Geometric Latent Diffusion Models for 3D Molecule Generation

Minkai Xu [1]   **Alexander S. Powers** [1 2]   **Ron O. Dror** [* 1]   **Stefano Ermon** [* 1]   **Jure Leskovec** [* 1]

Abstract

Generative models, especially diffusion models (DMs), have achieved promising results for generating feature-rich geometries and advancing foundational science problems such as molecule design. Inspired by the recent huge success of Stable (latent) Diffusion models, we propose a novel and principled method for 3D molecule generation named Geometric Latent Diffusion Models (GEOLDM). GEOLDM is the first latent DM model for the molecular geometry domain, composed of autoencoders encoding structures into continuous latent codes and DMs operating in the latent space. Our key innovation is that for modeling the 3D molecular geometries, we capture its critical roto-translational equivariance constraints by building a point-structured latent space with both invariant scalars and equivariant tensors. Extensive experiments demonstrate that GEOLDM can consistently achieve better performance on multiple molecule generation benchmarks, with up to 7% improvement for the valid percentage of large biomolecules. Results also demonstrate GEOLDM's higher capacity for controllable generation thanks to the latent modeling. Code is provided at https://github.com/MinkaiXu/GeoLDM.

1. Introduction

Generative modeling for feature-rich geometries is an important task for many science fields. Typically, geometries can be represented as point clouds where each point is embedded in the Cartesian coordinates and labeled with rich features. Such structures are ubiquitous in scientific domains, *e.g.*, we can represent molecules as atomic graphs in 3D (Schütt et al., 2017) and proteins as proximity spatial graphs over amino acids (Jing et al., 2021). Therefore, de-

veloping effective geometric generative models holds great promise for scientific discovery problems such as material and drug design (Pereira et al., 2016; Graves et al., 2020; Townshend et al., 2021). Recently, considerable progress has been achieved with machine learning approaches, especially deep generative models. For example, Gebauer et al. (2019); Luo & Ji (2021) and Satorras et al. (2021a) proposed data-driven methods to generate 3D molecules (in silico) with autoregressive and flow-based models respectively. However, despite great potential, the results are still unsatisfactory with low chemical validity and small molecule size, due to the insufficient capacity of the underlying generative models (Razavi et al., 2019).

Most recently, diffusion models (DMs) (Ho et al., 2020; Song et al., 2021) have emerged with surprising results on image synthesis (Meng et al., 2022) and beyond (Kong et al., 2021; Li et al., 2022). DMs define a diffusion process that gradually perturbs the data, and learn neural networks to reverse this corruption by progressive denoising. Then the denoising network can conduct generation by iteratively cleaning data initialized from random noise. Several studies have also applied such frameworks to the geometric domain, especially molecular structures (Hoogeboom et al., 2022; Wu et al., 2022; Anand & Achim, 2022). However, the existing models typically run DMs directly in the atomic feature space, which typically is composed of diverse physical quantities, *e.g.*, charge, atom types, and coordinates. These features are multi-modal with discrete, integer, and continuous variables, making unified Gaussian diffusion frameworks sub-optimal (Hoogeboom et al., 2022; Wu et al., 2022) or requiring sophisticated, decomposed modeling of different variables (Anand & Achim, 2022). Besides, the high dimensionality of input features also increases DM modeling difficulty, since the model's training and sampling require function forward and backward computation in the full input dimension. Therefore, the validity rate of generated molecules is still not satisfying enough, and an ideal approach would be a more flexible and expressive framework for modeling complex structures.

In this paper, we propose a novel and principled method to overcome the above limitations by utilizing a smoother latent space, named Geometric Latent Diffusion Models (GEOLDM). GEOLDM is set up as (variational) autoencoders (AEs) with DMs operating on the latent space. The
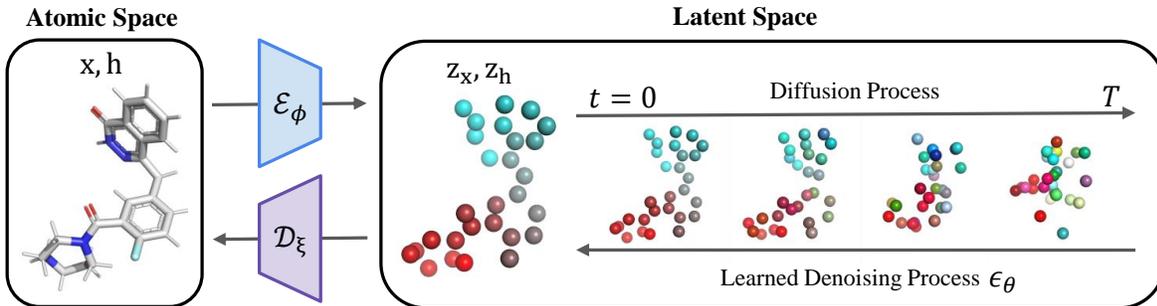
Atomic Space

Latent Space

Figure 1. Illustration of GEOLDM. The encoder $\mathcal{E}_\phi$ encodes molecular features $\mathbf{x}, \mathbf{h}$ into equivariant latent variables $\mathbf{z}_x, \mathbf{z}_h$, and the latent diffusion transitions $q(\mathbf{z}_{x,t}, \mathbf{z}_{h,t} | \mathbf{z}_{x,t-1}, \mathbf{z}_{h,t-1})$ gradually added noise until the latent codes converge to Gaussians. Symmetrically, for generation, an initial latent $\mathbf{z}_{x,T}, \mathbf{z}_{h,T}$ is sampled from standard normal distributions and progressively refined by equivariant denoising dynamics $\boldsymbol{\epsilon}_\theta(\mathbf{z}_x, \mathbf{z}_h)$. The final latents $\mathbf{z}_x, \mathbf{z}_h$ are further decoded back to molecular point clouds with the decoder $\mathcal{D}_\xi$.

encoder maps the raw geometries into a lower-dimensional representational space, and DMs learn to model the smaller and smoother distribution of latent variables. For modeling the 3D molecular geometry, our key innovation is constructing sufficient conditions for latent space to satisfy the critical 3D roto-translation equivariance constraints, where simply equipping latent variables with scalar-valued[1] (*i.e.*, invariant) variables lead to extremely poor generation quality. Technically, we realize this constraint by building the latent space as point-structured latents with both invariant and equivariant variables, which in practice is implemented by parameterizing encoding and decoding functions with advanced equivariant networks. To the best of our knowledge, we are the first work to incorporate equivariant features, *i.e.*, tensors, into the latent space modeling.

A unique advantage of GEOLDM is that unlike previous DM methods operating in the feature domain, we explicitly incorporate a latent space to capture the complex structures. This unified formulation enjoys several strengths. First, by mapping raw features into regularized latent space, the latent DMs learn to model a much smoother distribution. This alleviates the difficulty of directly modeling complex structures' likelihood, and is therefore more expressive. Besides, the latent space enables GEOLDM to conduct training and sampling with a lower dimensionality, which can also benefit the generative modeling complexity. Furthermore, the use of latent variables also allows for better control over the generation process, which has shown promising results in text-guided image generation (Rombach et al., 2022). This enables the user to generate specific types of molecules with desired properties. Finally, our framework is very general and can be extended to various downstream molecular problems where DMs have shown promising results, *i.e.*, target drug design (Lin et al., 2022) and antigen-specific antibody generation (Luo et al., 2022).

---

[1]In this paper, we will use "scalar" and "tensor" to interchangeably refer to type-0 (invariant) and type-1 (equivariant) features, following the common terminologies used in geometric literature.

We conduct detailed evaluations of GEOLDM on multiple benchmarks, including both unconditional and property-conditioned molecule generation. Results demonstrate that GEOLDM can consistently achieve superior generation performance on all the metrics, with up to 7% higher valid rate for large biomolecules. Empirical studies also show significant improvement for controllable generation thanks to latent modeling. All the empirical results demonstrate that GEOLDM enjoys a significantly higher capacity to explore the chemical space and generate structurally novel and chemically feasible molecules.

2. Related Work

Latent Generative Models. To improve the generative modeling capacity, a lot of research (Dai & Wipf, 2019; Yu et al., 2022) has been conducted to learn more expressive generative models over the latent space. VQ-VAEs (Razavi et al., 2019) proposed to discretize latent variables and use autoregressive models to learn an expressive prior there. Ma et al. (2019) instead employed flow-based models as the latent prior, with applications on non-autoregressive text generation. Another line of research is inspired by variational autoencoder's (VAE's) problem that the simple Gaussian priors cannot accurately match the encoding posteriors and therefore generate poor samples, and Dai & Wipf (2019); Aneja et al. (2021) therefore proposed to use VAEs and energy-based models respectively to learn the latent distribution. Most recently, several works successfully developed latent DMs with promising results on various applications, ranging from image (Vahdat et al., 2021), point clouds (Zeng et al., 2022), to text (Li et al., 2022) generation. Among them, the most impressive success is Stable Diffusion models (Rombach et al., 2022), which show surprisingly realistic text-guided image generation results. Despite the considerable progress we have achieved, existing latent generative methods mainly work on latent space only filled with typical *scalars*, without any consideration for equivariance. By contrast, we study the novel and challenging task that latent

space also contains equivariant *tensors*.

Molecule Generation in 3D. Although extensive prior work has focused on generating molecules as 2D graphs (Jin et al., 2018; Liu et al., 2018; Shi et al., 2020), interest has recently increased in 3D generation. G-Schnet and G-SphereNet (Gebauer et al., 2019; Luo & Ji, 2021) employed autoregressive approaches to build molecules by sequential attachment of atoms or molecular fragments. Similar frameworks have also been applied to structure-based drug design (Li et al., 2021; Peng et al., 2022; Powers et al., 2022). However, this autoregressive approach requires careful formulation of a complex action space and action ordering. Other studies utilized atomic density grids, by which the entire molecule can be generated in "one step" by outputting a density over the voxelized 3D space (Masuda et al., 2020). However, these density grids lack the desirable equivariance property and require a separate fitting algorithm. In the past year, DMs have attracted attention for molecule generation in 3D (Hoogeboom et al., 2022; Wu et al., 2022), with successful application in downstream tasks like target drug generation (Lin et al., 2022), antibody design (Luo et al., 2022), and protein design (Anand & Achim, 2022; Trippe et al., 2022). However, existing models mainly still work on the original atomic space, while our method works on the fundamentally different and more expressive latent space.

3. Background

3.1. Problem Definition

In this paper, we consider generative modeling of molecular geometries from scratch. Let d be the dimension of node features, then each molecule is represented as point clouds $\mathcal{G} = \langle \mathbf{x}, \mathbf{h} \rangle$, where $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_N) \in \mathbb{R}^{N \times 3}$ is the atom coordinates matrix and $\mathbf{h} = (\mathbf{h}_1, \ldots, \mathbf{h}_N) \in \mathbb{R}^{N \times d}$ is the node feature matrix, such as atomic type and charges. We consider the following two generation tasks:
(I) Unconditional generation. With a collection of molecules $\mathcal{G}$, learn parameterized generative models $p_\theta(\mathcal{G})$ which can generate diverse and realistic molecules $\hat{\mathcal{G}}$ in 3D.
(II) Controllable generation. With molecules $\mathcal{G}$ labeled with certain properties s, learn conditional generation models $p_\theta(\mathcal{G}|s)$ which can conduct controllable molecule generation given desired property value s.

3.2. Equivariance

Equivariance is ubiquitous for geometric systems such as molecules, where vector features like atomic forces or dipoles should transform accordingly *w.r.t.* the coordinates (Thomas et al., 2018; Weiler et al., 2018; Fuchs et al., 2020; Batzner et al., 2021). Formally, a function $\mathcal{F}$ is defined as equivariant *w.r.t* the action of a group G if $\mathcal{F} \circ S_g(\mathbf{x}) = T_g \circ \mathcal{F}(\mathbf{x}), \forall g \in G$ where S_g, T_g are trans-

formations for a group element g (Serre et al., 1977). In this work, we consider the Special Euclidean group SE(3), *i.e.*, the group of rotation and translation in 3D space, where transformations T_g and S_g can be represented by a translation $\boldsymbol{t}$ and an orthogonal matrix rotation $\mathbf{R}$.

In molecules the features $\mathbf{h}$ are SE(3)-invariant while the coordinates will be affected[2] as $\mathbf{R}\mathbf{x} + \boldsymbol{t} = (\mathbf{R}\mathbf{x}_1 + \boldsymbol{t}, \ldots, \mathbf{R}\mathbf{x}_N + \boldsymbol{t})$. This requires our learned likelihood to be invariant to roto-translations. Such property has been shown important for improving the generalization capacity of 3D geometric modeling (Satorras et al., 2021a; Xu et al., 2022).

3.3. Diffusion Models for Non-geometric Domains

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) are latent variable models that model the data $\mathbf{x}_0$ as Markov chains $\mathbf{x}_T \cdots \mathbf{x}_0$, with intermediate variables sharing the same dimension. DMs can be described with two Markovian processes: a forward *diffusion* process $q(\mathbf{x}_{1:T} \mid \mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ and a reverse *denoising* process $p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$. The forward process gradually adds Gaussian noise to data $\mathbf{x}_t$:

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t}\mathbf{x}_{t-1}, \beta_t \boldsymbol{I}), \quad (1)$$

where the hyperparameter $\beta_{1:T}$ controls the amount of noise added at each timestep t. The $\beta_{1:T}$ are chosen such that samples $\mathbf{x}_T$ can approximately converge to standard Gaussians, *i.e.*, $q(\mathbf{x}_T) \approx \mathcal{N}(0, \boldsymbol{I})$. Typically, this forward process q is predefined without trainable parameters.

The generation process of DMs is defined as learning a parameterized reverse *denoising* process, which aims to incrementally denoise the noisy variables $\mathbf{x}_{T:1}$ to approximate clean data $\mathbf{x}_0$ in the target data distribution:

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \rho_t^2 \boldsymbol{I}), \quad (2)$$

where the initial distribution $p(\mathbf{x}_T)$ is defined as $\mathcal{N}(0, \boldsymbol{I})$. The means $\boldsymbol{\mu}_\theta$ typically are neural networks such as U-Nets for images or Transformers for text, and the variances ρ_t typically are also predefined.

As latent variable models, the forward process $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$ can be viewed as a fixed posterior, to which the reverse process $p_\theta(\mathbf{x}_{0:T})$ is trained to maximize the variational lower bound of the likelihood of the data $\mathcal{L}_{vlb} = \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \Big[\log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p_\theta(\mathbf{x}_T)} + \sum_{t=2}^{T} \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_t)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \Big]$. However, directly optimizing this objective is known to suffer serious training instability (Nichol & Dhariwal, 2021). Instead, Song & Ermon (2019); Ho et al. (2020) suggest a simple

---

[2]We follow the convention to use $\mathbf{R}\mathbf{x}$ to denote applying group actions $\mathbf{R}$ on $\mathbf{x}$, which formally is calculated as $\mathbf{x}\mathbf{R}^T$.

surrogate objective up to irrelevant constant terms:

$$\mathcal{L}_{DM} = \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I}), t} \left[w(t) || \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) ||^2 \right], \quad (3)$$

where $\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}$, with $\alpha_t = \sqrt{\prod_{s=1}^{t}(1-\beta_s)}$ and $\sigma_t = \sqrt{1 - \alpha_t^2}$ are parameters from the tractable diffusion distributions $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \alpha_t \mathbf{x}_0, \sigma_t^2 \boldsymbol{I})$. $\boldsymbol{\epsilon}_\theta$ comes from the widely adopted parametrization of the means $\mu_\theta(\mathbf{x}_t, t) := \frac{1}{\sqrt{1-\beta_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\alpha_t^2}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right)$. The reweighting terms are $w(t) = \frac{\beta_t^2}{2\rho_t^2(1-\beta_t)(1-\alpha_t^2)}$, while in practice simply setting it as 1 often promotes the sampling quality. Intuitively, the model $\boldsymbol{\epsilon}_\theta$ is trained to predict the noise vector $\boldsymbol{\epsilon}$ to denoise diffused samples $\mathbf{x}_t$ at every step t towards a cleaner one $\mathbf{x}_{t-1}$. After training, we can draw samples with $\boldsymbol{\epsilon}_\theta$ by the iterative ancestral sampling:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{1-\beta_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\alpha_t^2}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \rho_t \boldsymbol{\epsilon}, \quad (4)$$

with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. The sampling chain is initialized from Gaussian prior $\mathbf{x}_T \sim p(x_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \boldsymbol{I})$.

4. Method

In this section, we formally describe Geometric Latent Diffusion Models (GEOLDM). Our work is inspired by the recent success of stable (latent) diffusion models (Rombach et al., 2022), but learning latent representations for the geometric domain is however challenging (Winter et al., 2021). We address these challenges by learning a faithful point-structured latent space with both invariant and equivariant variables, and elaborate on the design details of geometric autoencoding and latent diffusion in Section 4.1 and Section 4.2 respectively. Finally, we briefly summarize the simple training and sampling scheme in Section 4.3, and further discuss extensions for conditioning mechanisms in Section 4.4. A high-level schematic is provided in Figure 1.

4.1. Geometric Autoencoding

We are interested in first compressing the geometries $\mathcal{G} = \langle \mathbf{x}, \mathbf{h} \rangle \in \mathbb{R}^{N \times (3+d)}$ (see Section 3.1 for details) into lower-dimensional latent space. We consider the classic autoencoder (AE) framework, where the encoder $\mathcal{E}_\phi$ encodes $\mathcal{G}$ into latent domain $\mathbf{z} = \mathcal{E}_\phi(\mathbf{x}, \mathbf{h})$ and the decoder $\mathcal{D}_\xi$ learns to decode $\mathbf{z}$ back to data domain $\tilde{\mathbf{x}}, \tilde{\mathbf{h}} = \mathcal{D}_\xi(\mathbf{z})$. The whole framework can be trained by minimizing the reconstruction objective $d(\mathcal{D}(\mathcal{E}(\mathcal{G})), \mathcal{G})$, e.g., L_p norms.

However, this classic autoencoding scheme is non-trivial in the geometric domain. Considering we follow SE(3) group in this paper (see Section 3.2), the typical parameterization of latent space as invariant scalar-valued features (Kingma & Welling, 2013) is very challenging:

Proposition 4.1. (*Winter et al., 2022*) *Learning autoencoding functions $\mathcal{E}$ and $\mathcal{D}$ to represent geometries $\mathcal{G}$ in scalar-valued (i.e., invariant) latent space **necessarily** requires an additional **equivariant** function ψ to store **suitable** group actions such that $\mathcal{D}(\psi(\mathcal{G}), \mathcal{E}(\mathcal{G})) = T_{\psi(\mathcal{G})} \circ \hat{\mathcal{D}}(\mathcal{E}(\mathcal{G})) = \mathcal{G}$.*

The idea of this proposition is that Geometric AE requires an additional function ψ to represent appropriate group actions for encoding, and align output and input positions for decoding, to solve the reconstruction task. We leave a more detailed explanation with examples in Appendix A. For euclidean groups SE(n), Winter et al. (2022) suggests implementing ψ as equivariant ortho-normal vectors in the unit n-dimensional sphere S^n.

In our method, instead of separately representing and applying the equivariance with ψ, we propose to also incorporate equivariance into $\mathcal{E}$ and $\mathcal{D}$ by constructing latent features as point-structured variables $\mathbf{z} = \langle \mathbf{z}_x, \mathbf{z}_h \rangle \in \mathbb{R}^{N \times (3+k)}$, which holds 3-d equivariant and k-d invariant latent features $\mathbf{z}_x$ and $\mathbf{z}_h$ for each node. This in practice can be implemented by parameterizing $\mathcal{E}$ and $\mathcal{D}$ with equivariant graph neural networks (EGNNs) (Satorras et al., 2021b), which extract both invariant and equivariant embeddings with the property:

$$\mathbf{R}\mathbf{z}_x + \boldsymbol{t}, \mathbf{z}_h = \mathcal{E}_\phi(\mathbf{R}\mathbf{x} + \boldsymbol{t}, \mathbf{h}); \mathbf{R}\mathbf{x} + \boldsymbol{t}, \mathbf{h} = \mathcal{D}_\xi(\mathbf{R}\mathbf{z}_x + \boldsymbol{t}, \mathbf{z}_h), \quad (5)$$

for all rotations $\mathbf{R}$ and translations $\boldsymbol{t}$. We provide parameterization details of EGNNs in Appendix C. The latent points $\mathbf{z}_x$ can perform the role of ψ required in Proposition 4.1, to align the orientation of outputs towards inputs. Furthermore, this point-wise latent space follows the inherent structure of geometries $\mathcal{G}$, thereby achieving good reconstructions.

Then the encoding and decoding processes can be formulated by $q_\phi(\mathbf{z}_x, \mathbf{z}_h|\mathbf{x}, \mathbf{h}) = \mathcal{N}(\mathcal{E}_\phi(\mathbf{x}, \mathbf{h}), \sigma_0 \boldsymbol{I})$ and $p_\xi(\mathbf{x}, \mathbf{h}|\mathbf{z}_x, \mathbf{z}_h) = \prod_{i=1}^{N} p_\xi(x_i, h_i|\mathbf{z}_x, \mathbf{z}_h)$ respectively. Following Xu et al. (2022); Hoogeboom et al. (2022) that linear subspaces with the center of gravity always being zero can induce translation-invariant distributions, we also define distributions of latent $\mathbf{z}_x$ and reconstructed $\mathbf{x}$ on the subspace that $\sum_i \mathbf{z}_{x,i}$ (or $\mathbf{x}_i$) = 0. The whole framework can be effectively optimized by:

$$\begin{aligned} \mathcal{L}_{AE} &= \mathcal{L}_{recon} + \mathcal{L}_{reg}, \\ \mathcal{L}_{recon} &= -\mathbb{E}_{q_\phi(\mathbf{z}_x, \mathbf{z}_h|\mathbf{x}, \mathbf{h})} p_\xi(\mathbf{x}, \mathbf{h}|\mathbf{z}_x, \mathbf{z}_h), \end{aligned} \quad (6)$$

which is a reconstruction loss combined with a regularization term. The reconstruction loss in practice is calculated as L_2 norm or cross-entropy for continuous or discrete features. For the $\mathcal{L}_{reg}$ terms we experimented with two variants: *KL-reg* (Rombach et al., 2022), a slight Kullback-Leibler penalty of q_ϕ towards standard Gaussians similar to variational AE; and *ES-reg*, an early-stop q_ϕ training strategy to avoid a scattered latent space. The regularization prevents

latent embeddings from arbitrarily high variance and is thus more suitable for learning the latent DMs (LDMs).

4.2. Geometric Latent Diffusion Models

With the equivariant autoencoding functions $\mathcal{E}_\phi$ and $\mathcal{D}_\xi$, now we can represent structures $\mathcal{G}$ using lower-dimensional latent variables $\mathbf{z}$ while still keeping geometric properties. Compared with the original atomic features which are high-dimensional with complicated data types and scales, the encoded latent space significantly benefits likelihood-based generative models since: (i) as described in Section 4.1, our proposed AEs can be viewed as *regularized autoencoders* (Ghosh et al., 2020), where the latent space is more compact and smoothed, thereby improving DM's training; (ii) latent codes also enjoy lower dimensionality and benefit the generative modeling complexity, since DMs typically operate in the full dimension of inputs.

Existing latent generative models for images (Vahdat et al., 2021; Esser et al., 2021) and texts (Li et al., 2022) usually rely on typical autoregressive or diffusion models to model the scalar-valued latent space. By contrast, a fundamental challenge for our method is that the latent space $\mathbf{z}$ contains not only scalars (*i.e,* invariant features) $\mathbf{z}_\mathrm{h}$ but also tensors (*i.e,* equivariant features) $\mathbf{z}_\mathrm{x}$. This requires the distribution of latent DMs to satisfy the critical invariance:

$$p_\theta(\mathbf{z}_\mathrm{x}, \mathbf{z}_\mathrm{h}) = p_\theta(\mathbf{R}\mathbf{z}_\mathrm{x}, \mathbf{z}_\mathrm{h}), \ \forall \ \mathbf{R}. \tag{7}$$

Xu et al. (2022) proved that this can be achieved if the initial distribution $p(\mathbf{z}_{\mathrm{x},T}, \mathbf{z}_{\mathrm{h},T})$ is invariant while the transitions $p_\theta(\mathbf{z}_{\mathrm{x},t-1}, \mathbf{z}_{\mathrm{h},t-1}|\mathbf{z}_{\mathrm{x},t}, \mathbf{z}_{\mathrm{h},t})$ are equivariant:

$$p_\theta(\mathbf{z}_{\mathrm{x},t-1}, \mathbf{z}_{\mathrm{h},t-1}|\mathbf{z}_{\mathrm{x},t}, \mathbf{z}_{\mathrm{h},t}) =$$
$$p_\theta(\mathbf{R}\mathbf{z}_{\mathrm{x},t-1}, \mathbf{z}_{\mathrm{h},t-1}|\mathbf{R}\mathbf{z}_{\mathrm{x},t}, \mathbf{z}_{\mathrm{h},t}), \ \forall \ \mathbf{R}. \tag{8}$$

Xu et al. (2022); Hoogeboom et al. (2022) further show that this can be realized by implementing the denoising dynamics ϵ_θ with equivariant networks such that:

$$\mathbf{R}\mathbf{z}_{\mathrm{x},t-1} + \boldsymbol{t}, \mathbf{z}_{\mathrm{h},t-1} = \epsilon_\theta(\mathbf{R}\mathbf{z}_{\mathrm{x},t} + \boldsymbol{t}, \mathbf{z}_{\mathrm{h},t}, t), \ \forall \ \mathbf{R} \text{ and } \boldsymbol{t}. \tag{9}$$

which in practice we parameterize as time-conditional EGNNs. More model details are also provided in Appendix C. Similar to the encoding posterior, in order to keep translation invariance, all the intermediate states $\mathbf{z}_{\mathrm{x},t}, \mathbf{z}_{\mathrm{h},t}$ are also required to lie on the subspace by $\sum_i \mathbf{z}_{\mathrm{x},t,i} = 0$ by moving the center of gravity. Analogous to Equation (3), now we can train the model by:

$$\mathcal{L}_{LDM} = \mathbb{E}_{\mathcal{E}(\mathcal{G}), \epsilon \sim \mathcal{N}(0,\boldsymbol{I}), t}\big[w(t)||\epsilon - \epsilon_\theta(\mathbf{z}_{\mathrm{x},t}, \mathbf{z}_{\mathrm{h},t}, t)||^2\big], \tag{10}$$

with $w(t)$ simply set as 1 for all steps t.

Theoretical analysis. The combined objective for the whole framework, *i.e.*, $\mathcal{L}_{AE} + \mathcal{L}_{LDM}$, appears similar to the

---

Algorithm 1 Training Algorithm of GEOLDM

1: **Input:** geometric data $\mathcal{G} = \langle \mathbf{x}, \mathbf{h} \rangle$
2: **Initial:** encoder network $\mathcal{E}_\phi$, decoder network $\mathcal{D}_\xi$, denoising network ϵ_θ
3: **First Stage: Autoencoder Training**
4: **while** ϕ, ξ have not converged **do**
5:     $\boldsymbol{\mu}_\mathrm{x}, \boldsymbol{\mu}_\mathrm{h} \leftarrow \mathcal{E}_\phi(\mathbf{x}, \mathbf{h})$ {Encoding}
6:     $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
7:     Subtract center of gravity from ϵ_x in $\epsilon = [\epsilon_\mathrm{x}, \epsilon_\mathrm{h}]$
8:     $\mathbf{z}_\mathrm{x}, \mathbf{z}_\mathrm{h} \leftarrow \epsilon \odot \sigma_0 + \boldsymbol{\mu}$ {Reparameterization}
9:     $\tilde{\mathbf{x}}, \tilde{\mathbf{h}} \leftarrow \mathcal{D}_\xi(\mathbf{z}_\mathrm{x}, \mathbf{z}_\mathrm{h})$ {Decoding}
10:    $\mathcal{L}_{AE} = \mathrm{reconstrcution}([\tilde{\mathbf{x}}, \tilde{\mathbf{h}}], [\mathbf{x}, \mathbf{h}]) + \mathcal{L}_{reg}$
11:    $\phi, \xi \leftarrow \mathrm{optimizer}(\mathcal{L}_{AE}; \phi, \xi)$
12: **end while**
13: **Second Stage: Latent Diffusion Models Training**
14: Fix encoder parameters ϕ
15: **while** θ have not converged **do**
16:    $\mathbf{z}_{\mathrm{x},0}, \mathbf{z}_{\mathrm{h},0} \sim q_\phi(\mathbf{z}_\mathrm{x}, \mathbf{z}_\mathrm{h}|\mathbf{x}, \mathbf{h})$ {As lines 5-8}
17:    $t \sim \mathbf{U}(0, T), \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
18:    Subtract center of gravity from ϵ_x in $\epsilon = [\epsilon_\mathrm{x}, \epsilon_\mathrm{h}]$
19:    $\mathbf{z}_{\mathrm{x},t}, \mathbf{z}_{\mathrm{h},t} = \alpha_t[\mathbf{z}_{\mathrm{x},0}, \mathbf{z}_{\mathrm{h},0}] + \sigma_t \epsilon$
20:    $\mathcal{L}_{LDM} = ||\epsilon - \epsilon_\theta(\mathbf{z}_{\mathrm{x},t}, \mathbf{z}_{\mathrm{h},t}, t)||^2$
21:    $\theta \leftarrow \mathrm{optimizer}(\mathcal{L}_{LDM}; \theta)$
22: **end while**
23: **return** $\mathcal{E}_\phi, \mathcal{D}_\xi, \epsilon_\theta$

---

standard VAE objective with an additional regularization. We make the formal justification that considering neglecting the minor $\mathcal{L}_{reg}$ term, $\mathcal{L} = \mathcal{L}_{recon} + \mathcal{L}_{LDM}$ is theoretically an SE(3)-invariant variational lower bound of log-likelihood:

Theorem 4.2. *(informal) Let* $\mathcal{L} := \mathcal{L}_{recon} + \mathcal{L}_{LDM}$. *With certain weights* $w(t)$, $\mathcal{L}$ *is an SE(3)-invariant variational lower bound to the log-likelihood, i.e., for any geometries* $\langle \mathbf{x}, \mathbf{h} \rangle$, *we have:*

$$\mathcal{L}(\mathbf{x}, \mathbf{h}) \geq -\mathbb{E}_{p_{data}}[\log p_{\theta,\xi}(\mathbf{x}, \mathbf{h})], \ and$$
$$\mathcal{L}(\mathbf{x}, \mathbf{h}) = \mathcal{L}(\mathbf{R}\mathbf{x} + \boldsymbol{t}, \mathbf{h}), \ \forall \ rotation \ \mathbf{R} \ and \ translation \ \boldsymbol{t},$$

where $p_{\theta,\xi}(\mathbf{x}, \mathbf{h}) = \mathbb{E}_{p_\theta(\mathbf{z}_\mathrm{x}, \mathbf{z}_\mathrm{h})} p_\xi(\mathbf{x}, \mathbf{h}|\mathbf{z}_\mathrm{x}, \mathbf{z}_\mathrm{h})$ *is the marginal distribution of* $\langle \mathbf{x}, \mathbf{h} \rangle$ *under* GEOLDM *model.*

Furthermore, for the induced marginal distribution $p_{\theta,\xi}(\mathbf{x}, \mathbf{h})$, we also hold the equivariance property that:

Proposition 4.3. *With decoders and latent DMs defined with equivariant distributions, the marginal* $p_{\theta,\xi}(\mathbf{x}, \mathbf{h}) = \mathbb{E}_{p_\theta(\mathbf{z}_\mathrm{x}, \mathbf{z}_\mathrm{h})} p_\xi(\mathbf{x}, \mathbf{h}|\mathbf{z}_\mathrm{x}, \mathbf{z}_\mathrm{h})$ *is an SE(3)-invariant distribution.*

These theoretical analysis suggest that GEOLDM is parameterized and optimized in an SE(3)-invariant fashion, which is a critical inductive bias for geometric generative models (Satorras et al., 2021a; Xu et al., 2022) and provides explanations as to why our framework can achieve better 3D geometries generation quality. We provide the full statements and proofs in Appendix B
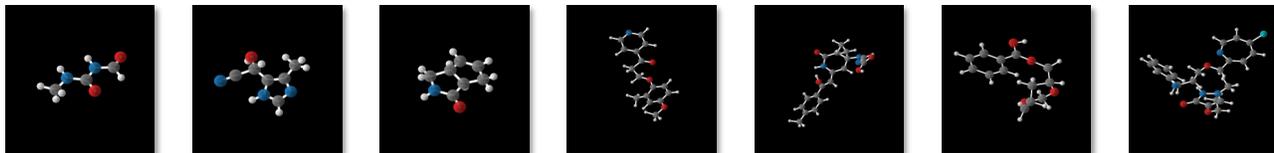
Figure 2. Molecules generated by GEOLDM trained on QM9 (left three) and DRUG (right four).

Algorithm 2 Sampling Algorithm of GEOLDM

1: **Input:** decoder network $\mathcal{D}_\xi$, denoising network ϵ_θ
2: $\mathbf{z}_{\mathrm{x},T}, \mathbf{z}_{\mathrm{h},T} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
3: **for** t in $T, T-1, \cdots, 1$ **do**
4:    $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ {Latent Denoising Loop}
5:    Subtract center of gravity from ϵ_{x} in $\epsilon = [\epsilon_{\mathrm{x}}, \epsilon_{\mathrm{h}}]$
6:    $\mathbf{z}_{t-1} = \frac{1}{\sqrt{1-\beta_t}}(\mathbf{z}_t - \frac{\beta_t}{\sqrt{1-\alpha_t^2}}\epsilon_\theta(\mathbf{z}_t, t)) + \rho_t \epsilon$
7: **end for**
8: $\mathbf{x}, \mathbf{h} \sim p_\xi(\mathbf{x}, \mathbf{h}|\mathbf{z}_{\mathrm{x},0}, \mathbf{z}_{\mathrm{h},0})$ {Decoding}
9: **return** $\mathbf{x}, \mathbf{h}$

4.3. Training and Sampling

With the proposed formulation and practical parameterization, we now present the training and sampling schemes for GEOLDM. While objectives for training Geometric AEs and LDMs are already defined in Equations (6) and (10), it is still unclear whether the two components should be trained one by one, or optimized simultaneously by backpropagation through reparameterizing (Kingma & Welling, 2013). Previous work about latent DMs for image generation (Sinha et al., 2021; Rombach et al., 2022) shows that the two-stage training strategy usually leads to better performance, and we notice similar phenomena in our experiments. This means we first train AE with regularization, and then train the latent DMs on the latent embeddings encoded by the pre-trained encoder. A formal description of the training process is provided in Algorithm 1.

With GEOLDM we can formally define a residual generative distribution $p_{\theta,\xi}(\mathbf{x}, \mathbf{h}, \mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}}) = p_\theta(\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}})p_\xi(\mathbf{x}, \mathbf{h}|\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}})$, where p_θ refers to the latent DM modeling the point-structured latent codes, and p_ξ denotes the decoder. We can generate molecular structures by first sampling equivariant latent embeddings from p_θ and then translating them back to the original geometric space with p_ξ. The pseudo-code of the sampling procedure is provided in Algorithm 2.

For the number of nodes N, in the above sections, we assume it to be predefined for each data point. In practice, we need to sample different numbers N for generating molecules of different sizes. We follow the common practice (Satorras et al., 2021a) to first count the distribution $p(N)$ of molecular sizes on the training set. Then for generation, we can first sample $N \sim p(N)$ and then generate latent variables and node features in size N.

4.4. Controllable Generation

Similar to other generative models (Kingma & Welling, 2013; Van Den Oord et al., 2016), DMs are also capable of controllable generation with given conditions s, by modeling conditional distributions $p(\mathbf{z}|s)$. This in DMs can be implemented with conditional denoising networks $\epsilon_\theta(\mathbf{z}, t, s)$, with the critical difference that it takes additional inputs s. In the molecular domain, desired conditions s typically are chemical properties, which are much lower-dimensional than the text prompts for image generations (Rombach et al., 2022; Ramesh et al., 2022). Therefore, instead of sophisticated cross-attention mechanisms used in text-guided image generation, we follow Hoogeboom et al. (2022) and simply parameterize the conditioning by concatenating s to node features. Besides, as a whole framework, we also adopt similar concatenation methods for the encoder and decoder, *i.e.*, $\mathcal{E}_\phi(\mathbf{x}, \mathbf{h}, s)$ and $\mathcal{D}_\xi(\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}}, s)$, to further shift the latent codes towards data distribution with desired properties s.

5. Experiments

In this section, we justify the advantages of GEOLDM with comprehensive experiments. We first introduce our experimental setup in Section 5.1. Then we report and analyze the evaluation results in Section 5.2 and Section 5.3, for unconditional and conditional generation respectively. We also provide further ablation studies in Appendix E to investigate the effect of several model designs. We leave more implementation details in Appendix D.

5.1. Experiment Setup

Evaluation Task. Following previous works on molecule generation in 3D (Gebauer et al., 2019; Luo & Ji, 2021; Satorras et al., 2021a; Hoogeboom et al., 2022; Wu et al., 2022), we evaluate GEOLDM by comparing with the state-of-the-art approaches on three comprehensive tasks. *Molecular Modeling and Generation* measures the model's capacity to learn the molecular data distribution and generate chemically valid and structurally diverse molecules. *Controllable Molecule Generation* concentrates on generating target molecules with desired chemical properties. For this task, we retrain the conditional version GEOLDM on molecular data with corresponding property labels.

Datasets. We first adopt *QM9* dataset (Ramakrishnan et al.,

Table 1. Results of atom stability, molecule stability, validity, and validity×uniqueness. A higher number indicates a better generation quality. Metrics are calculated with 10000 samples generated from each model. On QM9, we run the evaluation for 3 times and report the derivation. Note that, for DRUG dataset, molecule stability and uniqueness metric are omitted since they are nearly 0% and 100% respectively for all the methods. Compared with previous methods, the latent space with both invariant and equivariant variables enables GEOLDM to achieve up to 7% improvement for the validity of large molecule generation.

# Metrics	QM9				DRUG	
	Atom Sta (%)	Mol Sta (%)	Valid (%)	Valid & Unique (%)	Atom Sta (%)	Valid (%)
Data	99.0	95.2	97.7	97.7	86.5	99.9
ENF	85.0	4.9	40.2	39.4	-	-
G-Schnet	95.7	68.1	85.5	80.3	-	-
GDM	97.0	63.2	-	-	75.0	90.8
GDM-AUG	97.6	71.6	90.4	89.5	77.7	91.8
EDM	98.7	82.0	91.9	90.7	81.3	92.6
EDM-Bridge	98.8	84.6	92.0*	90.7	82.4	92.8*
GRAPHLDM	97.2	70.5	83.6	82.7	76.2	97.2
GRAPHLDM-AUG	97.9	78.7	90.5	89.5	79.6	98.0
GEOLDM	$\mathbf{98.9} \pm 0.1$	$\mathbf{89.4} \pm 0.5$	$\mathbf{93.8} \pm 0.4$	$\mathbf{92.7} \pm 0.5$	**84.4**	**99.3**

*Results obtained by our own experiments. Other results are borrowed from recent studies (Hoogeboom et al., 2022; Wu et al., 2022).

2014) for both unconditional and conditional molecule generation. QM9 is one of the most widely-used datasets for molecular machine learning research, which has also been adopted in previous 3D molecule generation studies (Gebauer et al., 2019; 2021). QM9 contains 3D structures together with several quantum properties for 130k small molecules, limited to 9 heavy atoms (29 atoms including hydrogens). Following (Anderson et al., 2019), we split the train, validation, and test partitions, with 100K, 18K, and 13K samples. For the molecule generation task, we also test GEOLDM on the *GEOM-DRUG* (Geometric Ensemble Of Molecules) dataset. The DRUG dataset consists of much larger organic compounds, with up to 181 atoms and 44.2 atoms on average, in 5 different atom types. It covers 37 million molecular conformations for around 450,000 molecules, labeled with energy and statistical weight. We follow the common practice (Hoogeboom et al., 2022) to select the 30 lowest energy conformations of each molecule for training.

5.2. Molecular Modeling and Generation

Evaluation Metrics. We measure model performances by evaluating the chemical feasibility of generated molecules, indicating whether the model can learn chemical rules from data. Given molecular geometries, we first predict bond types (single, double, triple, or none) by pair-wise atomic distances and atom types. Then we calculate the *atom stability* and *molecule stability* of the predicted molecular graph. The first metric captures the proportion of atoms that have the right valency, while the latter is the proportion of generated molecules for which all atoms are stable. In addition, We report *validity* and *uniqueness* metrics, which are the percentages of valid (measured by RDKIT) and unique

molecules among all the generated compounds.

Baselines. We compare GEOLDM to several competitive baseline models. *G-Schnet* (Gebauer et al., 2019) and Equivariant Normalizing Flows (*ENF*) (Satorras et al., 2021a) are previous equivariant generative models for molecules, based on autoregressive and flow-based models respectively. Equivariant Graph Diffusion Models (*EDM*) with its non-equivariant variant (*GDM*) (Hoogeboom et al., 2022) are recent progress on diffusion models for molecule generation. Most recently, Wu et al. (2022) proposed an improved version of EDM (*EDM-Bridge*), which further boosts the performance with well-designed informative prior bridges. To yield a fair comparison, all the baseline models use the same parameterization and training configurations as described in Section 5.1.

Results and Analysis. We generate $10,000$ samples from each method to calculate the above metrics, and the results are reported in Table 1. As shown in the table, GEOLDM outperforms competitive baseline methods on all metrics with an obvious margin. It is worth noticing that, for the DRUG dataset, even ground-truth molecules have 86.5% atom-level and nearly 0% molecule-level stability. This is because the DRUG molecules contain larger and more complex structures, creating errors during bond type prediction based on pair-wise atom types and distances. Furthermore, as DRUG contains many more molecules with diverse compositions, we also observe that *unique* metric is almost 100% for all methods. Therefore, we omit the *molecule stability* and *unique* metrics for the DRUG dataset. Overall, the superior performance demonstrates GEOLDM's higher capacity to model the molecular distribution and generate chemically realistic molecular geometries. We provide visualization of
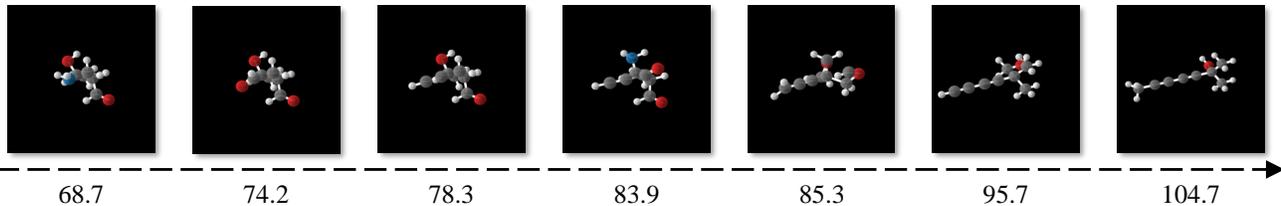
| | 68.7 | 74.2 | 78.3 | 83.9 | 85.3 | 95.7 | 104.7 |

Figure 3. Molecules generated by conditional GEOLDM. We conduct controllable generation with interpolation among different Polarizability α values with the same reparametrization noise ϵ. The given α values are provided at the bottom.

Table 2. Mean Absolute Error for molecular property prediction. A lower number indicates a better controllable generation result. Results are predicted by a pretrained EGNN classifier ω on molecular samples extracted from individual methods.

Property	α	$\Delta\varepsilon$	$\varepsilon_{\text{HOMO}}$	$\varepsilon_{\text{LUMO}}$	μ	C_v
Units	Bohr^3	meV	meV	meV	D	$\frac{\text{cal}}{\text{mol}}$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	2.76	655	356	584	1.111	1.101
GEOLDM	2.37	587	340	522	1.108	1.025

*The results of *QM9* and *Random* can be viewed as lower and upper bounds of MAE on all properties.

randomly generated molecules in Figure 2, and leave more visualizations in Appendix F.

Ablation Study. Furthermore, to verify the benefits of incorporating equivariant latent features, we conduct ablation studies with only invariant variables in the latent space, called Graph Latent Diffusion Models (GRAPHLDM). We run GRAPHLDM with the same configuration as our method, except that all modules (*i.e.*, encoder, decoder, and latent diffusion models) are instead equipped with typical non-equivariant graph networks. We also follow Hoogeboom et al. (2022) to test GDM-AUG and GRAPHLDM-AUG, where models are trained with data augmented by random rotations. Table 1 shows the empirical improvement of GEOLDM over these ablation settings, which verifies the effectiveness of our latent equivariance design.

5.3. Controllable Molecule Generation

Evaluation Metrics. In this task, we aim to conduct controllable molecule generation with the given desired properties. This can be useful in realistic settings of material and drug design where we are interested in discovering molecules with specific property preferences. We test our conditional version of GEOLDM on QM9 with 6 properties: polarizability α, orbital energies $\varepsilon_{\text{HOMO}}$, $\varepsilon_{\text{LUMO}}$ and their gap $\Delta\varepsilon$, Dipole moment μ, and heat capacity C_v. For evaluating the model's capacity to conduct property-conditioned generation, we follow Satorras et al. (2021a) to first split the

QM9 training set into two halves with $50K$ samples in each. Then we train a property prediction network ω on the first half, and train conditional models on the second half. Afterward, given a range of property values s, we conditionally draw samples from the generative models and then use ω to calculate their property values as $\hat{s}$. We report the *Mean Absolute Error (MAE)* between s and $\hat{s}$ to measure whether generated molecules are close to their conditioned property. We also test the MAE of directly running ω on the second half QM9, named *QM9* in Table 2, which measures the bias of ω. A smaller gap with *QM9* numbers indicates a better property-conditioning performance.

Baselines. We incorporate existing EDM as our baseline model. In addition, we follow Hoogeboom et al. (2022) to also list two baselines agnostic to ground-truth property s, named *Random* and N_{atoms}. *Random* means we simply do random shuffling of the property labels in the dataset and then evaluate ω on it. This operation removes any relation between molecule and property, which can be viewed as an upper bound of *MAE* metric. N_{atoms} predicts the molecular properties by only using the number of atoms in the molecule. The improvement over *Random* can verify the method is able to incorporate conditional property information into the generated molecules. And overcoming N_{atoms} further indicates the model can incorporate conditioning into molecular structures beyond the number of atoms.

Results and Analysis. We first provide a visualization of controlled molecule generation by GEOLDM in Figure 3, as qualitative assessments. We interpolate the conditioning property with different Polarizability values α while keeping the reparameterization noise ϵ fixed. Polarizability refers to the tendency of matter, when subjected to an electric field, to acquire an electric dipole moment in proportion to that applied field. Typically, less isometrically molecular geometries lead to larger α values. This is consistent with our observed phenomenon in Figure 3.

We report the numerical results in Table 2. As shown in the table, GEOLDM significantly outperforms baseline models, including the previous diffusion model running on atomic features (EDM), on all the property metrics. The results demonstrate that by modeling in the latent space, GEOLDM acquired a higher capacity to incorporate given property information into the generation process.

6. Conclusion and Future Work

We presented GEOLDM, a novel latent diffusion model for molecular geometry generation. While current models operate directly on high-dimensional, multi-modal atom features, GEOLDM overcomes their limitations by learning diffusion models over a continuous, lower-dimensional latent space. By building point-structured latent codes with both invariant scalars and equivariant tensors, GEOLDM is able to effectively learn latent representations while maintaining roto-translational equivariance. Experimental results demonstrate its significantly better capacity for modeling chemically realistic molecules. For future work, as a general and principled framework, GEOLDM can be extended to various 3D geometric generation applications, *e.g.*, apply GEOLDM in more realistic drug discovery scenarios with given protein targets, or scale up GEOLDM for more challenging 3D geometries such as peptides and proteins.

References

Anand, N. and Achim, T. Protein structure and sequence generation with equivariant denoising diffusion probabilistic models. *arXiv preprint arXiv:2205.15019*, 2022. 1, 3

Anderson, B., Hy, T. S., and Kondor, R. Cormorant: Covariant molecular neural networks. *Advances in neural information processing systems*, 32, 2019. 7

Aneja, J., Schwing, A., Kautz, J., and Vahdat, A. A contrastive learning approach for training variational autoencoder priors. *Advances in neural information processing systems*, 34:480–493, 2021. 2

Batzner, S., Smidt, T. E., Sun, L., Mailoa, J. P., Kornbluth, M., Molinari, N., and Kozinsky, B. Se (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *arXiv preprint arXiv:2101.03164*, 2021. 3

Dai, B. and Wipf, D. Diagnosing and enhancing VAE models. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1e0X3C9tQ. 2

Esser, P., Rombach, R., and Ommer, B. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021. 5

Fuchs, F., Worrall, D., Fischer, V., and Welling, M. Se(3)-transformers: 3d roto-translation equivariant attention networks. *NeurIPS*, 2020. 3

Gebauer, N., Gastegger, M., and Schütt, K. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. *Advances in neural information processing systems*, 32, 2019. 1, 3, 6, 7

Gebauer, N. W., Gastegger, M., Hessmann, S. S., Müller, K.-R., and Schütt, K. T. Inverse design of 3d molecular structures with conditional generative neural networks. *arXiv preprint arXiv:2109.04824*, 2021. 7

Ghosh, P., Sajjadi, M. S. M., Vergari, A., Black, M., and Scholkopf, B. From variational to deterministic autoencoders. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1g7tpEYDS. 5

Graves, J., Byerly, J., Priego, E., Makkapati, N., Parish, S. V., Medellin, B., and Berrondo, M. A review of deep learning methods for antibodies. *Antibodies*, 9(2):12, 2020. 1

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020. 1, 3

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, pp. 8867–8887. PMLR, 2022. 1, 3, 4, 5, 6, 7, 8, 16, 17

Jin, W., Barzilay, R., and Jaakkola, T. Junction tree variational autoencoder for molecular graph generation. *arXiv preprint arXiv:1802.04364*, 2018. 3

Jing, B., Eismann, S., Suriana, P., Townshend, R. J. L., and Dror, R. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2021. 1

Jo, J., Lee, S., and Hwang, S. J. Score-based generative modeling of graphs via the system of stochastic differential equations. *arXiv preprint arXiv:2202.02514*, 2022. 18

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *3nd International Conference on Learning Representations*, 2014. 17

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations*, 2013. 4, 6

Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021. 1

Landrum, G. Rdkit: open-source cheminformatics http://www. rdkit. org. 2016. 17

Li, X. L., Thickstun, J., Gulrajani, I., Liang, P., and Hashimoto, T. Diffusion-LM improves controllable text generation. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=3s9IrEsjLyk. 1, 2, 5

Li, Y., Pei, J., and Lai, L. Structure-based de novo drug design using 3d deep generative models. *Chemical science*, 12(41):13664–13675, 2021. 3

Lin, H., Huang, Y., Liu, M., Li, X., Ji, S., and Li, S. Z. Diffbp: Generative diffusion of 3d molecules for target protein binding. *arXiv preprint arXiv:2211.11214*, 2022. 2, 3

Liu, Q., Allamanis, M., Brockschmidt, M., and Gaunt, A. Constrained graph variational autoencoders for molecule design. In *Advances in neural information processing systems*, 2018. 3

Luo, S., Su, Y., Peng, X., Wang, S., Peng, J., and Ma, J. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=jSorGn2Tjg. 2, 3

Luo, Y. and Ji, S. An autoregressive flow model for 3d molecular geometry generation from scratch. In *International Conference on Learning Representations*, 2021. 1, 3, 6

Ma, X., Zhou, C., Li, X., Neubig, G., and Hovy, E. FlowSeq: Non-autoregressive conditional sequence generation with generative flow. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4282–4292, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1437. URL https://aclanthology.org/D19-1437. 2

Masuda, T., Ragoza, M., and Koes, D. R. Generating 3d molecular structures conditional on a receptor binding site with deep generative models. *arXiv preprint arXiv:2010.14442*, 2020. 3

Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.-Y., and Ermon, S. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=aBsCjcPu_tE. 1

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021. 3

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 17

Peng, X., Luo, S., Guan, J., Xie, Q., Peng, J., and Ma, J. Pocket2mol: Efficient molecular sampling based on 3d protein pockets. In *International Conference on Machine Learning*, 2022. 3

Pereira, J. C., Caffarena, E. R., and Dos Santos, C. N. Boosting docking-based virtual screening with deep learning. *Journal of chemical information and modeling*, 56(12): 2495–2506, 2016. 1

Powers, A. S., Yu, H. H., Suriana, P., and Dror, R. O. Fragment-based ligand generation guided by geometric deep learning on protein-ligand structure. *bioRxiv*, 2022. doi: 10.1101/2022.03.17.484653. URL https://www.biorxiv.org/content/early/2022/03/21/2022.03.17.484653. 3

Ramakrishnan, R., Dral, P. O., Rupp, M., and Von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014. 6

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 6

Razavi, A., Van den Oord, A., and Vinyals, O. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019. 1, 2

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022. 2, 4, 6, 18

Satorras, V. G., Hoogeboom, E., Fuchs, F. B., Posner, I., and Welling, M. E (n) equivariant normalizing flows for molecule generation in 3d. *arXiv preprint arXiv:2105.09016*, 2021a. 1, 3, 5, 6, 7, 8, 17

Satorras, V. G., Hoogeboom, E., and Welling, M. E(n) equivariant graph neural networks. In *International conference on machine learning*, pp. 9323–9332. PMLR, 2021b. 4, 16, 17

Schütt, K., Kindermans, P.-J., Sauceda Felix, H. E., Chmiela, S., Tkatchenko, A., and Müller, K.-R. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in Neural Information Processing Systems*, pp. 991–1001. Curran Associates, Inc., 2017. 1

Serre, J.-P. et al. *Linear representations of finite groups*, volume 42. Springer, 1977. 3

Shi, C., Xu, M., Zhu, Z., Zhang, W., Zhang, M., and Tang, J. Graphaf: a flow-based autoregressive model for molecular graph generation. *arXiv preprint arXiv:2001.09382*, 2020. 3

Sinha, A., Song, J., Meng, C., and Ermon, S. D2c: Diffusion-decoding models for few-shot conditional generation. *Advances in Neural Information Processing Systems*, 34:12533–12548, 2021. 6

Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. *arXiv preprint arXiv:1503.03585*, 2015. 3

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pp. 11918–11930, 2019. 3

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 1

Thomas, N., Smidt, T., Kearnes, S. M., Yang, L., Li, L., Kohlhoff, K., and Riley, P. Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds. *ArXiv*, 2018. 3

Townshend, R. J. L., Vögele, M., Suriana, P. A., Derry, A., Powers, A., Laloudakis, Y., Balachandar, S., Jing, B.,

Anderson, B. M., Eismann, S., Kondor, R., Altman, R., and Dror, R. O. ATOM3d: Tasks on molecules in three dimensions. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. URL https://openreview.net/forum?id=FkDZLpK1Ml2. 1

Trippe, B. L., Yim, J., Tischer, D., Broderick, T., Baker, D., Barzilay, R., and Jaakkola, T. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. *arXiv preprint arXiv:2206.04119*, 2022. 3

Vahdat, A., Kreis, K., and Kautz, J. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021. 2, 5

Van Den Oord, A., Kalchbrenner, N., and Kavukcuoglu, K. Pixel recurrent neural networks. In *International conference on machine learning*, pp. 1747–1756. PMLR, 2016. 6

Weiler, M., Geiger, M., Welling, M., Boomsma, W., and Cohen, T. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. In *NeurIPS*, 2018. 3

Winter, R., Noé, F., and Clevert, D.-A. Auto-encoding molecular conformations. *arXiv preprint arXiv:2101.01618*, 2021. 4

Winter, R., Bertolini, M., Le, T., Noe, F., and Clevert, D.-A. Unsupervised learning of group invariant and equivariant representations. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=47lpv23LDPr. 4, 13

Wu, L., Gong, C., Liu, X., Ye, M., and qiang liu. Diffusion-based molecule generation with informative prior bridges. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=TJUNtiZiTKE. 1, 3, 6, 7, 17

Xu, M., Yu, L., Song, Y., Shi, C., Ermon, S., and Tang, J. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022. 3, 4, 5, 16

Yu, J., Li, X., Koh, J. Y., Zhang, H., Pang, R., Qin, J., Ku, A., Xu, Y., Baldridge, J., and Wu, Y. Vector-quantized image modeling with improved VQGAN. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=pfNyExj7z2. 2

Zang, C. and Wang, F. Moflow: an invertible flow model for generating molecular graphs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 617–626, 2020. 18

Zeng, X., Vahdat, A., Williams, F., Gojcic, Z., Litany, O., Fidler, S., and Kreis, K. LION: Latent point diffusion models for 3d shape generation. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=tHK5ntjp-5K. 2, 18

A. Explanation of Proposition 4.1

We first explain the intuition behind the theoretical justification of Proposition 4.1 by an example here. Considering given a input geometry $\mathcal{G} = \langle \mathbf{h}, \mathbf{x} \rangle$, the encoder $\mathcal{E}$ and decoder $\mathcal{D}$, such that $\mathcal{G} = \mathcal{D}(\mathcal{E}(\mathcal{G}))$. Then we can transform $\mathcal{G}$ by an action g from SE(3)-group to $\hat{\mathcal{G}} = T_g\mathcal{G} = \langle \mathbf{h}, \mathbf{R}\mathbf{x} + \boldsymbol{t} \rangle$ and input it into the autoencoders. Since the encoding function is invariant, we have $\mathcal{E}(\mathcal{G}) = \mathcal{E}(\hat{\mathcal{G}})$, and thus the reconstructed geometry however will still be $\mathcal{G} = \mathcal{D}(\mathcal{E}(\hat{\mathcal{G}}))$ instead of $\hat{\mathcal{G}}$. This is problematic because we couldn't calculate the reconstruction error based on $\mathcal{G}$ and $\hat{\mathcal{G}}$, and a natural solution is that we need an additional function ψ to extract the group action g. Then after decoding, we can apply the group action on generated $\mathcal{G}$ to recover $\hat{\mathcal{G}}$, thereby solving the problem

Formally, the explanation is that all elements can be expressed in terms of coordinates with respect to a given basis. So we should consider a canonical basis for all orbits, and learn the equivariant function ψ to indicate to which orbit elements are decoded as "canonical". For detailed theoretical analysis, we refer readers to Winter et al. (2022).

B. Formal Statements and Proofs

B.1. Relationship to SE(3)-invariant Likelihood: Theorem 4.2

First, recall the informal theorem we provide in Section 4.2, which builds the connection between GEOLDM's objective and SE(3)-invariant maximum likelihood:

Theorem 4.2. *(informal) Let $\mathcal{L} := \mathcal{L}_{recon} + \mathcal{L}_{LDM}$. With certain weights $w(t)$, $\mathcal{L}$ is an SE(3)-invariant variational lower bound to the log-likelihood, i.e., for any geometries $\langle \mathbf{x}, \mathbf{h} \rangle$, we have:*

$$\mathcal{L}(\mathbf{x}, \mathbf{h}) \geq -\mathbb{E}_{p_{data}}[\log p_{\theta,\xi}(\mathbf{x}, \mathbf{h})], \text{ and} \tag{11}$$

$$\mathcal{L}(\mathbf{x}, \mathbf{h}) = \mathcal{L}(\mathbf{R}\mathbf{x} + \boldsymbol{t}, \mathbf{h}), \; \forall \text{ rotation } \mathbf{R} \text{ and translation } \boldsymbol{t}, \tag{12}$$

where $p_{\theta,\xi}(\mathbf{z}_\mathbf{x}, \mathbf{z}_\mathbf{h}) = \mathbb{E}_{p_\theta(\mathbf{z}_\mathbf{x}, \mathbf{z}_\mathbf{h})} p_\xi(\mathbf{x}, \mathbf{h}|\mathbf{z}_\mathbf{x}, \mathbf{z}_\mathbf{h})$ is the marginal distribution of $\langle \mathbf{x}, \mathbf{h} \rangle$ under GEOLDM *model.*

Before providing the proof, we first present a formal version of the theorem:

Theorem B.1. *(formal) For predefined valid $\{\beta_i\}_{i=0}^T$, $\{\alpha_i\}_{i=0}^T$, and $\{\rho_i\}_{i=0}^T$, let $w(t)$ satisfy:*

$$w(t) = \frac{\beta_t^2}{2\rho_t^2(1 - \beta_t)(1 - \alpha_t^2)}, \; \forall t \in [1, \cdots, T], \quad and \quad w(0) = -1. \tag{13}$$

Let $\mathcal{L}(\mathbf{x}, \mathbf{h}; \theta, \phi, \xi) := \mathcal{L}_{recon}(\mathbf{x}, \mathbf{h}; \phi, \xi) + \mathcal{L}_{LDM}(\mathbf{z}_\mathbf{x}, \mathbf{z}_\mathbf{h}; \theta)$. Then given the geometries $\langle \mathbf{x}, \mathbf{h} \rangle \in \mathbb{R}^{N \times (3+d)}$, we have:

$$\mathcal{L}(\mathbf{x}, \mathbf{h}) \geq -\mathbb{E}_{p_{data}}[\log p_{\theta,\xi}(\mathbf{x}, \mathbf{h})], \text{ and} \tag{14}$$

$$\mathcal{L}(\mathbf{x}, \mathbf{h}) = \mathcal{L}(\mathbf{R}\mathbf{x} + \boldsymbol{t}, \mathbf{h}), \; \forall \text{ rotation } \mathbf{R} \text{ and translation } \boldsymbol{t}, \tag{15}$$

$p_{\theta,\xi}(\mathbf{z}_\mathbf{x}, \mathbf{z}_\mathbf{h}) = \mathbb{E}_{p_\theta(\mathbf{z}_\mathbf{x}, \mathbf{z}_\mathbf{h})} p_\xi(\mathbf{x}, \mathbf{h}|\mathbf{z}_\mathbf{x}, \mathbf{z}_\mathbf{h})$ *is the marginal distribution of $\langle \mathbf{x}, \mathbf{h} \rangle$ under* GEOLDM *model.*

As shown in the theorem, the conclusion is composed of two statements, *i.e.*, Equation (14) and Equation (15). The first equation states that $\mathcal{L}$ is a variational lower bound of the log-likelihood, and the second equation shows that the objective $\mathcal{L}$ is further SE(3)-invariant, *i.e.*, invariant to any rotational and translational transformations. Here, we provide the full proofs of the two statements separately. We first present the proof for Equation (14):

Proof of Theorem 4.2 (Equation (14)*).* For analyzing the variational lower bound in Equation (14), we don't need to consider the different geometric properties of $\mathbf{x}$ and $\mathbf{h}$. Therefore, in this part, we use $\mathcal{G}$ to denote $\langle \mathbf{z}_\mathbf{x}, \mathbf{z}_\mathbf{h} \rangle$, and $\mathbf{z}_\mathcal{G}$ to denote $\langle \mathbf{z}_\mathbf{x}, \mathbf{z}_\mathbf{h} \rangle$. Besides, we interchangeably use $\mathbf{z}_\mathcal{G}$ and $\mathbf{z}_\mathcal{G}^{(0)}$ to denote the "clean" latent variables at timestep 0, and use $\mathbf{z}_\mathcal{G}^{(T)}$ to denote the "noisy" latent variables at timestep T. Then we have that:

$$\begin{aligned}
\mathbb{E}_{p_{\text{data}}(\mathcal{G})}[\log p_{\theta,\xi}(\mathcal{G})] &= \mathbb{E}_{p_{\text{data}}(\mathcal{G})}\left[\log \int_{\mathbf{z}_\mathcal{G}} p_\xi(\mathbf{x}|\mathbf{z}_\mathcal{G}) p_\theta(\mathbf{z}_\mathcal{G})\right] \\
&\geq \mathbb{E}_{p_{\text{data}}(\mathcal{G}), q_\phi(\mathbf{z}_\mathcal{G}|\mathcal{G})}[\log p_\xi(\mathcal{G}|\mathbf{z}_\mathcal{G}) + \log p_\theta(\mathbf{z}_\mathcal{G}) - \log q_\phi(\mathbf{z}_\mathcal{G}|\mathcal{G})] \quad \text{Jensen's inequality} \\
&= \mathbb{E}_{p_{\text{data}}(\mathcal{G}), q_\phi(\mathbf{z}_\mathcal{G}|\mathcal{G})}[\log p_\xi(\mathcal{G}|\mathbf{z}_\mathcal{G})] - D_{\text{KL}}(q_\phi(\mathbf{z}_\mathcal{G}|\mathcal{G})\|p_\theta(\mathbf{z}_\mathcal{G})). \quad \text{KL divergence}
\end{aligned} \tag{16}$$

Compared with the objective $\mathcal{L}$ for GEOLDM that:

$$
\begin{aligned}
\mathcal{L}(\mathcal{G}; \theta, \phi, \xi) &:= \mathcal{L}_{recon}(\mathcal{G}; \phi, \xi) + \mathcal{L}_{LDM}(\mathbf{z}_{\mathcal{G}}; \theta) \\
&= \mathbb{E}_{p_{\text{data}}(\mathcal{G}), q_\phi(\mathbf{z}_{\mathcal{G}}|\mathcal{G})}[\log p_\xi(\mathcal{G}|\mathbf{z}_{\mathcal{G}})] + \mathcal{L}_{LDM}(\mathbf{z}_{\mathcal{G}}; \theta),
\end{aligned}
\tag{17}
$$

it is clear that we can complete the proof if we have:

$$
\begin{aligned}
\mathcal{L}_{LDM}(\mathbf{z}_{\mathcal{G}}; \theta) &\geq D_{\mathrm{KL}}(q_\phi(\mathbf{z}_{\mathcal{G}}|\mathcal{G})\|p_\theta(\mathbf{z}_{\mathcal{G}})) \\
&= -H(q_\phi(\mathbf{z}_{\mathcal{G}}|\mathcal{G})) - \mathbb{E}_{q_\phi(\mathbf{z}_{\mathcal{G}}|\mathcal{G})}[\log p_\theta(\mathbf{z}_{\mathcal{G}})]
\end{aligned}
\tag{18}
$$

or since the Shannon entropy term $H(q_\phi(\mathbf{z}_{\mathcal{G}}|\mathcal{G}))$ is never negative, we can equivalently prove:

$$
\mathcal{L}_{LDM}(\mathbf{z}_{\mathcal{G}}; \theta) \geq -\mathbb{E}_{q_\phi(\mathbf{z}_{\mathcal{G}}|\mathcal{G})}[\log p_\theta(\mathbf{z}_{\mathcal{G}})]
\tag{19}
$$

Now we prove the inequality by analyzing the right side of the inequality. We first apply variational inference with an inference model $q(\mathbf{z}_{\mathcal{G}}^{(1:T)}|\mathbf{z}_{\mathcal{G}}^{(0)})$. Note that, now we change the notation of "clean" latent variable from $\mathbf{z}_{\mathcal{G}}$ to $\mathbf{z}_{\mathcal{G}}^{(0)}$, to highlight the timestep information of the latent diffusion model:

$$
\begin{aligned}
&\mathbb{E}_{q_\phi(\mathbf{z}_{\mathcal{G}}^{(0)}|\mathcal{G})}[\log p_\theta(\mathbf{z}_{\mathcal{G}}^{(0)})] \\
&= \mathbb{E}_{q_\phi(\mathbf{z}_{\mathcal{G}}^{(0)}|\mathcal{G})}[\log \int_{\mathbf{z}_{\mathcal{G}}^{(1:T)}} \left(p_\theta(\mathbf{z}_{\mathcal{G}}^{(T)}) \prod_{t=1}^{T} p_\theta(\mathbf{z}_{\mathcal{G}}^{(t-1)}|\mathbf{z}_{\mathcal{G}}^{(t)})\right)] \\
&\geq \mathbb{E}_{\mathbf{z}_{\mathcal{G}}^{(0:T)}}[\log p_\theta(\mathbf{z}_{\mathcal{G}}^{(T)}) + \sum_{t=1}^{T} \log p_\theta(\mathbf{z}_{\mathcal{G}}^{(t-1)}|\mathbf{z}_{\mathcal{G}}^{(t)}) - \log q(\mathbf{z}_{\mathcal{G}}^{(1:T)}|\mathbf{z}_{\mathcal{G}}^{(0)})] \\
&\geq \mathbb{E}_{\mathbf{z}_{\mathcal{G}}^{(0:T)}}\Big[\log p_\theta(\mathbf{z}_{\mathcal{G}}^{(T)}) - \log q(\mathbf{z}_{\mathcal{G}}^{(T)}|\mathbf{z}_{\mathcal{G}}^{(0)}) - \sum_{t=2}^{T} \underbrace{D_{\mathrm{KL}}(q(\mathbf{z}_{\mathcal{G}}^{(t-1)}|\mathbf{z}_{\mathcal{G}}^{(t)}, \mathbf{z}_{\mathcal{G}}^{(0)})\|p_\theta(\mathbf{z}_{\mathcal{G}}^{(t-1)}|\mathbf{z}_{\mathcal{G}}^{(t)}))}_{\mathcal{L}_{LDM}^{(t-1)}} + \log p_\theta(\mathbf{z}_{\mathcal{G}}^{(0)}|\mathbf{z}_{\mathcal{G}}^{(1)})\Big],
\end{aligned}
\tag{20}
$$

where we factorize $\log p_\theta$ into a sequence of KL divergences between $q(\mathbf{z}_{\mathcal{G}}^{(t-1)}|\mathbf{z}_{\mathcal{G}}^{(t)}, \mathbf{z}_{\mathcal{G}}^{(0)})$ and $p_\theta(\mathbf{z}_{\mathcal{G}}^{(t-1)}|\mathbf{z}_{\mathcal{G}}^{(t)})$. Now, for $t \geq 2$, let us consider transitions q and p_θ with the form in Equations (1) and (2) respectively, which are both Gaussian distributions with fixed variances. Then we can just set the standard deviation of $p_\theta(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})$ to be the same as that of $q(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}, \mathbf{x}^{(0)}))$. With this parameterization, the KL divergence for $\mathcal{L}_{LDM}^{(t-1)}$ is between two Gaussians with the same standard deviations and thus can be simply calculated as a weighted Euclidean distance between the means. Using the derivation results from in Section 3.3 that $\mu_\theta(\mathbf{x}_t, t) := \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\alpha_t^2}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right)$, we have that:

$$
\mathcal{L}_{LDM}^{(t-1)} = \mathbb{E}_{\mathbf{z}_0, \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})}\left[\frac{\beta_t^2}{2\rho_t^2(1-\beta_t)(1-\alpha_t^2)}\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{z}_{\mathcal{G}}^{(t)}, t)\|_2^2\right]
\tag{21}
$$

which gives us the weights of $w(t)$ for $t = 1, \cdots, T$. For $p_\theta(\mathbf{z}_{\mathcal{G}}^{(0)}|\mathbf{z}_{\mathcal{G}}^{(1)})$, we can directly analyze it in the Gaussian form with mean

$$
\mu_\theta(\mathbf{z}_{\mathcal{G}}^{(1)}, 1) = \frac{\mathbf{z}_{\mathcal{G}}^{(1)} - \sigma_1 \boldsymbol{\epsilon}_\theta(\mathbf{z}_{\mathcal{G}}^{(1)}, 1)}{\alpha_1}.
$$

And with

$$
\mathbf{z}_{\mathcal{G}}^{(0)} = \frac{\mathbf{z}_{\mathcal{G}}^{(1)} - \sigma_0 \boldsymbol{\epsilon}}{\alpha_1}
$$

we have that:

$$
\log p_\theta(\mathbf{z}_{\mathcal{G}}^{(0)}|\mathbf{z}_{\mathcal{G}}^{(1)}) = -\log Z^{-1} + \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{z}_{\mathcal{G}}^{(1)}, 1)\|_2^2 = -\mathcal{L}_{LDM}^{(0)}
\tag{22}
$$

with the normalization constant Z. This distribution gives us the weight of $w(0)$. Besides, we have:

$$
\mathbb{E}_{\mathbf{z}_{\mathcal{G}}^{(0:T)}}[\log p_\theta(\mathbf{z}_{\mathcal{G}}^{(T)}) - q(\mathbf{z}_{\mathcal{G}}^{(T)}|\mathbf{z}_{\mathcal{G}}^{(0)})] = 0
\tag{23}
$$

since $\mathbf{z}_{\mathcal{G}}^{(T)} \sim \mathcal{N}(0, \boldsymbol{I})$ for both p_θ and q. Therefore, without the constants, we have that:

$$\mathbb{E}_{q_\phi(\mathbf{z}_{\mathcal{G}}^{(0)}|\mathcal{G})}[\log p_\theta(\mathbf{z}_{\mathcal{G}}^{(0)})] = \sum_{t=2}^{T} \underbrace{D_{\mathrm{KL}}(q(\mathbf{z}_{\mathcal{G}}^{(t-1)}|\mathbf{z}_{\mathcal{G}}^{(t)}, \mathbf{z}_{\mathcal{G}}^{(0)})\|p_\theta(\mathbf{z}_{\mathcal{G}}^{(t-1)}|\mathbf{z}_{\mathcal{G}}^{(t)}))}_{\mathcal{L}_{LDM}^{(t-1)}} - \log p_\theta(\mathbf{z}_{\mathcal{G}}^{(0)}|\mathbf{z}_{\mathcal{G}}^{(1)})$$

$$\geq -\sum_{t=2}^{T} \mathcal{L}_{LDM}^{(t-1)} - \mathcal{L}_{LDM}^{(0)} = -\mathcal{L}_{LDM}$$

(24)

which completes our proof. $\qquad \square$

Proof of Theorem 4.2 (Equation (15)). Here we show that our derived lower bound is an SE(3)-invariant lower bound. Recall the objective function:

$$\mathcal{L}(\mathbf{x}, \mathbf{h}; \theta, \phi, \xi) := \underbrace{\mathbb{E}_{p_{\mathrm{data}}(\mathcal{G}), q_\phi(\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}}|\mathbf{x}, \mathbf{h})}\big[\log p_\xi(\mathbf{x}, \mathbf{h}|\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}})\big]}_{\mathcal{L}_{recon}(\mathbf{x}, \mathbf{h}; \phi, \xi)} +$$

$$\underbrace{\sum_{t=2}^{T} D_{\mathrm{KL}}(q(\mathbf{z}_{\mathcal{G}}^{(t-1)}|\mathbf{z}_{\mathcal{G}}^{(t)}, \mathbf{z}_{\mathcal{G}}^{(0)})\|p_\theta(\mathbf{z}_{\mathcal{G}}^{(t-1)}|\mathbf{z}_{\mathcal{G}}^{(t)})) - \log p_\theta(\mathbf{z}_{\mathcal{G}}^{(0)}|\mathbf{z}_{\mathcal{G}}^{(1)})}_{\mathcal{L}_{LDM}(\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}}; \theta)}.$$

(25)

Note that, we have $q_\phi(\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}}|\mathcal{G})$ and $\log p_\xi(\mathbf{x}, \mathbf{h}|\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}})$ are equivariant distributions, *i.e.*, $q_\phi(\mathbf{R}\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}}|\mathbf{R}\mathbf{x}, \mathbf{h})$ and $\log p_\xi(\mathbf{R}\mathbf{x}, \mathbf{h}|\mathbf{R}\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}})$ for all orthogonal $\mathbf{R}$. Then for $\mathcal{L}_{recon}(\mathbf{x}, \mathbf{h})$, we have:

$$\begin{aligned}
\mathcal{L}_{recon}(\mathbf{R}\mathbf{x}, \mathbf{h}) &= \mathbb{E}_{p_{\mathrm{data}}(\mathcal{G}), q_\phi(\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}}|\mathbf{R}\mathbf{x}, \mathbf{h})}\big[\log p_\xi(\mathbf{R}\mathbf{x}, \mathbf{h}|\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}})\big] \\
&= \int_{\mathcal{G}} q_\phi(\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}}|\mathbf{R}\mathbf{x}, \mathbf{h}) \log p_\xi(\mathbf{R}\mathbf{x}, \mathbf{h}|\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}}) \\
&= \int_{\mathcal{G}} q_\phi(\mathbf{R}\mathbf{R}^{-1}\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}}|\mathbf{R}\mathbf{x}, \mathbf{h}) \log p_\xi(\mathbf{R}\mathbf{x}, \mathbf{h}|\mathbf{R}\mathbf{R}^{-1}\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}}) &&\text{Multiply by } \mathbf{R}\mathbf{R}^{-1} = \mathbf{I} \\
&= \int_{\mathcal{G}} q_\phi(\mathbf{R}^{-1}\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}}|\mathbf{x}, \mathbf{h}) \log p_\xi(\mathbf{x}, \mathbf{h}|\mathbf{R}^{-1}\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}}) &&\text{Equivariance \& Invariance} \\
&= \int_{\mathcal{G}} q_\phi(\mathbf{y}, \mathbf{z}_{\mathrm{h}}|\mathbf{x}, \mathbf{h}) \log p_\xi(\mathbf{x}, \mathbf{h}|\mathbf{y}, \mathbf{z}_{\mathrm{h}}) \cdot \underbrace{\det \mathbf{R}}_{=1} &&\text{Change of Variables } \mathbf{y} = \mathbf{R}^{-1}\mathbf{z}_{\mathrm{x}} \\
&= \mathbb{E}_{p_{\mathrm{data}}(\mathcal{G}), q_\phi(\mathbf{y}, \mathbf{z}_{\mathrm{h}}|\mathbf{x}, \mathbf{h})}\big[\log p_\xi(\mathbf{x}, \mathbf{h}|\mathbf{y}, \mathbf{z}_{\mathrm{h}})\big] \\
&= \mathcal{L}_{recon}(\mathbf{x}, \mathbf{h}),
\end{aligned}$$

(26)

which shows that $\mathcal{L}_{recon}(\mathbf{x}, \mathbf{h})$ is invariant. And for $\mathcal{L}_{LDM}(\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}})$, given that $q(\mathbf{z}_{\mathcal{G}}^{(t-1)}|\mathbf{z}_{\mathcal{G}}^{(t)}, \mathbf{z}_{\mathcal{G}}^{(0)})$ and $p_\theta(\mathbf{z}_{\mathcal{G}}^{(t-1)}|\mathbf{z}_{\mathcal{G}}^{(t)})$ are equivariant distributions, we have that:

$$\begin{aligned}
\mathcal{L}_{LDM}(\mathbf{R}\mathbf{z}_{\mathrm{x}}^{(0)}, \mathbf{z}_{\mathrm{h}}^{(0)}) =& \mathbb{E}_{p_{\mathrm{data}}(\mathcal{G})}\Big[\sum_{t=2}^{T} D_{\mathrm{KL}}(q(\mathbf{z}_{\mathrm{x}}^{(t-1)}, \mathbf{z}_{\mathrm{h}}^{(t-1)}|\mathbf{z}_{\mathrm{x}}^{(t)}, \mathbf{z}_{\mathrm{h}}^{(t)}, \mathbf{R}\mathbf{z}_{\mathrm{x}}^{(0)}, \mathbf{z}_{\mathrm{h}}^{(0)})\|p_\theta(\mathbf{z}_{\mathrm{x}}^{(t-1)}, \mathbf{z}_{\mathrm{h}}^{(t-1)}|\mathbf{z}_{\mathrm{x}}^{(t)}, \mathbf{z}_{\mathrm{h}}^{(t)})) \\
&- \log p_\theta(\mathbf{R}\mathbf{z}_{\mathrm{x}}^{(0)}, \mathbf{z}_{\mathrm{h}}^{(0)}|\mathbf{z}_{\mathrm{x}}^{(1)}, \mathbf{z}_{\mathrm{h}}^{(1)})\Big] \\
=& \int_{\mathbf{z}_{\mathcal{G}}} \Big[\sum_{t=2}^{T} \log \frac{q(\mathbf{z}_{\mathrm{x}}^{(t-1)}, \mathbf{z}_{\mathrm{h}}^{(t-1)}|\mathbf{z}_{\mathrm{x}}^{(t)}, \mathbf{z}_{\mathrm{h}}^{(t)}, \mathbf{R}\mathbf{z}_{\mathrm{x}}^{(0)}, \mathbf{z}_{\mathrm{h}}^{(0)})}{p_\theta(\mathbf{z}_{\mathrm{x}}^{(t-1)}|\mathbf{z}_{\mathrm{x}}^{(t)}, \mathbf{z}_{\mathrm{h}}^{(t)}))} - \log p_\theta(\mathbf{R}\mathbf{z}_{\mathrm{x}}^{(0)}, \mathbf{z}_{\mathrm{h}}^{(0)}|\mathbf{z}_{\mathrm{x}}^{(1)}, \mathbf{z}_{\mathrm{h}}^{(1)})\Big] \\
=& \int_{\mathbf{z}_{\mathcal{G}}} \Big[\sum_{t=2}^{T} \log \frac{q(\mathbf{R}\mathbf{R}^{-1}\mathbf{z}_{\mathrm{x}}^{(t-1)}, \mathbf{z}_{\mathrm{h}}^{(t-1)}|\mathbf{R}\mathbf{R}^{-1}\mathbf{z}_{\mathrm{x}}^{(t)}, \mathbf{z}_{\mathrm{h}}^{(t)}, \mathbf{R}\mathbf{z}_{\mathrm{x}}^{(0)}, \mathbf{z}_{\mathrm{h}}^{(0)})}{p_\theta(\mathbf{R}\mathbf{R}^{-1}\mathbf{z}_{\mathrm{x}}^{(t-1)}, \mathbf{z}_{\mathrm{h}}^{(t-1)}|\mathbf{R}\mathbf{R}^{-1}\mathbf{z}_{\mathrm{x}}^{(t)}, \mathbf{z}_{\mathrm{h}}^{(t)}))} \\
&- \log p_\theta(\mathbf{R}\mathbf{z}_{\mathrm{x}}^{(0)}, \mathbf{z}_{\mathrm{h}}^{(0)}|\mathbf{z}_{\mathrm{x}}^{(1)}, \mathbf{R}\mathbf{R}^{-1}\mathbf{z}_{\mathrm{h}}^{(1)})\Big] \quad \text{(Multiply by } \mathbf{R}\mathbf{R}^{-1} = \mathbf{I}) \\
=& \int_{\mathbf{z}_{\mathcal{G}}} \Big[\sum_{t=2}^{T} \log \frac{q(\mathbf{R}^{-1}\mathbf{z}_{\mathrm{x}}^{(t-1)}, \mathbf{z}_{\mathrm{h}}^{(t-1)}|\mathbf{R}^{-1}\mathbf{z}_{\mathrm{x}}^{(t)}, \mathbf{z}_{\mathrm{h}}^{(t)}, \mathbf{z}_{\mathrm{x}}^{(0)}, \mathbf{z}_{\mathrm{h}}^{(0)})}{p_\theta(\mathbf{R}^{-1}\mathbf{z}_{\mathrm{x}}^{(t-1)}, \mathbf{z}_{\mathrm{h}}^{(t-1)}|\mathbf{R}^{-1}\mathbf{z}_{\mathrm{x}}^{(t)}, \mathbf{z}_{\mathrm{h}}^{(t)}))}
\end{aligned}$$

(27)

$$- \log p_\theta(\mathbf{z}_\mathbf{x}^{(0)}, \mathbf{z}_\mathbf{h}^{(0)} | \mathbf{R}^{-1}\mathbf{z}_\mathbf{x}^{(1)}, \mathbf{z}_\mathbf{h}^{(1)}) \Big] \quad \text{(Equivariance \& Invariance)}$$

$$= \mathbb{E}_{p_{\text{data}}(\mathcal{G})} \Big[\sum_{t=2}^{T} D_{\mathrm{KL}}(q(\mathbf{y}_\mathbf{x}^{(t-1)}, \mathbf{z}_\mathbf{h}^{(t-1)} | \mathbf{y}_\mathbf{x}^{(t)}, \mathbf{z}_\mathbf{h}^{(t)}, \mathbf{z}_\mathbf{x}^{(0)}, \mathbf{z}_\mathbf{h}^{(0)}) \| p_\theta(\mathbf{y}_\mathbf{x}^{(t-1)}, \mathbf{z}_\mathbf{h}^{(t-1)} | \mathbf{y}_\mathbf{x}^{(t)}, \mathbf{z}_\mathbf{h}^{(t)}))$$

$$- \log p_\theta(\mathbf{z}_\mathbf{x}^{(0)}, \mathbf{z}_\mathbf{h}^{(0)} | \mathbf{y}_\mathbf{x}^{(1)}, \mathbf{z}_\mathbf{h}^{(1)}) \Big]$$

$$= \mathcal{L}_{LDM}(\mathbf{z}_\mathbf{x}^{(0)}, \mathbf{z}_\mathbf{h}^{(0)}),$$

which shows that $\mathcal{L}_{LDM}(\mathbf{R}\mathbf{z}_\mathbf{x}^{(0)}, \mathbf{z}_\mathbf{h}^{(0)})$ is invariant. Furthermore, since we operate on the zero-mean subspace, the objectives are naturally also translationally invariant. Thus, we finish the proof. □

B.2. Invariant Marginal Distribution: Proposition 4.3

We also include proofs for key properties of the equivariant probabilistic diffusion model here to be self-contained (Xu et al., 2022; Hoogeboom et al., 2022). Note that, since here we are interested in the equivariant properties, we omit the trivial scalar inputs $\mathbf{h}$ and focus on analyzing tensor features $\mathbf{z}$. The proof shows that when the the initial distribution $p(\mathbf{z}^{(T)})$ is invariant and transition distributions $p(\mathbf{z}_\mathbf{x}^{(t-1)}|\mathbf{z}_\mathbf{x}^{(t)})$ are equivariant, then the marginal distributions $p(\mathbf{z}_\mathbf{x}^{(t)})$ will be invariant, importantly including $p(\mathbf{z}_\mathbf{x}^{(0)})$. Similarly, with decoder $p(\mathbf{x}|\mathbf{z}_\mathbf{x}^{(0)})$ also being equivariant, we can further have that our induced distribution $p(\mathbf{x})$ is invariant.

Proof. The justification formally can be derived as follow:

Condition: We are given that $p(\mathbf{z}_\mathbf{x}^T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ is invariant with respect to rotations, *i.e.*, $p(\mathbf{z}_\mathbf{x}^T) = p(\mathbf{R}\mathbf{z}_\mathbf{x}^T)$.

Derivation: For $t \in \{1, \cdots, T\}$, let $p(\mathbf{z}_\mathbf{x}^{t-1}|\mathbf{z}_\mathbf{x}^t)$ be equivariant distribution, *i.e.*, $p(\mathbf{z}_\mathbf{x}^{t-1}|\mathbf{z}_\mathbf{x}^t) = p(\mathbf{R}\mathbf{z}_\mathbf{x}^{t-1}|\mathbf{R}\mathbf{z}_\mathbf{x}^t)$ for all orthogonal $\mathbf{R}$. Assume $p(\mathbf{z}_\mathbf{x}^t)$ to be invariant distribution, *i.e.*, $p(\mathbf{z}_\mathbf{x}^t) = p(\mathbf{R}\mathbf{z}_\mathbf{x}^t)$ for all orthogonal $\mathbf{R}$, then we have:

$$p(\mathbf{R}\mathbf{z}_\mathbf{x}^{t-1}) = \int_{\mathbf{z}_\mathbf{x}^t} p(\mathbf{R}\mathbf{z}_\mathbf{x}^{t-1}|\mathbf{z}_\mathbf{x}^t)p(\mathbf{z}_\mathbf{x}^t) \qquad \text{Chain Rule}$$

$$= \int_{\mathbf{z}_\mathbf{x}^t} p(\mathbf{R}\mathbf{z}_\mathbf{x}^{t-1}|\mathbf{R}\mathbf{R}^{-1}\mathbf{z}_\mathbf{x}^t)p(\mathbf{R}\mathbf{R}^{-1}\mathbf{z}_\mathbf{x}^t) \qquad \text{Multiply by } \mathbf{R}\mathbf{R}^{-1} = \mathbf{I}$$

$$= \int_{\mathbf{z}_\mathbf{x}^t} p(\mathbf{z}_\mathbf{x}^{t-1}|\mathbf{R}^{-1}\mathbf{z}_\mathbf{x}^t)p(\mathbf{R}^{-1}\mathbf{z}_\mathbf{x}^t) \qquad \text{Equivariance \& Invariance}$$

$$= \int_{\mathbf{y}} p(\mathbf{z}_\mathbf{x}^{t-1}|\mathbf{y})p(\mathbf{y}) \cdot \underbrace{\det \mathbf{R}}_{=1} \qquad \text{Change of Variables } \mathbf{y} = \mathbf{R}^{-1}\mathbf{z}_\mathbf{x}^t$$

$$= p(\mathbf{z}_\mathbf{x}^{t-1}),$$

and therefore $p(\mathbf{z}_\mathbf{x}^{t-1})$ is invariant. By induction, $p(\mathbf{z}_\mathbf{x}^{T-1}), \ldots, p(\mathbf{z}_\mathbf{x}^0)$ are all invariant. Furthermore, since the decoder $p(\mathbf{x}|\mathbf{z}_\mathbf{x}^{(0)})$ is also equivariant, with the same derivation we also have that our induced distribution $p(\mathbf{x})$ is invariant. □

C. Model Architecture Details

In our implementation, all models are parameterized with EGNNs (Satorras et al., 2021b) as backbone. EGNNs are a class of Graph Neural Network that satisfies the equivariance property in Equations (5) and (9). In this work, we consider molecular geometries as point clouds, without specifying the connecting bonds. Therefore, in practice, we take the point clouds as fully connected graph G and model the interactions between all atoms $\mathbf{v}_i \in \mathcal{V}$. Each node $\mathbf{v}_i$ is embedded with coordinates $\mathbf{x}_i \in \mathbb{R}^3$ and atomic features $\mathbf{h}_i \in \mathbb{R}^d$. Then, EGNNs are composed of multiple Equivariant Convolutional

Layers $\mathbf{x}^{l+1}, \mathbf{h}^{l+1} = \mathrm{EGCL}[\mathbf{x}^l, \mathbf{h}^l]$, with each single layer defined as:

$$
\begin{aligned}
\mathbf{m}_{ij} &= \phi_e\left(\mathrm{h}_i^l, \mathrm{h}_j^l, d_{ij}^2, a_{ij}\right), \\
\mathrm{h}_i^{l+1} &= \phi_h(\mathrm{h}_i^l, \sum_{j \neq i} \tilde{e}_{ij} \mathbf{m}_{ij}), \\
\mathrm{x}_i^{l+1} &= \mathrm{x}_i^l + \sum_{j \neq i} \frac{\mathrm{x}_i^l - \mathrm{x}_j^l}{d_{ij} + 1} \phi_x\left(\mathrm{h}_i^l, \mathrm{h}_j^l, d_{ij}^2, a_{ij}\right),
\end{aligned}
\tag{28}
$$

where l denotes the layer index. $\tilde{e}_{ij} = \phi_{inf}(\mathbf{m}_{ij})$ acts as the attention weights to reweight messages passed from different edges. $d_{ij} = \|\mathrm{x}_i^l - \mathrm{x}_j^l\|_2$ represents the pairwise distance between atoms v_i and v_j, and a_{ij} are optional edge features. We follow previous work (Satorras et al., 2021a; Hoogeboom et al., 2022) to normalize the relative directions $\mathrm{x}_i^l - \mathrm{x}_j^l$ in Equation (28) by $d_{ij} + 1$, which empirically improved model stability. All learnable functions, *i.e.*, ϕ_e, ϕ_h, ϕ_x and ϕ_{inf}, are parameterized by Multi Layer Perceptrons (MLPs). Then a complete EGNN model can be realized by stacking L EGCL layers such that $\mathbf{x}^L, \mathbf{h}^L = \mathrm{EGNN}[\mathbf{x}^0, \mathbf{h}^0]$, which can satisfy the required equivariant constraint in Equations (5) and (9).

D. Featurization and Implementation Details

We use the open-source software RDKIT (Landrum, 2016) to preprocess molecules. For QM9 we take atom types (H, C, N, O, F) and integer-valued atom charges as atomic features, while for Drugs we only use atom types. The results reported in Sections 5.2 and 5.3 are based on the *ES-reg* regularization strategy (Section 4.3), where the encoder is only optimized with 1000 iterations of warm-up training and then fixed. For the diffusion process (Equation (1)), we use the polynomial noise schedule (Hoogeboom et al., 2022; Wu et al., 2022), where α linearly decays from $10^3/T$ to 0 *w.r.t.* time step t. And for the denoising process (Equation (2)), the variances are defined as $\rho_t = \sqrt{\frac{\sigma_{t-1}}{\sigma_t}} \beta_t$.

All neural networks used for the encoder, latent diffusion, and decoder are implemented with EGNNs (Satorras et al., 2021b) by PyTorch (Paszke et al., 2017) package, as introduced in Appendix C. We set the dimension of latent invariant features k to 1 for QM9 and 2 for DRUG, which extremely reduces the atomic feature dimension. For the training of latent denoising network ϵ_θ: on QM9, we train EGNNs with 9 layers and 256 hidden features with a batch size 64; and on GEOM-DRUG, we train EGNNs with 4 layers and 256 hidden features, with batch size 64. For the autoencoders, we parameterize the decoder $\mathcal{D}_\xi$ in the same way as ϵ_θ, but implement the encoder $\mathcal{E}_\phi$ with a 1 layer EGNN. The shallow encoder in practice constrains the encoding capacity and helps regularize the latent space. All models use SiLU activations. We train all the modules until convergence. For all the experiments, we choose the Adam optimizer (Kingma & Ba, 2014) with a constant learning rate of 10^{-4} as our default training configuration. The training on QM9 takes approximately 2000 epochs, and on DRUG takes 20 epochs.

E. Ablation Studies

In this section, We provide additional experimental results on QM9 to justify the effect of several model designs. Specifically, we perform ablation studies on two key model designs: *autoencoder regularization method* and *latent space dimension* k. The results are reported in Table 3.

Table 3. Results of ablation study with different model designs. Metrics are calculated with 10000 samples generated from each setting.

# Metrics	Atom Sta (%)	Mol Sta (%)	Valid (%)	Valid & Unique (%)
GEOLDM ($k = 1$, *KL-reg*)*	95.45	40.7	83.7	83.5
GEOLDM ($k = 16$, *ES-reg*)	98.6	86.0	92.4	92.2
GEOLDM ($k = 8$, *ES-reg*)	98.7	87.1	92.1	92.0
GEOLDM ($k = 4$, *ES-reg*)	98.8	87.4	92.6	92.5
GEOLDM ($k = 1$, *ES-reg*)	**98.9** $\pm$ 0.1	**89.4** $\pm$ 0.5	**93.8** $\pm$ 0.4	**92.7** $\pm$ 0.5

*Note that this reported result is already the best result we achieved for *KL-reg*.

We first discuss the effect of different autoencoder regularization methods, *i.e.*, *KL-reg* and *ES-reg* (see details in 4.1), with the latent invariant feature dimension fixed as 1. Following previous practice in latent diffusion models of image and point

clouds domains (Rombach et al., 2022; Zeng et al., 2022), for *KL-reg*, we also weight the KL term with a small factor 0.01. However, during our initial experiments where we naturally first try the *KL-reg* method, we observed unexpected failure with extremely poor performance, as shown in the first row in Table 3. Note that, this reported result is already the best result we achieved for *KL-reg*, with searching over a large range of KL term weights and latent space dimension. In practice, we even notice the *KL-reg* is unstable for training, which often suffers from numerical errors during training. Our closer observation of the experimental results suggests that the *equivariant latent feature* part always tends to converge to highly scattered means and extremely small variances, which leads to the numerical issue for calculating KL term and also is not suitable for LDM training. Therefore, we turned to constraining the encoder, more precisely, constraining the value scale of encoded latent features, by early stopping the training encoder. This easy strategy turned out to work pretty well in practice as shown in Table 3, and we leave the further study of *KL-reg* as future work in this area.

We further study the effect of latent invariance feature dimension k, and the results are also reported in Table 3. As shown in the table, we observe that generally GEOLDM shows better performance with lower k. This phenomenon verifies our motivation that a lower dimensionality can alleviate the generative modeling complexity and benefit the training of LDM. Specifically, the performances of GEOLDM on QM9 with k set as 1 or 2 are very similar, so we only report $k = 1$ as representative in Table 3. In practice, we set k as 1 for QM9 dataset and 2 for DRUG which contains more atom types.

F. More Visualization Results

In this section, we provide more visualizations of molecules generated from GEOLDM. Samples drawn from models trained on QM9 and DRUG are provided in Figure 4 and Figure 5 respectively. These examples are randomly generated without any cherry pick. Therefore, the generated geometries might be difficult to see in some figures due to imperfect viewing direction.

As shown in the two figures, the model is always able to generate realistic molecular geometries for both small and large size molecules. An outlier case is that the model occasionally generates disconnected components, as shown in the rightest column of Figure 5, which happens more often when trained on the large molecule DRUG dataset. However, this phenomenon actually is not a problem and is common in all non-autoregressive molecule generative models (Zang & Wang, 2020; Jo et al., 2022), and can be easily fixed by just filtering the smaller components.
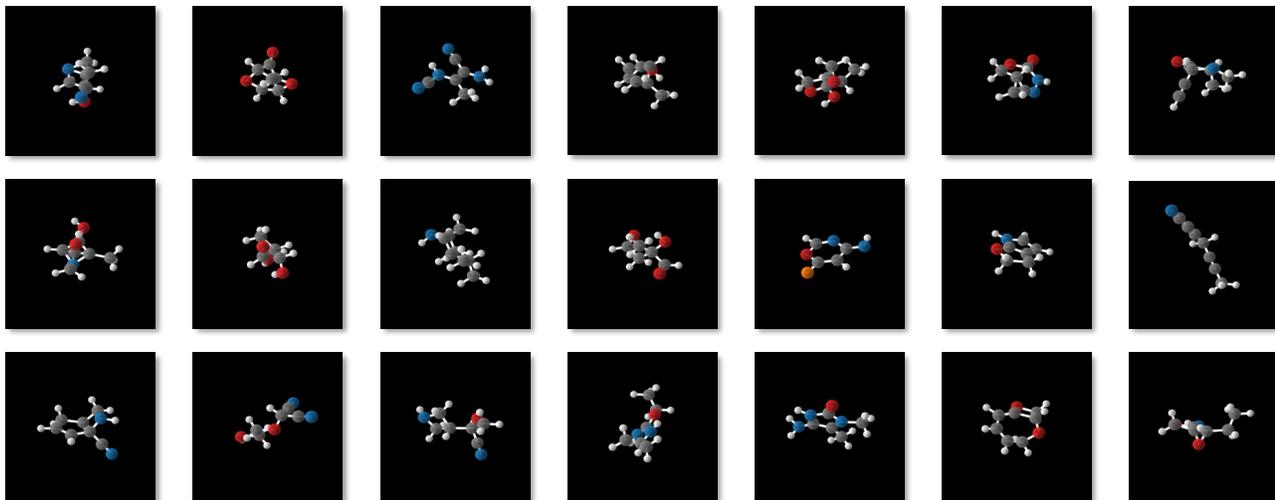

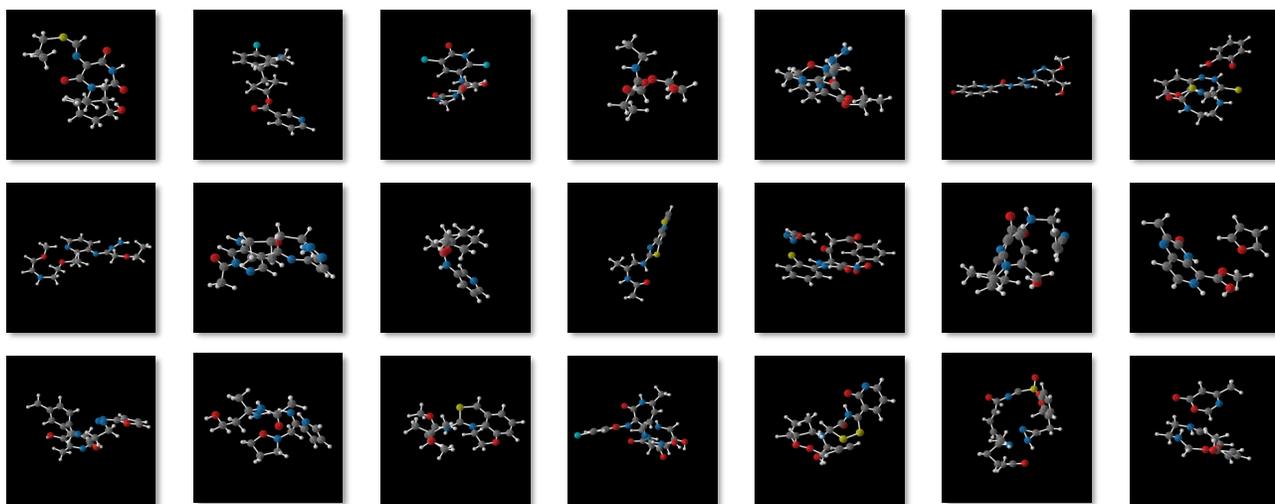*Figure 4.* Molecules generated from GEOLDM trained on QM9.

Figure 5. Molecules generated from GEOLDM trained on DRUG.