FlashMol: High-Quality Molecule Generation in as Few as Four Steps

Xinyuan Wei[1,2*] **Zian Li**[1,3*] **Shaoheng Yan**[1,2] **Cai Zhou**[4] **Muhan Zhang**[1,5†]

[1]Institute for Artificial Intelligence, Peking University
[2]Yuanpei College, Peking University
[3]School of Intelligence Science and Technology, Peking University
[4]Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology
[5]State Key Laboratory of General Artificial Intelligence, Peking University

Project Page: `https://sylvan-wesley.github.io/flashmol.github.io/`

Abstract

Generating chemically valid 3D molecular conformations is critical for computational drug discovery. Classical diffusion-based models like GeoLDM perform well but require hundreds of steps, making large-scale in silico screening impractical. Recent efforts on few-step molecular generation have accelerated this process to 12-50 steps, but they often largely sacrifice sample stability. In this work, we present FlashMol, an ultra-fast molecule generative model producing high-quality molecular conformations in as few as 4 steps. To achieve this, we adapt distribution matching distillation (DMD) — a reverse KL-divergence minimization objective — to the molecular domain for effective distillation. Considering the local minimization behavior of DMD, we respace the molecule generation timesteps, providing the generator with much better initialization and enables effective distillation. Additionally, to mitigate the mode-seeking behavior of DMD and improve diversity, we further regularize it with a Jensen-Shannon divergence term, which incorporates the mean-seeking behavior of the forward KL divergence. Extensive experiments on QM9 and GEOM-DRUG datasets demonstrate that FlashMol matches and even surpasses the original 1000-step teacher: by applying respacing to the GeoLDM teacher (GeoLDM-respaced), we achieve comparable results to the 1000-NFE teacher with as few as 16 NFEs. After further distillation, FlashMol is able to sample with only 4 NFEs while maintaining high molecular quality, achieving up to 250x acceleration compared to the GeoLDM teacher.

1 Introduction

Molecule generative models have emerged as a cornerstone of computational molecular design, offering the prospect of exploring chemical space at scales that are infeasible for traditional physics-based simulation [14, 49, 24, 16]. Among these approaches, diffusion-based molecule generative models such as EDM [14] and GeoLDM [49] produce stable and valid molecules through iterative denoising, demonstrating strong potential in domains such as drug discovery [47, 22, 4], protein engineering [52], and materials science [9].

However, generating a molecule with diffusion-based methods typically requires hundreds to thousands of sequential network evaluations. For example, classic baselines including EDM [14] and GeoLDM [49] require 1000 NFEs, and even more efficient flow-matching variants such as SemlaFlow [16] and ET-Flow [10] still require more than a hundred steps. Although recent few-step

* Equal Contribution.
† Correspondence to Muhan Zhang <muhan@pku.edu.cn>.

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

generative methods [56, 29, 24] have reduced NFE to the range of 12–50 steps, they often come with substantially lower molecule stability, and no existing method, to the best of our knowledge, achieves reliable generation in 10 steps or fewer, a regime that is increasingly common in image and video domains [15, 7, 39]. Sampling speed is therefore a major bottleneck preventing these models from being deployed in the high-throughput screening pipelines essential for drug discovery.

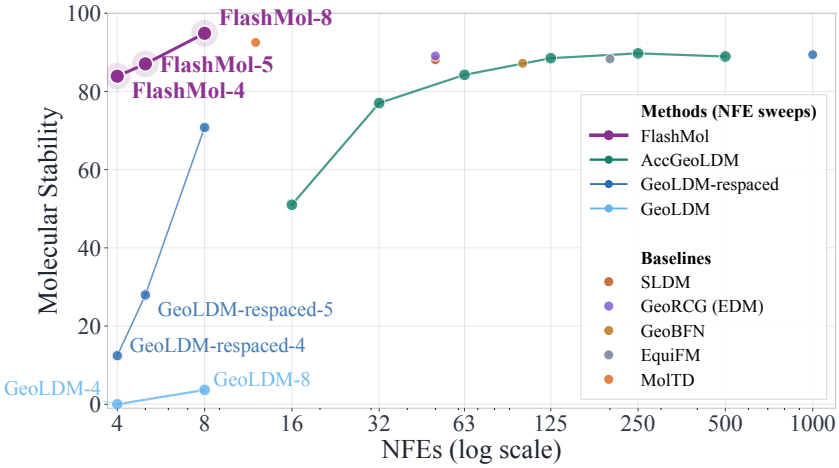

Figure 1: **Molecule generation quality on QM9.** We achieve comparable or superior performance on QM9 with only 4 to 8 steps, outperforming recent few-step molecule generative models. GeoLDM-respaced uses our respaced timesteps (Section 4.2); GeoLDM uses the original DDPM timesteps.

In this work, we propose FlashMol, an ultra-fast molecule generative model that generates high-quality molecules in **as few as 4 steps**, as shown in Figure 1. To achieve this, we distill a few-step student from its 1000-step teacher via distribution matching distillation [54, 53], a reverse-KL method that encourages high-fidelity samples and has shown strong promise in image and video domains [54, 53, 18, 15]. We identify two key obstacles to applying DMD in molecular generation. First, the local optimization behavior of DMD requires a well-initialized few-step student to provide useful optimization signals. Yet molecular structures are far less tolerant of generation errors than images: small deviations in bond lengths and angles can destroy chemical validity, making the initial few-step student much less likely to produce reasonable samples (nearly 0 % stability). To address this issue, we draw inspiration from the EDM design space analysis [20] and **redesign the discrete timestep schedule at inference and distillation training time**, without any retraining. This respacing substantially improves few-step quality and provides a viable student initialization for DMD. Second, the mode-seeking behavior of DMD gives the student a smaller probability support than the teacher, reducing molecular diversity. This issue is especially pronounced on QM9 [33], which nearly enumerates the space of small molecules [49]. To address this issue, we leverage a Jensen-Shannon divergence [51], which **combines forward and reverse KL divergence, to regularize support shrinkage**, and treat it as a regularization term with a small coefficient, successfully improving the generation diversity. To summarize, our contributions are as follows:

- We integrate distribution matching distillation into molecular generation using the latent molecular generative model GeoLDM [49].
- We identify inference-time timestep scheduling as a key bottleneck for effective distribution distillation and introduce a respacing strategy inspired by EDM [20] that substantially improves the initial few-step generator even without training, and enables effective distillation.
- We extend the vanilla DMD objective with a Jensen-Shannon divergence term and provide a unified integration of KL-divergence-related objectives in molecular settings.
- We achieve **83.89% molecular stability** on QM9 and **84.72% atom stability** on GEOM-DRUG with only **4 steps**.

2 Related Work

Diffusion and Flow Models for 3D Molecule Generation Advances in molecular generation can be broadly categorized based on the output modality: 2D graph generation and 3D coordinate generation. Early work on molecular generation primarily focused on 2D graph generation [45, 32, 57], where molecules are represented as graphs of atoms and bonds [17]. However, recent research has shifted towards generating 3D coordinates of atoms, allowing models to capture the spatial arrangement of molecules, which is crucial for their chemical properties. Prominent models such as the Equivariant Diffusion Model (EDM) [14] and GeoLDM [49] have demonstrated impressive performance in both conditional and unconditional molecule generation tasks by adopting diffusion models. Despite their successes, these models often require a thousand function evaluations (NFEs) to generate a single sample. To address this limitation, flow-based models [16, 42, 6] reduce the number of NFEs required, yet they still typically require over a hundred evaluations. Other methods, such as GeoRCG [24] and REED [46], use molecular representations to enhance generation and can generate stable molecules in as few as 50 steps. A recent method [56] introduces molecule-specific trajectory diagnosis, which accelerates sampling and achieves strong performance with as few as 12 steps.

Distillation and Distribution Matching for Accelerated Sampling Efforts to accelerate diffusion models generally focus on two strategies. The first involves progressive distillation [35], consistency models [39, 38], shortcut models [27, 7, 25, 58], and flow map distillation [3, 43], where a student model is trained to replicate a teacher model's denoising trajectory. While this approach reduces the required steps, it suffers from error accumulation as the trajectory length increases. The second strategy, distribution matching distillation [54, 53, 18, 51, 43], uses a reverse KL-divergence objective to match the output distributions of the teacher and the student models. The reverse KL-divergence objective leads to mode-seeking behavior and preserves the sharpness of main modes, achieving high-quality individual samples [53]. This line of distribution matching work has led to state-of-the-art performance in one-step generation and, since molecules are less tolerant to noise, we identify distribution matching objective as particularly suited to the molecule domain.

3 Preliminaries

3.1 Molecule Diffusion Models

In this paper, a molecule with N atoms is represented as a point cloud $\mathcal{G} = \langle x, h \rangle$, where $x \in \mathbb{R}^{N\times 3}$ denotes 3D coordinates and $h \in \mathbb{R}^{N\times d}$ denotes atom features. FlashMol adopts GeoLDM [49] as its backbone due to its simplicity, which encodes $\mathcal{G}$ into a continuous latent molecule $z_0 = \langle z_x, z_h \rangle$ and decodes the final latent sample back to molecular coordinates and features. The coordinate latent z_x is kept in the zero-center-of-mass subspace to preserve translation invariance.

The latent diffusion process corrupts z_0 using a predefined noise schedule (α_t, σ_t): $z_t = \alpha_t z_0 + \sigma_t \epsilon, \epsilon \sim \mathcal{N}(0, I)$. The denoising networks are implemented with equivariant architectures [14], so rotations of coordinate latents induce corresponding rotations in the predicted coordinates while atom-feature channels remain invariant. Our method keeps this GeoLDM latent-space formulation, but distills a few-step student generator to replace the expensive multi-step teacher. Additional details on the original molecule diffusion formulation are provided in Appendix A.1.

3.2 Distribution Matching Distillation

Distribution matching distillation (DMD) [53, 54] aims to minimize the KL divergence between the few-step student model and the multi-step teacher model $\mathrm{KL}(p_{\text{fake}} \| p_{\text{real}})$. Let $F(x, t) = \alpha_t x + \sigma_t \epsilon$ denote the corruption process for non-geometric objects x, with $\epsilon \sim \mathcal{N}(0, I)$. The corresponding DMD loss has a gradient equivalent to:

$$\nabla \mathcal{L}_{\text{DMD}} = \mathbb{E}_t \left(\nabla_\theta \mathrm{KL}(p_{\text{fake},t} \| p_{\text{real},t}) \right) \tag{1}$$

$$= -\mathbb{E}_t \left(\int \left(s_{\text{real}}(F(G_\theta(z), t), t) - s_{\text{fake},\phi}(F(G_\theta(z), t), t) \right) \frac{dG_\theta(z)}{d\theta} \, dz \right). \tag{2}$$

Here, $G_\theta(z)$ denotes the few-step student generator, starting from noise $z \sim \mathcal{N}(0, I)$, whose generated samples x follow $x \sim p_{\text{fake}}$. The distributions $p_{\text{fake},t}$ and $p_{\text{real},t}$ correspond to the diffusion

distributions defined by the forward operator F of the student and teacher models, respectively. The score functions [40] are defined as $s_{\text{real}}(x,t) = \nabla \log p_{\text{real},t}(x)$ and $s_{\text{fake},\phi}(x,t) = \nabla \log p_{\text{fake},t}(x)$, where $s_{\text{fake},\phi}(x,t)$ serves as a proxy multi-step score for the few-step student generator and is updated using the diffusion loss computed on samples produced by G_θ [54] while $s_{\text{real}}(x,t)$ is approximated by the teacher model and is frozen during the training process.

The teacher model is parameterized as μ_{real} and $s_{\text{fake},\phi}$ as $\mu_{\text{fake},\phi}$, which are both noise-prediction diffusion models [53]; the denoising output ϵ_ϕ of $\mu_{\text{fake},\phi}$ is converted to $s_{\text{fake},\phi}$ via the score-based SDE framework [40].Following DMD2 [53], in each iteration, $\mu_{\text{fake},\phi}$ is first updated for 5 steps using the diffusion loss on the generated sample of the student model:

$$\mathcal{L}_{\text{fake}} = \mathbb{E}_{x\sim p_{\text{fake}},\epsilon\sim\mathcal{N}(0,I)}\big[w(t)||\epsilon_\phi(F(x,t),t) - \epsilon||^2\big] \tag{3}$$

where x is the sample generated by the student model G_θ and $\epsilon_\phi(F(x,t),t)$ is the output of μ_{fake}. The student generator G_θ is updated by the gradient of the DMD loss $\mathcal{L}_{\text{DMD}}$ for one step afterwards.

As in DMD2 [53], few-step generator samples in a consistency-sampling manner [39]. Given a generator G, the final samples are obtained by repeatedly applying:

$$x^{t+1} = G(z^t), \quad z^{t+1} = \alpha_{t+1}x^{t+1} + \sigma_{t+1}\epsilon \tag{4}$$

where $\epsilon, z^0 \sim \mathcal{N}(0,I)$ and α_t, σ_t are fixed noise schedules. This iterative refinement significantly improves the sample quality of the alternative one-step generator [53].

3.3 f-Divergences for Distribution Matching Distillation

DMD optimizes the reverse KL loss between student and teacher distributions. More generally, one can consider f-divergences of the form $D_f(p||q) = \int q(x) f(\frac{p(x)}{q(x)})dx$. As derived in Xu et al. [51],

$$\nabla_\theta D_f(p_{\text{real}}||p_{\text{fake}}) = \mathbb{E}_{\epsilon,t}\big[-\int f''(r)r^2\big(s_{\text{real}}(F(G_\theta(z),t),t) - s_{\text{fake},\phi}(F(G_\theta(z),t),t)\big)\frac{dG_\theta(z)}{d\theta}\,dz\big] \tag{5}$$

where $r = \frac{p_{\text{real}}(x)}{p_{\text{fake}}(x)}$. For the Jensen-Shannon divergence, $f(r) := r\log r - (r+1)\log\frac{r+1}{2}$. To approximate r, we can introduce a discriminator D trained with the GAN objective, which yields the approximation $r \approx D(x)/(1-D(x))$.

4 Methods

4.1 Geometric Distribution Matching Distillation

We now describe the key designs of the FlashMol framework, which adapts the DMD approach to the molecular domain with carefully tailored mechanisms, achieving remarkable few-step generation performance. We choose GeoLDM [49], a strong latent molecule generative model, as the backbone of FlashMol. The overall training framework is illustrated in Figure 2. Following DMD [54], our training uses three models: the few-step generator G_θ, the frozen multi-step teacher score estimator μ_{real}, and the proxy score estimator μ_{fake} for the student generator. In each training iteration, the student generator G_θ first produces latent molecules $\mathcal{G} = \langle x, h\rangle$ from Gaussian noise z via K-step consistency sampling, as defined in Equation (4), without decoding to the molecule space. During sampling, the coordinate component x is always projected onto the zero-mean subspace to preserve translation invariance. To let μ_{fake} faithfully approximate the score of the student generator, we first update μ_{fake} according to the diffusion loss in Equation (3) for 5 steps following [53]. Then, the generated latent molecules are passed through μ_{real} and μ_{fake} to estimate the scores $\nabla \log p_t(\langle x_t, h_t\rangle)$, which yield the DMD gradients following Equation (1) for updating the student generator G_θ.

Notably, in the geometric formulation of DMD, equivariance is always preserved: the resulting DMD gradient is invariant to rotation of the student samples. This property improves data efficiency in few-step distillation for equivariant molecular generators. For simplicity, we assume G to be a one-step generator. Formally,

Proposition 1 (invariance)**.** *The distribution matching loss*

$$\nabla\mathcal{L}_{DMD}(z;\theta) = -\mathbb{E}_t\left(\int\big(s_{real}(F(G_\theta(z),t),t) - s_{fake,\phi}(F(G_\theta(z),t),t)\big)\frac{dG_\theta(z)}{d\theta}\,dz\right)$$

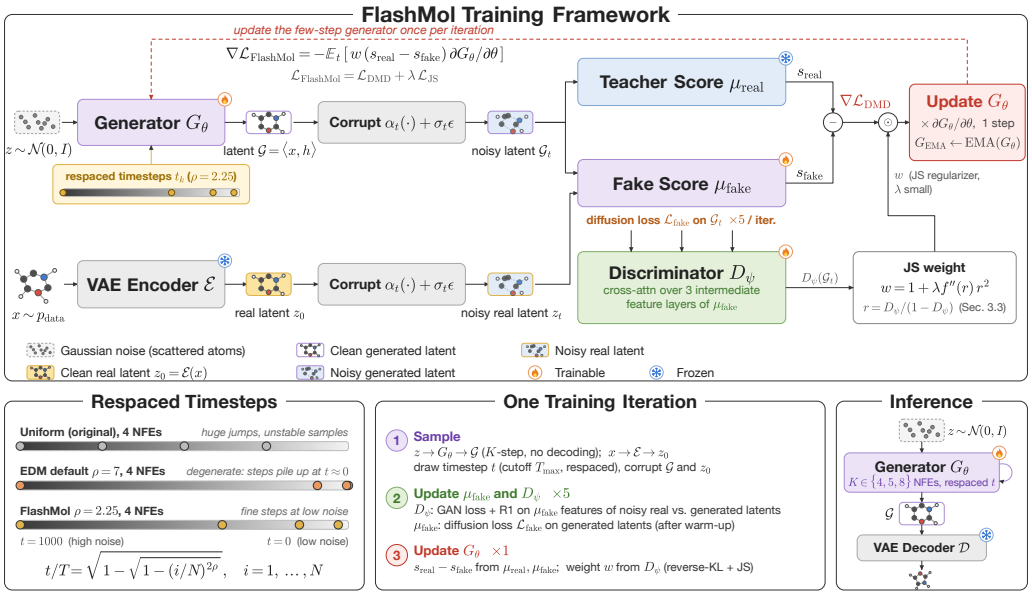

Figure 2: **Training framework for FlashMol.** In each iteration, the few-step generator G produces a batch of sampled molecules. Then, the μ_{fake} model which approximates fake score s_{fake} and the discriminator D are first updated using diffusion loss and GAN loss respectively for 5 steps. After that, μ_{fake} and the discriminator D's output are used to compute the DMD and Jensen-Shannon divergence gradient. Lastly, the gradients are mixed and update the generator G.

is invariant *to SE(3) group, where Gaussian noise* $z = \langle z_x, z_h \rangle \in \mathbb{R}^{N\times(3+d)}$ *and the group transformation applies only on* z_x*. Formal definitions and proof are in Appendix B.*

At inference, the few-step generator samples in consistency sampling manner with K-steps, and the sampled latent molecules are decoded using the VAE inherited from the pretrained GeoLDM model.

4.2 Escaping Poor Initial Modes via Respaced Timesteps

Directly adapting DMD to molecules poses significant optimization challenges. In this subsection, we identify the key issue and show how the proposed timestep respacing strategy addresses it.

Local Optimization Behavior of DMD Theoretically, the DMD loss is a complete loss in the sense that its optimal solution aligns the student and teacher distributions nearly perfectly, assuming unlimited model capacity. In practice, however, the reverse KL divergence, $\mathrm{KL}(p_{\text{fake}} \| p_{\text{real}}) = \mathbb{E}_{x\sim p_{\text{fake}}}\big[\log p_{\text{fake}}(x) - \log p_{\text{real}}(x)\big]$, induces a local optimization behavior [18, 2, 55], because **the samples are drawn from the student model**, rather than the teacher or data distribution as in the case of the forward KL. Consequently, if the initial student model cannot generate reasonable (even if imperfect) samples, the DMD loss in Equation (1) provides limited optimization signals.

Different from other modalities such as images, molecule generation exhibits two distinctive characteristics: (1) its process is highly sensitive to noise [29]; and (2) molecular bond lengths and angles must be encoded with precise fine-grained detail, meaning that low-noise regions require careful handling. Directly initializing a few-step student model from a multi-step teacher model and sampling with uniformly spaced timesteps [37] makes huge jump in the denoising trajectory and introduces severe discretization error. On the other hand, higher NFEs allow the model to further refine its details in later steps [12], leading to better quality. As a result, the few-step initial student produces poor-quality molecular samples. As shown in the top row of Figure 3, the 4-step student samples generated with the original timestep schedule are highly noisy and chemically invalid—they often contain disconnected atoms due to large deviations in atom-pair distances. Eventually, directly optimizing such a model with the DMD loss will lead to ineffective few-step distillation. In our initial experiments, we observe that this causes gradient explosions even before the first DMD training epoch

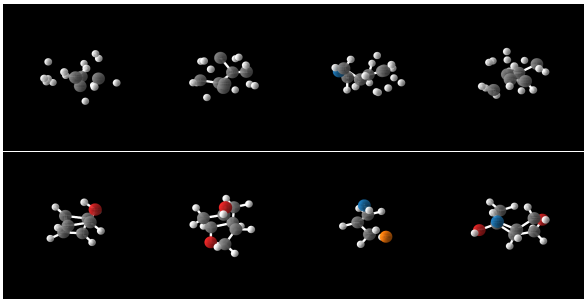

Figure 3: Random 4-step samples from the multi-step teacher without additional training. Top: original GeoLDM schedule. Bottom: our respaced schedule.

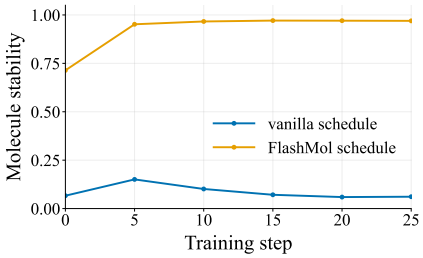

Figure 4: Distribution matching distillation training dynamics under different sampling noise schedules.

is completed and results in poor-quality samples after training. As shown in Figure 4 (blue line), an 8-step student model sampled with uniform timesteps fails to achieve effective DMD optimization.

Respaced Timesteps for GeoLDM To address this challenge, we devise a respaced timestep inspired by EDM [20], which shows that the optimal schedule, in the sense of minimized truncation error, for the noise levels $\{\sigma_i\}_{i=0}^{N-1}$ takes the form:

$$\sigma_i = \left(\sigma_{\max}^{1/\rho} + \frac{i}{N-1}\left(\sigma_{\min}^{1/\rho} - \sigma_{\max}^{1/\rho}\right)\right)^{\rho}, \quad i = 0, \ldots, N-1. \tag{6}$$

Crucially, this schedule front-loads large denoising steps at high noise levels and reserves many small steps for the low-noise regime, where fine geometric detail is recovered.

However, GeoLDM inherits the variance preserving [40] schedule $\sigma_t = \sqrt{1-\alpha_t^2}$ with $\alpha_t = 1-(t/T)^2$ and $T = 1000$, which allocates timesteps roughly uniformly in noise-variance space. This conflicts with the EDM-style timestep. We therefore select the timesteps according to the noise-level schedule in Equation (6). Specifically, we first choose noise levels uniformly from Equation (6), and then select the corresponding timesteps from the GeoLDM schedule to align with these noise levels. Under this design, the discrete step $t \in [0, T]$ corresponding to the continuous-time noise level σ_i satisfies $\sigma_i^2 = 1-\alpha_t^2 = 1-(1-(t/T)^2)^2$. Inverting this relationship gives the formula:

$$\frac{t}{T} = \sqrt{1-\sqrt{1-(i/(N-1))^{2\rho}}}, \tag{7}$$

and we apply this respaced schedule in both the few-step generator G and the score estimator μ_{real} and μ_{fake}, ensuring compatibility during training.[1]

Furthermore, we investigate the selection of ρ in the schedule. We observe that the original configuration with $\rho = 7$ in Equation (7) leads to the diffusion step $t \approx 0$ for more than 38% of the 1000 denoising steps, which is a degenerate case and demonstrates poor sampling quality in molecule generation. This is visualized in Figure 5. Thus, we experiment and obtain the optimal $\rho = 2.25$, which distinguishes between the t values in $[0, 300]$ while remaining concentrated on low-noise regions. This is justified by our ablation experiments in Section 5.4.

Benefits of Respacing After timestep respacing, the generated molecules are dramatically more stable (bottom row of Figure 3). Although not perfect before distillation, they provide a good starting point for escaping the poor initial modes in DMD training. Consequently, DMD training is stabilized, with no significant gradient explosion even in later stages as shown in Figure 4 (orange line).

4.3 Improving Diversity via f-Divergence Regularization

DMD adopts a reverse KL-divergence objective which severely **penalizes the student generator for placing probability mass outside the teacher's support**, thereby damaging sampling diversity. By choosing the Jensen-Shannon divergence , the resulting objective $D_{\text{JS}}(p_{\text{real}}||p_{\text{fake}})$ can be interpreted

[1] In practice, we adopt $\frac{t}{T} = \sqrt{1-\sqrt{1-(i/(N))^{2\rho}}}$ to avoid square-rooting a negative number.

as the mean of $\mathrm{KL}(p_{\text{fake}} \,\|\, (p_{\text{fake}} + p_{\text{real}})/2)$ and $\mathrm{KL}(p_{\text{real}} \,\|\, (p_{\text{fake}} + p_{\text{real}})/2)$, which incorporates both forward- and reverse-KL components [26]. This adds mean-seeking behavior via the forward-KL component, counteracting DMD's reverse-KL mode seeking and improving sample diversity. In FlashMol, we introduce this f-divergence as a regularization term complementary to the DMD loss, resulting in the final loss:

$$\mathcal{L}_{\text{FlashMol}} = \mathcal{L}_{\text{DMD}} + \lambda \cdot \mathcal{L}_{\text{JS}}, \tag{8}$$

where $\mathcal{L}_{\text{JS}}$ follows the definition in Equation (5) with $f(r) := r \log r - (r+1) \log \frac{r+1}{2}$. However, direct differentiation of the loss requires the computation of $\frac{p_{\text{fake}}(\mathcal{G})}{p_{\text{real}}(\mathcal{G})}$ given a molecule $\mathcal{G}$, which is intractable [51]. Therefore, we approximate this term using output of a GAN discriminator [8] according to the formula in Section 3.3. Following DMD2 [53], we train the discriminator on both the generated molecules of the few-step generator and real molecules in the dataset. To better capture both high-level semantics and low-level details, we extract three intermediate layers of features from the μ_{fake} network. Then, for each layer, we perform cross attention [44] with a learnable token query for the real/fake logits. We leave the detailed implementation in Appendix C.1 and the final algorithm in Algorithm C.1.

5 Experiments

In this section, we show the remarkable performance of FlashMol on various molecule generative tasks. We first describe the experimental settings in Section 5.1. Then, we report quantitative results and acceleration-quality trade-offs across different datasets and metrics in Section 5.2 and Section 5.3. Finally, we provide an ablation study on the key designs of FlashMol, including time respacing and distillation loss, in Section 5.4.

5.1 Experimental Settings

Teacher Models For unconditional generation tasks, the teacher model is obtained from GeoLDM [49]. For conditional generation, we train the conditional GeoLDM teacher from scratch. Implementation details can be found in Appendix C. We also train another molecule generative model (SemlaFlow [16]) to demonstrate the transferability to another backbone and present the results in Appendix D.4.

Tasks and Metrics We experiment with both unconditional and conditional generation following standard evaluation protocols [14, 56]. Specifically, we evaluate FlashMol on QM9 [33] dataset, which consists of 130K small molecules (with up to 29 atoms including hydrogens each), for both unconditional and conditional generation, and on GEOM-DRUG [1] dataset, which consists of 450K larger molecules with an average atom number of 44 and maximum of 181, for unconditional generation. For unconditional generation, we report the generated molecules' 2D graph properties, including atom stability, molecule stability, validity and uniqueness following GeoLDM, SLDM, and MOLTD [49, 29, 56]. Since our models output atom types and atoms' 3D coordinates directly, bond types are computed via look-up tables according to pairwise distances and atom types [14]. In addition to those metrics, we record PoseBusters metrics in Table D.1 following [5, 19], which evaluate 3D conformational properties, including bond length, bond angles and ring flatness. Furthermore, we report the number of function evaluations (NFE) to compare sampling efficiency across various models. For conditional generation, we report the mean absolute error (MAE) of the specific property between the generated molecule and the specified condition value.

5.2 Unconditional Generation on QM9 and GEOM-DRUG

Baselines Our direct baseline is GeoLDM [49], the multi-step teacher of FlashMol. To highlight the effectiveness of the distillation training, we compare directly to the GeoLDM with respaced timesteps (Section 4.2), denoted as GeoLDM-respaced. We further compare against competitive baselines spanning multiple generative paradigms, including: 1) Diffusion-based models, including EDM and its non-equivariant counterpart (GDM) [14], as well as EDM-Bridge [48] which extends EDM by incorporating informative prior bridges; 2) flow-based models, such as EquiFM [42], GOAT [13], GeoBFN [41] and ADiT [19]; 3) advanced few-step molecule generative methods, including AccGeoLDM [23], SLDM [29], GeoRCG [24] and MOLTD [56]. Notably, several methods

Table 1: Unconditional molecule generation results of FlashMol. We present models of three different step counts and our method achieves 250× to 125× acceleration as well as state-of-the-art performance on various datasets. We highlight metrics in which our student model surpasses the respaced teacher using the same NFEs. GEOLDM-respaced results are obtained using our sampling method; see Section 4.2. The MOLTD V&U entry is estimated by us.

		QM9				DRUG	
# Metrics	NFE	Atom Sta (%)	Mol Sta (%)	Valid (%)	Valid & Unique (%)	Atom Sta (%)	Valid (%)
Data	-	99.0	95.2	97.7	97.7	86.5	99.9
GDM-AUG	1000	97.6	71.6	90.4	89.5	77.7	91.8
EDM	1000	98.7	82.0	91.9	90.7	81.3	92.6
EDM-Bridge	1000	98.8	84.6	92.0	90.7	82.4	92.8
GeoLDM	1000	98.9	89.4	93.8	**92.7**	84.4	99.3
EquiFM	200	98.9	88.3	94.7	93.5	84.1	98.9
GeoBFN	100	98.6	87.2	93.0	91.5	78.9	93.1
GOAT	90	99.2	-	92.9	92.0	84.8	96.2
SLDM	50	98.70	88.09	92.84	89.53	-	-
GeoRCG (EDM)	50	98.75	89.08	95.05	-	81.44	95.70
AccGeoLDM	16	92.08	51.02	73.22	72.39	-	-
MOLTD	12	99.40	92.53	96.04	92.0	86.88	95.33
GEOLDM-respaced	8	95.49 ± 0.1	70.78 ± 0.4	87.13 ± 0.3	81.48 ± 0.4	64.35	90.44
FLASHMOL	**8**	**99.55** ± 0.0	**94.87** ± 0.5	**96.97** ± 0.2	87.51 ± 0.4	**89.98**	**99.86**
GEOLDM-respaced	5	82.11 ± 0.1	27.96 ± 0.2	62.10 ± 0.1	59.20 ± 0.1	18.27	82.07
FLASHMOL	**5**	98.73 ± 0.0	87.05 ± 0.4	92.67 ± 0.3	78.83 ± 0.2	87.69	99.69
GEOLDM-respaced	4	72.35 ± 0.1	12.43 ± 0.3	45.24 ± 0.3	42.53 ± 0.3	8.30	84.65
FLASHMOL	**4**	97.70 ± 0.1	83.89 ± 0.3	90.70 ± 0.2	72.70 ± 0.3	84.72	99.72

that directly generate 2D bonds, including SemlaFlow [16], are not compared against our models regarding 2D properties for fairness. This is because metrics such as stability and validity are essentially 2D metrics; they can easily reach very high values with 2D generative methods, yet this cannot reflect a method's ability to learn 3D distributions. Therefore, we only include them to compare 3D Posebusters [5] metrics in Table D.1.

Results and Analysis Table 1 presents the main results, and we highlight several important observations as below.

First, with **the fewest NFEs**, FlashMol achieves **the best stability and validity** on both QM9 and GEOM-DRUG. Specifically, on QM9, FlashMol achieves 94.87% molecule stability with only 8 NFEs, substantially outperforming GeoLDM-respaced, which achieves 70.78%. It also surpasses strong few-step generative methods, including GeoRCG (89.08% with 50 NFEs) and MOLTD (92.53% with 12 NFEs). On GEOM-DRUG, FlashMol even exceeds the dataset-level atom stability of 86.5%, achieving 89.98% atom stability.

Second, under extremely smaller NFE budgets, such as 4 and 5 NFEs, FlashMol significantly **lifts the near-unstable GeoLDM-respaced baseline** to a performance level **comparable to 1000-NFE baselines**. On QM9, without FlashMol training, GeoLDM-respaced achieves only 12.43% and 27.96% molecule stability with 4 and 5 NFEs, indicating that almost no stable molecules can be generated with such few sampling steps. In contrast, FlashMol achieves 83.89% and 87.05% molecule stability, approaching advanced performance with 250× and 200× less computation compared to the original 1000-step GeoLDM baseline. The gain is even more pronounced on GEOM-DRUG, where the 5-step FlashMol maintains the best performance among all compared methods.

Third, FlashMol shows the **diversity metric V&U** to a level **comparable with advanced methods**. For example, the 8-step FlashMol achieves 87.51% V&U, compared with approximately 90% for the baselines. It is known that reverse-KL-based objectives can lead to substantial diversity loss for the improvement of fidelity [54, 18], which is also validated in our setting as shown in Figure D.1. This is because samples from the student distribution that lie outside the support of the teacher distribution incur a large penalty, encouraging p_{fake} to concentrate on the high-probability modes of the teacher distribution while discarding low-probability regions. However, FlashMol successfully mitigates this effect by introducing an f-divergence loss, whose effectiveness is further validated in Section 5.4. We note that although the V&U score of FlashMol remains slightly lower than those of some baselines, we view this **as an opportunity rather than a limitation** as discussed in DMDR [18]: steering the optimized distribution toward high-confidence regions is often desirable in practical few-step generation scenarios where sample quality is prioritized over exhaustive distribution coverage.

Table 2: Conditional molecule generation results are evaluated on various properties of the molecule. We report the MAE(↓) between generated molecules. *QM9* and *Random* results serve as lower and upper bounds of MAE across all properties.

Property Units	NFE -	α Bohr^3	$\Delta\varepsilon$ meV	$\varepsilon_{\text{HOMO}}$ meV	$\varepsilon_{\text{LUMO}}$ meV	μ D	C_v $\frac{\text{cal}}{\text{mol K}}$
QM9	-	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	1000	2.76	655	356	584	1.111	1.101
GeoLDM	1000	2.37	587	340	522	1.108	1.025
GeoBFN	100	2.34	577	**328**	516	0.998	0.949
GeoLDM-respaced	**32**	2.14	565	342	491	**0.985**	0.933
FLASHMOL	**32**	**2.12**	**556**	330	**489**	0.988	**0.926**

Table 3: Ablation studies. **Left**: comparisons of using different ρ's during sampling. **Middle**: ablation on the divergence regularizer. **Right**: ablation on the sampling method.

Schedule exponent ρ

ρ	Mol Sta (%)	V&U (%)
2	69.93	81.27
2.25	**70.78**	**81.48**
2.75	68.46	81.35
5	35.23	61.95

Divergence regularizer

Divergence Term	Mol Stab (%)	V&U (%)
DMD only	81.51	68.84
Forward KL ($\lambda = 0.1$)	58.09	59.10
JS only	83.34	72.36
DMD + JS ($\lambda = 0.1$)	**83.89**	**72.70**

Sampling Method

NFE	Respaced	DDIM	Consistency
4	**12.43**±0.30	0.01±0.01	0.00±0.00
8	**70.78**±0.39	5.06±0.09	6.71±0.39
16	**94.61**±0.18	41.84±0.42	59.94±0.49

5.3 Conditional Generation on Molecular Properties

Conditional generation evaluates the model's ability to generate molecules that satisfy a specific property. Following GeoLDM [49], for each property condition, we inject it as a node feature in the few-step student, the score estimator μ_{fake} and the teacher model. The property values of the generated molecules are predicted by a pretrained property-specific classifier. Since the evaluation of this task relies solely on the 3D coordinates of the generated molecule, higher precision is required. As more NFEs yield finer-grained details, we choose a higher NFE budget (32 NFEs), which is still, to our knowledge, fewer than latest baselines in conditional generation on QM9. We provide implementation details and experiment results of models with fewer steps in Appendix C.4 and Appendix D.2.

As presented in Table 2, FlashMol **achieves comparable or better results** than other methods in almost all metrics with **the fewest NFEs**. In $\varepsilon_{\text{HOMO}}$ and $\varepsilon_{\text{LUMO}}$, FlashMol yields 330meV and 489meV MAE respectively compared to 340meV and 522meV produced by the 1000-step GeoLDM teacher. This demonstrates the applicability of FlashMol to various scenarios.

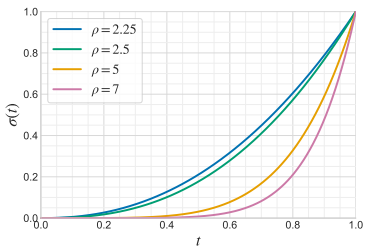

Figure 5: Noise schedules for different values of the exponent ρ in the respaced timesteps in Equation (7).

5.4 Ablation Studies

Effect of Respaced Schedule on Sampling Quality To isolate the contribution of the noise schedule, we sample with the GeoLDM teacher on QM9 without distillation under three conditions and report molecule stability: (1) uniform schedule with DDIM [37], as in GeoLDM [49]; (2) uniform schedule with consistency sampling (Consistency); (3) FlashMol's respaced schedule with consistency sampling (Respaced). As shown in Table 3, condition (3) **achieves 94.61% molecule stability** and surpasses trained few-step models of MOLTD [56] at 16 NFEs **without any distillation training**. Consistency sampling alone yields a small additional gain over DDIM.

Sensitivity to the Schedule Exponent ρ To characterize the sensitivity of the sampling process to ρ in Equation (7), we vary ρ and sample using the GeoLDM teacher on QM9 without distillation training; results are shown in Table 3. $\rho = 2.25$ yields the best stability and uniqueness. An excessively large ρ concentrates sampling steps in the low-noise regime, losing the coarse structural deformation that early high-noise steps provide: sampling with $\rho = 5$ yields only 35.23% molecule

stability, compared to 70.78% at $\rho = 2.25$. Conversely, smaller ρ's closely resemble the original DDIM schedule, limiting their ability to improve sample quality.

Effect of f-Divergence Choice on Stability and Diversity We compare the FlashMol objective in Equation (8) with its variants: 1) DMD only ($\lambda = 0$); 2) Forward-KL, substituting $f(r) := r \log r$ in Equation (5), which corresponds to the forward-KL objective $\mathrm{KL}(p_{\text{real}}||p_{\text{fake}})$ [51]; 3) Jensen-Shannon only ($\mathcal{L}_{\text{JS-only}} = \mathcal{L}_{\text{JS}}$). As shown in Table 3 and Table D.3, combining JS-divergence regularization with the DMD loss yields the best overall performance: JS-divergence **boosts Valid & Unique by approximately 4% at 4 NFEs** by increasing sampling diversity, while DMD loss alone improves molecule stability but **reduces Valid & Unique**, indicating collapsed probability support. Forward KL-divergence causes dramatic instability due to high-variance estimation [51], incompatible with DMD's reverse-KL objective. More details can be found in Appendix D.3.

6 Conclusion

We propose FlashMol, a few-step generative model for 3D molecules that achieves up to $250\times$ acceleration over its 1000-step GeoLDM teacher while matching or exceeding teacher-level sample quality. Our approach adapts distribution matching distillation to the molecular domain through two key designs: an EDM-inspired respacing of generation timesteps that supplies the few-step student with a viable initialization, and a Jensen-Shannon divergence term that regularizes the reverse-KL DMD objective with a forward-KL component to mitigate mode-seeking behavior. FlashMol attains comparable generation quality in as few as 4 steps and state-of-the-art molecular stability and validity on QM9 and GEOM-DRUG in 8 steps.

Acknowledgement

This work is supported by the Beijing Natural Science Foundation (Z260022) and National Natural Science Foundation of China (62561160152).

References

[1] Simon Axelrod and Rafael Gómez-Bombarelli. Geom, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 2022.

[2] Lichen Bai, Zikai Zhou, Shitong Shao, Wenliang Zhong, Shuo Yang, Shuo Chen, Bojun Chen, and Zeke Xie. Optimizing few-step generation with adaptive matching distillation. *arXiv preprint arXiv:2602.07345*, 2026.

[3] Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation. *arXiv preprint arXiv:2505.18825*, 2025.

[4] Michal Brylinski and Grover Waldrop. Computational redesign of bacterial biotin carboxylase inhibitors using structure-based virtual screening of combinatorial libraries. *Molecules*, 2014.

[5] Martin Buttenschoen, Garrett M Morris, and Charlotte M Deane. Posebusters: Ai-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical science*, 15(9):3130–3139, 2024.

[6] Ian Dunn and David R. Koes. Flowmol3: flow matching for 3d de novo small-molecule generation. *Digital Discovery*, 2026.

[7] Zhengyang Geng, Mingyang Deng, Xingjian Bai, Jeremy Z. Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025.

[8] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 2014.

[9] Rebecca L. Greenaway and Kim E. Jelfs. Integrating computational and experimental workflows for accelerated organic materials discovery. *Advanced Materials*, 2021.

[10] Majdi Hassan, Nikhil Shenoy, Jungyoon Lee, Hannes Stark, Stephan Thaler, and Dominique Beaini. Equivariant flow matching for molecular conformer generation. *ICML 2024 Workshop*, 2024.

[11] Stephen R Heller, Alan McNaught, Igor Pletnev, Stephen Stein, and Dmitrii Tchekhovskoi. Inchi, the iupac international chemical identifier. *Journal of cheminformatics*, 7(1):23, 2015.

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

[13] Haokai Hong, Wanyu Lin, and Kay Chen Tan. Accelerating 3d molecule generation via jointly geometric optimal transport. *arXiv preprint arXiv:2405.15252*, 2024.

[14] Emiel Hoogeboom, Victor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. *Proceedings of the 39th International Conference on Machine Learning*, 2022.

[15] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025.

[16] Ross Irwin, Alessandro Tibo, Jon Paul Janet, and Simon Olsson. Semlaflow–efficient 3d molecular generation with latent attention and equivariant flow matching. *arXiv preprint arXiv:2406.07266*, 2024.

[17] Yunhui Jang, Dongwoo Kim, and Sungsoo Ahn. Hierarchical graph generation with K^2-trees. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023.

[18] Dengyang Jiang, Dongyang Liu, Zanyi Wang, Qilong Wu, Liuzhuozheng Li, Hengzhuang Li, Xin Jin, David Liu, Changsheng Lu, Zhen Li, Bo Zhang, Mengmeng Wang, Steven Hoi, Peng Gao, and Harry Yang. Distribution matching distillation meets reinforcement learning. *arXiv preprint arXiv:2511.13649*, 2025.

[19] Chaitanya K Joshi, Xiang Fu, Yi-Lun Liao, Vahe Gharakhanyan, Benjamin Kurt Miller, Anuroop Sriram, and Zachary W Ulissi. All-atom diffusion transformers: Unified generative modelling of molecules and materials. *arXiv preprint arXiv:2503.03965*, 2025.

[20] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 2022.

[21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] Xiangzhe Kong, Wenbing Huang, Zhixing Tan, and Yang Liu. Molecule generation by principal subgraph mining and assembling. *Advances in Neural Information Processing Systems*, 2022.

[23] Romain Lacombe and Neal Vaidya. Accelerating the generation of molecular conformations with progressive distillation of equivariant latent diffusion models. *arXiv preprint arXiv:2404.13491*, 2024.

[24] Zian Li, Cai Zhou, Xiyuan Wang, Xingang Peng, and Muhan Zhang. Geometric representation condition improves equivariant molecule generation. *arXiv preprint arXiv:2410.03655*, 2024.

[25] Haitao Lin, Peiyan Hu, Minsi Ren, Zhifeng Gao, Zhi-Ming Ma, Guolin ke, Tailin Wu, and Stan Z. Li. On the design of one-step diffusion via shortcutting flow paths. *arXiv preprint arXiv:2512.11831*, 2025.

[26] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 2002.

[27] Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. *arXiv preprint arXiv:2410.11081*, 2024.

[28] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[29] Yuyan Ni, Shikun Feng, Haohan Chi, Bowen Zheng, Huan ang Gao, Wei-Ying Ma, Zhi-Ming Ma, and Yanyan Lan. Straight-line diffusion model for efficient 3d molecular generation. *arXiv preprint arXiv:2503.02918*, 2025.

[30] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. *NeurIPS 2017 Workshop on Autodiff*, 2017.

[31] Peter Potaptchik, Jason Yim, Adhi Saravanan, Peter Holderrieth, Eric Vanden-Eijnden, and Michael S Albergo. Discrete flow maps. *arXiv preprint arXiv:2604.09784*, 2026.

[32] Yiming Qin, Manuel Madeira, Dorina Thanou, and Pascal Frossard. Defog: Discrete flow matching for graph generation. *arXiv preprint arXiv:2410.04263*, 2024.

[33] Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp, and Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 2014.

[34] Anthony K Rappé, Carla J Casewit, Kent S Colwell, William A Goddard III, and W Mason Skiff. Uff, a full periodic table force field for molecular mechanics and molecular dynamics simulations. *Journal of the American chemical society*, 114(25):10024–10035, 1992.

[35] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

[36] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

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

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

[39] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[40] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[41] Yuxuan Song, Jingjing Gong, Yanru Qu, Hao Zhou, Mingyue Zheng, Jingjing Liu, and Wei-Ying Ma. Unified generative modeling of 3d molecules with bayesian flow networks. *The Twelfth International Conference on Learning Representations*, 2024.

[42] Yuxuan Song, Jingjing Gong, Minkai Xu, Ziyao Cao, Yanyan Lan, Stefano Ermon, Hao Zhou, and Wei-Ying Ma. Equivariant flow matching with hybrid probability transport for 3d molecule generation. *Advances in Neural Information Processing Systems*, 2023.

[43] Shangyuan Tong, Nanye Ma, Saining Xie, and Tommi Jaakkola. Flow map distillation without data. *arXiv preprint arXiv:2511.19428*, 2025.

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 2017.

[45] Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. *arXiv preprint arXiv:2209.14734*, 2022.

[46] Chenyu Wang, Cai Zhou, Sharut Gupta, Zongyu Lin, Stefanie Jegelka, Stephen Bates, and Tommi Jaakkola. Learning diffusion models with flexible representation guidance. *arXiv preprint arXiv:2507.08980*, 2025.

[47] Wendy A. Warr, Marc C. Nicklaus, Christos A. Nicolaou, and Matthias Rarey. Exploration of ultralarge compound collections for drug discovery. *Journal of Chemical Information and Modeling*, 2022.

[48] Lemeng Wu, Chengyue Gong, Xingchao Liu, Mao Ye, and Qiang Liu. Diffusion-based molecule generation with informative prior bridges. *Advances in Neural Information Processing Systems*, 2022.

[49] Minkai Xu, Alexander Powers, Ron Dror, Stefano Ermon, and Jure Leskovec. Geometric latent diffusion models for 3d molecule generation. *Proceedings of the 40th International Conference on Machine Learning*, 2023.

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

[51] Yilun Xu, Weili Nie, and Arash Vahdat. One-step diffusion models with f-divergence distribution matching. *arXiv preprint arXiv:2502.15681*, 2025.

[52] Zehra Yildirim, Kyle Swanson, Xuekun Wu, James Zou, and Joseph Wu. Next-gen therapeutics: pioneering drug discovery with ipscs, genomics, ai, and clinical trials in a dish. *Annual Review of Pharmacology and Toxicology*, 2025.

[53] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in Neural Information Processing Systems*, 2024.

[54] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[55] Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Frédo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[56] Zhilong Zhang, Yuxuan Song, Yichun Wang, Jingjing Gong, Hanlin Wu, Dongzhan Zhou, Hao Zhou, and Wei-Ying Ma. Accelerating 3d molecule generative models with trajectory diagnosis. *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[57] Cai Zhou, Xiyuan Wang, and Muhan Zhang. Unifying generation and prediction on graphs with latent graph diffusion. *Advances in Neural Information Processing Systems*, 2024.

[58] Linqi Zhou, Mathias Parger, Ayaan Haque, and Jiaming Song. Terminal velocity matching. *arXiv preprint arXiv:2511.19797*, 2025.

Appendix

A Additional Preliminaries

A.1 Molecule Diffusion Models

We provide additional details on the molecule diffusion model summarized in Section 3. Following GeoLDM [49], a molecule with N atoms is represented as $\mathcal{G} = \langle x, h \rangle$, where $x \in \mathbb{R}^{N\times 3}$ denotes 3D atom coordinates and $h \in \mathbb{R}^{N\times d}$ denotes node features. GeoLDM first maps the molecule into a continuous latent representation with an encoder $\mathcal{E}_\phi$:

$$q_\phi(z_x, z_h \mid x, h) = \mathcal{N}(\mathcal{E}_\phi(x, h), \sigma_0^2 I).$$

Here, z_x is the coordinate latent and z_h is the feature latent. The coordinate latent is constrained to the zero-center-of-mass subspace, i.e., $\sum_i z_{x,i} = 0$, which removes global translation degrees of freedom. A decoder $\mathcal{D}_\varphi$ maps latent samples back to molecular coordinates and atom features.

The diffusion model operates entirely in this latent space. At each diffusion step, Gaussian noise is added to both latent coordinates and latent features according to a predefined schedule (α_t, σ_t):

$$q(z_{x,t}, z_{h,t} \mid z_{x,t-1}, z_{h,t-1}) = \mathcal{N}(z_{x,t}; \sqrt{\alpha_t} z_{x,t-1}, \sigma_t^2 I) \cdot \mathcal{N}(z_{h,t}; \sqrt{\alpha_t} z_{h,t-1}, \sigma_t^2 I).$$

Equivalently, the noised latent at timestep t can be written as

$$z_t = \alpha_t z_0 + \sigma_t \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I),$$

where $z_t = \langle z_{x,t}, z_{h,t} \rangle$. During this process, the coordinate component is projected back to the zero-center-of-mass subspace so that the latent representation remains translation invariant.

The reverse denoising process is parameterized by a neural network that predicts the added noise. The original diffusion training objective can be written as

$$\mathcal{L}_{DM} = \mathbb{E}_{(z_{x,0}, z_{h,0}) \sim q_\phi,\, \epsilon \sim \mathcal{N}(0,I),\, t} \left[\|\epsilon_x - \epsilon_{\theta,x}(z_{x,t}, z_{h,t}, t)\|^2 + \|\epsilon_h - \epsilon_{\theta,h}(z_{x,t}, z_{h,t}, t)\|^2 \right].$$

The denoiser therefore learns to recover both geometric and feature perturbations in latent space. In FlashMol, the teacher model μ_{real}, the few-step student generator G_θ, and the fake score estimator μ_{fake} all inherit this latent-space parameterization.

For geometric consistency, GeoLDM uses equivariant networks such as EGNN [36] in the encoder, decoder, and denoising model. These networks preserve SO(3)-equivariance for coordinate channels and SO(3)-invariance for feature channels. Informally, rotating the coordinate input rotates the coordinate output in the same way, while the feature output remains unchanged. Together with the zero-center-of-mass constraint and an invariant Gaussian prior, this yields a latent molecular distribution that is invariant to global rigid transformations [50]. This geometric structure is the basis for the invariance statement used in our distribution matching formulation.

At inference time, GeoLDM starts from Gaussian latent noise, applies iterative denoising in latent space, and decodes the final latent sample with $\mathcal{D}_\varphi$. FlashMol preserves this inference interface but distills the denoising process into a small number of consistency-sampling steps, as described in Sections 3.2 and 4.2.

B Formal Statements and Proofs

We provide the formal definition of invariance as follows:

Definition 1. *For a sample $z = \langle z_x, z_h \rangle \in \mathbb{R}^{N\times(3+d)} \sim \mathcal{N}(0, I)$ in the data distribution and the model parameter θ, a loss that is computed according to $\mathcal{L}(z;\theta)$ is* invariant *to SO(3) [2] group if and only if:*

$$\nabla_\theta \mathcal{L}(z;\theta) = \nabla_\theta \mathcal{L}(\mathbf{R}z;\theta)$$

for any $\mathbf{R} \in SO(3)$.

We use the notation $\mathbf{R}z$ to indicate group action of $\mathbf{R}$ on z, which is a matrix multiplication that gives $\mathbf{R}z = \langle z_x\mathbf{R}, z_h \rangle$ where $\mathbf{R} \in SO(3)$.

B.1 Proof of Proposition 1

Proof. $\forall \mathbf{R} \in SO(3), \mathbf{R}\mathbf{R}^T = I, \det(\mathbf{R}) = 1$, thus:

$$\nabla\mathcal{L}_{\text{DMD}}(\mathbf{R}z;t) = -\int p(\epsilon)\big(s_{\text{real}}(\alpha_t G_\theta(\mathbf{R}z) + \sigma_t\epsilon, t) - s_{\text{fake}}(\alpha_t G_\theta(\mathbf{R}z) + \sigma_t\epsilon, t)\big)\frac{dG_\theta(\mathbf{R}z)}{d\theta}\,d\epsilon$$

Given the equivariant property of EGNN [36], we denote $z' = \langle z'_x, z'_h \rangle = G_\theta(z)$, then $G_\theta(\mathbf{R}z) = \langle z'_x\mathbf{R}, z'_h \rangle$. Therefore, we denote $\bar{\mathbf{R}} = \text{diag}\{\mathbf{R}, I_d\}$, then $G_\theta(\mathbf{R}z) = G_\theta(z)\bar{\mathbf{R}}$, $\bar{\mathbf{R}}\bar{\mathbf{R}}^T = I$ and $\det(\bar{\mathbf{R}}) = 1$.

$$\begin{aligned}
\nabla\mathcal{L}_{\text{DMD}}(\mathbf{R}z;t) &= -\int p(\epsilon)\big(s_{\text{real}}(\alpha_t G_\theta(z) + \sigma_t\epsilon\bar{\mathbf{R}}^T, t) - s_{\text{fake}}(\alpha_t G_\theta(z) + \sigma_t\epsilon\bar{\mathbf{R}}^T, t)\big)\bar{\mathbf{R}}\bar{\mathbf{R}}^T\frac{dG_\theta(z)}{d\theta}\,d\epsilon \\
&= -\int p(\epsilon)\big(s_{\text{real}}(\alpha_t G_\theta(z) + \sigma_t\epsilon\bar{\mathbf{R}}^T, t) - s_{\text{fake}}(\alpha_t G_\theta(z) + \sigma_t\epsilon\bar{\mathbf{R}}^T, t)\big)\frac{dG_\theta(z)}{d\theta}\,d\epsilon \\
&= -\int p(\epsilon\bar{\mathbf{R}}^T)\big(s_{\text{real}}(\alpha_t G_\theta(z) + \sigma_t\epsilon\bar{\mathbf{R}}^T, t) - s_{\text{fake}}(\alpha_t G_\theta(z) + \sigma_t\epsilon\bar{\mathbf{R}}^T, t)\big)\frac{dG_\theta(z)}{d\theta}\,d\epsilon \\
&= -\int p(\epsilon)\big(s_{\text{real}}(\alpha_t G_\theta(z) + \sigma_t\epsilon, t) - s_{\text{fake}}(\alpha_t G_\theta(z) + \sigma_t\epsilon, t)\big)\frac{dG_\theta(z)}{d\theta}\,d\epsilon \\
&= \nabla\mathcal{L}_{\text{DMD}}(z;t)
\end{aligned}$$

□

It is worth noting that the ϵ added during the Markovian forward process is *not* equivariant, we need to prove that in expectation sense this does not matter. Given that $det(\mathbf{R}) = 1, \forall \mathbf{R} \in SO(3)$,

$$\begin{aligned}
\nabla\mathcal{L}_{\text{DMD}}(z;t) &= -\int p(\epsilon)\big(s_{\text{real}}(\alpha_t G_\theta(z) + \sigma_t\epsilon, t) - s_{\text{fake}}(\alpha_t G_\theta(z) + \sigma_t\epsilon, t)\big)\frac{dG_\theta(z)}{d\theta}\,d\epsilon \\
&= -\int p(\epsilon')\big(s_{\text{real}}(\alpha_t G_\theta(z) + \sigma_t\epsilon'\bar{\mathbf{R}}^{-1}, t) - s_{\text{fake}}(\alpha_t G_\theta(z) + \sigma_t\epsilon'\bar{\mathbf{R}}^{-1}, t)\big)\frac{dG_\theta(z)}{d\theta}\,d\epsilon'
\end{aligned}$$

where $\epsilon' = \epsilon\bar{\mathbf{R}}$.

[2] We only need to prove that the model is invariant to SO(3) only. This is due to the fact that we subtract the mean of all equivariant features with respect to the molecule $\mathcal{G}$ in each step of the EGNN backbone and therefore operate on a zero-center-of-mass space as defined in EDM [14], which is invariant to translation by definition.

Algorithm C.1 Training procedure

Require: Frozen teacher μ_{real}, generator G, auxiliary score model μ_{fake}, discriminator D, EMA generator G_{EMA}

Initialize G with respaced schedule
for each epoch e **do**
  for each molecular minibatch x **do**
    Preprocess x; construct masks and optional context c
    $x_e \leftarrow \text{VAE.encode}(x)$
    Sample step number $k \in \mathcal{K}$
    $t \leftarrow$ sample diffusion timestep with cutoff $T_{\max}$
    $t \leftarrow \text{EDM_schedule}(t, k, s{=}10^{-4}, \rho{=}2.25)$
    $z_0 \leftarrow G(c, k)$

// Update auxiliary score model and discriminator
    for $j = 1$ **to** N_{ratio} **do**
      $z_t^f \leftarrow \text{corrupt}(\text{sg}(z_0), t)$
      $z_t^r \leftarrow \text{corrupt}(\text{sg}(x_e), t)$
      Evaluate μ_{fake} on z_t^f and z_t^r
      Update D using real and fake intermediate features from μ_{fake}
      if e is after warm-up **then**
        Update μ_{fake} with the fake-sample diffusion objective
      end if
    end for

// Update generator
    $z_t \leftarrow \text{corrupt}(z_0, t)$
    $s_{\text{real}} \leftarrow \mu_{\text{real}}(z_t, t), \quad s_{\text{fake}} \leftarrow \mu_{\text{fake}}(z_t, t)$
    Estimate divergence weight from the output of D
    Update G with the reweighted DMD objective
    $G_{\text{EMA}} \leftarrow \text{EMA}(G_{\text{EMA}}, G)$
  end for
end for

C Experimental Settings

C.1 Discriminator Design

The discriminator D_ψ takes as input three intermediate layers of μ_{fake}: the second, fifth, and seventh layers for QM9, and the zeroth, first, and second layers for GEOM-DRUG. We denote the selected representations of a molecule $x \in \mathbb{R}^{N \times D}$ by `mu_fake_out_0`, `mu_fake_out_1`, and `mu_fake_out_2`, respectively.

For each extracted representation $\texttt{mu_fake_out_i} \in \mathbb{R}^{N \times D}$, where $i \in \{0, 1, 2\}$, N is the number of atoms, and D is the hidden dimension, we introduce a learnable query token $\texttt{q_i} \in \mathbb{R}^D$. This query token aggregates information from the corresponding layer through cross-attention [44]:

$$\texttt{output_i} = \text{CrossAttn}\left(\text{query} = \texttt{q_i}, \text{key} = \texttt{mu_fake_out_i}, \text{value} = \texttt{mu_fake_out_i}\right).$$

The resulting representation is then passed through a layer-specific MLP to obtain the discriminator logit:

$$\texttt{logits_i} = \text{MLP}_i(\texttt{output_i}).$$

Each logit is converted into a probability score `p_i` using the softmax activation. The final discriminator score $D_\psi(x)$ is computed as the average of the three probability scores:

$$\texttt{p} = \frac{1}{3} \sum_{i=0}^{2} \texttt{p_i}.$$

To achieve faster convergence and increase training stability, we also adopt an R1 regularization loss [28]:

$$R_1(\psi) = \mathbb{E}_{x\sim p_{\text{data}}}\big[||\nabla D_\psi(x)||^2\big].$$

C.2 Implementation details for unconditional generation

QM9 We implement our distillation framework on top of GeoLDM [49] within the PyTorch framework [30], using the EGNN-based equivariant score network [36] as both the teacher and the student generator. Following the GeoLDM implementation, we operate in the learned continuous latent space with invariant feature dimensionality of 2, and we retain the teacher's architectural hyperparameters for the student generator G_θ, the fake score network μ_{fake}, and the discriminator backbone. The student is initialized from the pretrained teacher's weights, while μ_{fake} is initialized as a copy of the teacher and subsequently updated online during distillation.

We adopt the Adam optimizer [21] for all three networks. We observe that 4-step generation training typically suffers from under-fitting and requires a larger learning rate to achieve a better minimum. For the 5-step and 8-step regimes, we use $G_{lr} = 8\times10^{-7}$, $\mu_{\text{fake},lr} = 32\times10^{-7}$, and $\text{disc}_{lr} = 16\times10^{-5}$. For the 4-step regime, we scale all three learning rates by a factor of 10, yielding $G_{lr} = 8\times10^{-6}$, $\mu_{\text{fake},lr} = 32\times10^{-6}$, and $\text{disc}_{lr} = 16\times10^{-4}$. The relative ratio between the three learning rates $(1:4:200)$ is preserved across both regimes, which we found important for maintaining stable adversarial dynamics between the generator, the fake score estimator, and the discriminator. An exponential moving average of the student weights is maintained with decay 0.9999 and used for all evaluation.

We provide more details on the self-rollout procedure. We also note that using 1, 1, and 3 respectively for generation of 4 steps, 5 steps, and 8 steps as the lower bound for the number of generated samples passed to μ_{fake} is optimal. During each training iteration, the number of generation steps K is sampled from one of three curricula: $K \in \{1,2,3,4\}$, $K \in \{1,2,3,4,5\}$, or $K \in \{3,4,5,6,7,8\}$. The first two curricula target few-step inference, while the third targets higher-fidelity multi-step generation. Sampling K uniformly within a curriculum exposes the student to a range of rollout horizons and encourages consistent generation quality across the supported inference budgets. At each step of the rollout, we apply the distribution matching gradient following DMD2 [53], with the fake score network updated via a standard denoising objective on the student's generated samples.

The total training loss combines the distribution matching objective with two auxiliary terms: an adversarial GAN loss and a Jensen-Shannon divergence loss computed between the teacher and student score distributions. Across all experiments, we fix the GAN loss coefficient for μ_{fake} to $\lambda_{\text{GAN}} = 0.2$ and the Jensen-Shannon loss coefficient to $\lambda_{\text{JS}} = 0.1$. We observed that both auxiliary losses improve sample quality and stability but must be kept subordinate to the primary distribution matching loss to avoid mode collapse. Training is performed on RTX 4090 GPUs. For QM9, we use a batch size of 32 and the student converges within 5 epochs (15,000 iterations).

GEOM-DRUG We implement our distillation framework on top of GeoLDM [49] in PyTorch [30]. Following GeoLDM, both the teacher and student models use an EGNN-based equivariant score network [36]. For GEOM- DRUG, we perform distillation in the learned continuous latent space with latent feature dimensionality 2. Unless otherwise stated, the student generator G_θ, the fake score network μ_{fake}, and the discriminator backbone follow the same architectural hyperparameters as the pretrained teacher. In particular, we use hidden dimension 256, four EGNN layers, normalization factors $[1, 4, 10]$, and a global normalization factor of 1. An exponential moving average of the student weights is maintained with decay 0.9999 and used for all evaluation.

The student is initialized from the pretrained teacher checkpoint, while μ_{fake} is initialized as a copy of the teacher and updated online during distillation. We use the polynomial noise schedule `polynomial_2` with noise precision 10^{-5} and train with the ℓ_2 diffusion loss. The minimum diffusion time is set to $T_{\min} = 0.02$, with the same value used during pretraining. For the GEOM-DRUG experiment reported here, we target 5-step generation and set the rollout step number to $K = 5$.

We optimize the student generator, fake score network, and discriminator using Adam [21]. The learning rates are set to

$$G_{\text{lr}} = 5\times10^{-6}, \qquad \mu_{\text{fake,lr}} = 4\times10^{-6}, \qquad D_{\text{lr}} = 2\times10^{-4}.$$

The R1 regularization weight is set to 0.001, with perturbation scale $\sigma = 0.01$.

The total objective combines the distribution matching loss with auxiliary adversarial and divergence terms. For this GEOM-DRUG setting, the adversarial network coefficient, which is related to both μ_{fake} and D_ψ, is set to 0.47. We additionally enable the Jensen-Shannon divergence objective with coefficient 0.03. The consistency regularization coefficient and additional regularization coefficient are both set to 0.

Training is performed with distributed data parallelism on 8 A100 GPUs. We use a per-device batch size of 6, corresponding to an effective batch size of 48 and the student converges within 20,000 iterations, with 3,000 iterations of discriminator warmup.

C.3 Quality-stability trade-off

The experiments are conducted using the same parameters as used to train the models in Appendix C.2. However, we simply elongate the training time. Due to the higher magnitude of the learning rate for the generation of 4 steps, longer training leads to overshooting, and we do not report them in ablation on training time.

C.4 Conditional Generation

Following previous conventions [49], we evaluate conditional generation results on 6 properties. The condition is added as an additional node feature in the molecule representation. For each molecule property, we train a GeoLDM teacher for 3000 epochs conditioning on it and then distill a FlashMol student model accordingly. We judge the results by training a classifier separately that outputs the value of this property given a molecule. Since generally speaking conditional generation is more difficult, we present the results of our 32 step models.

We train FlashMol on the QM9 dataset, conditioning on the $\alpha, \Delta\epsilon$, HOMO, LUMO, μ, and Cv respectively following [49] and [14]. The student distills a pre-trained conditional GeoLDM teacher (`exp_cond_alpha` for instance) into a 32-step generator using the f-divergence variant of DMD [51] with Jensen–Shannon divergence as the final objective as specified in Section 4.

C.4.1 Architecture

The student inherits the teacher's architecture and is initialized from the teacher weights. Both networks operate on continuous point-structured geometric latents $z = \langle z_x, z_h \rangle$ with $k = 1$ invariant channel per node, parameterized by an EGNN denoiser with 9 layers and hidden width 192. Feature normalization factors are $[1, 8, 1]$ for $(x, h_{\text{cat}}, h_{\text{int}})$, and atomic charges are excluded from the input. An exponential moving average of the student weights is maintained with decay 0.9999 and used for all evaluation.

Table C.1: Architecture hyperparameters.

Hyperparameter	Value
Hidden features (`nf`)	192
Number of layers (`n_layers`)	9
Latent features (`latent_nf`)	1
Normalization factors	$[1, 8, 1]$
EMA decay	0.9999

C.4.2 Diffusion Settings

The teacher uses a polynomial-2 noise schedule with precision 10^{-5} and an L2 ϵ-prediction objective. $[T_{\min}, T]$ with $T_{\min} = 0.02$ to avoid the unstable low-noise tail is applied to the timestep sampler used for the fake-score network's denoising-score-matching update.

The discriminator is regularized with an R1 gradient penalty [28] of weight 10^{-3} at noise scale $\sigma = 0.01$.

Table C.2: Diffusion settings.

Parameter	Value
Noise schedule	polynomial-2
Noise precision	10^{-5}
Diffusion loss	L2
Distillation steps K	32
Minimum timestep for score estimation $T_{\min}$	0.02

Table C.3: Loss coefficients and regularization.

Parameter	Value
f-divergence type	Jensen–Shannon (`use_js`)
f-divergence coefficient	0.1
GAN coefficient (μ_{fake} and discriminator, λ_f)	0.2
R1 gradient penalty weight	10^{-3}
R1 noise scale σ	0.01

Table C.4: Learning rates.

Component	Learning rate
Generator G_θ	8×10^{-7}
Fake-score network μ_{fake}	3.2×10^{-6}
Discriminator D	1.6×10^{-4}

Table C.5: Training and evaluation configuration.

Parameter	Value
Dataset	QM9 (second half, 50K)
Include charges	No
Batch size	32
Stability samples	10,000
GPU	NVIDIA RTX 4090

C.4.3 Optimization

Following DMD2 [53] and f-distill [51], the three networks are optimized at distinct learning rates with Adam: the generator G_θ at 8×10^{-7}, the fake-score network μ_{fake} at 3.2×10^{-6}, and the discriminator at 1.6×10^{-4}. The auxiliary networks (μ_{fake} and D) are updated 5 times per generator update (`step_ratio` $= 5$), to better track the moving student distribution before each f-distill gradient is computed. The lower bound of the self-rollout step number is set to 3 for 8, 12 and 32 NFEs while 1 for 5 NFEs.

C.4.4 Training and Evaluation

Training is performed on the QM9 second-half split (50K molecules) with batch size 32 on a single NVIDIA RTX 4090 GPU. Evaluation is performed by running three runs, drawing $10{,}000$ conditional samples each, and we report the conditional MAE against a pretrained EGNN property predictor ω trained on the first 50K split, following the protocol of EDM [14] and GeoLDM [49].

Table D.1: PoseBusters evaluation on QM9 (a) and GEOM-DRUGS (b). On QM9, we compare against the 1000-NFE GeoLDM teacher under 8 NFEs. On GEOM-DRUGS, we underline the best entry in each row (the fewest NFEs for NFE). We highlight metrics in which our student model FlashMol surpasses GeoLDM-respaced under the same NFEs.

(a) QM9

Metrics (% pass)	GeoLDM-respaced	FlashMol	GeoLDM teacher
Atoms connected	98.65	99.81	99.89
Bond angles	99.80	100.00	100.00
Bond lengths	99.64	99.99	99.98
Ring flat	100.00	100.00	100.00
Double bond flat	97.02	98.53	99.30
Internal energy	92.90	96.66	96.45
No steric clash	99.60	99.95	99.87
Validity	78.29	94.40	90.20
Uniqueness	96.10	94.53	98.90
NFE	8	8	1000

(b) GEOM-DRUGS

Metric	FlashMol	GeoLDM-respaced	SemlaFlow	ADiT
Validity	99.66	96.42	93.90	95.30
Uniqueness	85.84	96.34	100.00	100.00
Novelty	85.43	96.25	–	–
Atoms connected	100.00	100.00	92.30	93.00
Bond angles	91.20	17.69	94.80	92.30
Bond lengths	99.20	75.18	94.60	92.50
Ring flat	100.00	99.98	94.90	95.40
Double bond flat	100.00	100.00	94.20	95.30
Internal energy	88.30	15.41	94.80	91.30
No steric clash	97.61	63.38	92.00	91.80
PoseBusters valid	85.31	9.05	87.50	85.30
NFE	8	8	100	500/1000

D Results

Sampling protocol Following previous conventions, all outcomes below on both QM9 and DRUG dataset are obtained by running the test for 3 times. We sample 10000 molecules for each test. We report the mean for each metric. Our models are trained until Valid & Unique drops.

D.1 Unconditional Molecule Generation Results

PoseBusters Evaluation PoseBusters [5] evaluate 3D conformation properties of the generated molecules. They include the following metrics: 1. Atoms connected (which checks if all generated atoms form a single connected molecule); 2. Bond angles (which checks if every bond angle is within tolerance of the range expected from distance geometry); 3. Bond lengths (which checks if every bond length is within tolerance of its reference value); 4. Internal steric clash (which checks if any pair of non-bonded atoms is closer than allowed); 5. Aromatic ring flatness (which checks if all atoms in each aromatic ring lie in one plane); 6. Double bond flatness (which checks if the atoms around each carbon–carbon double bond lie in one plane); and 7. Internal energy (which checks if the conformer's UFF [34] energy is not unreasonably higher than that of a force-field-relaxed version of the same molecule). Notably, validity means the total percentage of molecules that pass all these checks and are InChI [11] convertable, which is different from the validity metrics used in GeoLDM [49].

As presented in Table D.1, FlashMol surpasses the teacher and untrained model in all stability metrics on QM9 and GEOM-DRUGS. Besides, FlashMol also achieves better generation quality in terms of bond lengths, ring flatness, double bond flatness and no-steric-clash percentage compared to SemlaFlow [16] and ADiT [19] with far fewer NFEs (only 8).

Stability and Diversity Trade-Offs We train the 8-NFE and 5-NFE model using the configurations in Appendix C. During training, although molecule stability generated by the models keeps increasing, Valid & Unique metrics keeps declining by a small margin as shown in Figure D.1.

D.2 Conditional Molecule Generation Results

Here, we present the test results of conditional generation trained with fewer NFEs. All hyperparameters are the same for the 12-step and 8-step models. We train each model until convergence: 5 epochs for the 12-NFE model, 6 epochs for the 8-NFE model, and 10 epochs for the 5-NFE model.

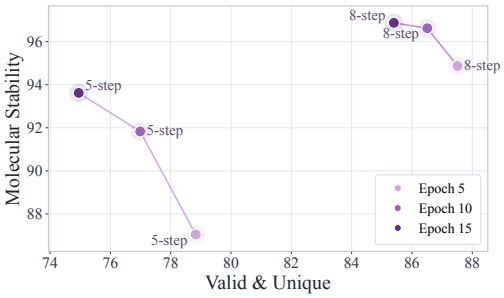

Figure D.1: Training dynamics of the distilled student across epochs. Darker points correspond to later training epochs. While atom and molecule stability improve with training, uniqueness decreases monotonically

Table D.2: Conditional generation results are evaluated on various properties of the molecule. We report the MAE(↓) between generated molecules. The property of each generated molecule is predicted by a pretrained EGNN classifier. *QM9* and *Random* results can be viewed as lower and upper bounds of MAE on all properties. We highlight metrics in which our student model surpasses the respaced teacher using the same NFEs.

Property Units	NFE -	α Bohr^3	$\Delta\varepsilon$ meV	$\varepsilon_{\text{HOMO}}$ meV	$\varepsilon_{\text{LUMO}}$ meV	μ D	C_v $\frac{\text{cal}}{\text{mol K}}$
QM9	-	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	1000	2.76	655	356	584	1.111	1.101
GeoLDM	1000	2.37	587	340	522	1.108	1.025
GeoBFN	100	2.34	577	328	516	0.998	0.949
GeoLDM-respaced	12	2.70	670	383	591	1.058	1.087
FLASHMOL	12	2.70	646	367	578	1.082	1.117
GeoLDM-respaced	8	3.20	750	423	686	1.111	1.301
FLASHMOL	8	2.98	714	399	611	1.111	1.230
GeoLDM-respaced	**5**	5.24	935	591	910	1.343	1.706
FLASHMOL	**5**	3.88	814	538	747	1.830	1.428

D.3 Ablation Study Results

Effect of f-Divergence Choice on Stability and Diversity We use the same configuration as elaborated in Appendix C.2 and compare our full objective (DMD + JS) against a Jensen-Shannon term only, reverse KL + forward KL, or a DMD only variant. The corresponding loss objectives are: $\mathcal{L}_{\text{DMD}}$, $\mathcal{L}_{\text{DMD}} + \lambda \cdot \mathcal{L}_{\text{forward}}$ and $\mathcal{L}_{\text{JS}}$ respectively, where $\lambda = 0.1$ and $\mathcal{L}_{\text{forward}}$ is computed with $f := r \log r$ in Equation (5). We train all models until V&U metric drops (which is 5 epochs for all models). Higher is better for all metrics. Results are reported as means over 3 sampling runs of 10000 molecules each on QM9 dataset. The results are shown in Table 3.

We also report the results trained without the Jensen-Shannon objective across 4, 5 and 8 steps in Table D.3. All models are distilled under the configurations as specified in Appendix C.2 from the same GeoLDM teacher with $K = 4, 5, 8$ generation steps respectively and evaluated at epoch 5 where all training converge. Results trained with regularizers can be found in Table 1. Higher is better for all metrics. Results are reported as means over three sampling runs of 10000 molecules each. The results are shown in Table 3.

D.4 Transferability of FlashMol to Other Backbones

To demonstrate that FlashMol transfers accross backbones, we choose another strong molecule generative model, SemlaFlow [16], as the baseline. DMD [54] is designed for continuous space

Table D.3: DMD-only variant

NFE	Mol Stab (%)	V&U (%)
4	81.51	68.84
5	88.60	78.19
8	95.32	86.89

generation. However, SemlaFlow involves both generation in continuous space (for 3D coordinates) and in discrete space (for atom types, bond orders and formal charges). Therefore, we integrate Discrete Flow Maps [31] to provide supervision signal for the discrete part of the output, whereas we use our f-divergence objective for the continuous part. We present the adaptation of Discrete Flow Maps and our results for SemlaFlow below.

Adapatation of Discrete Flow Maps DMD is well defined only for the Gaussian coordinate path. Atom types, bond orders, and formal charges in SemlaFlow [16] instead follow a discrete corruption process $\mathbf{x}_s \sim \mathrm{Cat}\big(s\,\mathbf{p}_1 + (1-s)\,\mathcal{U}\big)$, whose state space admits no score, so the DMD gradient cannot provide signal to the categorical heads of the student. To address this issue, we distill the discrete heads with the **progressive self-distillation (PSD) loss of Discrete Flow Maps** [31], which operates directly on the probability simplex rather than forcing one-hot states into continuous space. PSD parameterizes the few-step map by the mean denoiser $\psi_{s,t}(\mathbf{x}) \in \Delta^{K-1}$, the time-averaged posterior over the clean token, and enforces the semigroup identity

$$\psi_{s,t}(\mathbf{x}) = \alpha_{s,u,t}\,\psi_{s,u}(\mathbf{x}) + \beta_{s,u,t}\,\psi_{u,t}\big(X_{s,u}(\mathbf{x})\big), \qquad X_{s,u}(\mathbf{x}) = \tfrac{1-u}{1-s}\,\mathbf{x} + \tfrac{u-s}{1-s}\,\psi_{s,u}(\mathbf{x}), \quad \text{(D.1)}$$

for $s < u < t$, with $\alpha_{s,u,t} = \frac{(u-s)(1-t)}{(t-s)(1-u)}$ and $\beta_{s,u,t} = \frac{(t-u)(1-s)}{(t-s)(1-u)}$. Since $\alpha + \beta = 1$, the right-hand side is a convex combination of probability vectors and therefore a valid distillation target, matched to the student by a forward KL:

$$\mathcal{L}_{\text{PSD}} = \mathbb{E}_{s,u,t,\mathbf{x}_s}\Big[\mathrm{KL}\Big(\mathrm{sg}\big[\alpha_{s,u,t}\,\hat{\psi}_{s,u}(\mathbf{x}_s) + \beta_{s,u,t}\,\hat{\psi}_{u,t}(\hat{X}_{s,u}(\mathbf{x}_s))\big]\;\Big\|\;\hat{\psi}_{s,t}(\mathbf{x}_s)\Big)\Big]. \quad \text{(D.2)}$$

The SemlaFlow backbone is conditioned on a single time and always predicts the clean endpoint, so our student represents only $\psi_{s,1}$. We therefore fix $t = 1$, which collapses Equation (D.1) to $\alpha = 0$, $\beta = 1$: the target reduces to the two-hop prediction $\psi_{u,1}(X_{s,u}(\mathbf{x}_s))$, and PSD becomes a consistency condition between a long jump to $t = 1$ from s and a shorter jump from the midpoint $u = (s+1)/2$. We realize the first hop from s to u with the frozen teacher, transporting the full molecular state—coordinates, atom types, bonds, and charges—along the linear path to u, and the second hop from u to $t = 1$ with an EMA copy of the student, which stabilizes the target as the student drifts away from its initialization. Both calls are stop-gradient; the trainable student is evaluated once, directly at s. We apply PSD loss on all three discrete generation heads,

$$\mathcal{L}_{\text{PSD}} = \sum_{c \in \{\text{atom, bond, charge}\}} \lambda_c\;\mathbb{E}_{s,\,\mathbf{x}_s}\Big[\mathrm{KL}\Big(\mathrm{sg}\big[\hat{\psi}^{(c)}_{\text{EMA}}\big(\hat{X}^{\text{teacher}}_{s,u}(\mathbf{x}_s),\,u\big)\big]\;\Big\|\;\hat{\psi}^{(c)}_{\theta}(\mathbf{x}_s,\,s)\Big)\Big], \quad \text{(D.3)}$$

where s is drawn from the same respaced schedule as the coordinate DMD term, atom and charge terms are averaged over valid nodes, and the bond term over valid node pairs. Because μ_{fake} is regressed on coordinate velocity only, categorical supervision is owned entirely by Equation (D.3), and the full objective is $\mathcal{L} = \mathcal{L}_{\text{DMD}} + \lambda\,\mathcal{L}_{\text{JS}} + \mathcal{L}_{\text{PSD}}$.

SemlaFlow Results We apply the aforementioned distillation recipe to SemlaFlow [16] and compare against SemlaFlow-respaced under a 4-NFE budget on QM9. After further tuning the ρ for respacing, we choose $\rho = 7$, and FlashMol-SemlaFlow achieves **88.51% molecule stability**, compared to 70.08% for SemlaFlow-respaced and 83.89% for the 4-step FlashMol-GeoLDM, while maintaining 89.75% uniqueness (Table D.4). The gain extends to the 3D-aware metrics: strain per atom drops from 5.83 to 2.15 and optimized RMSD from 0.376 to 0.252, indicating that the improvement reflects better-learned conformations rather than inflated 2D statistics. This demonstrates that FlashMol **transfers to other molecule generation backbones** without architecture-specific modification.

Table D.4: FlashMol-SemlaFlow training results on QM9 with 4 NFEs. Entries where FlashMol-SemlaFlow surpasses SemlaFlow-respaced are in bold.

Metric	4-Step SemlaFlow-respaced	4-Step FlashMol-SemlaFlow
Atom Stability (↑)	97.68%	**98.98%**
Molecule Stability (↑)	70.08%	**88.51%**
Validity (↑)	79.52%	**90.31%**
Connected Validity (↑)	73.98%	**89.18%**
Uniqueness (↑)	91.74%	89.75%
Novelty (↑)	96.14%	90.69%
Energy (↓)	130.31	**62.03**
Energy per Atom (↓)	7.43	**3.51**
Energy Validity (↑)	75.38%	**89.40%**
Optimized Energy Validity (↑)	75.38%	**89.40%**
Strain (↓)	101.83	**37.53**
Strain per Atom (↓)	5.83	**2.15**
Optimized RMSD (↓)	0.376	**0.252**

E Limitations and Broader Impact

E.1 Limitations

Uniqueness and Diversity Trade-Offs Despite the training efficiency and generation quality of FlashMol, we observe a gradual reduction in sample diversity as distillation training proceeds. This is not unique to our method: it is a known characteristic of score-distillation objectives, including the DMD family [53, 54], which trade stochastic uncertainty in sampling for stable, high-quality output. In the molecular generation setting, diversity loss is particularly consequential: a generative model used for virtual screening must cover a wide region of chemical space to maximize the probability of discovering viable lead compounds. We observe this effect primarily as reduced uniqueness scores over extended training, while atom-level and molecule-level stability remain largely unaffected.

Generalization to other noise-sensitive molecular backbones. The core insight motivating FlashMol—that certain equivariant architectures exhibit structured sensitivity to noise in latent space that can be exploited by distribution matching distillation and timestep respacing technique —is not specific to the GeoLDM backbone [49]. We expect similar benefits to extend to other 3D molecular generative models that operate over continuous latent representations of atomic coordinates, such as protein–ligand co-design or conformation generation models. Adapting FlashMol to these settings may require re-examining the latent-space geometry and the construction of the fake score network, particularly for architectures with different equivariance groups (*e.g.*, $\mathrm{SE}(3)$ vs. $\mathrm{E}(3)$) or non-Gaussian priors. We leave systematic benchmarking on broader molecular modalities to future work.

E.2 Broader Impact

This work accelerates the sampling of 3-D molecular generative models by applying distribution matching distillation to equivariant latent diffusion models. The primary intended application is computational drug discovery and materials design, where generative models are used to propose novel molecular structures for subsequent experimental validation.

Positive impacts. Reducing the inference cost of high-quality molecular generative models lowers the computational barrier to large-scale virtual screening—a step that currently limits the practical deployment of diffusion-based molecular design tools in academic and resource-constrained industrial settings. Faster generation enables higher-throughput exploration of chemical space within fixed compute budgets, which could accelerate the early-stage identification of therapeutically relevant molecules. More broadly, FlashMol is a general distillation framework applicable to any equivariant latent diffusion model, and may therefore benefit downstream tasks such as protein structure prediction, antibody design, and materials discovery.

Risks and limitations. As with all generative models for molecular design, there is a potential for misuse: accelerated generation of structurally valid molecules could in principle be directed toward the design of harmful compounds, including those with toxic, environmentally persistent, or weaponisable properties. We note, however, that FlashMol operates as a distillation of an existing teacher model and introduces no novel generative capacity beyond what is already present in the teacher; the risk profile of FlashMol is therefore no greater than that of the teacher molecular diffusion model itself. We recommend that downstream applications incorporate appropriate safety filters—for example, standard toxicity predictors (*e.g.*, hERG cardiotoxicity screening, Ames mutagenicity tests) applied to generated structures prior to any experimental follow-through.

Societal context. The acceleration of AI-driven drug discovery has complex distributional effects: it may reduce time-to-drug for diseases with well-funded research programs while potentially increasing the concentration of early-stage discovery power in organisations with large compute resources. We hope that the reduced inference cost introduced by FlashMol partially counteracts this trend by making high-quality molecular generation more accessible to the broader research community.