IAGA: Identity-Aware Gaussian Approximation for Efficient 3D Molecular Generation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Gaussian Probability Path based Generative Models (GPPGMs) generate data by reversing a stochastic process that progressively corrupts samples with Gaussian noise. While these models have achieved state-of-the-art performance in 3D molecular generation, their practical deployment remains constrained by the high computational cost of long generative trajectories, involving hundreds to thousands of steps during model training and sampling. In this work, we introduce a novel method that improves the efficiency of 3D molecular generation without sacrificing training granularity or inference fidelity. Our key insight is that different data modalities will exhibit markedly different rates of convergence to Gaussianity in the forward process of GPPGMs. We analytically identify a characteristic step at which the data has acquired sufficient Gaussianity, and then replace the remaining generation trajectory with a closed-form Gaussian approximation. Unlike existing techniques that accelerate the generation process via reformulating or coarsening the trajectories, our method preserves the full resolution of learning dynamics while avoiding redundant distributional transport with little data identity remained. Empirical results across different 3D molecular generation datasets demonstrate substantial improvements in both sample quality and computational efficiency.

1 Introduction

Generative models, particularly Gaussian Probability Path based Generative Models (GPPGMs), have demonstrated impressive performance across diverse domains such as images [Li et al., 2019], text [Austin et al., 2021], and molecules [Zhang et al., 2023]. However, the generative trajectories are typically modeled as the solution to a stochastic differential equation (SDE) or ordinary differential equation (ODE), which are often represented by hundreds to thousands of steps for better learning granularity. The heavy computational demand thus becomes one of their key limitations, especially for 3D molecular data. To improve the efficiency, prior work has largely focused on sampling acceleration, for example, coarsening trajectories with reduced-step solvers [Song et al., 2020, Lu et al., 2022, Karras et al., 2022] and retrieval-based methods [Zhang et al., 2025]. While effective for inference, these approaches either compromise trajectory granularity or leave training costs unaffected. Efforts closer to training, such as adaptive priors [Lee et al., 2021, Vignac et al., 2022] and leapfrog initializers for trajectory prediction [Mao et al., 2023], still depend on modifications of the noising process or specialized architectures, rendering them domain-specific and difficult to apply to 3D molecular generation.

In this work, we propose a novel method that improves both training and sampling efficiency of GPPGMs via Gaussian Approximation (GA). A key feature of our framework is that it naturally applies to zero-mean invariant modalities, a broad and practically important class including molecular graphs and 3D geometric data, where zero-mean is a common data regularization method without information loss. Rather than coarsening the generative trajectories or modifying the predefined noise

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

schedule, our method identifies a characteristic time step T^* at which the input data distribution has effectively lost its specific identity while gaining sufficient Gaussianity. *Based on this point, the generation trajectory can be truncated, and the final distribution can be approximated by a tractable Gaussian reference distribution with analytically derived mean and variance, as shown in Fig. 1.* This design yields two key merits absent in existing methods: **(1) ability for training acceleration via eliminating ineffective optimization on over-noised inputs**, and **(2) sampling fidelity preservation by maintaining the accuracy and granularity of the original generative trajectories.** We empirically validate our method across different 3D molecular datasets, demonstrating significant improvements in both sampling and training efficiency with high-quality generation.

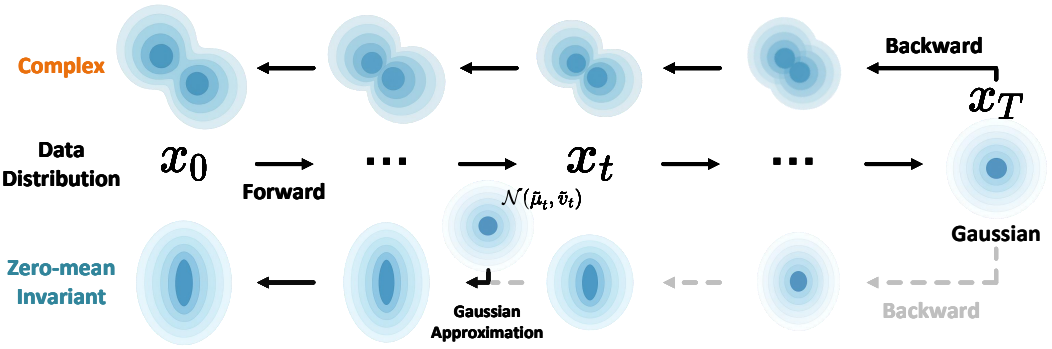

Figure 1: The flowchart of the IAGA. When the noised data distribution $\boldsymbol{x}_t$ has lost its identity at timestep t, we approximate it with a reference Gaussian $\mathcal{N}(\tilde{\mu}_t, \tilde{v}_t)$. In such case, the length of the generative trajectory can be reduced from T steps to t steps.

2 Preliminaries

2.1 Gaussian Probability Path based Generative Models

GPPGMs construct complex data distributions by learning to reverse a reference stochastic process that progressively corrupts clean data with Gaussian noise. Given a data sample $\boldsymbol{x}_0$ drawn from the target distribution $p_{\text{data}}(\boldsymbol{x})$, we define a forward (noising) process that maps $\boldsymbol{x}_0$ to a sequence of latent states $\{\boldsymbol{x}_t\}_{t=1}^T$. A commonly used instantiation of this noising process is a time-indexed Gaussian perturbation:

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t \mid \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0,\, \bar{\sigma}_t^2\mathbf{I}), \tag{1}$$

where $\bar{\alpha}_t \in [0,1]$ controls the decay of the signal power over time. Typically, $\bar{\alpha}_t$ is defined as $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$, with $\alpha_t \in (0,1)$ monotonically decreasing such that $\bar{\alpha}_0 \approx 1$ and $\bar{\alpha}_T \approx 0$, ensuring that $\boldsymbol{x}_T$ approaches a tractable reference distribution, often taken to be $\mathcal{N}(\mathbf{0}, \bar{\sigma}_T^2\mathbf{I})$.

In the case of variance-preserving (VP) forward processes, defined by $\bar{\sigma}_t = \sqrt{1-\bar{\alpha}_t}$, the forward process admits the following Markov factorization:

$$q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0) = \prod_{t=1}^T q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) = \prod_{t=1}^T \mathcal{N}(\boldsymbol{x}_t \mid \alpha_{t|t-1}\boldsymbol{x}_{t-1},\, \sigma_{t|t-1}^2\mathbf{I}), \tag{2}$$

where $\alpha_{t|t-1} = \bar{\alpha}_t/\bar{\alpha}_{t-1}$ and $\sigma_{t|t-1}^2 = 1-\alpha_{t|t-1}^2$. The VP forward process is the most commonly used formulation in the design of GPPGMs. Unless otherwise specified, we adopt the VP noising schedule throughout this work.

The reverse (denoising) process, which models $p(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$, admits a closed-form expression under the Gaussian assumption:

$$q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_{t-1} \mid \boldsymbol{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0),\, \tilde{\sigma}_t^2\mathbf{I}), \tag{3}$$

where

$$\boldsymbol{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\,(1-\bar{\alpha}_t)}{1-\bar{\alpha}_{t-1}}\boldsymbol{x}_0 + \frac{\sqrt{\alpha_t}\,(\bar{\alpha}_{t-1}-\bar{\alpha}_t)}{1-\bar{\alpha}_{t-1}}\boldsymbol{x}_t, \quad \tilde{\sigma}_t^2 = \frac{(1-\bar{\alpha}_t)\,(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_{t-1}}.$$

This Gaussian formulation facilitates a tractable variational objective and enables efficient sampling algorithms that are central to GPPGMs.

Learning the Reverse Process A key feature of GPPGMs is that the reverse-time generative process is constructed to approximate the true posterior $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0)$. Since the original sample $\boldsymbol{x}_0$ is unavailable during generation, it is replaced by a neural estimate $\hat{\boldsymbol{x}}_0 = \phi(\boldsymbol{x}_t, t)$ inferred from the current noisy observation. The generative transition distribution is then defined as:

$$p(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t-1} \mid \boldsymbol{\mu}_t(\boldsymbol{x}_t, \hat{\boldsymbol{x}}_0),\, \tilde{\sigma}_t^2 \mathbf{I}), \tag{4}$$

where the mean and variance retain the form of the true posterior, with $\boldsymbol{x}_0$ replaced by its approximation $\hat{\boldsymbol{x}}_0$. Given this generative model, we can derive a variational lower bound on the marginal log-likelihood:

$$\log p(\boldsymbol{x}_0) \geq \mathcal{L}_0 + \mathcal{L}_{\text{base}} + \sum_{t=1}^{T} \mathcal{L}_t, \tag{5}$$

where $\mathcal{L}_0 = \log p(\boldsymbol{x}_0 \mid \boldsymbol{x}_1)$ is the terminal reconstruction term, $\mathcal{L}_{\text{base}} = -\text{KL}(q(\boldsymbol{x}_T \mid \boldsymbol{x}_0) \,\|\, p(\boldsymbol{x}_T))$ regularizes the marginal at the final time step, and

$$\mathcal{L}_t = -\text{KL}(q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_0, \boldsymbol{x}_t) \,\|\, p(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)), \quad \text{for } t = 1, \ldots, T. \tag{6}$$

In practice, the base KL term $\mathcal{L}_{\text{base}}$ becomes negligible when $\alpha_T \approx 0$, and the data term $\mathcal{L}_0$ is often near zero for discrete $\boldsymbol{x}_0$ when $\alpha_0 \approx 1$. Meanwhile, Ho et al. [2020] found it more stable to parameterize ϕ as a noise predictor: rather than outputting $\hat{\boldsymbol{x}}_0$ directly, the network predicts the noise vector $\hat{\boldsymbol{\epsilon}}$ such that $\boldsymbol{x}_t \approx \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}$. In this case, $\hat{\boldsymbol{x}}_0$ can be recovered via $\hat{\boldsymbol{x}}_0 = \frac{1}{\alpha_t}(\boldsymbol{x}_t - \sigma_t \hat{\boldsymbol{\epsilon}})$. This formulation leads to a simplified training objective, where each KL term $\mathcal{L}_t$ reduces to a weighted denoising score-matching loss:

$$\mathcal{L}_t = \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[\frac{1}{2} w_t \left\| \boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}} \right\|^2 \right], \tag{7}$$

where w_t is a scalar weight derived from the noise schedule. This structure naturally extends to various GPPGM frameworks, including diffusion models [Ho et al., 2020] and flow-matching models [Lipman et al., 2022], both of which aim to approximate the conditional dynamics of the reverse process via supervised regression on progressively removed noise.

2.2 Zero-Mean Invariance

A data modality is *zero-mean invariant* if centering each sample by subtracting its empirical mean preserves all the information necessary for downstream modeling. Formally, let $x \in \mathbb{R}^d$ denote a data sample, and define its centered version as:

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} - \frac{1}{d} \sum_{i=1}^{d} \boldsymbol{x}_i \cdot \mathbf{1}_d, \tag{8}$$

where $\mathbf{1}_d \in \mathbb{R}^d$ is the vector of all 1-s. A data modality is said to satisfy zero-mean invariance if, for all $\boldsymbol{x}$ in the support of the data distribution $p(\boldsymbol{x})$, the transformation $\boldsymbol{x} \mapsto \tilde{\boldsymbol{x}}$ retains the semantic or structural information of the original input.

This property is common in domains where only internal relationships among dimensions carry information, while global offsets are irrelevant or redundant. 3Typical examples include any representations defined up to an affine baseline or possessing a shift-symmetric structure, such as configurations invariant to global alignment, label encodings invariant to additive bias, or features embedded in contrastive spaces. We provide some detailed examples and the corresponding analysis in Appendix. A. Zero-mean invariance permits generative models to operate in a reduced subspace orthogonal to the mean direction, eliminating redundant degrees of freedom. In 3D molecular data, zero-mean invariance is widely employed due to its translational invariance [Hoogeboom et al., 2022, Hong et al., 2025, Xu et al., 2023].

3 Identity-Aware Gaussian Approximation

Building on the preliminaries, we now introduce our framework for shortening the generative trajectory in GPPGMs via truncation. Rather than executing the full generation trajectories, we identify a characteristic timestep T^* at which the data effectively loses its identity and exhibits sufficient Gaussianity. This enables an analytic truncation, whereby the remaining trajectory is replaced with a direct Gaussian approximation. It significantly improves computational efficiency without compromising generative fidelity.

3.1 Gaussian Approximation

Gaussian approximations (GA) are commonly employed in statistics to represent intractable conditional or marginal distributions [Berry, 1941, Deng and Zhang, 2020, Chernozhukov et al., 2013]. This modeling choice facilitates closed-form expressions for critical quantities, including transition densities, posterior distributions, and variational bounds, which are essential for both optimization and sampling procedures. In GPPGMs, the forward process can be interpreted as progressively pushing the data toward a Gaussian distribution. As illustrated in (1), the marginal and transition densities of the trajectories at any finite time index remain Gaussian:

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t \mid \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0;\, \Sigma_t), \quad \text{where} \quad \Sigma_t := (1-\bar{\alpha}_t)\mathbf{I}. \tag{9}$$

However, the intractability of data distribution prevent us from directly calculating the mean and variance of approximated Gaussian. For data modalities that are *zero-mean invariant*, such as molecular coordinates, point clouds, or categorical embeddings, the difficulty of estimating mean can be avoided by enforcing zero-centering as a preprocessing step. Such centering preserves structural information and symmetries (e.g., translational invariance) [Hoogeboom et al., 2022], while consistently ensuring $\hat{\mu} = 0$, as analyzed in Appendix A.

In addition, the variance remains intractable to obtain exactly. In this paper, we estimate it through the per-sample statistics. Given a dataset $\mathcal{D} = \{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$ with $\boldsymbol{x}^{(i)} \in \mathbb{R}^d$, we compute

$$v^{(i)} = \frac{1}{d-1}\sum_{j=1}^{d}(\boldsymbol{x}_j^{(i)} - \mu^{(i)})^2, \quad \text{where} \quad \mu^{(i)} = \frac{1}{d}\sum_{j=1}^{d}\boldsymbol{x}_j^{(i)}, \tag{10}$$

and aggregate across the dataset to obtain the *average per-sample variance*: $\hat{v} = \frac{1}{N}\sum_{i=1}^{N} v^{(i)}$. This estimator is unbiased under mild moment conditions [Vershynin, 2012], and we empirically verify that it's an available choice for GA in GPPGMs.

Under the variance-preserving (VP) forward process on zero-meaned data, these choices yield the following analytic form for the mean and variance of $\boldsymbol{x}_t$:

$$\tilde{\mu}_t = \mathbf{0}, \quad \tilde{v}_t = 1 - \bar{\alpha}_t(1-\hat{v}). \tag{11}$$

Consequently, for zero-meaned data, once sufficient noise has been injected at timestep T^*, the marginal distribution of $\boldsymbol{x}_{T^*}$ can be approximated by $\mathcal{N}(\mathbf{0}, \tilde{v}_{T^*}\mathbf{I})$ which serves as the foundation for our trajectory-shortening strategy.

3.2 Gaussian Approximation and Initial Data Distribution

The analysis above shows that, once sufficient noise is injected, the forward process admits a tractable Gaussian approximation. Nevertheless, the following question arises:

> ***(Q)*** *How do we determine T^* at which the injected noise becomes sufficient for this approximation? Is it the same across different tasks?*

To answer this question, we first present Proposition 3.1 to show that T^* is related to properties of the initial data distribution.

Proposition 3.1. *Given $t \in [0, T)$ and $K \geq 3$, and the Gaussianity evaluation functional*

$$\mathcal{H}^{(K)}(x) \;:=\; \beta\left\|\Pi_{\mathsf{D}^\perp}(\mathrm{Cov}(x))\right\|_F \;+\; \sum_{k=3}^{K} w_k \left\|C^{(k)}(x)\right\|_F. \tag{12}$$

where $\beta > 0$ and $w_k > 0$ ($k \geq 3$). $\mathsf{D} := \{\mathrm{Diag}(v) : v \in \mathbb{R}^d\}$ is the diagonal subspace and

$$\Pi_{\mathsf{D}}(\Sigma) := \mathrm{Diag}(\mathrm{diag}\,\Sigma), \quad \Pi_{\mathsf{D}^\perp}(\Sigma) := \Sigma - \Pi_{\mathsf{D}}(\Sigma). \tag{13}$$

are the orthogonal projections. $Cov(\cdot)$ and $C^{(k)}(X)$ are the covariance calculator and the k-th cumulant tensor, respectively. Let A, B be two initial data distribution, where

$$\mathcal{H}^{(m)}(\boldsymbol{x}_t^A) \;\leq\; \mathcal{H}^{(m)}(\boldsymbol{x}_t^B) \quad \textit{for all } m = 2, 3, \ldots, K \tag{14}$$

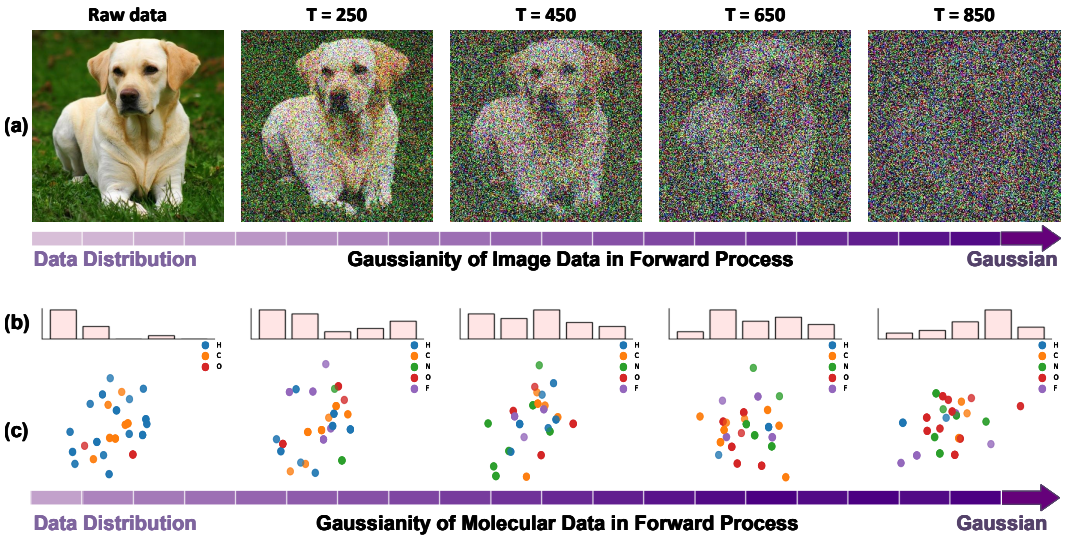

Figure 2: Comparisons of the forward noising process across different data modalities. (a) shows a continuous-valued image matrix, while (b) and (c) illustrate the distribution of molecular data consisting of one-hot vectors for atom types and 3D Euclidean coordinates for atom positions, respectively. **The same noise schedule is applied across all modalities, with the number of steps T up to 1000.** Despite identical signal-to-noise ratios, image data retains recognizable identities for significantly more steps, whereas molecular data lost it much earlier.

holds with at least one strict inequality. Then for every $s > t$,

$$\mathcal{H}^{(K)}(\boldsymbol{x}_s^A) \;<\; \mathcal{H}^{(K)}(\boldsymbol{x}_s^B).$$

Consequently, for every $\epsilon > 0$,

$$T_A^* = \inf\{s > t : \mathcal{H}^{(K)}(\boldsymbol{x}_s^A) \le \epsilon\} \;<\; \inf\{s > t : \mathcal{H}^{(K)}(\boldsymbol{x}_s^B) \le \epsilon\} = T_B^*. \tag{15}$$

A formal proof is provided in Appendix B. This proposition establishes that if the initial data distribution is inherently closer to Gaussian, then the corrupted samples achieve sufficient Gaussianity earlier, and the corresponding GA timestep T^* can be smaller. In particular, sparse molecular coordinates around equilibrium are closer to Gaussian [Frenkel and Smit, 2023], then approximation can start at a smaller T^*, as shown in Fig. 2. As different initial data distribution induces different GA time steps, we need a principled way to identify the precise T^*. Therefore, in Sec. 3.3, we develop a statistical Gaussianity evaluator that serves as an operational test, combining dependency-sensitive functionals and distributional similarity criteria to precisely identify the GA timestep T^*.

3.3 Evaluating Gaussianity: Data Identity and Distributional Similarity

While the preceding analysis suggests that $\boldsymbol{x}_t$ may be approximated by a Gaussian, the validity of this approximation fundamentally depends on whether the $\boldsymbol{x}_{T^*}$ has gained sufficient Gaussianity for GA. In this section, we present the Gaussianity evaluation method from the perspectives of data identity and distributional similarity for our IAGA framework.

Data Identity Decay. The timestep at which data loses its structural identity under progressive noise perturbation is critical for establishing a valid Gaussian approximation. Since the Gaussian distribution in GA is independent, the disappearance of identity in $\boldsymbol{x}_t$ directly indicates that the data has lost its dependency, which can be well approximated by $\mathcal{N}(\tilde{\mu}_t, \tilde{v}_t I)$. As illustrated in Fig. 2, the rate at which identity vanishes strongly depends on the underlying data modality. Monitoring the decay of data identity thus provides a principled criterion for determining the characteristic timestep T^* at which Gaussian approximation becomes valid.

Since structural identity is inherently reflected by the presence of dependencies among variables, we quantify identity decay by measuring statistical dependency in $\boldsymbol{x}_t$. Concretely, we adopt the mutual information (MI) test [Kraskov et al., 2004] as our dependency functional. Because exact

independence occurs only at the terminal prior $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we adopt a tolerance $\varepsilon_{\text{dep}} > 0$ and define the identity-loss timestep as

$$T_{\text{ID}} := \min \left\{ t \;\middle|\; \text{Dep}(\boldsymbol{x}_t) \leq \varepsilon_{\text{dep}} \right\}, \tag{16}$$

where $\text{Dep}(\cdot)$ denotes the MI-based dependency evaluator. The implementation details are provided in Appendix C.1. This evaluator provides a concrete condition under which GA becomes valid from the perspective of data identity.

Distributional Similarity. While data identity decay captures the disappearance of dependence, Gaussian approximation also requires that the marginals of $\boldsymbol{x}_t$ align with those of a Gaussian distribution. To assess this, we measure the distributional similarity between $\boldsymbol{x}_t$ and the reference Gaussian $\mathcal{N}(\tilde{\mu}_t, \tilde{v}_t I)$ with matching variance using the Kolmogorov–Smirnov (KS) distance [Massey Jr, 1951]. Concretely, for each dimension $\boldsymbol{x}_t^{(j)}$, we compare its empirical cumulative distribution function (CDF) $F_{t,j}(x)$ with the Gaussian CDF $\Phi_{\tilde{v}_t}(x)$, and average across all dimensions:

$$D_t = \frac{1}{d} \sum_{j=1}^{d} D_{t,j}, \quad \text{where} \quad D_{t,j} = \sup_x \left| F_{t,j}(x) - \Phi_{\tilde{v}_t}(x) \right|. \tag{17}$$

A smaller D_t indicates closer alignment with Gaussian marginals and therefore stronger justification for approximation by $\mathcal{N}(\tilde{\mu}_t, \tilde{v}_t \mathbf{I})$. Since exact convergence only holds at the terminal prior $\boldsymbol{x}_T \sim \mathcal{N}(\tilde{\mu}_t, \mathbf{I})$, we adopt a tolerance $\varepsilon_{\text{DS}} > 0$ and define the distributional-similarity timestep as

$$T_{\text{DS}} := \min \left\{ t \;\middle|\; D_t \leq \varepsilon_{\text{DS}} \right\}. \tag{18}$$

As illustrated in Fig. 2, molecular datasets exhibit rapid decay in D_t, reflecting fast convergence to Gaussian marginals, while image datasets maintain larger D_t values over many more noise steps. From the perspectives of dependency decay and distributional similarity, we obtain a concrete and quantitative characterization of the Gaussianity of $\boldsymbol{x}_t$, ensuring that the approximated $\boldsymbol{x}_{T^*}$ is both sufficiently independent and marginally Gaussian. In addition, we define the operational Gaussian-approximation timestep as

$$T^* = \max(T_{\text{ID}}, T_{\text{DS}}), \tag{19}$$

which ensures that $\boldsymbol{x}_{T^*}$ is both sufficiently independent and marginally Gaussian.

4 Experiments

In this section, we empirically evaluate the proposed method on standard molecular generation benchmarks. We present the experimental setup, define the evaluation metrics, and report quantitative results on both generation quality and efficiency. Additional details on the Gaussianity tests and experimental configurations are provided in Appendix C and Appendix D, respectively.

4.1 Experimental Setup

Datasets. We conduct experiments on widely-used molecular datasets, QM9 [Ramakrishnan et al., 2014] and GEOM-Drugs [Axelrod and Gomez-Bombarelli, 2022]. QM9 contains 130k small molecules with up to 29 atoms, while GEOM-Drugs comprises 450k drug-like molecules with an average of 44 and up to 181 atoms. The configuration of datasets follows Hoogeboom et al. [2022] for regular generation and Xu et al. [2023] for latent-space generation, respectively.

Baselines. For molecule generation, we conduct comparison experiments on several competitive baselines. G-Schnet [Gebauer et al., 2019] and Equivariant Normalizing Flows (ENF) [Garcia Satorras et al., 2021] employ autoregressive models for molecule generation. Equivariant Graph Diffusion Model (EDM) [Hoogeboom et al., 2022], Geometric Latent Diffusion Model (GeoLDM) [Xu et al., 2023], and Equivariant Flow Matching model (EquiFM) [Song et al., 2023] are three representative GPPGMs from different perspectives for molecule generation, including regular diffusion, latent diffusion, and flow-matching, respectively. Moreover, the invariant versions of EDM (GDM) and GeoLDM (GraphLDM) are also employed for comparison.

Metrics We evaluate our method on standard molecular generation benchmarks using two broad classes of metrics: generation quality and efficiency. For generation quality, we report **validity** (the proportion of chemically valid molecules according to standard valency checks), **uniqueness** (the proportion of distinct molecules among generated samples), **molecular stability** (the fraction of generated molecules satisfying correct valency constraints), and **atom stability** (the fraction of generated atoms satisfying correct valency constraints). Following prior work [Hong et al., 2025], these metrics are computed using RDKit-based validation and duplicate filtering over 10,000 generated samples. For efficiency, we record the average **sampling time (S-Time)** in GPU seconds per sample and total **training time (T-Time)** in GPU days, both measured on identical hardware before and after applying our Gaussian approximation strategy. Moreover, the **trajectory length (Steps)** T^* is also shown in the results. These metrics collectively quantify the fidelity, diversity, and practical computational benefits of our method.

4.2 Quantitative Performance

Table 1: Quantitative results on the QM9 dataset. The best results are shown in **bold**. Metrics are calculated using 10,000 samples generated from each model. We run the evaluation for 3 times and report the mean value. Compared with previous methods, GA benefits all methods, achieving up to a 2.3% improvement in the Valid * Uniq metric, and significantly reduces the generation trajectory length by 40% without harming learning accuracy. All GA-compared baselines are tested using our implementation. The best results are shown in **bold**.

	Generation Performance				Efficiency		
Model	**Atom Sta** (%)	**Mol Sta** (%)	**Valid** (%)	**Valid * Uniq** (%)	**S-Time** (GPU sec.)	**T-Time** (GPU day)	**Traj. Len.** (Steps)
Data	99.0	95.2	97.7	97.7	-	-	-
ENF	85.0	4.9	40.2	39.4	-	-	-
G-Schnet	95.7	68.1	85.5	80.3	-	-	-
GDM-AUG	97.6	71.6	90.4	89.5	0.52	2.9	1000
GraphLDM	97.2	70.5	83.6	82.7	0.36	5.7	1000
EDM	98.4	81.8	91.9	90.7	0.65	5.6	1000
EDM + IAGA	**98.9**	**85.6**	**94.7**	**92.0**	**0.36**	**3.1**	**550**
GeoLDM	98.9	89.8	94.0	91.9	0.64	11.7	1000
GeoLDM + IAGA	**99.2**	**92.3**	**96.7**	**94.4**	**0.42**	**7.2**	**650**
EquiFM	98.5	87.3	94.9	93.4	0.17	6.2	1000
EquiFM + IAGA	**99.0**	**91.2**	**96.2**	**93.7**	**0.15**	**4.9**	**800**

'-' denotes the invalid or not recorded setting in the original publication.

We evaluate the effectiveness of the proposed Gaussian Approximation (GA) across multiple molecular generative baselines on both the QM9 and GEOM-Drugs datasets. As shown in Tables 1 and 2, GA consistently improves generation quality while significantly reducing both training and sampling cost. Crucially, our method shortens the diffusion trajectory, by up to 40%, without degrading the learning accuracy of the generative model. This is because GA does not alter the original noise schedule or variance scaling used during training; instead, it exploits the observation that molecular data loses its identity rapidly in the diffusion process, allowing training and sampling to begin from an earlier noise step without violating the underlying stochastic process. On the QM9 dataset, GA yields up to a 2.3% improvement in the *Valid * Uniq* metric, reflecting gains in both chemical correctness and diversity of the generated molecules

Table 2: Quantitative results on GEOM dataset. Metrics are calculated using 10,000 samples generated from each model. We run the evaluation for 3 times and report the mean value. In general, GA improves generation performance and provides better efficiency across models. The best results are shown in **bold**.

	Generation Performance		Efficiency	
Model	**Atom Sta** (%)	**Valid (%)**	**S-Time** (GPU sec.)	**Traj. Len.** (Steps)
Data	86.5	99.9	–	–
GDM-AUG	77.7	91.8	–	1000
GraphLDM	76.2	97.2	–	1000
EDM	81.3	92.6	10.9	1000
EDM + IAGA	**84.3**	**93.4**	**6.4**	**650**
GeoLDM	84.4	**99.3**	10.2	1000
GeoLDM + IAGA	**89.3**	98.0	**7.1**	**650**

'-' denotes the invalid or not recorded setting in the original publication.

Similar benefits are observed on the more challenging GEOM-Drugs dataset. In this experiment, we omit metrics such as uniqueness (which is consistently close to 100%) and molecule stability (which remains near 0%) due to their limited discriminative value across different methods. Overall, GA consistently improves atom-level stability and reduces the training and sampling time across various baselines. These improvements are particularly noteworthy given that GA requires no changes to model architecture or parameters. Instead, it modifies only the generation trajectory length by leveraging the rapid identity decay characteristic of molecular structures. This enables the model to focus on denoising stages where the molecular structure begins to emerge, leading to faster convergence without incurring additional transport cost or unnecessary noise inference from the skipped steps.

5 Related Work

Probability Path-based Generative Models (PPGMs). PPGMs generate samples over data distributions by learning a transport process that maps simple prior distributions to complex data distributions through a sequence of structured transformations, i.e., the probability path. Specifically, diffusion-based generative models simulate this sequential transformation via stochastic differential equations, which have emerged as a powerful paradigm for multi-modal data synthesis [Croitoru et al., 2023, Kementzidis et al., 2025, Xu et al.]. However, their iterative sampling (often requiring hundreds of steps) poses a significant speed bottleneck. A variety of techniques aim to accelerate diffusion sampling, such as progressive distillation [Salimans and Ho, 2022] and learned noising schedules [Williams et al., 2024]. Nevertheless, the training process still typically requires hundreds of steps. Beyond diffusion models, Flow Matching offers a fresh perspective on acceleration. Flow Matching trains a continuous normalizing flow by regressing an optimal vector field along prescribed paths. Lipman et al. [2022] showed that using diffusion-style Gaussian paths in flow matching yields more robust training and faster ODE-based sampling. However, the nonlinear and high-curvature nature of learned transport fields makes it challenging to accurately approximate such trajectories with few discretization steps during training [Hassan et al., 2024, Eijkelboom et al., 2024].

Gaussian Approximation (GA). GA has long been a cornerstone in machine learning theory and practice. The Central Limit Theorem provides a classical justification: aggregates of many random factors tend toward a Gaussian distribution, which often explains why high-dimensional features or latent codes appear approximately normal [Hazra et al., 2021, Düker et al., 2024]. Some researchers have explored the potential of the Gaussian approximation in generative modeling. For instance, Wang and Vastola observe that at high noise levels, the learned diffusion score can be well-approximated by a linear Gaussian model. Therefore, they can skip 15–30% of the sampling steps without degrading output fidelity. Such findings reinforce the idea that Gaussian assumptions can serve as an effective proxy for complex distributions in certain regimes, providing practical speedups without significant fidelity loss.

6 Conclusion and Future Work

In this work, we introduced a principled framework for efficient GPPGMs on 3D molecular generation. By leveraging zero-mean preprocessing and empirical variance estimation, we proposed an analytic Gaussian approximation that identifies a characteristic time step T^* at which data identity effectively vanishes and the forward process becomes distributionally Gaussian. This approximation enables the truncation of redundant noise steps, which are inefficient transport between "Gaussian-like" distributions. Therefore, our IAGA can improve the efficiency of both sampling and training and yields consistent improvements in generation quality across multiple molecular generation benchmarks.

Future Work. Despite its empirical success, the current framework assumes that the data modality is zero-mean invariant. While this assumption holds in many geometric and categorical domains, it is not valid for modalities like natural images or videos, where the absolute mean carries semantic information. Extending our methodology to such domains requires further methods for determining identity loss and Gaussianity without relying on zero-mean centering.

References

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Simon Axelrod and Rafael Gomez-Bombarelli. Geom, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9(1):185, 2022.

Andrew C Berry. The accuracy of the gaussian approximation to the sum of independent variates. *Transactions of the american mathematical society*, 49(1):122–136, 1941.

Victor Chernozhukov, Denis Chetverikov, and Kengo Kato. Gaussian approximations and multiplier bootstrap for maxima of sums of high-dimensional random vectors. 2013.

Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9): 10850–10869, 2023.

Hang Deng and Cun-Hui Zhang. Beyond gaussian approximation. *The Annals of Statistics*, 48(6): 3643–3671, 2020.

Marie-Christine Düker, Robert Lund, and Vladas Pipiras. High-dimensional latent gaussian count time series: Concentration results for autocovariances and applications. *Electronic Journal of Statistics*, 18(2):5484–5562, 2024.

Floor Eijkelboom, Grigory Bartosh, Christian Andersson Naesseth, Max Welling, and Jan-Willem van de Meent. Variational flow matching for graph generation. *Advances in Neural Information Processing Systems*, 37:11735–11764, 2024.

Daan Frenkel and Berend Smit. *Understanding molecular simulation: from algorithms to applications*. Elsevier, 2023.

Victor Garcia Satorras, Emiel Hoogeboom, Fabian Fuchs, Ingmar Posner, and Max Welling. E (n) equivariant normalizing flows. *Advances in Neural Information Processing Systems*, 34:4181–4192, 2021.

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

Majdi Hassan, Nikhil Shenoy, Jungyoon Lee, Hannes Stärk, Stephan Thaler, and Dominique Beaini. Et-flow: Equivariant flow-matching for molecular conformer generation. *Advances in Neural Information Processing Systems*, 37:128798–128824, 2024.

Arnab Hazra, Raphaël Huser, and Árni V Jóhannesson. Latent gaussian models for high-dimensional spatial extremes. *arXiv preprint arXiv:2110.02680*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020.

Haokai Hong, Wanyu Lin, and Kay Chen Tan. Accelerating 3d molecule generation via jointly geometric optimal transport. In *The Thirteenth International Conference on Learning Representations*, 2025.

Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pages 8867–8887. PMLR, 2022.

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

Georgios Kementzidis, Erin Wong, John Nicholson, Ruichen Xu, and Yuefan Deng. An iterative framework for generative backmapping of coarse grained proteins. *arXiv preprint arXiv:2505.18082*, 2025.

Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 69(6):066138, 2004.

Greg Landrum et al. Rdkit: Open-source cheminformatics, 2016.

Sang-gil Lee, Heeseung Kim, Chaehun Shin, Xu Tan, Chang Liu, Qi Meng, Tao Qin, Wei Chen, Sungroh Yoon, and Tie-Yan Liu. Priorgrad: Improving conditional denoising diffusion models with data-dependent adaptive prior. *arXiv preprint arXiv:2106.06406*, 2021.

Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip Torr. Controllable text-to-image generation. *Advances in neural information processing systems*, 32, 2019.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

Weibo Mao, Chenxin Xu, Qi Zhu, Siheng Chen, and Yanfeng Wang. Leapfrog diffusion model for stochastic trajectory prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5517–5526, 2023.

Frank J Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

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

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.

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

Yuxuan Song, Jingjing Gong, Minkai Xu, Ziyao Cao, Yanyan Lan, Stefano Ermon, Hao Zhou, and Wei-Ying Ma. Equivariant flow matching with hybrid probability transport for 3d molecule generation. *Advances in Neural Information Processing Systems*, 36:549–568, 2023.

Roman Vershynin. How close is the sample covariance matrix to the actual covariance matrix? *Journal of Theoretical Probability*, 25(3):655–686, 2012.

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.

Binxu Wang and John Vastola. The unreasonable effectiveness of gaussian score approximation for diffusion models and its applications. *Transactions on Machine Learning Research*.

Christopher Williams, Andrew Campbell, Arnaud Doucet, and Saifuddin Syed. Score-optimal diffusion schedules. *arXiv preprint arXiv:2412.07877*, 2024.

Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations*.

Minkai Xu, Alexander S Powers, Ron O Dror, Stefano Ermon, and Jure Leskovec. Geometric latent diffusion models for 3d molecule generation. In *International Conference on Machine Learning*, pages 38592–38610. PMLR, 2023.

Hui Zhang, Zuxuan Wu, Zhen Xing, Jie Shao, and Yu-Gang Jiang. Adadiff: Adaptive step selection for fast diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 9914–9922, 2025.

Xuan Zhang, Limei Wang, Jacob Helwig, Youzhi Luo, Cong Fu, Yaochen Xie, Meng Liu, Yuchao Lin, Zhao Xu, Keqiang Yan, Keir Adams, Maurice Weiler, Xiner Li, Tianfan Fu, Yucheng Wang, Haiyang Yu, YuQing Xie, Xiang Fu, Alex Strasser, Shenglong Xu, Yi Liu, Yuanqi Du, Alexandra Saxton, Hongyi Ling, Hannah Lawrence, Hannes Stärk, Shurui Gui, Carl Edwards, Nicholas Gao, Adriana Ladera, Tailin Wu, Elyssa F. Hofgard, Aria Mansouri Tehrani, Rui Wang, Ameya Daigavane, Montgomery Bohde, Jerry Kurtin, Qian Huang, Tuong Phung, Minkai Xu, Chaitanya K. Joshi, Simon V. Mathis, Kamyar Azizzadenesheli, Ada Fang, Alán Aspuru-Guzik, Erik Bekkers, Michael Bronstein, Marinka Zitnik, Anima Anandkumar, Stefano Ermon, Pietro Liò, Rose Yu, Stephan Günnemann, Jure Leskovec, Heng Ji, Jimeng Sun, Regina Barzilay, Tommi Jaakkola, Connor W. Coley, Xiaoning Qian, Xiaofeng Qian, Tess Smidt, and Shuiwang Ji. Artificial intelligence for science in quantum, atomistic, and continuum systems. *arXiv preprint arXiv:2307.08423*, 2023.

Appendix

A Examples and Analysis of Zero-mean Invariant Data

We aim to show that for data modalities satisfying zero-mean invariance, the operation of mean-centering preserves all structural information relevant to generative modeling. We mainly discuss the Euclidean and non-uniform one-hot cases, which are tested in our experiments.

Euclidean Data. Let $\{\boldsymbol{x}_i\}_{i=1}^n \subset \mathbb{R}^d$ denote a collection of n vectors (e.g., 3D Euclidean coordinates of atoms). Define the sample mean $\bar{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i$, and let $\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i - \bar{\boldsymbol{x}}$ be the centered representation. We claim that pairwise Euclidean distances are invariant under mean-centering:

$$\|\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{x}}_j\|_2 = \|(\boldsymbol{x}_i - \bar{\boldsymbol{x}}) - (\boldsymbol{x}_j - \bar{\boldsymbol{x}})\|_2 = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2. \tag{20}$$

Hence, all geometric properties that depend on inter-point distances, such as adjacency structures, bond lengths, or conformational shapes, are preserved exactly under centering. Consequently, zero-mean projection retains full information about the relational structure of the data.

Non-Uniform One-Hot Categorical Vectors. Let $h_i \in \{0,1\}^d$ denote a one-hot encoded vector satisfying $\sum_{j=1}^d (h_i)_j = 1$, and let $\bar{h} = \frac{1}{n}\sum_{i=1}^n h_i$ be the sample mean across a batch of n such vectors. Define the centered vector $\tilde{h}_i = h_i - \bar{h}$. Note that each $\tilde{h}_i \in \mathbb{R}^d$ lies in a subspace orthogonal to the constant vector $\mathbf{1}_d$, since:

$$\sum_{j=1}^d (\tilde{h}_i)_j = \sum_{j=1}^d (h_i - \bar{h})_j = 1 - \sum_{j=1}^d \bar{h}_j = 0. \tag{21}$$

Moreover, the inner product between two centered vectors $\tilde{h}_i$ and $\tilde{h}_j$ satisfies:

$$\langle \tilde{h}_i, \tilde{h}_j \rangle = \langle h_i, h_j \rangle - \langle h_i, \bar{h} \rangle - \langle \bar{h}, h_j \rangle + \langle \bar{h}, \bar{h} \rangle, \tag{22}$$

from which it follows that pairwise centered dot products retain sufficient information to distinguish between original categorical identities once the category set is not degenerate (e.g., uniform). Since each one-hot vector h_i is uniquely defined by a single active index, subtracting the global mean $\bar{h}$ merely induces a translation within the categorical simplex. The position of the maximal entry in $\tilde{h}_i$ still identifies the active class as long as $\bar{h}$ does not collapse distinct h_i vectors onto the same centered value. Therefore, for any non-uniform categorical data embedded via one-hot encoding, mean-centering preserves the identity of the active component up to an affine transformation of the ambient space. As a result, zero-mean preprocessing retains the categorical semantics necessary for generative modeling under Euclidean approximation schemes.

B Proof of Proposition 3

Cumulants. Let $X \in \mathbb{R}^d$ have moment generating function (mgf) $M_X(u) = \mathbb{E}[e^{u^\top X}]$ and cumulant generating function $K_X(u) = \log M_X(u)$, $u \in \mathbb{R}^d$. The *k-th cumulant tensor* is

$$\big(C^{(k)}(X)\big)_{i_1,\dots,i_k} = \left.\frac{\partial^k K_X(u)}{\partial u_{i_1} \cdots \partial u_{i_k}}\right|_{u=0}, \qquad k \geq 1.$$

In particular $C^{(1)}(X) = \mu := \mathbb{E}[X]$, $C^{(2)}(X) = \Sigma := \mathrm{Cov}(X)$, and $C^{(k)}(G) = 0$ for all $k \geq 3$ if G is Gaussian.

Setup (VP forward map). Let $\{\boldsymbol{x}_t\}_{t=0}^T$ be the forward (noising) trajectory under a variance-preserving schedule. Fix $t \in [0,T)$ and $s > t$. Then

$$\boldsymbol{x}_s \;=\; \sqrt{\bar{\alpha}_{s|t}}\,\boldsymbol{x}_t \;+\; \sqrt{1-\bar{\alpha}_{s|t}}\,\boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I}),\;\; \boldsymbol{\varepsilon} \perp \boldsymbol{x}_t,\;\; \bar{\alpha}_{s|t} := \bar{\alpha}_s/\bar{\alpha}_t \in (0,1). \tag{23}$$

Gaussianity–and–independence functional (definition). Let $\mathsf{D} := \{\mathrm{Diag}(v) : v \in \mathbb{R}^d\}$ be the diagonal subspace and define the orthogonal projections

$$\Pi_{\mathsf{D}}(\Sigma) := \mathrm{Diag}(\mathrm{diag}\,\Sigma), \qquad \Pi_{\mathsf{D}^\perp}(\Sigma) := \Sigma - \Pi_{\mathsf{D}}(\Sigma).$$

For weights $\beta > 0$ and $w_k > 0$ ($k \geq 3$), define for any random vector $x \in \mathbb{R}^d$

$$\mathcal{H}^{(K)}(x) \;:=\; \beta \left\|\Pi_{\mathsf{D}^\perp}(\mathrm{Cov}(x))\right\|_F \;+\; \sum_{k=3}^{K} w_k \left\|C^{(k)}(x)\right\|_F. \tag{24}$$

Lemma B.1 (VP propagation of moments/cumulants and contraction of $\mathcal{H}^{(K)}$)**.** *Let $t \in [0, T)$ and $s > t$, and write $a := \bar{\alpha}_{s|t} \in (0, 1)$. Under* (23)*, with $\Sigma_t := \mathrm{Cov}(\boldsymbol{x}_t)$ and $B_{k,t} := \left\|C^{(k)}(\boldsymbol{x}_t)\right\|_F$ for $k \geq 3$,*

$$\mu_s = \sqrt{a}\,\mu_t, \qquad \Sigma_s = a\,\Sigma_t + (1-a)\,\mathbf{I}, \qquad \left\|C^{(k)}(\boldsymbol{x}_s)\right\|_F = a^{k/2} B_{k,t} \;\; (k \geq 3).$$

Consequently,

$$\mathcal{H}^{(K)}(\boldsymbol{x}_s) = \beta\, a \left\|\Pi_{\mathsf{D}^\perp}(\Sigma_t)\right\|_F + \sum_{k=3}^{K} w_k\, a^{k/2}\, B_{k,t}, \tag{25}$$

and

$$\frac{\partial}{\partial a}\,\mathcal{H}^{(K)}(\boldsymbol{x}_s) = \beta \left\|\Pi_{\mathsf{D}^\perp}(\Sigma_t)\right\|_F + \sum_{k=3}^{K} w_k\, \frac{k}{2}\, a^{k/2-1}\, B_{k,t} \;>\; 0$$

whenever $\left\|\Pi_{\mathsf{D}^\perp}(\Sigma_t)\right\|_F + \sum_{k=3}^{K} B_{k,t} > 0$. Hence, since $a = \bar{\alpha}_{s|t}$ decreases strictly in s for a VP schedule, $\mathcal{H}^{(K)}(\boldsymbol{x}_s)$ is strictly decreasing in s unless $\boldsymbol{x}_t$ is already an independent Gaussian (in which case $\mathcal{H}^{(K)}(\boldsymbol{x}_s) \equiv 0$).

Proof. First, $\mu_s = \mathbb{E}[\boldsymbol{x}_s] = \sqrt{a}\,\mu_t + \sqrt{1-a}\,\mathbb{E}[\boldsymbol{\varepsilon}] = \sqrt{a}\,\mu_t$. For the covariance, write $\boldsymbol{x}_s = \sqrt{a}\,\boldsymbol{x}_t + \sqrt{1-a}\,\boldsymbol{\varepsilon}$ and center by the means:

$$\boldsymbol{x}_s - \mu_s = \sqrt{a}\,(\boldsymbol{x}_t - \mu_t) + \sqrt{1-a}\,\boldsymbol{\varepsilon}.$$

Independence and $\mathbb{E}[\boldsymbol{\varepsilon}] = 0$ give

$$\Sigma_s = \mathbb{E}\big[(\boldsymbol{x}_s - \mu_s)(\boldsymbol{x}_s - \mu_s)^\top\big] = a\,\Sigma_t + (1-a)\,\mathbb{E}[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^\top] = a\,\Sigma_t + (1-a)\,\mathbf{I}.$$

Linearity of $\Pi_{\mathsf{D}^\perp}$ and $\mathbf{I} \in \mathsf{D}$ yield

$$\Pi_{\mathsf{D}^\perp}(\Sigma_s) = \Pi_{\mathsf{D}^\perp}(a\,\Sigma_t) + (1-a)\,\Pi_{\mathsf{D}^\perp}(\mathbf{I}) = a\,\Pi_{\mathsf{D}^\perp}(\Sigma_t),$$

hence $\|\Pi_{\mathsf{D}^\perp}(\Sigma_s)\|_F = a\,\|\Pi_{\mathsf{D}^\perp}(\Sigma_t)\|_F$.

For cumulants, independence implies additivity: $C^{(k)}(X+Y) = C^{(k)}(X) + C^{(k)}(Y)$ when $X \perp Y$ (this follows from $K_{X+Y}(u) = K_X(u) + K_Y(u)$). Homogeneity follows from $K_{cX}(u) = \log \mathbb{E}[e^{u^\top cX}] = K_X(cu)$ and the chain rule:

$$\frac{\partial^k}{\partial u_{i_1} \cdots \partial u_{i_k}} K_{cX}(u)\bigg|_{u=0} = c^k\, \frac{\partial^k}{\partial u_{i_1} \cdots \partial u_{i_k}} K_X(u)\bigg|_{u=0} \Rightarrow\; C^{(k)}(cX) = c^k C^{(k)}(X).$$

Because a Gaussian has $C^{(k)}(\boldsymbol{\varepsilon}) = 0$ for $k \geq 3$,

$$C^{(k)}(\boldsymbol{x}_s) = C^{(k)}(\sqrt{a}\,\boldsymbol{x}_t) + C^{(k)}(\sqrt{1-a}\,\boldsymbol{\varepsilon}) = a^{k/2} C^{(k)}(\boldsymbol{x}_t),$$

so $\|C^{(k)}(\boldsymbol{x}_s)\|_F = a^{k/2}\|C^{(k)}(\boldsymbol{x}_t)\|_F$. Plugging these identities into (24) gives (25). Finally, since $\beta > 0, w_k > 0$ and $B_{k,t} \geq 0$, the displayed derivative is > 0 whenever not all terms vanish. As a strictly decreases in s for VP, $\mathcal{H}^{(K)}(\boldsymbol{x}_s)$ strictly decreases in s unless already identically zero. □

Lemma B.2 (θ-decomposition via prefix sums)**.** *For $a \in (0, 1)$, define*

$$\theta_2(a) := a - a^{3/2}, \qquad \theta_m(a) := a^{m/2} - a^{(m+1)/2} \;\; (3 \leq m \leq K-1), \qquad \theta_K(a) := a^{K/2},$$

and for $m \in \{2, 3, \dots, K\}$ define the prefix functionals

$$\mathcal{H}^{(m)}(\boldsymbol{x}_t) := \beta \left\| \Pi_{\mathsf{D}^\perp}(\Sigma_t) \right\|_F + \sum_{k=3}^{m} w_k \left\| C^{(k)}(\boldsymbol{x}_t) \right\|_F.$$

Then $\theta_m(a) > 0$ for all m and $a \in (0,1)$, and the closed form (25) *admits*

$$\mathcal{H}^{(K)}(\boldsymbol{x}_s) = \sum_{m=2}^{K} \theta_m(a)\, \mathcal{H}^{(m)}(\boldsymbol{x}_t), \qquad \textit{with} \quad \sum_{m=j}^{K} \theta_m(a) = a^{j/2} \ \textit{for each } j \in \{2, 3, \dots, K\}. \tag{26}$$

Proof. For $0 < a < 1$, $\theta_m(a) = a^{m/2}(1 - a^{1/2}) > 0$ for $m \le K-1$ and $\theta_K(a) = a^{K/2} > 0$. To prove (26), expand the right-hand side:

$$\sum_{m=2}^{K} \theta_m(a)\, \mathcal{H}^{(m)}(\boldsymbol{x}_t) = \Big(\sum_{m=2}^{K} \theta_m(a) \Big) \beta \|\Pi_{\mathsf{D}^\perp}(\Sigma_t)\|_F + \sum_{k=3}^{K} \Big(\sum_{m=k}^{K} \theta_m(a) \Big) w_k \|C^{(k)}(\boldsymbol{x}_t)\|_F.$$

Hence it suffices to show the *tail-sum identities* $\sum_{m=2}^{K} \theta_m(a) = a$ and $\sum_{m=k}^{K} \theta_m(a) = a^{k/2}$ for each $k \in \{3, \dots, K\}$. For $k \le K-1$,

$$\sum_{m=k}^{K-1} \big(a^{m/2} - a^{(m+1)/2} \big) + a^{K/2} = \big(a^{k/2} - a^{(k+1)/2} \big) + \cdots + \big(a^{(K-1)/2} - a^{K/2} \big) + a^{K/2} = a^{k/2},$$

a telescoping sum; the case $k = K$ is immediate. The identity for $k = 2$ is the same computation with $k = 2$. Substituting these tail-sums into the expansion recovers (25). □

Lemma B.3 (Order preservation under prefix dominance)**.** *Let A, B be two classes of data set. Assume the* prefix dominance

$$\mathcal{H}^{(m)}(\boldsymbol{x}_t^A) \ \le \ \mathcal{H}^{(m)}(\boldsymbol{x}_t^B) \quad \textit{for all } m = 2, 3, \dots, K, \tag{27}$$

with at least one strict inequality. Then, for every $s > t$ (equivalently, every $a \in (0,1)$),

$$\mathcal{H}^{(K)}(\boldsymbol{x}_s^A) = \sum_{m=2}^{K} \theta_m(a)\, \mathcal{H}^{(m)}(\boldsymbol{x}_t^A) < \sum_{m=2}^{K} \theta_m(a)\, \mathcal{H}^{(m)}(\boldsymbol{x}_t^B) = \mathcal{H}^{(K)}(\boldsymbol{x}_s^B),$$

and the inequality is strict because all $\theta_m(a) > 0$ for $a \in (0,1)$.

Proof. By Lemma B.2, $\mathcal{H}^{(K)}(\boldsymbol{x}_s) = \sum_{m=2}^{K} \theta_m(a)\, \mathcal{H}^{(m)}(\boldsymbol{x}_t)$ with $\theta_m(a) > 0$. Applying (27) termwise gives $\mathcal{H}^{(K)}(\boldsymbol{x}_s^A) \le \mathcal{H}^{(K)}(\boldsymbol{x}_s^B)$. Strictness follows because at least one index $m^\star$ satisfies $\mathcal{H}^{(m^\star)}(\boldsymbol{x}_t^A) < \mathcal{H}^{(m^\star)}(\boldsymbol{x}_t^B)$ and $\theta_{m^\star}(a) > 0$, hence the weighted sum is strictly smaller. □

Lemma B.1 shows that for each initialization, $s \mapsto \mathcal{H}^{(K)}(\boldsymbol{x}_s)$ is strictly decreasing (unless already at an independent Gaussian). Lemma B.3 states that if A is *prefix-dominant* over B at time t, then $\mathcal{H}^{(K)}(\boldsymbol{x}_s^A) < \mathcal{H}^{(K)}(\boldsymbol{x}_s^B)$ for every $s > t$. Therefore, for any threshold $\varepsilon > 0$, the hitting times

$$T_X(\varepsilon) := \inf\{ s > t : \mathcal{H}^{(K)}(\boldsymbol{x}_s^X) \le \varepsilon \}$$

satisfy $T_A(\varepsilon) < T_B(\varepsilon)$, which formalizes that under VP the speed to gain sufficient Gaussianity for A is faster than for B whenever A starts closer to Gaussian in the sense of (27).

Moreover, if there exist nondecreasing functions $\varphi_{\mathrm{dep}}, \varphi_{\mathrm{ks}} : [0,\infty) \to [0,\infty)$ with $\varphi_{\mathrm{dep}}(0) = \varphi_{\mathrm{ks}}(0) = 0$ such that for every $s > t$,

$$\mathrm{Dep}(\boldsymbol{x}_s) \ \le \ \varphi_{\mathrm{dep}}\big(\mathcal{H}^{(K)}(\boldsymbol{x}_s)\big), \qquad D(\boldsymbol{x}_s) \ \le \ \varphi_{\mathrm{ks}}\big(\mathcal{H}^{(K)}(\boldsymbol{x}_s)\big), \tag{28}$$

then, for any tolerances $\varepsilon_{\mathrm{dep}}, \varepsilon_{\mathrm{DS}} > 0$,

$$T_{\mathrm{ID}}^A := \inf\{ s > t : \mathrm{Dep}(\boldsymbol{x}_s^A) \le \varepsilon_{\mathrm{dep}} \} \ \le \ T_{\mathrm{ID}}^B, \qquad T_{\mathrm{DS}}^A := \inf\{ s > t : D(\boldsymbol{x}_s^A) \le \varepsilon_{\mathrm{DS}} \} \ \le \ T_{\mathrm{DS}}^B,$$

and hence $T_A^* := \max(T_{\mathrm{ID}}^A, T_{\mathrm{DS}}^A) \ \le \ T_B^*$, with strict inequality if (27) is strict for some m.

C Gaussianity Test Details

C.1 Identity Test

By treating $\boldsymbol{x}_t$ as a sample from an intractable noised data distribution, we estimate the empirical mutual information (MI) across both the sample-wise and feature-wise dimensions of the data tensor. MI quantifies the degree of statistical dependency between random variables by measuring the divergence between their joint distribution and the product of their marginals. For two random vectors X and Y, the mutual information is defined as:

$$I(X;Y) = \int\int p_{X,Y}(x,y) \log \frac{p_{X,Y}(x,y)}{p_X(x)\, p_Y(y)}\, dx\, dy, \tag{29}$$

where $p_{X,Y}(x,y)$ denotes the joint probability density, and $p_X(x), p_Y(y)$ are the marginal densities of X and Y, respectively.

Given a data matrix representation of $\boldsymbol{x}_t$, we estimate the dependency across dimensions (features) within each data point and across data points for each dimension to represent the identity. Formally, let $\mathcal{X}_{\text{rows}}$ and $\mathcal{X}_{\text{cols}}$ denote the sets of row-wise and column-wise slices, respectively. Then, the empirical MI scores are given by:

$$\text{MI}_{\text{rows}} = \frac{1}{|\mathcal{X}_{\text{rows}}|} \sum_{x \in \mathcal{X}_{\text{rows}}} I(x), \quad \text{MI}_{\text{cols}} = \frac{1}{|\mathcal{X}_{\text{cols}}|} \sum_{x \in \mathcal{X}_{\text{cols}}} I(x), \tag{30}$$

where $I(x)$ denotes the estimated mutual information of the given vector x across its components. As t increases, these statistics decay toward zero, indicating diminishing dependency and the emergence of approximate independence in $\boldsymbol{x}_t$.

C.2 Distributional Similarity via KS-Test

To evaluate whether the noised data x_t has become sufficiently similar to the Gaussian distribution $\mathcal{N}(0, \tilde{v}_t I)$, we perform statistical testing based on the Kolmogorov–Smirnov (KS) criterion. At each test timestep t, we apply the one-sample KS test dimension-wise to the components of x_t after zero-centering, treating each variable as an independent sample drawn from the empirical distribution. Specifically, for each dimension $j \in \{1, \ldots, d\}$, we compute the empirical cumulative distribution function (CDF) $F_{t,j}(x)$ and compare it against the theoretical CDF $\Phi_{\tilde{v}_t}(x)$ of a univariate normal distribution with zero mean and variance $\tilde{v}_t$, derived analytically from the forward noise schedule. The test statistic is defined as:

$$D_{t,j} = \sup_x \left| F_{t,j}(x) - \Phi_{\tilde{v}_t}(x) \right|. \tag{31}$$

For each coordinate j, we test $H_0 : , x_t^{(j)} \sim \mathcal{N}(0, \tilde{v}_t)$ against $H_1 : , x_t^{(j)} \not\sim \mathcal{N}(0, \tilde{v}_t)$. Under H_0, with sample size n, the scaled statistic $\sqrt{n}, D_{t,j}$ converges to the Kolmogorov distribution with CDF $1 - 2\sum_{k=1}^{\infty}(-1)^{k-1}\exp(-2k^2\lambda^2)$, yielding the 5% critical threshold $D_{t,j} > c_{0.05}/\sqrt{n}$ (with $c_{0.05} \approx 1.36$ asymptotically). We declare that timestep t satisfies the Gaussianity criterion if at least 95% of coordinates fail to reject H_0, i.e., x_t is statistically indistinguishable from the reference $\mathcal{N}(0, \tilde{v}_t I)$ at 95% confidence level.

D Experimental Settings

D.1 Backbone model

In our experiments, all molecular generation baselines utilize the Equivariant Graph Neural Network (EGNN) [Satorras et al., 2021] as the backbone architecture for generative processing. EGNN operates on graphs embedded in Euclidean space and are designed to be equivariant under rigid-body transformations from the special Euclidean group $\text{SE}(3)$, including rotations and translations. This property ensures that molecular outputs transform consistently with the input geometry, preserving critical physical symmetries.

Formally, consider a molecule represented as a fully connected graph with N nodes, where each node i has coordinates $\boldsymbol{x}_i \in \mathbb{R}^3$ and associated atom features $\boldsymbol{h}_i \in \mathbb{R}^d$. At each EGNN layer, node features and positions are updated through message-passing operations:

$$\begin{aligned} \boldsymbol{m}_{ij} &= \phi_e(\boldsymbol{h}_i, \boldsymbol{h}_j, \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2), \\ \boldsymbol{h}_i' &= \phi_h\left(\boldsymbol{h}_i, \sum_{j \neq i} \alpha_{ij} \boldsymbol{m}_{ij}\right), \\ \boldsymbol{x}_i' &= \boldsymbol{x}_i + \sum_{j \neq i} \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\| + \epsilon} \phi_x(\boldsymbol{h}_i, \boldsymbol{h}_j, \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2), \end{aligned} \tag{32}$$

where ϕ_e, ϕ_h, and ϕ_x are learnable functions (typically MLPs), and α_{ij} is an optional attention or reweighting term. The update rule guarantees that output features are equivariant with respect to $\mathrm{SE}(3)$ transformations. This equivariant structure is critical for molecular generative tasks, as the physical properties of molecules are invariant to coordinate shifts and rotations.

D.2 Implementation Details

For all baseline models, we follow the official open-sourced codebases and retain their default hyperparameters unless otherwise specified. Gaussian Approximation is applied after the truncation step T^*, as estimated via our KS and MI-based Gaussianity evaluation.

All molecular generation evaluation metrics are computed on 10,000 generated molecules using RDKit [Landrum et al., 2016]. Validity and atom stability are defined by valency correctness, and uniqueness is computed as the percentage of distinct canonical SMILES. Sampling time is measured as the average GPU seconds to generate one molecule, while training time reflects total GPU days until the last pre-defined epochs in the official repositories.

All experiments are conducted on a computing cluster equipped with NVIDIA RTX 3090 GPUs, each with 24 GB memory. Training is parallelized across 2 GPUs using PyTorch DDP framework, while inference experiments are executed on a single GPU for fair comparison of sampling speed. The CPUs are Intel(R) Core(TM) i9-12900KF. Unless otherwise specified, we report sampling time as the average GPU seconds per generated sample, and training time in GPU days until the max epochs from the baselines' official repositories. All baseline implementations use their official code, pre-trained weights (if available) and hyperparameters to ensure comparability.