}_t = (1-t)\boldsymbol{x}_0 + \sigma\boldsymbol{\epsilon}$
  Minimize $||\boldsymbol{\epsilon} - \phi^{(x)}(\boldsymbol{x}_t, t)||^2 + 1_{t\le t_n}|\boldsymbol{h}_0 - \phi^{(h)}(\boldsymbol{x}_t, t)|$
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 $\boldsymbol{x}_1 \sim \sigma\cdot\mathcal{N}_{\text{CoM}}(\boldsymbol{0}, I_{3M})$
for $i = T-1$ **to** 1 **do**
  $t = i/T$, $\Delta t = 1/T$
  $\boldsymbol{x}_{t-\Delta t} = \frac{1-(t-\Delta t)}{1-t}\cdot(\boldsymbol{x}_t - \sigma\cdot\phi(\boldsymbol{x}_t, t))$
  Projecting $\boldsymbol{x}_{t-\Delta t}$ into zero CoM space
  if $i > 1$ **then**
    Sample $\boldsymbol{\epsilon} \sim \mathcal{N}_{\text{CoM}}(\boldsymbol{0}, I_{3M})$
    $\boldsymbol{x}_{t-\Delta t} = \boldsymbol{x}_{t-\Delta t} + t^{\nu}\sqrt{2}\cdot\sigma\cdot\boldsymbol{\epsilon}$
  else
    $\boldsymbol{h}_0 = \phi^{(h)}(\boldsymbol{x}_0, 0)$
  end if
end for
Return: $\boldsymbol{z}_0 = [\boldsymbol{x}_0, \boldsymbol{h}_0]$

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

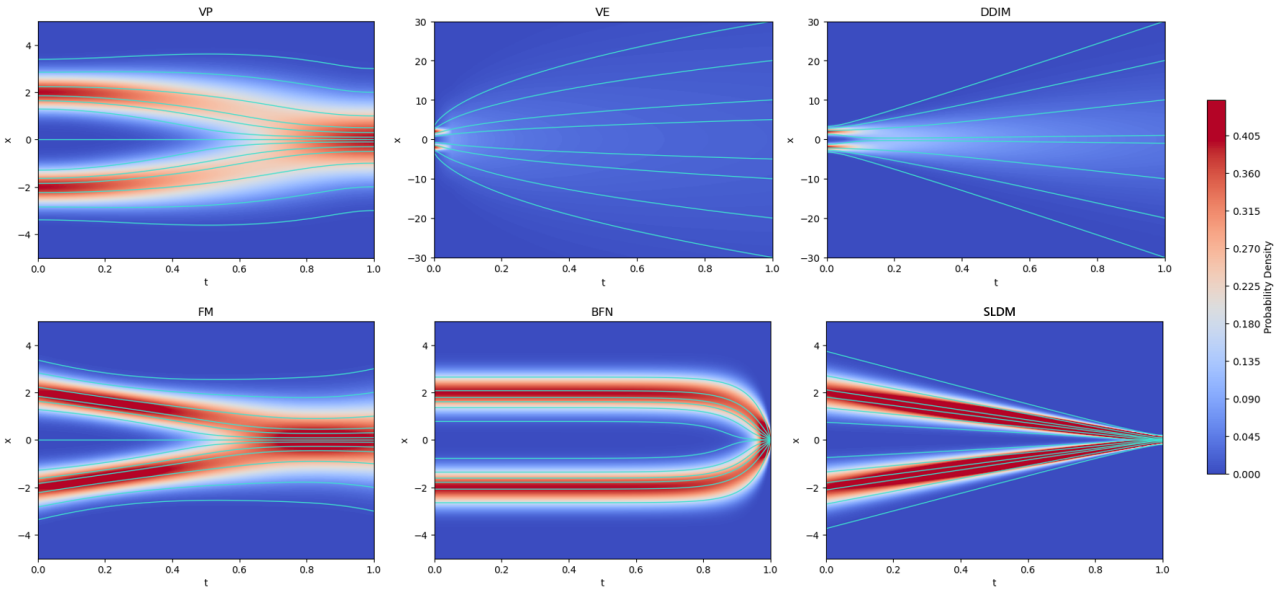

Figure 6. The trajectory of the ODE in equation 3 of various diffusion-based models. The data distribution is a mixture of Gaussians in one dimension, defined as $x_0 \sim 0.5 \cdot \mathcal{N}(2, 1/4) + 0.5 \cdot \mathcal{N}(-2, 1/4)$. The background color indicates the value of probability density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

trajectory that is consistently closer to a straight line compared to other approaches, leading to smaller truncation 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.

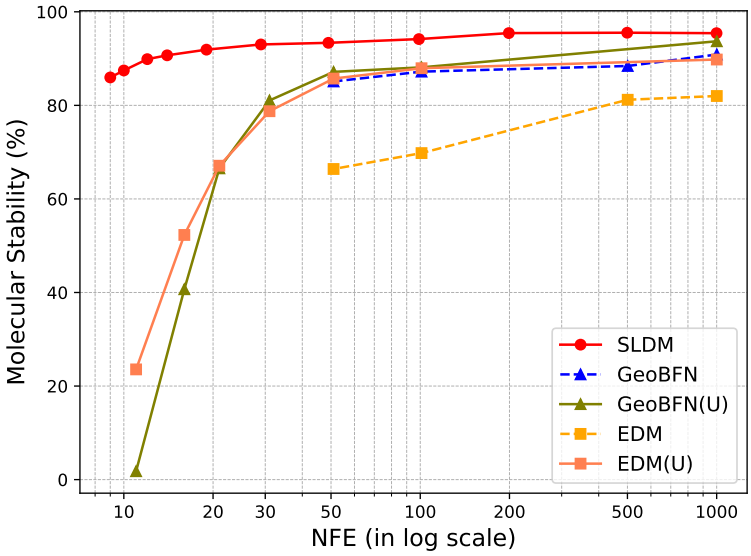

Figure 7. Comparison of molecule stability (↑) across diffusion-based molecular generation models with UniGEM framework on QM9 unconditional generation, evaluated with respect to the Number of Function Evaluations (NFE) during the sampling process.

C.3. 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, 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.4. Ablation Study for Temperature Control

Theoretically, a larger temperature annealing rate v corresponds to a faster cooling scheme. The results in Table 4 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.

Table 4. 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

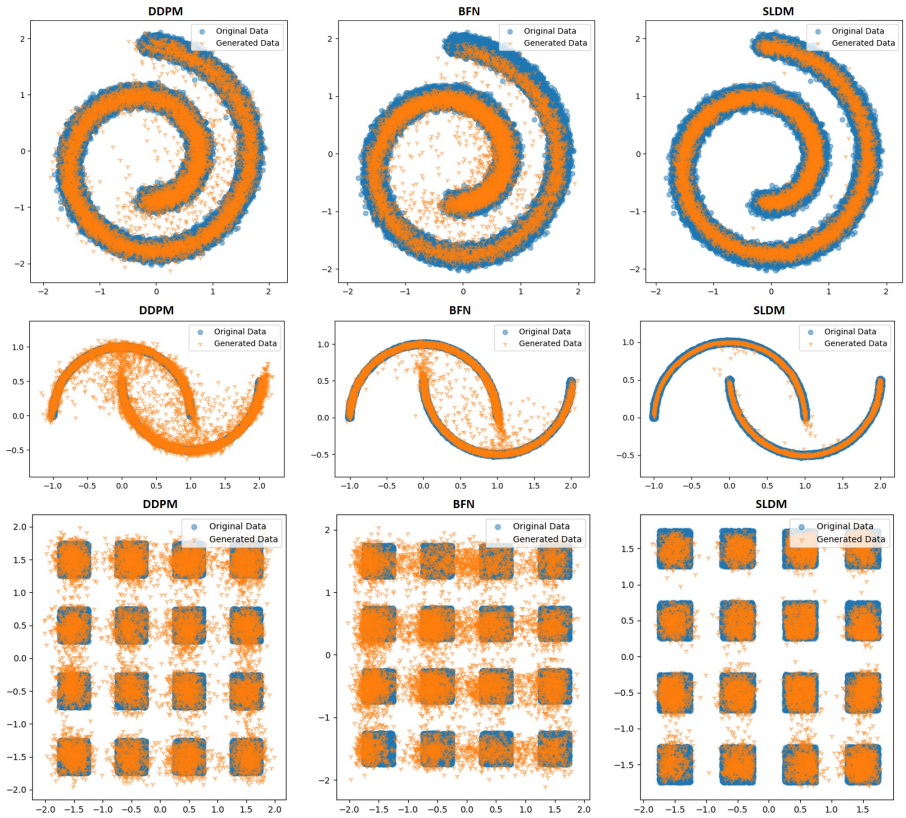

Figure 8. Generative performance comparison of Straight-Line Diffusion, DDPM, and BFN on 2D toy datasets.

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 & 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. Hoogeboom et al. (Hoogeboom 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., 2023b) 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., 2023b), 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., 2023b; Le et al., 2024b; Irwin et al., 2024)(Huang et al., 2023a; 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., 2023a), EQGAT-diff (Le et al., 2024b), 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 (Hoogeboom 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., 2023b), adheres to this evaluation standard. In contrast, another line of studies (Vignac et al., 2023; Le et al., 2024b; 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 & Dhariwal, 2021; Le et al., 2024a), time discretization strategies (Xue et al., 2024b; 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 $\boldsymbol{\mu}(\mathbf{t})$. 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 & Ho, 2022), consistency model (Song et al., 2023a; Luo et al., 2023) and optimal flow matching (Kornilov et al., 2024). Thus, 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 5. 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.

For QM9, it takes approximately 10 days on a single A100 GPU. For GEOM-drugs, it takes approximately 16 days on four A100 GPUs. 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

Table 5. 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 $T > 13$

values as additional inputs during training and sampling them from a prior distribution during inference. The results are measured using a property classifier 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.