QUOTIENT-SPACE DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion-based generative models have reformed generative AI, and have enabled new capabilities in the science domain, for example, generating 3D structures of molecules. Due to the intrinsic problem structure of certain tasks, there is often a *symmetry* in the system, which identifies objects that can be converted by a group action as equivalent, hence the target distribution is essentially defined on the *quotient space* with respect to the group. In this work, we establish a formal framework for diffusion modeling on a general quotient space, and apply it to molecular structure generation which follows the special Euclidean group $\mathrm{SE}(3)$ symmetry. The framework reduces the necessity of learning the component corresponding to the group action, hence simplifies learning difficulty over conventional group-equivariant diffusion models, and the sampler guarantees recovering the target distribution, while heuristic alignment strategies lack proper samplers. The arguments are empirically validated on structure generation for small molecules and proteins, indicating that the principled quotient-space diffusion model provides a new framework that outperforms previous symmetry treatments.

1 INTRODUCTION

Diffusion models have emerged as the dominant approach for modeling distributions in high-dimensional spaces. Building on their success in real-world domains such as images (Ho et al., 2020; Song et al., 2021), audios (Kong et al., 2021; Evans et al., 2024), and videos (Ho et al., 2022; Li et al., 2023), diffusion models are now increasingly adopted in scientific applications, ranging from fluid field solving (Bastek et al., 2025), electronic structure prediction (Kim et al., 2025), molecular structure generation (Xu et al., 2022; Abramson et al., 2024; Hassan et al., 2024; Geffner et al., 2025), and thermodynamic ensemble modeling (Zheng et al., 2024; Lewis et al., 2025).

Compared with general tasks, scientific applications often exhibit inherent *symmetry* structures, wherein objects that can be related through specific transformations are regarded as equivalent. Consider molecular structure generation as a representative example. A molecular structure can be represented as a vector in $\mathbb{R}^{3N}$ by concatenating the 3D coordinates of its N atoms. However, because the choice of coordinate system is arbitrary, vectors in $\mathbb{R}^{3N}$ that differ only by a global 3D translation or rotation of all atoms correspond to the same underlying structure. Mathematically, such transformations typically form a Lie group — for example, the special Euclidean group $\mathrm{SE}(3)$ in the case of molecular structures, which formally characterizes the symmetry.

The common treatment is putting the target distribution in the original space but assigning the same probability to equivalent objects, resulting in a distribution that is invariant under group action. This can be implemented by augmenting training data by applying randomly chosen group actions (Abramson et al., 2024), or using a group equivariant model (Xu et al., 2022; Hoogeboom et al., 2022b), which guarantees invariance if the starting prior distribution is invariant (Köhler et al., 2020). Nevertheless, we shall show that this treatment still has room to improve, as the neural network model, which is intended for updating the sample in each diffusion simulation step, still needs to learn a *specific* movement within the equivalent class (*e.g.*, rotating a molecular structure), which is unnecessary as *any* such a movement does not update the intrinsic system state (*e.g.*, the shape of a molecular structure) hence is acceptable. In hope to remove this redundancy, there are a few heuristic treatments using alignment, *i.e.*, adjusting the prediction target within its equivalent class according to a reference to remove these equivalent degrees of freedom (Xu et al., 2022; Abramson et al., 2024). But we find that the corresponding sampling process becomes incompatible with such training strategies, even with heuristic fix attempts (Wohlwend et al., 2025).

Table 1: Comparison among different training strategies in presence of a symmetry group. Learning difficulty is measured by whether the need to predict in the equivalent degrees of freedom (DOFs), induced by the group actions, is removed, and (if not) whether the variance on the equivalent DOFs is removed. Sampling compatibility means whether there is a sampler that exactly reproduces the target distribution. The denoising form of diffusion model $\mathbf{D}_\theta$ is used to express the loss functions, where $\mathcal{A}_\mathbf{y}(\mathbf{x})$ (Eq. (11)) represents aligning $\mathbf{x}$ towards $\mathbf{y}$, and $\bar{\theta}$ denotes treating θ as constant (*i.e.*, stop-gradient). The conclusions hold using either an equivariant architecture or a general architecture with data augmentation. See Sec. 3.4 for details.

Training strategy for $\mathbf{D}_\theta$	Optimal solution of $\mathbf{D}_\theta$	Reduction of learning difficulty		Sampling compatibility
		Removal of equivalent DOFs	Removal of variance on equivalent DOFs	
Conventional loss $\mathbb{E}\|\mathbf{D}_\theta(\mathbf{x}_t,t)-\mathbf{x}_1\|^2$	$\mathbb{E}[\mathbf{x}_1\|\mathbf{x}_t]$	✗	✗	✓
GeoDiff alignment loss $\mathbb{E}\|\mathbf{D}_\theta(\mathbf{x}_t,t)-\mathcal{A}_{\mathbf{x}_t}(\mathbf{x}_1)\|^2$	$\mathbb{E}[\mathcal{A}_{\mathbf{x}_t}(\mathbf{x}_1)\|\mathbf{x}_t]$	✗	✓	✗
AF3 alignment loss $\mathbb{E}\|\mathbf{D}_\theta(\mathbf{x}_t,t)-\mathcal{A}_{\mathbf{D}_{\bar{\theta}}(\mathbf{x}_t,t)}(\mathbf{x}_1)\|^2$	$g\cdot\mathbb{E}[\mathcal{A}_{\mathbf{x}_t}(\mathbf{x}_1)\|\mathbf{x}_t]$ for arbitrary $g\in\mathcal{G}$	✓	✓	✗
quotient-space diffusion loss $\mathbb{E}\|P_{\mathbf{x}_t}(\mathbf{D}_\theta(\mathbf{x}_t,t)-\mathbf{x}_1)\|^2$	$\mathbb{E}[P_{\mathbf{x}_t}(\mathbf{x}_1)\|\mathbf{x}_t]+\mathbf{v}^\mathcal{V}$ for arbitrary $\mathbf{v}^\mathcal{V}\in\mathrm{Ker}(P_{\mathbf{x}_t})$	✓	✓	✓

In this work, we develop a principled approach to building a diffusion model considering the intrinsic symmetry of the system. In particular, we leverage the concept of *quotient space*, in which a set of equivalent objects (equivalent class) are treated as one element. It is the formal mathematical construction that reflects the intrinsic variability of the system. We first derive the diffusion process on a general quotient space based on the correspondence between the Wiener processes on the two spaces. Considering that the quotient space is generally not Euclidean, hence it is hard to directly carry out a simulation on it, we further leverage the mathematical construction of horizontal lift to induce a diffusion process back in the original space that embeds[1] the quotient-space diffusion process. The resulting process effectively amounts to projecting the update vector in the original diffusion process onto the subspace that does not induce a movement within the equivalent class (*e.g.*, rotation). We show that this process *guarantees producing the correct target distribution*, meanwhile *reduces learning difficulty* by removing the necessity to learn a specific movement within an equivalent class. A visualization example in the 2-dimensional plane with $\mathrm{SO}(2)$ symmetry is shown in Fig. 1. In this example, the lifted process only has radial movements (Fig. 1(Left)) as the quotient space $\mathbb{R}^2/\mathrm{SO}(2)$ is isomorphic to the half real line and recovers the correct target distribution as conventional equivariant diffusion models (Fig. 1(Middle, Right)). A conceptual comparison with existing methods is shown in Table 1. The quotient-space diffusion admits either an equivariant model or a general model with data augmentation.

As a representative application, we deduce the specific training and sampling algorithms in the $\mathbb{R}^{3N}/\mathrm{SE}(3)$ scenario for molecular structure generation, which relaxes the model from learning a translation and rotation movement, while the sampling process keeps the structure with constant position and orientation. We study the empirical performance of quotient-space diffusion models on small molecule structure generation and protein backbone design tasks. The results show that our methods can consistently improve the generation performance in these applications over conventional equivariant diffusion models and using alignment strategies. Our method achieves 9%-23% relative improvements of ET-Flow(Hassan et al., 2024) on GEOM-QM9 and GEOM-DRUGS datasets, surpassing previous heuristic alignment methods. For the protein structure generation task, our method surpasses the state-of-the-art Proteína model (Geffner et al., 2025) with the same parameter scale (60M) in a large margin and also outperforms the much larger model (200M) on most key distributional metrics.

2 Background

2.1 Diffusion-based Generative Models on Euclidean Space

The main idea of diffusion models is to construct a step-by-step transformation from a simple prior distribution to a complex target distribution. In this paper, we follow the Stochastic Interpolant

[1]This "embedding" is meant for intuitive understanding; in the mathematical sense, the quotient space is unnecessarily able to be embedded in the original space.

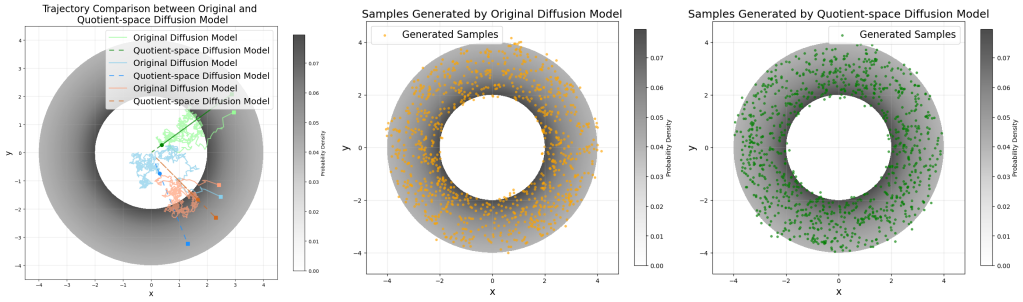

Figure 1: A visualization of the behavior of the original equivariant diffusion model and the quotient-space comparison between the original diffusion model and the quotient-space diffusion model in the $\mathcal{M} = \mathbb{R}^2, \mathcal{G} = \mathrm{SO}(2)$ case. The data distribution is restricted in the region $r \in [2, 4]$, where r is the radius on $\mathbb{R}^2$. The probability density function is shown in the color bar. **(Left)** The sampling trajectory comparison between the original diffusion model and the quotient-space diffusion model. The quotient-space diffusion model always diffuses on a straight line because the quotient space $\mathbb{R}^2/\mathrm{SO}(2)$ is isomorphic to the half real line, while the original diffusion model diffuses on a 2-D plane. This motivates us to reduce the learning task corresponding to the movement in the equivalent class (movement on a circle in this case). **(Middle)** The samples generated by the original diffusion model. **(Right)** The samples generated by the quotient-space diffusion model, which match the data distribution as well.

framework (Albergo et al., 2023), which unifies diffusion models and flow matching models (Lipman et al., 2023; Liu et al., 2023). Let $p_{\text{target}}(\mathbf{x})$ be the target distribution. The following linear interpolation is constructed:

$$\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \beta_t \mathbf{x}_1 + \gamma_t \boldsymbol{\epsilon}, \quad (\mathbf{x}_0, \mathbf{x}_1) \sim p_{\text{joint}}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}), \quad t \in [0, 1] \tag{1}$$

where p_{joint} is a pre-defined joint distribution of $(\mathbf{x}_0, \mathbf{x}_1)$ with marginals $\mathbf{x}_0 \sim p_{\text{prior}}$ and $\mathbf{x}_1 \sim p_{\text{target}}$. The coefficients $\alpha_t, \beta_t, \gamma_t$ satisfy the boundary conditions $\alpha_0 = 1$, $\beta_0 = 0$, $\gamma_0 = 0$, and $\alpha_1 = 0$, $\beta_1 = 1$, $\gamma_1 = 0$. Under these conditions, the following ordinary differential equation (ODE) can transform p_{prior} to p_{target} (Albergo et al., 2023, Cor. 2.18):

$$\mathrm{d}\mathbf{x}_t = \mathbf{v}(\mathbf{x}_t, t)\,\mathrm{d}t, \quad \text{where} \quad \mathbf{v}(\mathbf{x}_t, t) := \mathbb{E}[\alpha_t' \mathbf{x}_0 + \beta_t' \mathbf{x}_1 + \gamma_t' \boldsymbol{\epsilon} \mid \mathbf{x}_t]. \tag{2}$$

The velocity vector field $\mathbf{v}(\mathbf{x}_t, t)$ is typically trained with the objective: $\mathcal{L}(\theta) := \mathbb{E}_{p(t)} w(t) \mathbb{E}_{p_{\text{joint}}(\mathbf{x}_0, \mathbf{x}_1) p(\boldsymbol{\epsilon})} \|\mathbf{v}_\theta(\mathbf{x}_t, t) - (\alpha_t' \mathbf{x}_0 + \beta_t' \mathbf{x}_1 + \gamma_t' \boldsymbol{\epsilon})\|^2$, where the prime denotes the time derivative, and $p(t)$ and $w(t)$ control the sampling distribution and weighting over time. There is also a stochastic process for sample generation, given by :

$$\mathrm{d}\mathbf{x}_t = (\mathbf{v}(\mathbf{x}_t, t) + \eta_t \mathbf{s}(\mathbf{x}_t, t))\,\mathrm{d}t + \sqrt{2\eta_t}\,\mathrm{d}\mathbf{w}_t, \quad \text{where} \quad \mathbf{s}(\mathbf{x}_t, t) := \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) \tag{3}$$

is called the score function, and $\eta_t \geq 0$ is a non-negative smooth function (Albergo et al., 2023, Cor. 2.10). In the special case where $p_{\text{prior}} = \mathcal{N}(\mathbf{0}, \mathbf{I})$ (the *one-sided stochastic interpolant* (Albergo et al., 2023, Def. 3.4)), contributions of $\mathbf{x}_0$ and $\boldsymbol{\epsilon}$ can be combined as $\mathbf{x}_t = \hat{\alpha}_t \boldsymbol{\epsilon} + \beta_t \mathbf{x}_1$, where $\hat{\alpha}_t = \sqrt{\alpha_t^2 + \gamma_t^2}$, and the score function can be expressed by the velocity field: $\mathbf{s}(\mathbf{x}_t, t) = \frac{\beta_t' \mathbf{x}_t - \beta_t \mathbf{v}(\mathbf{x}_t, t)}{\hat{\alpha}_t(\hat{\alpha}_t' \beta_t - \hat{\alpha}_t \beta_t')}$.

A convenient variant to formulate the learning task is to define the $\mathbf{v}_\theta(\mathbf{x}_t, t)$ model with a neural network $\mathbf{D}_\theta(\mathbf{x}_t, t)$ which reformulates the objective:

$$\mathbf{v}_\theta(\mathbf{x}_t, t) := \frac{\hat{\alpha}_t' \mathbf{x}_t - (\hat{\alpha}_t' \beta_t - \hat{\alpha}_t \beta_t') \mathbf{D}_\theta(\mathbf{x}_t, t)}{\hat{\alpha}_t}, \tag{4}$$

$$\mathcal{L}(\theta) := \mathbb{E}_{p(t)} w(t) \frac{(\hat{\alpha}_t' \beta_t - \hat{\alpha}_t \beta_t')^2}{\hat{\alpha}_t^2} \mathbb{E}_{p(\mathbf{x}_1, \mathbf{x}_t)} \|\mathbf{D}_\theta(\mathbf{x}_t, t) - \mathbf{x}_1\|^2, \tag{5}$$

where $p(\mathbf{x}_1, \mathbf{x}_t)$ is derived from Eq. (1) by integrating out $\mathbf{x}_0$ and $\boldsymbol{\epsilon}$. This objective conveys the intuition of recovering the clean-data sample $\mathbf{x}_1$ from a noisy sample $\mathbf{x}_t$, hence $\mathbf{D}_\theta(\mathbf{x}_t, t)$ is called a denoising model and suits prevalent architectures. We adopt this form of a diffusion model below.

2.2 From Euclidean Space to Quotient Manifold

Tasks in scientific domains often involve inherent symmetry, where objects related by certain transformations are considered equivalent. A formal and inclusive description of symmetry in a system

requires both the geometry of the configuration space and the algebraic structure of the transformations, which leads to the concepts of manifolds and Lie groups.

Manifold and Lie groups. A (smooth) manifold is a geometric object that generalizes the Euclidean space to allow spatial heterogeneity. Typically, a manifold is endowed with a Riemannian metric, *i.e.*, an inner product in each tangent space, which leads to common concepts like curve length, distance, measure, gradient, Laplacian, and Wiener process on the manifold (Appx. B.1). Symmetries are formally represented by transformations that connect equivalent (*i.e.*, symmetric) objects, which constitute a group. A continuously-parameterized group that is also a manifold is called a Lie group.

We consider the general case where the configuration space of the system is an M-dimensional Riemannian manifold $\mathcal{M}$. The symmetry of the system is represented by a G-dimensional Lie group $\mathcal{G}$ acting on $\mathcal{M}$. A distribution p on $\mathcal{M}$ is said $\mathcal{G}$-invariant if $p(g \cdot \mathbf{x}) = p(\mathbf{x}), \forall g \in \mathcal{G}, \mathbf{x} \in \mathcal{M}$. This invariance implies that all equivalent points $\{g \cdot \mathbf{x} \mid g \in \mathcal{G}\}$, collectively called an equivalent class, are assigned with the same probability.

Quotient space. The symmetry group defines an equivalent relation in $\mathcal{M}$, *i.e.*, $\mathbf{x}_1$ and $\mathbf{x}_2$ are equivalent, if there exists a group action $g \in \mathcal{G}$ such that $g \cdot \mathbf{x}_1 = \mathbf{x}_2$, which is indeed an equivalent relation due to properties of a group. The quotient space $\mathcal{Q} := \mathcal{M}/\mathcal{G}$ treats equivalent objects under the action of $\mathcal{G}$ as one element, hence reflects the intrinsic variability of the system. There is a natural mapping called the projection connecting the two spaces: $\pi(\mathbf{x}) := \{g \cdot \mathbf{x} \mid g \in \mathcal{G}\}$. Under appropriate conditions, the quotient space is a smooth manifold with dimension $M - G$ (Appx. C). However, defining a diffusion process on this space is non-trivial, necessitating the extension of "velocity" and Wiener process from Euclidean space to the manifold.

Tangent vector. On a manifold $\mathcal{M}$, the velocity of a process at a certain point $\mathbf{x}$ is represented as a tangent vector at $\mathbf{x}$, intuitively representing an infinitesimal movement. All tangent vectors at $\mathbf{x}$ constitute a linear space $T_{\mathbf{x}}\mathcal{M}$ called the tangent space at $\mathbf{x}$. Since a manifold is typically curved, tangent spaces at different points are regarded as different linear spaces, but with a transformation on the manifold, *e.g.*, a group action g, an associated mapping $g_{*\mathbf{x}} : T_{\mathbf{x}}\mathcal{M} \to T_{g\cdot\mathbf{x}}\mathcal{M}$ between the tangent spaces can be defined, which can be intuitively perceived as $g_{*\mathbf{x}}(\mathbf{v}) := \lim_{h\to 0} \frac{g\cdot(\mathbf{x}+h\mathbf{v})-g\cdot\mathbf{x}}{h}$.[2] With this construction, we can define that a vector field on $\mathcal{M}$ is $\mathcal{G}$-equivariant if it is unchanged under the group action: $g_{*\mathbf{x}}\mathbf{v}(\mathbf{x}) = \mathbf{v}(g \cdot \mathbf{x})$. For the projection mapping π onto the quotient space, we can similarly define $\pi_{*\mathbf{x}} : T_{\mathbf{x}}\mathcal{M} \to T_{\pi(\mathbf{x})}\mathcal{Q}$ as the projection for tangent vectors.

Wiener process on a manifold. In Euclidean space, the Wiener process is generated by the Laplacian operator $\frac{1}{2}\Delta$. The Laplace-Beltrami operator, defined from a Riemannian metric, serves as a counterpart on a manifold, and defines the Wiener process to the manifold. Under a symmetry group $\mathcal{G}$, we require a meaningful stochastic process on the manifold $\mathcal{M}$ as $\mathcal{G}$-invariant, meaning that its marginal distribution is $\mathcal{G}$-invariant at any time step. See Appx. B for details.

3 Methods

As the quotient space represents the "essential states" of a system with symmetry, a principled diffusion model for the system is expected to be built on it. In this section, we unroll the development of the quotient-space diffusion model by deriving the projected diffusion process onto the quotient space, then lift it back into the total space (*i.e.*, the original space) for convenient implementation. We then derive the specialization in the $\mathbb{R}^{3N}/\mathrm{SE}(3)$ case for molecular structure generation, followed by training and sampling algorithms. We highlight the merit of the quotient-space diffusion in reducing training difficulty and sampler soundness with a comparative analysis with existing treatments considering symmetry.

3.1 Diffusion Process on a General Quotient Space

If the diffusion process in $\mathcal{M}$ is $\mathcal{G}$-invariant, the distribution at any time step can be viewed as a distribution in the quotient space $\mathcal{Q}$, then we can view the process as a stochastic process in $\mathcal{Q}$. By leveraging the projection mapping $\pi : \mathcal{M} \to \mathcal{Q}$, we can map a diffusion process $\{\mathbf{x}_t\}_{t\in[0,T]}$ in $\mathcal{M}$

[2] This is the understanding from a Euclidean-space perspective. In general, there are no "addition/subtraction" operations on a general manifold. The formal definition is by defining tangent vectors a directional derivative operators, and the push-forward mapping $g_{*\mathbf{x}}$ is defined by function composition. See Appx. B for details.

(Eq. (3)) onto the quotient space as $\{\mathbf{y}_t := \pi(\mathbf{x}_t)\}_{t\in[0,T]}$. This is a stochastic process on $\mathcal{Q}$, but its expression as a diffusion process on $\mathcal{Q}$ using specifiers defining the diffusion process of $\mathbf{x}_t$ is desired. The following theorem gives an explicit answer.

Theorem 1. *Assume $\{\mathbf{x}_t\}_{t\in[0,T]}$ is a diffusion process on $\mathcal{M}$, specified by the following SDE:*

$$\mathrm{d}\mathbf{x}_t = \mathbf{b}_t(\mathbf{x}_t)\,\mathrm{d}t + \sigma_t\,\mathrm{d}\mathbf{w}_t, \quad \mathbf{x}_0 \sim p_{\text{prior}}, \tag{6}$$

where $\mathbf{b}_t$ is a $\mathcal{G}$-equivariant time-dependent vector field on $\mathcal{M}$, $\mathbf{w}_t$ is the Wiener process on $\mathcal{M}$ that is also $\mathcal{G}$-invariant, and p_{prior} is a $\mathcal{G}$-invariant distribution. Then the projected process $\{\mathbf{y}_t := \pi(\mathbf{x}_t)\}_{t\in[0,T]}$ onto the quotient space $\mathcal{Q} := \mathcal{M}/\mathcal{G}$ is the solution to the following SDE:

$$\mathrm{d}\mathbf{y}_t = \left((\pi_*\mathbf{b}_t)(\mathbf{y}_t) - \frac{\sigma_t^2}{2}\mathbf{h}(\mathbf{y}_t)\right)\mathrm{d}t + \sigma_t\,\mathrm{d}\boldsymbol{\omega}_t, \quad \mathbf{y}_0 \sim \pi_{\#}p_{\text{prior}}, \tag{7}$$

*where $\pi_*\mathbf{b}_t$ is the projected vector field of $\mathbf{b}_t$ induced by π, $\mathbf{h}(\mathbf{y}_t)$ is the mean curvature vector field of $\mathcal{Q}$ reflecting the geometry of $\mathcal{Q}$, $\boldsymbol{\omega}_t$ is the Wiener process on $\mathcal{Q}$, and $\pi_{\#}p_{prior}$ is the pushed-forward distribution of p_{prior} (*i.e.*, $\mathbf{y}_0 = \pi(\mathbf{x}_0)$ where $\mathbf{x}_0 \sim p_{\text{prior}}$).*

See Appx. D.1 for formal definitions of the concepts and the proof. Thm. 1 shows that the projected process is indeed a diffusion process on $\mathcal{Q}$, which consists of the projected vector field and corresponding Wiener diffusion process, and perhaps unexpectedly, an additional vector field reflecting the curvature of $\mathcal{Q}$. As the quotient space squeezes an equivalent class as one point, a process viewed on the quotient space should accommodate for the change of the volume of the equivalent class along the movement. This additional vector is the gradient (*i.e.*, the change rates in all movement directions) of the volume of the equivalent class.

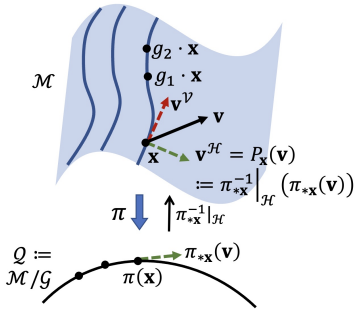

Figure 2: Illustration of the relation between the total space $\mathcal{M}$ and the quotient space $\mathcal{Q}$ and the correspondence of tangent vectors among them.

Although the diffusion process on the quotient space is defined, it is not convenient to simulate it in the quotient space directly due to the non-trivial geometric structure of $\mathcal{Q}$. Nevertheless, the quotient-space diffusion enables us a principled view to reduce the unnecessary movement within equivalent classes. A key observation from Thm. 1 is that if $\mathbf{b}_1 = \mathbf{v} + \mathbf{b}_2$ where $\mathbf{v}_{\mathbf{x}} \in \operatorname{Ker}\pi_{*\mathbf{x}} := \{\mathbf{v} \in T_{\mathbf{x}}\mathcal{M} \mid \pi_{*\mathbf{x}}(\mathbf{v}) = \mathbf{0}\}, \forall \mathbf{x} \in \mathcal{M}$, then the corresponding SDE in Eq. (13) has the same projection in the quotient space. This implies that the components in $\operatorname{Ker}\pi_{*\mathbf{x}}$ are not really necessary.

For better characterization of the necessary component, we focus on the tangent space of $\mathcal{M}$ at $\mathbf{x}$. The tangent space $T_{\mathbf{x}}\mathcal{M}$ is a linear space with the same dimensionality as $\mathcal{M}$. Define the vertical space $\mathcal{V}_{\mathbf{x}} := \operatorname{Ker}\pi_{*\mathbf{x}}$ (G-dimensional) corresponding to the infinitesimal action of the group $\mathcal{G}$. Since $T_{\mathbf{x}}\mathcal{M}$ has an inner product (because $\mathcal{M}$ is a Riemannian manifold), we can define the horizontal space $\mathcal{H}_{\mathbf{x}} := (\operatorname{Ker}\pi_{*\mathbf{x}})^{\perp}$ as the orthogonal complement of $\mathcal{V}_{\mathbf{x}}$. Then any tangent vector in $T_{\mathbf{x}}\mathcal{M}$ has an orthonormal decomposition $\mathbf{v} = \mathbf{v}^{\mathcal{V}} + \mathbf{v}^{\mathcal{H}}$, where $\mathbf{v}^{\mathcal{V}}$, $\mathbf{v}^{\mathcal{H}}$ is the vertical and horizontal component respectively; see Fig. 2 for visualization. Thus $\mathbf{v}^{\mathcal{H}}$ is the necessary part of the vector field $\mathbf{v}$.

Thanks to the quotient structure, we can leverage a correspondence between the diffusion process on $\mathcal{M}$ and $\mathcal{Q}$. For a diffusion process $\mathbf{y}_t$, there exists a diffusion process $\tilde{\mathbf{x}}_t$ in $\mathcal{M}$ such that $\pi(\tilde{\mathbf{x}}_t) = \mathbf{y}_t$ and $\tilde{\mathbf{x}}_t$ only has horizontal movement, which is called the horizontal lift of $\mathbf{y}_t$ (see Appx. D.2 for formal definitions). The horizontal lift of $\mathbf{y}_t$ is given explicitly in the following theorem.

Theorem 2. *The horizontal lift of Eq. (14) has the following explicit expression:*

$$\mathrm{d}\tilde{\mathbf{x}}_t = \left(P_{\tilde{\mathbf{x}}_t}(\mathbf{b}_t(\tilde{\mathbf{x}}_t)) - \frac{\sigma_t^2}{2}\tilde{\mathbf{h}}(\tilde{\mathbf{x}}_t)\right)\mathrm{d}t + \sigma_t\mathrm{d}\tilde{\mathbf{w}}_t, \quad \tilde{\mathbf{x}}_0 \sim p_{\text{prior}}, \tag{8}$$

where $P_{\mathbf{x}}(\mathbf{v}) = \mathbf{v}^{\mathcal{H}}$ is the horizontal projection on the tangent space of $\mathcal{M}$, $\tilde{\mathbf{h}}$ is the horizontal lift of the mean curvature vector $\mathbf{h}$ in Eq. (14), $\tilde{\mathbf{w}}_t$ is the horizontal lift of the Wiener process on $\mathcal{Q}$.

See Appx. D.2 for the proof. Comparing the expression between Eq. (13) and Eq. (8), we can observe that the lifted process is not simply given by adding a horizontal projection $P_{\mathbf{x}}$ on each term of the SDE, and an additional term depending on the curvature of the quotient space arises. This

term arises in Eq. (14) and remains after the horizontal lift. The horizontal projection $P_{\mathbf{x}}$ and the mean curvature vector field can be calculated in specific cases, so Eq. (8) has explicit form when $\mathcal{Q}$ is specified.

As mentioned, Eq. (8) only has horizontal movements, in other words, it does not have any movement in the equivalent class. This process reduces unnecessary movement and helps to reduce sampling trajectory length. From this viewpoint, previous methods do not reduce these unnecessary movements, although they have the equivalent diffusion process in the quotient space. The formal results are summarized in the following corollary. See Appx. D.2 for proof.

Corollary 3. *$\tilde{\mathbf{x}}_1$ (defined by Eq. (8)) has the same distribution on $\mathcal{Q}$ with $\mathbf{x}_1$ (defined by Eq. (13)). When $\sigma_t = 0$, $\forall \mathbf{x}_0 \in \mathcal{M}$, Eq. (8) has shorter trajectory length than Eq. (13).*

3.2 Special Case: the Shape Space

The abstract results in the previous section give the direction for practical implementations. In this subsection, we focus on the special case of quotient space $\mathbb{R}^{3N}/\mathrm{SE}(3)$. First, we need to define the quotient structure in this case (Appx. C). Let $\mathbf{x} := (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(N)}) \in \mathbb{R}^{3N}$, with $\mathbf{x}^{(i)} \in \mathbb{R}^3$, denote a configuration (or point cloud) of N points in $\mathbb{R}^3$. Let $\mathcal{M} := \{\mathbf{x} \in \mathbb{R}^{3N} \mid \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}^{(i)} = \mathbf{0}\}$ be the center-of-mass (COM) subspace of $\mathbb{R}^{3N}$. Let $\mathrm{SO}(3)$ be the special orthonormal group and we construct $\mathbb{R}^{3N}/\mathrm{SE}(3)$ as $\mathcal{M}/\mathrm{SO}(3)$ because the translation-invariant distribution does not exist (Yim et al., 2023). An element of the $\mathrm{SO}(3)$ group is given by a 3-dimensional rotation matrix $g \in \mathbb{R}^{3\times 3}$. The natural action of g on $\mathbf{x}$ is defined as $g \cdot \mathbf{x} := \left(g\mathbf{x}^{(1)}, g\mathbf{x}^{(2)}, \cdots, g\mathbf{x}^{(N)}\right)$, *i.e.*, the rotation is acted on each point of the system. Under certain conditions, the quotient space $\mathcal{Q} := \mathcal{M}/\mathrm{SO}(3)$ is a smooth manifold. Now we can consider the correspondence between the diffusion process in $\mathcal{M}$ (Eq. (13)) and the its horizontal lift from the quotient space projection (Eq. (8)). The results are summarized in the following theorem.

Theorem 4. *Assume $\mathbf{x}_t$ is a diffusion process in the COM subspace $\mathcal{M} \subset \mathbb{R}^{3N}$, given by the following SDE: $\mathrm{d}\mathbf{x}_t = \mathbf{b}_t(\mathbf{x}_t)\,\mathrm{d}t + \sigma_t\,\mathrm{d}\mathbf{w}_t, \mathbf{x}_0 \sim p_{prior}$ where $\mathbf{b}_t(\mathbf{x}_t)$ is a $\mathrm{SO}(3)$-equivariant vector field,$\forall t \in [0, T]$, p_{prior} is the $\mathcal{G}$-invariant prior distribution, $\mathbf{w}_t$ is the standard Wiener process on COM. The horizontal lift of the process $\pi(\mathbf{x}_t)$ is :*

$$\mathrm{d}\tilde{\mathbf{x}}_t = \left(P_{\tilde{\mathbf{x}}_t}(\mathbf{b}_t(\tilde{\mathbf{x}}_t)) - \frac{\sigma_t^2}{2}\tilde{\mathbf{h}}(\tilde{\mathbf{x}}_t)\right)\mathrm{d}t + \sigma_t P_{\tilde{\mathbf{x}}_t}\,\mathrm{d}\mathbf{w}_t, \quad \tilde{\mathbf{x}}_0 \sim p_{\text{prior}}, \tag{9}$$

where the $P_{\mathbf{x}}$ is the horizontal projection operator at $\mathbf{x}$ and $\tilde{\mathbf{h}}(\mathbf{x})$ is the horizontal lift of mean curvature vector. The explicit expressions of P and $\tilde{\mathbf{h}}$ are shown as follows:

$$P_{\mathbf{x}}\mathbf{v} = \mathbf{v} - \mathcal{I}^{-1}\left(\frac{1}{N}\sum\nolimits_{i=1}^{N} \mathbf{x}^{(i)} \times \mathbf{v}^{(i)}\right) \times \mathbf{x}, \quad \forall \mathbf{v} \in T_{\mathbf{x}}\mathcal{M},$$

$$\tilde{\mathbf{h}}(\mathbf{x}) = -(\mathrm{tr}(\mathcal{I}^{-1})I - \mathcal{I}^{-1}) \cdot \mathbf{x}, \quad \text{where } \mathcal{I} = \left(\frac{1}{N}\sum\nolimits_{i=1}^{N} \|\mathbf{x}^{(i)}\|^2\mathbf{I} - \frac{1}{N}\sum\nolimits_{i=1}^{N} \mathbf{x}^{(i)}\mathbf{x}^{(i)\top}\right).$$

See Appx. D.3 for proof. From the results of Thm. 4, we can deduce that $\pi(\mathbf{x}_t)$ has the same marginal distribution with $\pi(\tilde{\mathbf{x}}_t)$ in Eq. (9) (Cor. 3). If we consider the generation process in Eq. (2) or Eq. (3) as $\mathbf{x}_t$, we can construct the corresponding horizontal process $\tilde{\mathbf{x}}_t$ that can generated the same target distribution on the quotient space. Motivated by this fact, we can improve the training and inference method of diffusion based generative models by leveraging the quotient structure.

3.3 Practical Implementations

Previous results describe how we can construct a diffusion process in the quotient space using the coordinates in the total space. If we have a diffusion process on the total space, we can construct the horizontal lift of its projection process, which has no vertical velocity along its trajectory and the two processes are the same on quotient space. This fact implies that the vertical components of the original diffusion process are not dispensable and enables us to design a more efficient training and sampling algorithm of the diffusion model based on the quotient structure. In practice, we often set the total space as the Euclidean space. Next, we show the training and sampling methods for the special case $p_{\text{prior}} = \mathcal{N}(\mathbf{0}, \mathbf{I})$, and the general case is shown in Appx. E.

Training objective. The diffusion model on the total space $\mathcal{M}$ is trained by the objective Eq. (5). Since the vertical components of the velocity are not strictly needed, we propose to supervise the model only on the horizontal components and allow arbitrary vertical output of the model. We lever-

age the horizontal projection operator $P_{\mathbf{x}}$ (Thm. 4) and construct the horizontal training objective:

$$\mathcal{L}(\theta) := \mathbb{E}_{p(t)} w(t) \mathbb{E}_{p(\mathbf{x}_1, \mathbf{x}_t)} \| P_{\mathbf{x}_t} \left(\mathbf{D}_\theta(\mathbf{x}_t, t) - \mathbf{x}_1 \right) \|^2. \tag{10}$$

We can see that $\mathbf{D}_\theta + \mathbf{v}^{\mathcal{V}}$ has the same loss value with $\mathbf{D}_\theta$, where $\mathbf{v}^{\mathcal{V}}$ is an arbitrary vertical vector.

ODE sampler. After the training stage, $P_{\mathbf{x}_t}(\mathbf{D}_\theta(\mathbf{x}_t, t))$ is an approximation of the ground truth denoiser in the horizontal subspace. For the ODE sampler, we simulate the horizontal lift of the projected ODE, which is given by $\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = P_{\mathbf{x}_t} \mathbf{v}_\theta(\mathbf{x}_t, t) \mathrm{d}t$, where $\mathbf{v}_\theta(\mathbf{x}_t, t)$ is given by Eq. (4). In practice, the ODE process is approximated by numerical solvers.

SDE sampler. For the stochastic sampler, the we need to simulate the horizontal lift of the projected original SDE in Eq. (3). According to Thm. 1 and Thm. 4, the lifted process is given by

$$\mathrm{d}\mathbf{x}_t = P_{\mathbf{x}_t} \left(\mathbf{v}_\theta(\mathbf{x}_t, t) + g_t \mathbf{s}_\theta(\mathbf{x}_t, t) \right) \mathrm{d}t + \gamma \eta_t \mathbf{h}(\mathbf{x}_t) \mathrm{d}t + \sqrt{2\gamma\eta_t} P_{\mathbf{x}_t} \mathrm{d}\mathbf{w}_t,$$

where $\mathbf{s}_\theta(\mathbf{x}_t, t) = -\frac{\mathbf{x}_t - \beta_t \mathbf{D}_\theta(\mathbf{x}_t, t)}{\bar{\alpha}_t^2}$ and we introduce the hyperparameter γ for protein generation following Geffner et al. (2025). The details are summarized in Algorithm 1 and 3.

3.4 Analysis on Existing Treatments for Symmetry

In this section, we make a detailed analysis on existing methods that handle symmetry, and verify the conclusions in Table 1. In contrast to our quotient-space diffusion, we find that they either have not fully leveraged the symmetry to reduce model-learning difficulty, or do not have a proper sampler.

Conventional equivariant diffusion models and data augmentation. A common treatment is by assigning equal probability to equivalent objects, resulting in an invariant target distribution $p(\mathbf{x}_1)$. This can be implemented by augmenting data samples by applying randomly chosen group actions, mimicking sampling from the invariant distribution, or using an invariant prior distribution and an equivariant architecture securing $\mathbf{D}_\theta(g \cdot \mathbf{x}, t) = g \cdot \mathbf{D}_\theta(\mathbf{x}, t)$. The training strategy is the same as modeling a general distribution in the original space following Eq. (5), and the standard samplers by Eqs. (2, 3) remain valid. For each value of $\mathbf{x}_t$, this objective asks the model to minimize the average of $\|\mathbf{D}_\theta(\mathbf{x}_t, t) - \mathbf{x}_1\|^2$ terms where $\mathbf{x}_1$ come from $p(\mathbf{x}_1|\mathbf{x}_t)$, so the optimal solution is the conditional expectation $\mathbb{E}[\mathbf{x}_1|\mathbf{x}_t]$.

Fig. 3 shows an example and reveals characteristics of the training strategy. The example considers generating the structure of a diatomic molecule, where the target distribution $p(\mathbf{x}_1)$ concentrates on a single structure $\mathbf{x}^\star$ up to a uniform random orientation (Left). For a given $\mathbf{x}_t$, samples of $p(\mathbf{x}_1|\mathbf{x}_t)$ are $\mathbf{x}^\star$ structures posed in orientations distributed around the orientation of $\mathbf{x}_t$ (Middle). Indeed, an $\mathbf{x}_1$ sample more closely oriented with $\mathbf{x}_t$ would have a higher probability to produce the given $\mathbf{x}_t$ in the diffusion process, so there is a specific orientation correspondence between the learning target $\mathbb{E}[\mathbf{x}_1|\mathbf{x}_t]$ and $\mathbf{x}_t$. So the model is still asked to learn a correspondence in the equivalent degrees of freedom (DOFs) (*i.e.*, rotation of the output), in contrast to the quotient-space case in Eq. (10) where the model is unconstrained in the vertical space (*i.e.*, tangent space of the rotation group). Moreover, the $\mathbf{x}_1$ samples are not all posed in the orientation of $\mathbf{x}_t$ because $\mathbf{x}^\star$ in other orientations can also generate this $\mathbf{x}_t$ through the diffusion process. So the model learns the correspondence in the equivalent DOFs from samples with a variance, leading to another aspect of learning difficulty.

GeoDiff alignment. To reduce the learning difficulty, some heuristic treatments are proposed based on alignment. The first representative alignment used in GeoDiff (Xu et al., 2022) uses the following training loss: $\mathbb{E}_{p(\mathbf{x}_1, \mathbf{x}_t)} \|\mathbf{D}_\theta(\mathbf{x}_t, t) - \mathcal{A}_{\mathbf{x}_t}(\mathbf{x}_1)\|^2$, where the alignment operation is defined as:

$$\mathcal{A}_{\mathbf{y}}(\mathbf{x}) := \mathrm{argmin}_{\mathbf{x}' \in \{g \cdot \mathbf{x} | g \in \mathcal{G}\}} d(\mathbf{x}', \mathbf{y}), \tag{11}$$

where $d(\cdot, \cdot)$ is the distance metric on $\mathcal{M}$. With an illustration in Fig. 3(Right), the learning task can be understood as that for a given value of $\mathbf{x}_t$, the model output needs to fit $\mathcal{A}_{\mathbf{x}_t}(\mathbf{x}_1)$ samples, which are all posed in the orientation of $\mathbf{x}_t$, and they all coincide with the $\mathbf{x}^\star$ structure in the orientation of $\mathbf{x}_t$. This supervises the model to the target $\mathbb{E}[\mathcal{A}_{\mathbf{x}_t}(\mathbf{x}_1)|\mathbf{x}_t]$ from samples with no variance in the equivalent DOFs (*i.e.*, rotation of the output), hence reduces certain learning difficulty. Nevertheless, this target still requires the model to learn a specific mapping in the equivalent DOFs, hence does not enjoy the learning advantage in the quotient-space case that relaxes the learning in the DOFs.

A caveat of this alignment approach is that a proper sampler needs to be developed, as the conventional samplers still require a model targeting $\mathbb{E}[\mathbf{x}_1|\mathbf{x}_t]$, which is different from $\mathbb{E}[\mathcal{A}_{\mathbf{x}_t}(\mathbf{x}_1)|\mathbf{x}_t]$.

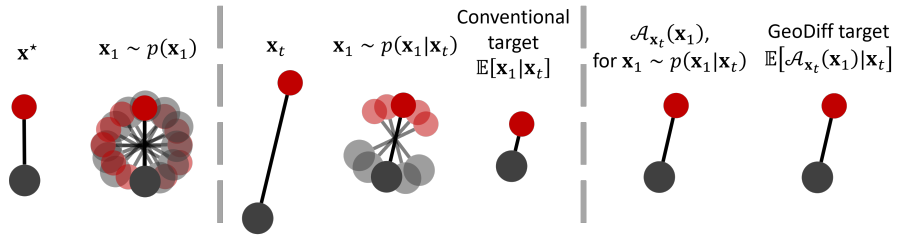

Figure 3: Illustration of denoising-model learning target using conventional training and using GeoDiff alignment. **(Left)** The example considers the structure distribution $p(\mathbf{x}_1)$ of a diatomic molecule, which concentrates on a single structure $\mathbf{x}^\star$ up to a uniform random orientation. **(Middle)** Given an $\mathbf{x}_t$ sample, the corresponding $\mathbf{x}_1$ samples distribute with a variance, and their expectation $\mathbb{E}[\mathbf{x}_1|\mathbf{x}_t]$ is the conventional learning target, which is *not* equivalent to $\mathbf{x}^\star$ (the bond is shorter). **(Right)** Given an $\mathbf{x}_t$ sample, all the $\mathbf{x}_1$ samples after alignment coincide with $\mathbf{x}^\star$ posed in the orientation of $\mathbf{x}_t$, which is also the learning target of GeoDiff $\mathbb{E}[\mathcal{A}_{\mathbf{x}_t}(\mathbf{x}_1)|\mathbf{x}_t]$.

Fig. 3 illustrates this difference: $\mathbb{E}[\mathbf{x}_1|\mathbf{x}_t]$ averages diversely oriented $\mathbf{x}^\star$ structures, resulting in a different shape than $\mathbf{x}^\star$ (the bond is shorter), while $\mathbb{E}[\mathcal{A}_{\mathbf{x}_t}(\mathbf{x}_1)|\mathbf{x}_t]$ is just $\mathbf{x}^\star$ in the orientation of $\mathbf{x}_t$.

AF3 alignment. Another alignment approach, which is used in Alphafold 3 (AF3) (Abramson et al., 2024), aligns the $\mathbf{x}_1$ samples towards the model output: $\mathbb{E}_{p(\mathbf{x}_1,\mathbf{x}_t)}\left\|\mathbf{D}_\theta(\mathbf{x}_t,t) - \mathcal{A}_{\mathbf{D}_{\bar\theta}(\mathbf{x}_t,t)}(\mathbf{x}_1)\right\|^2$, where $\bar\theta$ is treated constant in optimization. This loss function allows the model output to differ by an arbitrary group action (*e.g.*, rotation), hence removes the need to learn a specific target in the equivalent DOFs. Indeed, for an arbitrary group action $g_{\mathbf{x}_t,t}$, a new denoising model $g_{\mathbf{x}_t,t}\cdot\mathbf{D}_\theta(\mathbf{x}_t,t)$ achieves the same loss since $\|g_{\mathbf{x}_t,t}\cdot\mathbf{D}_\theta(\mathbf{x}_t,t) - \mathcal{A}_{g_{\mathbf{x}_t,t}\cdot\mathbf{D}_{\bar\theta}(\mathbf{x}_t,t)}(\mathbf{x}_1)\|^2 = \|g_{\mathbf{x}_t,t}\cdot\mathbf{D}_\theta(\mathbf{x}_t,t) - g_{\mathbf{x}_t,t}\cdot\mathcal{A}_{\mathbf{D}_{\bar\theta}(\mathbf{x}_t,t)}(\mathbf{x}_1)\|^2 = \|\mathbf{D}_\theta(\mathbf{x}_t,t) - \mathcal{A}_{\mathbf{D}_{\bar\theta}(\mathbf{x}_t,t)}(\mathbf{x}_1)\|^2$, where the last equality holds since the group preserves metric (Appx. C). Up to this DOF, the learning target is the same as GeoDiff's $\mathbb{E}[\mathcal{A}_{\mathbf{x}_t}(\mathbf{x}_1)|\mathbf{x}_t]$, since all the $\mathbf{x}_1$ samples are averaged after aligned with the same reference.

In the sampling process, the arbitrariness in the equivalent DOFs (*e.g.*, orientation) of the learned model $\mathbf{D}_\theta(\mathbf{x}_t,t)$ leads to an arbitrariness[3] in the vector field $\mathbf{v}_\theta(\mathbf{x}_t,t)$ through Eq. (4). Hence there is no guarantee of recovering the target distribution using conventional samplers. This problem is also noted by Boltz-1 (Wohlwend et al., 2025), which proposes to align the prediction $\mathbf{D}_\theta(\mathbf{x}_t,t)$ towards $\mathbf{x}_t$ in the sampling process. As the AF3 target is the same as GeoDiff's up to an arbitrary rotation, this amounts to using the GeoDiff model for sampling, which still cannot guarantee producing the target distribution as concluded above. These discussions are summarized in Table 1.

4 Experiments

In this section, we study the empirical performance of our quotient-space diffusion model. We carefully conduct several experiments covering different types of data, scales and scenarios. To evaluate our quotient space diffusion model framework for real-world applications, we focus on the molecule structure generation protein backbone design tasks, in which we consider the diffusion models on $\mathbb{R}^{3N}/\mathrm{SE}(3)$ (Sec. 3.2). The details of all experiments are shown in Appx. G.

4.1 Structure Generation for Small Molecules

Datasets. First, we evaluate our framework on the molecule structure generation task. In this scenario, our goal is to generate the 3D coordinates of a molecule given the graph structure of the molecule. We conduct the experiments on the GEOM datasets (Axelrod & Gomez-Bombarelli, 2022), which provides structure ensembles generated by metadynamics in CREST (Pracht et al., 2024) and we focus on the GEOM-QM9 and GEOM-DRUGS datasets. Following the data processing and splits from (Hassan et al., 2024), we use the random splits with train/validation/test of 243473/30433/1000 for GEOM-DRUGS and 106586/13323/1000 for GEOM-QM9. In addition, data with disconnect molecule graph are removed for GEOM-DRUGS (Hassan et al., 2024).

Setting. We primarily follow the setting in (Hassan et al., 2024). We use an equivariant graph transformer architecture from ET-Flow (Hassan et al., 2024) and set the Gaussian distribution as prior distribution on GEOM-QM9 and use the harmonic prior for GEOM-DRUGS (Volk et al., 2023). We fix the architecture as ET-Flow($\mathrm{SO}(3)$) for experiments on GEOM-QM9, and use the ET-

[3]This is not even an arbitrary group action (*e.g.*, rotation) since $\mathbf{x}_t$ does not vary together with the arbitrariness of $\mathbf{D}_\theta(\mathbf{x}_t,t)$.

Table 3: Molecule structure generation results on GEOM-DRUGS ($\delta = 0.75$Å). We use the ET-Flow(SO(3)) and ET-Flow(O(3)) architecture. We use the same sampling steps of 50 NFEs for fair comparison. Best results are marked in **bold**. Best results for the same architecture are underlined.

	Recall				Precision			
	Coverage ↑		AMR ↓		Coverage ↑		AMR ↓	
	mean	median	mean	median	mean	median	mean	median
GeoDiff	42.10	37.80	0.835	0.809	24.90	14.50	1.136	1.090
GeoMol	44.60	41.40	0.875	0.834	43.00	36.40	0.928	0.841
Torsional Diff.	72.70	80.00	0.582	0.565	55.20	56.90	0.778	0.729
MCF - S (13M)	79.4	87.5	0.512	0.492	57.4	57.6	0.761	0.715
MCF - B (62M)	84.0	91.5	0.427	0.402	64.0	66.2	0.667	0.605
MCF - L (242M)	**84.7**	**92.2**	**0.390**	**0.247**	66.8	71.3	0.618	0.530
ET-Flow (8.3M)	79.53	84.57	0.452	0.419	**74.38**	**81.04**	**0.541**	**0.470**
+ reproduction	78.94	84.24	0.489	0.472	66.24	70.42	0.651	0.595
+ Quotient-space diffusion	79.86	85.71	0.459	0.433	72.70	79.63	0.565	0.501
ET-Flow(SO(3)) (9.1M)	78.18	83.33	0.480	0.459	67.27	71.15	0.637	0.567
+ reproduction	74.91	80.90	0.541	0.515	60.33	62.71	0.724	0.665
+ Geodiff alignment	75.11	80.74	0.545	0.526	59.58	60.48	0.734	0.678
+ AF3 alignment	71.66	76.09	0.572	0.570	52.21	50.00	0.828	0.793
+ Quotient-space diffusion	78.50	84.20	0.477	0.455	67.35	71.42	0.635	0.563

Flow(O(3)), ET-Flow(SO(3)) architecture on the GEOM-DRUGS dataset. Following (Jing et al., 2022; Xu et al., 2022), we report the RMSD-based metrics, e.g. Coverage and Average Minimum RMSD (AMR) between the generated and ground truth structure ensembles.

Results. The results are presented in Table 2 and Table 3 for the GEOM-QM9 and GEOM-DRUGS datasets, respectively. As shown, our proposed quotient-space diffusion framework consistently outperforms prior methods and alignment techniques in terms of generation quality on both datasets. Our framework reduces learning difficulty by removing redundant components, enabling us to further improve the performance of the ET-Flow framework[4] on both datasets. On the GEOM-QM9 dataset, our quotient-space diffusion model framework surpasses strong baselines such as MCF (Wang et al., 2023) and the ET-Flow framework with other heuristic alignment methods among most of the RMSD-based metrics. On the GEOM-DRUGS dataset, our framework not only significantly surpasses the ET-Flow baseline with heuristic alignment methods, since these methods are incompatible with training, but also achieves competitive performance against the larger MCF-L (242M) model (Wang et al., 2023) on the Precision metrics.

Table 2: Molecule structure generation results on GEOM-QM9 ($\delta = 0.5$Å). We use the ET-Flow(SO(3)) architecture. We use the same sampling steps of 50 NFEs for fair comparison.

	Recall				Precision			
	Coverage ↑		AMR ↓		Coverage ↑		AMR ↓	
	mean	median	mean	median	mean	median	mean	median
CGCF	69.47	96.15	0.425	0.374	38.20	33.33	0.711	0.695
GeoDiff	76.50	**100.00**	0.297	0.229	50.00	33.50	1.524	0.510
GeoMol	91.50	**100.00**	0.225	0.193	87.60	**100.00**	0.270	0.241
Torsional Diff.	92.80	**100.00**	0.178	0.147	92.70	**100.00**	0.221	0.195
MCF	95.0	**100.00**	0.103	0.044	93.7	**100.00**	0.119	0.055
ET-Flow(SO(3))	95.98	**100.00**	0.076	0.030	92.10	**100.00**	0.110	0.047
+ Geodiff alignment	95.71	**100.00**	0.085	0.040	**95.20**	**100.00**	0.098	0.050
+ AF3 alignment	92.67	**100.00**	0.131	0.070	84.38	**100.00**	0.205	0.146
+ Quotient-space diffusion	**96.40**	**100.00**	**0.069**	**0.024**	93.30	**100.00**	**0.096**	**0.036**

4.2 Protein Backbone Design

Setting. To demonstrate the advantage of our quotient-space diffusion model for larger and more relevant molecules, we perform a comparative analysis on the task of protein structure generation against the state-of-the-art Proteína model (Geffner et al., 2025). We select their most efficient variant $\mathcal{M}_{\text{FS}}^{\text{small}}$, a 60M parameter transformer trained on the Foldseek AFDB clusters (D_{FS}) that forgoes triangle layers and pair representation updates, as a strong and relevant baseline. We train the quotient-space diffusion model from scratch using the identical architecture on the identical dataset. For evaluation, both our model and the officially released Proteína checkpoint are sampled using 400 steps with self-conditioning. We explore the designability-diversity trade-off by testing

[4]We reproduce the results using the released configurations: https://github.com/shenoynikhil/ETFlow. Due to changes in the data processing pipeline, our reproduced results do not exactly match those reported in the original paper.

Table 4: Performance comparison of the most efficient Proteína model against other baselines. The Proteína model is evaluated in two settings: sampling in the standard Euclidean space ($\mathbb{R}^{3N}$) and in our proposed quotient space ($\mathbb{R}^{3N}/\mathrm{SE}(3)$) for both ODE and SDE sampling. Best results are marked in **bold**.

Model	Designability(%)↑	FPSD vs.		fS	fJSD vs.	
		PDB↓	AFDB↓	(C/A/T)↑	PDB↓	AFDB↓
FrameDiff	65.4	194.2	258.1	2.46/5.78/23.35	1.04	1.42
FoldFlow (base)	96.6	601.5	566.2	1.06/1.79/9.72	3.18	3.10
FoldFlow (stoc.)	97.0	543.6	520.4	1.21/2.09/11.59	3.69	2.71
FoldFlow (OT)	97.2	431.4	414.1	1.35/3.10/13.62	2.90	2.32
FrameFlow	88.6	129.9	159.9	2.52/5.88/27.00	0.68	0.91
ESM3	22.0	933.9	855.4	3.19/6.71/17.73	1.53	0.98
Chroma	74.8	189.0	184.1	2.34/4.95/18.15	1.00	1.08
RFDiffusion	94.4	253.7	252.4	2.25/5.06/19.83	1.21	1.13
Proteus	94.2	225.7	226.2	2.26/5.46/16.22	1.41	1.37
Genie2	95.2	350.0	313.8	1.55/3.66/11.65	2.21	1.70
SDE Sampling						
Proteína $\mathcal{M}_{\mathrm{FS}}^{\mathrm{small}}, \gamma = 0.35$	96.0	386.5	378.2	1.77/4.97/17.78	2.17	1.73
+ Quotient-space diffusion	**97.6**	274.7	277.1	2.24/6.69/20.99	1.68	1.55
Proteína $\mathcal{M}_{\mathrm{FS}}^{\mathrm{small}}, \gamma = 0.45$	92.2	332.9	320.4	1.83/5.01/20.22	1.93	1.49
+ Quotient-space diffusion	92.6	244.5	246.3	2.24/6.68/23.47	1.43	1.28
Proteína $\mathcal{M}_{\mathrm{FS}}^{\mathrm{small}}, \gamma = 0.50$	89.2	306.2	290.8	1.86/4.92/21.15	1.81	1.36
+ Quotient-space diffusion	90.2	228.0	228.7	2.25/6.59/25.24	1.32	1.17
ODE Sampling						
Proteína $\mathcal{M}_{\mathrm{FS}}$	19.6	85.4	21.4	2.51/5.65/27.35	0.59	**0.09**
Proteína $\mathcal{M}_{\mathrm{FS}}^{\mathrm{small}}$	13.8	83.2	21.9	2.45/5.63/31.76	0.58	0.12
+ AF3 alignment	3.8	229.0	82.4	2.18/4.30/14.28	1.35	0.36
+ Quotient-space diffusion	15.6	**69.9**	**17.6**	**2.57/6.40/32.14**	**0.41**	0.11

a range of noise scales, $\gamma \in \{0.35, 0.45, 0.5\}$[5]. To faithfully evaluate the distributional metrics proposed in (Geffner et al., 2025), we utilize ODE sampling.

Results. The results in Table 4 highlight the superiority of our quotient space framework, which, unlike alignment-based strategies (adapted from AF3 and Boltz-1), provides a theoretical guarantee for sampling the correct target distribution. The alignment-based methods fail to recover this distribution, with performance metrics falling short of even data-augmented, semi-equivariant baselines. We attribute this failure to a fundamental incompatibility between their samplers and the learned density. Furthermore, our formulation effectively reduces learning difficulty by removing redundant spatial transformations, enabling the model to capture key structural features more efficiently than standard semi-equivariant baselines. This advantage of efficiency leads to significant results: our 60M parameter model not only surpasses its direct baseline across both SDE at all noise scales and ODE sampling setting, but also outperforms the much larger 200M $\mathcal{M}_{\mathrm{FS}}$ model on most key distributional metrics. This provides compelling evidence that a quotient space framework ensuring both sampling fidelity and learning efficiency is key to advancing generative protein models.

5 Conclusion

In this work, we formally construct a framework for building diffusion models on the quotient space over a group, in hope for a principled approach to handle symmetry in a generative task. We explicitly give the expression of the diffusion process on the quotient space, then also construct a corresponding diffusion process in the original space for easier implementation. The resulting training algorithm reduces learning difficulty by removing the need to predict the tangent vector in the direction along group action, and the resulting sampling process guarantees producing the target distribution while removes the unnecessary movement in the group-action direction. We instantiate the method in the case of $\mathbb{R}^{3N}/\mathrm{SE}(3)$ for molecular structure generation. Empirical results on structure sampling for small molecules from the GEOM-QM9 and GEOM-DRUGS datasets and protein backbone generation demonstrate the better generation quality and design success rate over existing conventional equivariant diffusion models and alignment-based approaches given equal or fewer training epochs, demonstrating the practical advantages from this principled framework to handling symmetry in diffusion models.

[5]Due to a known bug in a previous version of Foldseek (Daras et al., 2025, Appendix B), our comparative analysis in the main text is focused solely on the designability. More comprehensive metrics evaluating our self-sampled structures are provided in Table 6.

6 Ethics Statement

This work adheres to the ICLR Code of Ethics. Our study does not involve human subjects, personal data, or sensitive demographic information. All experiments are conducted on publicly available benchmark datasets, which are widely used in the machine learning community. No new data collection or human/animal experimentation was performed.

7 Reproducibility Statement

To facilitate the reproducibility of our research, we provide comprehensive details throughout the paper and its supplementary materials. We begin by establishing the necessary foundational knowledge in Sec. 2.1 and Appx. B. For all theoretical claims and proofs presented in the main text, we offer detailed step-by-step derivations in Appx. D. Our experiments are thoroughly documented; the datasets, training procedures, and evaluation protocols are carefully described in Sec. 4 and Appx. G. Upon acceptance of this paper, we commit to making our full codebase and all model checkpoints publicly available to ensure that the community can fully reproduce our results.

8 The use of Large Language models (LLMs)

In the preparation of this manuscript, LLMs were employed as a writing assistant to refine the language and improve the grammar. Furthermore, we utilized LLMs to assist in verifying our mathematical formulas for notational consistency. Following this process, all textual and mathematical content was meticulously reviewed, revised, and validated by the authors, who assume full responsibility for the final work presented.

References

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, pp. 1–3, 2024.

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

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.

Jan-Hendrik Bastek, WaiChing Sun, and Dennis Kochmann. Physics-informed diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tpYeermigp.

Fabrice Baudoin, Nizar Demni, and Jing Wang. *Stochastic areas, horizontal Brownian motions, and hypoelliptic heat kernels*. EMS Press, 2024.

Isaac Chavel. *Riemannian geometry: a modern introduction*. Number 108. Cambridge university press, 1995.

Ricky TQ Chen and Yaron Lipman. Flow matching on general geometries. *arXiv preprint arXiv:2302.03660*, 2023.

François Cornet, Federico Bergamin, Arghya Bhowmik, Juan Maria Garcia Lastra, Jes Frellsen, and Mikkel N Schmidt. Kinetic langevin diffusion for crystalline materials generation. *arXiv preprint arXiv:2507.03602*, 2025.

Giannis Daras, Jeffrey Ouyang-Zhang, Krithika Ravishankar, William Daspit, Costis Daskalakis, Qiang Liu, Adam Klivans, and Daniel J Diaz. Ambient proteins: Training diffusion models on low quality structures. *bioRxiv*, pp. 2025–07, 2025.

Valentin De Bortoli, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. Riemannian score-based generative modelling. *Advances in neural information processing systems*, 35:2406–2422, 2022.

Zach Evans, Cj Carr, Josiah Taylor, Scott H. Hawley, and Jordi Pons. Fast timing-conditioned latent audio diffusion. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12652–12665, 2024. URL https://proceedings.mlr.press/v235/evans24a.html.

Octavian Ganea, Lagnajit Pattanaik, Connor Coley, Regina Barzilay, Klavs Jensen, William Green, and Tommi Jaakkola. Geomol: Torsional geometric generation of molecular 3d conformer ensembles. *Advances in Neural Information Processing Systems*, 34:13757–13769, 2021.

Tomas Geffner, Kieran Didi, Zuobai Zhang, Danny Reidenbach, Zhonglin Cao, Jason Yim, Mario Geiger, Christian Dallago, Emine Kucukbenli, Arash Vahdat, and Karsten Kreis. Proteina: Scaling flow-based protein structure generative models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=TVQLu34bdw.

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.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

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

Emiel Hoogeboom, Víctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3D. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8867–8887. PMLR, 17–23 Jul 2022b.

Elton P Hsu. *Stochastic analysis on manifolds*. Number 38. American Mathematical Soc., 2002.

Chenqing Hua, Sitao Luan, Minkai Xu, Zhitao Ying, Jie Fu, Stefano Ermon, and Doina Precup. Mudiff: Unified diffusion for complete molecule generation. In *Learning on Graphs Conference*, pp. 33–1. PMLR, 2024.

Chin-Wei Huang, Milad Aghajohari, Joey Bose, Prakash Panangaden, and Aaron C Courville. Riemannian diffusion models. *Advances in Neural Information Processing Systems*, 35:2750–2761, 2022.

Bowen Jing, Gabriele Corso, Jeffrey Chang, Regina Barzilay, and Tommi Jaakkola. Torsional diffusion for molecular conformer generation. *Advances in Neural Information Processing Systems*, 35:24240–24253, 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.

Seongsu Kim, Nayoung Kim, Dongwoo Kim, and Sungsoo Ahn. High-order equivariant flow matching for density functional theory Hamiltonian prediction. *arXiv preprint arXiv:2505.18817*, 2025.

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

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan C. Catanzaro. DiffWave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=a-xFK8Ymz5J.

John M Lee. Smooth manifolds. In *Introduction to smooth manifolds*, pp. 1–29. Springer, 2003.

John M Lee. *Introduction to Riemannian manifolds*, volume 2. Springer, 2018.

Sarah Lewis, Tim Hempel, José Jiménez-Luna, Michael Gastegger, Yu Xie, Andrew Y. K. Foong, Victor García Satorras, Osama Abdin, Bastiaan S. Veeling, Iryna Zaporozhets, Yaoyi Chen, Soojung Yang, Adam E. Foster, Arne Schneuing, Jigyasa Nigam, Federico Barbero, Vincent Stimper, Andrew Campbell, Jason Yim, Marten Lienen, Yu Shi, Shuxin Zheng, Hannes Schulz, Usman Munir, Roberto Sordillo, Ryota Tomioka, Cecilia Clementi, and Frank Noé. Scalable emulation of protein equilibrium ensembles with generative deep learning. *Science*, 389(6761):eadv9817, 2025. doi: 10.1126/science.adv9817. URL https://www.science.org/doi/abs/10.1126/science.adv9817.

Xin Li, Wenqing Chu, Ye Wu, Weihang Yuan, Fanglong Liu, Qi Zhang, Fu Li, Haocheng Feng, Errui Ding, and Jingdong Wang. VideoGen: A reference–guided latent diffusion approach for high definition text-to-video generation. *arXiv preprint arXiv:2309.00398*, 2023. URL https://arxiv.org/abs/2309.00398.

Peijia Lin, Pin Chen, Rui Jiao, Qing Mo, Cen Jianhuan, Wenbing Huang, Yang Liu, Dan Huang, and Yutong Lu. Equivariant diffusion for crystal structure prediction. In *Forty-first International Conference on Machine Learning*, 2024.

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

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Philipp Pracht, Stefan Grimme, Christoph Bannwarth, Fabian Bohle, Sebastian Ehlert, Gereon Feldmann, Johannes Gorges, Marcel Müller, Tim Neudecker, Christoph Plett, et al. Crest—a program for the exploration of low-energy molecular chemical space. *The Journal of Chemical Physics*, 160(11), 2024.

Arne Schneuing, Charles Harris, Yuanqi Du, Kieran Didi, Arian Jamasb, Ilia Igashov, Weitao Du, Carla Gomes, Tom L Blundell, Pietro Lio, et al. Structure-based drug design with equivariant diffusion models. *Nature Computational Science*, 4(12):899–909, 2024.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Anton Thalmaier. Stochastic riemannian geometry. 2023.

Jos Torge, Charles Harris, Simon V Mathis, and Pietro Lio. Diffhopp: A graph diffusion model for novel drug design via scaffold hopping. *arXiv preprint arXiv:2308.07416*, 2023.

Amanda A Volk, Robert W Epps, Daniel T Yonemoto, Benjamin S Masters, Felix N Castellano, Kristofer G Reyes, and Milad Abolhasani. AlphaFlow: autonomous discovery and optimization of multi-step chemistry using a self-driven fluidic lab guided by reinforcement learning. *Nature Communications*, 14(1):1403, 2023.

Yuyang Wang, Ahmed A Elhag, Navdeep Jaitly, Joshua M Susskind, and Miguel Angel Bautista. Swallowing the bitter pill: Simplified scalable conformer generation. *arXiv preprint arXiv:2311.17932*, 2023.

Jeremy Wohlwend, Gabriele Corso, Saro Passaro, Noah Getz, Mateo Reveiz, Ken Leidal, Wojtek Swiderski, Liam Atkinson, Tally Portnoi, Itamar Chinn, et al. Boltz-1 democratizing biomolecular interaction modeling. *BioRxiv*, pp. 2024–11, 2025.

Lemeng Wu, Chengyue Gong, Xingchao Liu, Mao Ye, and Qiang Liu. Diffusion-based molecule generation with informative prior bridges. *Advances in neural information processing systems*, 35:36533–36545, 2022.

Minkai Xu, 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*, 2022.

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

Jason Yim, Brian L Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet, Regina Barzilay, and Tommi Jaakkola. SE(3) diffusion model with application to protein backbone generation. In *International Conference on Machine Learning*, pp. 40001–40039, 2023.

Shuxin Zheng, Jiyan He, Chang Liu, Yu Shi, Ziheng Lu, Weitao Feng, Fusong Ju, Jiaxi Wang, Jianwei Zhu, Yaosen Min, He Zhang, Shidi Tang, Hongxia Hao, Peiran Jin, Chi Chen, Frank Noé, Haiguang Liu, and Tie-Yan Liu. Predicting equilibrium distributions for molecular systems with deep learning. *Nature Machine Intelligence*, 2024. ISSN 2522-5839. doi: 10.1038/s42256-024-00837-3.

Yuchen Zhu, Tianrong Chen, Lingkai Kong, Evangelos A Theodorou, and Molei Tao. Trivialized momentum facilitates diffusion generative modeling on lie groups. *arXiv preprint arXiv:2405.16381*, 2024.

APPENDIX

The organization of the appendix are as follows. In Appx. A, we briefly discuss the related work relevant to our research. In Appx. B, we review some background knowledge of Riemannian geometry and stochastic calculus on the manifold. In Appx. C, we give the details of the Riemannian structures of the quotient space. In Appx. D, we give all the proofs of the theorems in the main text. In Appx. E, we show our methods for the general case. In Appx. F, we give some additional results and discussions. Finally, the details of the experiments are given in Appx. G.

A RELATED WORK

Diffusion models on Riemannian manifolds. As the quotient has the Riemannian manifold structure, several previous works construct the diffusion model on the Riemannian manifolds. De Bortoli et al. (2022) constructs diffusion models using different overlapping local coordinate systems of the manifold and requires geodesic random walk to simulate the forward process. Huang et al. (2022); Chen & Lipman (2023) construct diffusion models in an embedding space which allows a global representation but requires explicit geodesic formula of the manifold. Zhu et al. (2024) constructs the reverse of kinetic Langevin dynamics on a Lie group to perform generative modeling. In this framework, the Brownian motion term is added only on the tangent space of the Lie group, which is trivialized as an Euclidean space. In our quotient space case, the specialty with a quotient structure enables us to construct diffusion models using the coordinate systems of the total space without relying on an embedding of the quotient in the total space (unnecessarily an embedding space), which is more practical to implement yet still general.

Geometric diffusion models. To ensure physical symmetry in the generation process, a mainstream strategy integrates fundamental physical constraints, such as SE(3) equivariance, directly into the diffusion model's architecture. This approach, pioneered by models like EDM (Hoogeboom et al., 2022a), typically employs an EGNN to operate directly on atomic coordinates, using techniques like zero center of mass adjustments to guarantee translational invariance. This foundational concept was subsequently extended in several directions. For instance, the approach was adapted for Diffusion Bridges in models like EDM-Bridge (Wu et al., 2022) and for diffusion in a latent space in models like GeoLDM (Xu et al., 2023). These equivariant diffusion techniques have been successfully applied across a range of molecular tasks. For structure generation, models like GeoDiff (Xu et al., 2022) predict 3D structures from molecular graphs. In molecular optimization, methods such as DiffHopp (Torge et al., 2023) refine existing molecules to enhance desired properties. For de novo design, a key advancement has been to combine discrete diffusion models (D3PM) (Austin et al., 2021) for 2D topology with continuous equivariant diffusion for 3D geometry, enabling joint generation as seen in models like DiffSBDD (Schneuing et al., 2024) and MUDiff (Hua et al., 2024). A similar problem is also considered in crystalline structure generation, where the intrinsic periodic translation symmetry is crucial for generative modeling. Lin et al. (2024) highlighted the intrinsic periodic translation symmetry that has been omitted for a long time in the field of periodic crystalline structure generation. The work designed a modified diffusion process that induces a transition kernel that is invariant under periodic translation. The resulting optimization problem, while keeping the simplicity of no data augmentation, leads to a learning target for the score model that is invariant under periodic translation. Cornet et al. (2025) proposes a novel method that generalizes the Trivialized Diffusion Model framework for fractional coordinates to model the intrinsic periodic translation symmetry using flat coordinates. The proposed method considers the process with the velocity restricted to the mean-free linear subspace. Although considering different generation tasks, both of these works have a similar motivation to reduce the learning difficulty of the model using the intrinsic symmetry of the data distribution.

Learning with alignment To reduce learning difficulty, some heuristic treatments (learning with alignment) have been proposed to reduce the degrees of freedom corresponding to the symmetry group action. The alignment strategy used in GeoDiff (Xu et al., 2022) aligns the target structure with the noisy input by finding an optimal rigid transformation that minimizes the distance between them. Another approach, used in AlphaFold 3 (AF3) (Abramson et al., 2024), aligns the target samples towards the model's output. However, such alignment-based training frameworks can be incompatible with the sampling process and lack a mathematical guarantee for recovering the correct

target distribution. Boltz-1 (Wohlwend et al., 2025), an open-source version of AF3, in an attempt to improve performance, introduces an input-alignment step as a sampling technique.

B Background in Riemannian Geometry and Stochastic Calculus

B.1 Riemannian Geometry

In this section, we review some background on differential geometry and Riemannian geometry. For a systematic treatment of the subject, please refer to standard textbooks Lee (2003; 2018).

First, we give the formal definition of the smooth manifold. A manifold is a general topological space that locally has a Euclidean structure.

Definition 5. An M**-dimensional topological manifold** is a topological space $\mathcal{M}$ such that:

- $\mathcal{M}$ is locally Euclidean, i.e. locally homeomorphic to $\mathbb{R}^M$. Formally, $\forall p \in \mathcal{M}$, there exists an open neighborhood $p \in \mathcal{U} \subset \mathcal{M}$ that is homeomorphic to some open set $\mathcal{V} \subset \mathcal{M}$. We call the homeomorphism $\phi : \mathcal{U} \to \mathcal{V} \subset \mathbb{R}^M$ a **coordinate system** or a chart.
- $\mathcal{M}$ is a Hausdorff topological space.
- $\mathcal{M}$ has a countable basis for its topology.

A smooth manifold is a topological manifold with an additional smooth structure, which is defined as follows.

Definition 6. A smooth structure on a M-dimensional topological space $\mathcal{M}$ is a collection of coordinate systems $C = \{(\mathcal{U}_\alpha, \phi_\alpha) : \alpha \in A\}$ which satisfies the following properties:

- The collection C covers $\mathcal{M}$: $\bigcup_{\alpha \in A} \mathcal{U}_\alpha = \mathcal{M}$;
- For any $\alpha, \beta \in A$, the transition function $\phi_\alpha \circ \phi_\beta^{-1}$ is a smooth map;
- C is a maximal collection, i.e. if $(\mathcal{U}, \phi)$ is a coordinate system such that for all $\alpha \in A$ that the maps $\phi \circ \phi_\alpha^{-1}$ and $\phi_\alpha \circ \phi^{-1}$ are smooth, then $(\mathcal{U}, \phi) \in \mathcal{C}$.

The pair $(\mathcal{M}, C)$ is called a **smooth manifold** of dimension M.

With the smooth structure, we can define a smooth function on the manifold and a smooth mapping between smooth manifolds.

Definition 7. Let $\mathcal{M}, \mathcal{N}$ be smooth manifolds with dimensions M, N respectively.

- A function $f : \mathcal{M} \to \mathbb{R}$ is called a **smooth function** if $f \circ \phi^{-1} : \phi^{-1}(\mathcal{U}) \to \mathbb{R}$ is smooth on $\phi^{-1}(\mathcal{U}) \subset \mathbb{R}^m$ for all smooth coordinate systems $(\mathcal{U}, \phi)$ of $\mathcal{M}$. Denote all the smooth functions on $\mathcal{M}$ as $C^\infty(\mathcal{M})$.
- A map $\Phi : \mathcal{M} \to \mathcal{N}$ is called a **smooth map** if $\psi \circ \Phi \circ \phi^{-1} : \phi^{-1}(\mathcal{U}) \to \psi(\mathcal{V})$ is smooth for all smooth coordinate systems $(\mathcal{U}, \phi)$ of $\mathcal{M}$ and $(\mathcal{V}, \psi)$.

A smooth map $\Phi : \mathcal{M} \to \mathcal{N}$ which is invertible and whose inverse is smooth is called a diffeomorphism. In this case we say that $\mathcal{M}$ and $\mathcal{N}$ are diffeomorphic manifolds.

To define movement on a smooth manifold $\mathcal{M}$, we need to define tangent vectors on the manifold.

Definition 8. Let $\mathcal{M}$ be a smooth manifold, and $p \in \mathcal{M}$ is a point. A linear map $\mathbf{v}_p : C^\infty(\mathcal{M}) \to \mathbb{R}$ is called a derivative at p if it satisfies

$$\mathbf{v}_p(fg) = f(p)\mathbf{v}_p(g) + g(p)\mathbf{v}_p(f), \quad \forall f, g \in C^\infty(\mathcal{M}).$$

The set of all the derivations of $C^\infty(\mathcal{M})$ in p, denoted by $T_p\mathcal{M}$, is a vector space called the **tangent space** to $\mathcal{M}$ at p. An element of $T_p\mathcal{M}$ is called a **tangent vector** at p.

The **tangent bundle** $T\mathcal{M}$ is the union of the tangent spaces of each points, i.e. $T\mathcal{M} := \bigsqcup_{p \in \mathcal{M}} T_p\mathcal{M}$. Similar to the total derivative of the smooth map in Euclidean space, the differential of a smooth map between smooth manifolds is a linear map between tangent spaces.

Definition 9. Let $\mathcal{M}, \mathcal{N}$ be smooth manifolds and $F : \mathcal{M} \to \mathcal{N}$ be a smooth map. The **differential of** F **at** $p \in \mathcal{M}$, denoted by $F_{p*} : T_p\mathcal{M} \to T_{F(p)}\mathcal{N}$, is defined as

$$F_{p*}(\mathbf{v}_p)f = \mathbf{v}_p(f \circ F), \quad \forall f \in C^\infty(\mathcal{N}), \mathbf{v}_p \in T_p\mathcal{M}.$$

A **vector field** $\mathbf{v}$ on a smooth manifold $\mathcal{M}$ is a correspondence that associates to each point $p \in \mathcal{M}$ a vector $\mathbf{v}_p \in T_p\mathcal{M}$. The vector field is smooth if the mapping $\mathbf{v} : \mathcal{M} \to T\mathcal{M}$ is smooth. Denote all the smooth vector fields on $\mathcal{M}$ by $\mathscr{X}(\mathcal{M})$. With the definition of a vector field, we can define the solution of ordinary differential equation (ODE) on the manifold. The idea is similar to the definition in Euclidean space, the solution of the ODE is a curve whose velocity at each point is the same as the vector field.

Definition 10. Let $\mathbf{v}$ be a smooth vector field on the smooth manifold $\mathcal{M}$. An **integral curve of** $\mathbf{v}$ is a differentiable curve $\gamma : [0, T] \to \mathcal{M}$ whose velocity at each point is equal to the value of $\mathbf{v}$ at that point:

$$\gamma'(t) = \mathbf{v}_{\gamma(t)}, \quad \forall t \in [0, T].$$

Let $T_p^*\mathcal{M}$ be the dual space of $T_p\mathcal{M}$, which is called the cotangent space of $\mathcal{M}$ at p. The **cotangent bundle** $T^*\mathcal{M}$ is the union of the cotangent space of each points, i.e. $T^*\mathcal{M} := \bigsqcup_{p\in\mathcal{M}} T_p^*\mathcal{M}$.

Definition 11. A **1-form** Θ on smooth manifold $\mathcal{M}$ is a correspondence that associates to each point $p \in \mathcal{M}$ a covector $\Theta_p \in T_p^*\mathcal{M}$. The 1-form is smooth if the mapping $\Theta : \mathcal{M} \to T^*\mathcal{M}$ is smooth.

With the definition of a smooth manifold, we can define a continuous group with good properties.

Definition 12. A **Lie group** is a smooth manifold $\mathcal{G}$ that is also a group with the property that the multiplication map $\mathcal{G} \times \mathcal{G} \to \mathcal{G}, (g, h) \mapsto g \cdot h$ and the inversion map $\mathcal{G} \to \mathcal{G}, g \mapsto g^{-1}$ are both smooth.

Define the left multiplication mapping $L_g(h) = gh$. A vector field $\mathbf{v}$ on $\mathcal{G}$ is said to be left-invariant if it's invariant under all left multiplications, i.e. $(L_{g*})_{g'}(\mathbf{v}_{g'}) = \mathbf{v}_{gg'}$.

Definition 13. A Lie algebra is a real vector space $\mathfrak{g}$ endowed with a map called the bracket $[\cdot, \cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ that satisfies the following properties for all $X, Y, Z \in \mathfrak{g}$:

- Bilinearity: $\forall a, b \in \mathbb{R}$,
$$[aX + bY, Z] = a[X, Z] + b[Y, Z], [Z, aX + bY] = a[Z, X] + b[Z, Y];$$
- Antisymmetry: $[X, Y] = -[Y, X]$;
- Jacobi Identity: $[X, [Y, Z]] + [Y, [Z, X]] + [Z, [X, Y]] = 0$.

The Lie algebra of all smooth left-invariant vector fields on a Lie group $\mathcal{G}$ is called the **Lie algebra of** $\mathcal{G}$, which has the same dimension with $\mathcal{G}$.

Example 14. The Lie algebra of the group $\mathrm{SO}(3)$, denoted by $\mathfrak{so}(3)$, is given by all the 3-dimensional antisymmetric matrices $\mathfrak{so}(3) = \{A \in \mathbb{R}^{3\times3} | A + A^T = 0\}$.

Smooth manifold is a topological structure. If we want to define the "length of the velocity" and distance between two points on the manifold, a metric on the tangent space is required. Such a metric endows the metric with an additional geometry structure. The formal definitions are as follows.

Definition 15. A **Riemannian metric** on a smooth manifold is a correspondence which associates to each point p of $\mathcal{M}$ an inner product $\langle \cdot, \cdot \rangle_p^{\mathcal{M}}$ that varies smoothly on $\mathcal{M}$. In other words, for any two smooth vector fields $\mathbf{u}, \mathbf{v}$, $\langle \mathbf{u}, \mathbf{v} \rangle^{\mathcal{M}}$ is a smooth function on $\mathcal{M}$. A smooth manifold with a given Riemannian metric is called a **Riemannian manifold**.

To define the "difference" between tangent space at different points, we need to introduce a concept called affine connection.

Definition 16. An **affine connection** ∇ on a Riemannian manifold is a mapping

$$\nabla : \mathscr{X}(\mathcal{M}) \times \mathscr{X}(\mathcal{M}) \to \mathscr{X}(\mathcal{M})$$

which is denoted by $(\mathbf{u}, \mathbf{v}) \to \nabla_{\mathbf{u}}\mathbf{v}$ which satisfies the following properties:

- $\nabla_{\mathbf{u}}\mathbf{v}$ is linear over $C^\infty(\mathcal{M})$ in $\mathbf{u}$: $\forall f_1, f_2 \in C^\infty(\mathcal{M})$ and $\mathbf{u}_1, \mathbf{u}_2 \in \mathscr{X}(\mathcal{M})$,
$$\nabla_{f_1\mathbf{u}_1+f_2\mathbf{u}_2}\mathbf{v} = f_1\nabla_{\mathbf{u}_1}\mathbf{v} + f_2\nabla_{\mathbf{u}_2}\mathbf{v};$$
- $\nabla_{\mathbf{u}}\mathbf{v}$ is linear over $\mathbb{R}$ in $\mathbf{v}$: $\forall a_1, a_2 \in \mathbb{R}$ and $\mathbf{v}_1, \mathbf{v}_2 \in \mathscr{X}(\mathcal{M})$,
$$\nabla_{\mathbf{u}_1}(a_1\mathbf{v}_1 + a_2\mathbf{v}_2) = a_1\nabla_{\mathbf{u}}\mathbf{v}_1 + a_2\nabla_{\mathbf{u}}\mathbf{v}_2;$$
- ∇ satisfies the following product rule: $\forall f \in C^\infty(\mathcal{M})$,
$$\nabla_{\mathbf{u}}(f\mathbf{v}) = f\nabla_{\mathbf{u}}\mathbf{v} + (\mathbf{u}f)\mathbf{v}.$$

A connection is called the **Levi-Civita connection** if satisfies the following additional properties:

- ∇ is compatible with metric: $\nabla_{\mathbf{u}}\langle\mathbf{v}_1, \mathbf{v}_2\rangle = \langle\nabla_{\mathbf{u}}\mathbf{v}_1, \mathbf{v}_2\rangle + \langle\mathbf{v}_1, \nabla_{\mathbf{u}}\mathbf{v}_2\rangle$;
- ∇ is torsion-free: $\nabla_{\mathbf{u}}\mathbf{v} - \nabla_{\mathbf{v}}\mathbf{u} = \mathbf{u}(\mathbf{v}(\cdot)) - \mathbf{v}(\mathbf{u}(\cdot))$.

The Levi-Civita connection is the connection with nice properties. Its existence and uniqueness is a fundamental result of Riemannian geometry.

Theorem 17. *(Fundamental Theorem of Riemannian Geometry (Lee, 2018, Thm. 5.10)) Assume $(\mathcal{M}, \langle\cdot,\cdot\rangle^{\mathcal{M}})$ is a Riemannian manifold. Then there exists a unique Levi-Civita connection.*

As the end of this subsection, we introduce the Laplace-Beltrami operator on the manifold, which is used to define the Wiener process on the manifold.

Definition 18. Let ∇ be the Levi-Civita connection on $\mathcal{M}$. The Hessian of $f \in C^\infty(\mathcal{M})$ is defined by
$$\mathrm{Hess}(f)(\mathbf{u}, \mathbf{v}) := \mathbf{v}(\mathbf{u}(f)) - (\nabla_{\mathbf{v}}\mathbf{u})f, \quad \forall \mathbf{u}, \mathbf{v} \in \mathscr{X}(\mathcal{M}).$$
The Laplace-Beltrami operator $\Delta^{\mathcal{M}}$ is defined as the trace of Hessian. In other words, $\Delta^{\mathcal{M}} f := \sum_{i=1}^{M} \mathrm{Hess}(e_i, e_i)$ where $\{e_1, ..., e_M\}$ is some orthonormal basis for $T_x\mathcal{M}$.

B.2 Stochastic Calculus on a Manifold

With the Riemannian structure defined in the previous section, we can consider the definition of stochastic differential equations (SDE) and diffusion processes on the manifold. For a systematic treatment of the subject, please refer to standard textbooks Hsu (2002); Thalmaier (2023). First, we recall the definition of SDE and diffusion process in Euclidean space.

Definition 19. (Generator of a Process) The infinitesimal generator $\mathcal{A}_t$ of a stochastic process $(\mathbf{x}_t)$ for a function $\phi(x)$ is
$$\mathcal{L}_t\phi(x) = \lim_{s\to 0^+} \frac{\mathbb{E}[\phi(\mathbf{x}_{t+s})|\mathbf{x}_t = x] - \phi(x)}{s},$$
where ϕ is a suitably regular function. For an Itô process defined as the solution to the SDE $\mathrm{d}\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)\mathrm{d}t + \Sigma(\mathbf{x}_t, t)\mathrm{d}\mathbf{w}_t$, the generator is
$$\mathcal{L}_t = \sum_{i=1}^{d} \mathbf{f}^i(x,t)\frac{\partial}{\partial x_i} + \frac{1}{2}\sum_{i,j=1}^{d} \left(\Sigma(\mathbf{x}_t, t)\Sigma(\mathbf{x}_t, t)^\top\right)_{i,j} \frac{\partial^2}{\partial x_i \partial x_j}.$$

On the other hand, the diffusion process can also be defined by its generator.

Definition 20. A d-dimensional stochastic process $\mathbf{x}_t$ with continuous sample path defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is called a diffusion process generated by a smooth second-order elliptic operator $\mathcal{L}_t$ if the following hold: $\forall f \in C^\infty(\mathbb{R}^d)$, the process
$$M_t^f = f(\mathbf{x}_t) - f(\mathbf{x}_0) - \int_0^t \mathcal{L}_s f(\mathbf{x}_s)\,\mathrm{d}s$$
is a $\mathcal{F}_t$-martingale.

To generalize the definition of SDE to a Riemannian manifold $\mathcal{M}$, we need to define the second-order differential operator on the manifold. Let $\mathcal{M}$ be an M-dimensional Riemannian manifold. A

second order partial differential operator (PDO) on $\mathcal{M}$ is of the form

$$\mathcal{L} = \mathbf{v}_0 + \sum_{i=1}^{r} \mathbf{v}_i^2, \quad \text{where} \quad \mathbf{v}_i \in \mathscr{X}(\mathcal{M}), r \in \mathbb{N}^+.$$

The square of a vector field is understand by the decomposition of derivatives, i.e.

$$\mathbf{v}_i^2(f) = \mathbf{v}_i(\mathbf{v}_i(f)), \quad \forall f \in C^\infty(\mathcal{M}).$$

$\mathbf{v}$ can also be generalized to a time-dependent vector field. Now we can define the diffusion process on the manifold.

Definition 21. (Thalmaier, 2023, Def. 1.1.3) Let $(\Omega, \mathcal{F}, \mathbb{P}; (\mathcal{F})_{t\geq 0})$ be a probability space equipped with increasing sequence of sub-σ-algebra $\mathcal{F}_y \subset \mathcal{F}$. An adapted continuous process $\mathbf{x}_t$ taking values in $\mathcal{M}$, is called $\mathcal{L}_t$-diffusion if for all test functions $f \in C_c^\infty(\mathcal{M})$, the process

$$N_t^f := f(\mathbf{x}_t) - f(\mathbf{x}_0) - \int_0^t (\mathcal{L}_s f)(\mathbf{x}_s)\mathrm{d}s, \quad t \geq 0,$$

is a martingale, i.e. $\mathbb{E}[N_t^f - N_s^f \mid \mathcal{F}_s] = 0, \quad \forall s \leq t.$

For a special case, we can define the Wiener process on the Riemannian manifold $\mathcal{M}$.

Definition 22. A Wiener process $\mathbf{w}_t$ on $\mathcal{M}$ is a diffusion process with generator $\frac{1}{2}\Delta^{\mathcal{M}}$, where $\Delta^{\mathcal{M}}$ is the Laplace-Beltrami operator of $\mathcal{M}$, i.e. $\mathbf{w}_t$ is a continuous stochastic process on $\mathcal{M}$ such that for any $f \in C^\infty(\mathcal{M})$,

$$f(\mathbf{x}_t) - \frac{1}{2}\int_0^t \Delta^{\mathcal{M}} f(\mathbf{w}_t)\mathrm{d}s, 0 \leq t < e,$$

is a local martingale, where e is the lifetime of $\mathbf{w}_t$ on $\mathcal{M}$.

For stochastic differential geometry, the Stratonovitch integral is more useful than the Itô Integral, because it satisfies the ordinary chain rule of calculus. This property enables a clear correspondence between the diffusion process under a diffeomorphism between Riemannian manifolds. Next, we give the definition of the Stratonovitch integral on the Euclidean space and its generalization to Riemannian manifolds.

Definition 23. For continuous real-valued semimartingales $\mathbf{x}$ and $\mathbf{y}$, let $\mathbf{x} \circ \mathbf{y} := \mathbf{x}\mathrm{d}\mathbf{y} + \frac{1}{2}\mathrm{d}[\mathbf{x}, \mathbf{y}]$ be the Stratonovitch differential. Here $\mathbf{x}\mathrm{d}\mathbf{y}$ is the usual Itô differential and $\mathrm{d}[\mathbf{x}, \mathbf{y}] = \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y}$ is the quadratic covariation of $\mathbf{x}$ and $\mathbf{y}$. The integral

$$\int_0^t \mathbf{x} \circ \mathbf{y} = \int_0^t \mathbf{x}\mathrm{d}\mathbf{y} + \frac{1}{2}[\mathbf{x}, \mathbf{y}]_t$$

is called Stratonovitch integral of $\mathbf{x}$ with respect to $\mathbf{y}$. The Stratonovitch integral satisfies the following properties:

- Associativity: $\mathbf{x} \circ (\mathbf{y} \circ \mathrm{d}\mathbf{z}) = (\mathbf{x}\mathbf{y}) \circ \mathrm{d}\mathbf{z}$;
- Product rule: $\mathrm{d}(\mathbf{x}\mathbf{y}) = \mathbf{x} \circ \mathrm{d}\mathbf{y} + \mathbf{y} \circ \mathrm{d}\mathbf{x}$.

Proposition 24. *(Itô-Stratonovitch formula (Thalmaier, 2023, Prop. 1.2.10)). Let $\mathbf{x}$ be a continuous $\mathbb{R}^d$-valued semimartingale and $f \in C^\infty(\mathbb{R}^d)$. Then $\langle \nabla f(\mathbf{x}), \circ \mathrm{d}\mathbf{x} \rangle$.*

Proposition 25. *(Thalmaier, 2023, Prop. 1.2.11) Solutions to the Stratonovitch SDE*

$$\mathrm{d}\mathbf{x}_t = \mathbf{b}(\mathbf{x}_t, t)\mathrm{d}t + \Sigma(\mathbf{x}_t, t) \circ \mathrm{d}\mathbf{w}_t$$

define $\mathcal{L}_t$-diffusions for the operator

$$\mathcal{L}_t = \mathbf{v}_0 + \frac{1}{2}\sum_{i=1}^{d} \mathbf{v}_i^2, \quad \textit{where} \quad \mathbf{v}_0 = \sum_{i=1}^{d} \mathbf{b}^i \frac{\partial}{\partial x_i}, \quad \mathbf{v}_k = \sum_{i=1}^{d} \Sigma_{ik} \frac{\partial}{\partial x_i}.$$

Now we can generalize the definition of SDE to the Riemannian manifold case. A SDE on manifold $\mathcal{M}$ is defined by vector fields $\mathbf{v}_0, \mathbf{v}_1, ..., \mathbf{v}_d$ on $\mathcal{M}$. Let $\mathbf{w}$ be the $\mathbb{R}^d$-valued Wiener process and $\mathbf{x}_0$ be a $\mathcal{M}$-valued random variable serving as the initial value of the solution. The equation is

symbolically written as

$$\mathrm{d}\mathbf{x}_t = \mathbf{v}_0(\mathbf{x}_t, t)\mathrm{d}t + \sum_{i=1}^{d} \mathbf{v}_i(\mathbf{x}_t, t) \circ \mathrm{d}\mathbf{w}_i(t). \tag{12}$$

Definition 26. An $\mathcal{M}$-valued semimartingale $\mathbf{x}$ defined up to a stopping time τ is a solution of SDE Eq. (12) up to τ if for all $f \in C^\infty(\mathcal{M})$,

$$f(\mathbf{x}_t) = f(\mathbf{x}_0) + \int_0^t \left(\mathbf{v}_0(f)(\mathbf{x}_s, s)\mathrm{d}s + \sum_{i=1}^{d} \mathbf{v}_i(f)(\mathbf{x}_s, s) \circ \mathrm{d}\mathbf{x}_i \right), \quad 0 \le t < \tau.$$

Proposition 27. *(Thalmaier, 2023, Cor. 1.2.19) Let $\mathcal{L}_t = \mathbf{v}_0 + \frac{1}{2}\sum_{i=1}^{d} \mathbf{v}_i^2$, and $\mathbf{x}_t$ be the solution of SDE Eq. (12). Then for all $f \in C^\infty(\mathcal{M})$,*

$$N_t^f := f(\mathbf{x}_t) - f(\mathbf{x}_0) - \int_0^t (\mathcal{L}_s f)(\mathbf{x}_s)\mathrm{d}s, \quad t \ge 0,$$

is a martingale. In other words, the solution of SDE Eq. (12) is a $\mathcal{L}_t$ diffusion to the operator $\mathcal{L}_t = \mathbf{v}_0 + \frac{1}{2}\sum_{i=1}^{d} \mathbf{v}_i^2$.

C Construction of Quotient Space

In this section, we give the rigorous construction of the quotient space and endow it with the manifold structure. Please refer to the standard textbooks Lee (2018) for the systematic treatments. Assume that the total space $\mathcal{M}$ is a Riemannian manifold and $\mathcal{G}$ is a compact Lie group. First we give the formal definition of the group action.

Definition 28. Let $\mathcal{G}$ be a group and $\mathcal{M}$ is a Riemannian manifold. A left action of $\mathcal{G}$ on $\mathcal{M}$ is a map $\mathcal{G} \times \mathcal{M} \to \mathcal{M}, (g, \mathbf{x}) \mapsto g \cdot \mathbf{x}$, satisfying $g_1 \cdot (g_2 \cdot \mathbf{x}) = (g_1 g_2) \cdot \mathbf{x}$ and $id \cdot \mathbf{x} = \mathbf{x}, \forall g_1, g_2 \in \mathcal{G}, \mathbf{x} \in \mathcal{M}$. An action is smooth if its defining map $\mathcal{G} \times \mathcal{M} \to \mathcal{M}$ is smooth.

Definition 29. A smooth action is said to be free if $g \cdot \mathbf{x} = \mathbf{x}$ for some $g \in \mathcal{G}, \mathbf{x} \in \mathcal{M}$, then $g = e$. A smooth action is said to be proper if the map $\mathcal{G} \times \mathcal{M} \to \mathcal{M} \times \mathcal{M}, (g, \mathbf{x}) \mapsto (g \cdot \mathbf{x}, \mathbf{x})$ is a proper map, meaning that the preimage of every compact set is compact. The action is said to be an isometric action if the map $L_g : \mathcal{M} \to \mathcal{M}, \mathbf{x} \mapsto g \cdot \mathbf{x}$ is an isometry for any $g \in \mathcal{G}$, i.e. $\langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{x}} = \langle L_{g*}\mathbf{u}, L_{g*}\mathbf{v} \rangle_{g \cdot \mathbf{x}}$.

The proper property is a technical assumption to ensure the topological structure of the quotient space. The following technical characterization is usually the easiest way to prove that a given action is proper.

Proposition 30. *(Lee, 2018, Prop. C.15) Assume $\mathcal{G}$ is a Lie group acting smoothly on the smooth manifold $\mathcal{M}$. The action is proper if and only if the following condition is satisfied: if (p_i) is a sequence in $\mathcal{M}$ and (g_i) is a sequence in $\mathcal{G}$ such that both (p_i) and $(g_i \cdot p_i)$ converge, then a subsequence of (g_i) converges. Thus every smooth action by a compact Lie group on a smooth manifold is proper.*

We define a equivalence relation $\sim$ on $\mathcal{M}$ by $\mathbf{x}_1 \sim \mathbf{x}_2$ if and only if $\exists g \in \mathcal{G}, \mathbf{x}_1 = g \cdot \mathbf{x}_2$. The quotient space $\mathcal{Q} := \mathcal{M}/\sim$ is defined as the set of equivalence classes under the relation $\sim$. The quotient space inherits the Riemannian structure of the total space under certain conditions.

Theorem 31. *(Lee, 2018, Cor. 2.29) Let $\mathcal{M}$ be a Riemmanian manifold, and $\mathcal{G}$ is a Lie group acting smoothly, freely, properly, and isometrically on $\mathcal{M}$. Then the orbit $\mathcal{M}/\mathcal{G}$ has a unique smooth manifold structure and Riemannian metric such that π is a Riemannian submersion.*

With the Riemannian submersion structure, we can define two subspaces of the tangent space $T_{\mathbf{x}}\mathcal{M}$ as follows. The vertical tangent space $\mathcal{V}_{\mathbf{x}} := \mathrm{Ker}\ \pi_{*\mathbf{x}}$, and the horizontal tangent space is its orthogonal complement $\mathcal{H}_{\mathbf{x}} := (\mathrm{Ker}\, \pi_{*\mathbf{x}})^{\perp}$. A vector field on $\mathcal{M}$ is said to be a horizontal vector field if its value at each point lies in the horizontal subspace at that point, a vertical vector field is defined similarly.

Definition 32. Given a vector field $\mathbf{v}$ on $\mathcal{Q}$, a vector field $\mathbf{u}$ on $\mathcal{M}$ is called a **horizontal lift** of $\mathbf{v}$ if $\mathbf{u}$ is a horizontal vector field and $\mathbf{u}$ is π-related to $\mathbf{v}$, where the latter property means that $\pi_{*\mathbf{x}}\mathbf{u}_{\mathbf{x}} = \mathbf{v}_{\pi(\mathbf{x})}$.

The horizontal lift is unique and always exists. We summarized the properties of horizontal vector fields in the following proposition.

Proposition 33. *(Lee, 2018, Prop. 2.25) Assume $\pi : \mathcal{M} \to \mathcal{Q}$ is a smooth submersion, then we have:*

- *Every smooth vector field $\mathbf{u}$ on $\mathcal{M}$ can be expressed uniquely in the form $\mathbf{u} = \mathbf{u}^{\mathcal{H}} + \mathbf{v}^{\mathcal{V}}$, where $\mathbf{u}^{\mathcal{H}}$ is horizontal and $\mathbf{u}^{\mathcal{V}}$ is vertical and both $\mathbf{u}^{\mathcal{H}}$ and $\mathbf{u}^{\mathcal{V}}$ are smooth;*
- *Every smooth vector field on $\mathcal{Q}$ has a unique smooth horizontal lift to $\mathcal{M}$.*
- *For every $\mathbf{x} \in \mathcal{M}$ and $\mathbf{v} \in \mathcal{H}_{\mathbf{x}}$, there is a vector field $\mathbf{u} \in \mathscr{X}(\mathcal{Q})$ whose horizontal lift $\tilde{\mathbf{u}}$ satisfies $\tilde{\mathbf{u}}_{\mathbf{x}} = \mathbf{v}$.*

According to the first property of Prop. 33 we can define the horizontal projection within $T_{\mathbf{x}}\mathcal{M}$ itself: $P_{\mathbf{x}}(\mathbf{v}) := \mathbf{v}^{\mathcal{H}}$ and P is a smooth mapping. The result of Thm. 31 shows that π is a Riemannian submersion, i.e. the Riemannian metric of $\mathcal{Q}$ can be pulled back from total space $\mathcal{M}$ using $\pi_{*\mathbf{x}}^{-1}|_{\mathcal{H}}$, i.e. $\langle \mathbf{u}_1, \mathbf{u}_2 \rangle_{\mathbf{y}}^{\mathcal{Q}} := \langle \pi_{*\mathbf{x}}^{-1}|_{\mathcal{H}}(\mathbf{u}_1), \pi_{*\mathbf{x}}^{-1}|_{\mathcal{H}}(\mathbf{u}_2) \rangle_{\mathbf{x}}^{\mathcal{M}}$, which is the same for any $\mathbf{x} \in \pi^{-1}(\mathbf{y})$ (due to the isometry property of the group action).

Proposition 34. *(Lee, 2018, Exercise. 5.6) Let $\nabla^{\mathcal{M}}$ and $\nabla^{\mathcal{Q}}$ denote the Levi-Civita connection of $\mathcal{M}, \mathcal{Q}$ respectively. Then for any $\mathbf{u}, \mathbf{v} \in \mathcal{Q}$, let $\tilde{\mathbf{u}}, \tilde{\mathbf{v}}$ be the horizontal lift of $\mathbf{u}, \mathbf{v}$. Then we have*

$$\widetilde{\nabla_{\mathbf{u}}^{\mathcal{Q}} \mathbf{v}} = (\nabla_{\tilde{\mathbf{u}}}^{\mathcal{M}} \tilde{\mathbf{v}})^{\mathcal{H}}.$$

For a concrete example, we consider the example of shape space, i.e. the total space $\mathbb{R}^{3n}$ with the $\mathrm{SE}(3)$ symmetry. Let

$$\mathbf{x} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(N)}) \in \mathbb{R}^{3N}, \quad \text{with } \mathbf{x}^{(i)} \in \mathbb{R}^3,$$

denote a configuration (or point cloud) of N points in $\mathbb{R}^3$. Since the translation group is not compact thus there does not exist a probability distribution that is translation invariant. To solve this issue, we first let $\bar{\mathcal{M}}$ be the center of mass subspace (COM) and consider the $\mathrm{SO}(3)$ action on it. Formally, let $\bar{\mathcal{M}} := \{\mathbf{x} \in \mathbb{R}^{3n} \mid \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i = \mathbf{0}\}$. $\mathcal{M}$ is a linear subspace of $\mathbb{R}^{3n}$, so obviously it is a Riemmannian manifold. We endow $\bar{\mathcal{M}}$ with the standard inner product of $\mathbb{R}^{3n}$. An element of the $\mathrm{SO}(3)$ group is given by a 3-dimensional rotation matrix $g \in \mathbb{R}^{3\times 3}$. The natural action of g on $\mathbf{x}$ is defined as $g \cdot \mathbf{x} = \big(g\mathbf{x}^{(1)},\ g\mathbf{x}^{(2)},\ \cdots,\ g\mathbf{x}^{(N)}\big)$, i.e. the rotation is acted on each point of the system.

Unfortunately, $\mathrm{SO}(3)$ does not act freely on $\bar{\mathcal{M}}$ in some degenerate cases, e.g. all the coordinates of the points are in a straight line. So we define the subset $\mathcal{D} \subset \bar{\mathcal{M}}$ that $\mathrm{SE}(3)$ does not have free action on it. $\bar{\mathcal{M}} \backslash \mathcal{D}$ is a smooth manifold as $\mathcal{D}$ is a low dimensional subspace of $\bar{\mathcal{M}}$. Now $\mathrm{SO}(3)$ acts freely and smoothly on $\mathcal{M} := \bar{\mathcal{M}} \backslash \mathcal{D}$, and it's obvious that the $\mathrm{SE}(3)$ action is isometric in the Euclidean space. Since $\mathrm{SO}(3)$ is a compact group, by Prop. 30, the action is proper and we have checked that the action is smooth, proper, isometric and free. Then by Thm. 31, the quotient space $\mathcal{Q} := \mathcal{M}/\mathrm{SO}(3)$ is a Riemannian manifold and the projection $\pi : \mathcal{M} \to \mathcal{Q}$ is a Riemannian submersion. Again, we denote $\pi : \mathcal{M} \to \mathcal{Q}$ as the projection operator, $\pi(\mathbf{x}) = [\mathbf{x}]$, where $[\mathbf{x}] \in \mathcal{Q}$ is the equivalent class that $\mathbf{x} \in \mathcal{M}$ belongs to.

Since $\mathcal{M}$ is a Riemannian manifold with standard Euclidean inner product, we can uniquely decompose the tangent space of $\mathcal{M}$ as the orthogonal direct sum of the horizontal subspace and the vertical subspace, i.e. $T_{\mathbf{x}}\mathcal{M} = \mathcal{V}_{\mathbf{x}} \oplus \mathcal{H}_{\mathbf{x}}$. The vertical space $\mathcal{V}_{\mathbf{x}} := \mathrm{Ker}\, \pi_{*\mathbf{x}}$ captures the infinitesimal movement of the group action, which is defined by the Lie algebra of the Lie group $\mathcal{G}$ (Appx. B). For $\mathcal{G} = \mathrm{SO}(3)$, the Lie algebra $\mathfrak{so}(3)$ is given by the antisymmetric matrices in $\mathbb{R}^{3\times 3}$. So the vertical tangent space is given by:

$$\mathcal{V}_{\mathbf{x}} = \{(\mathbf{A}\mathbf{x}^{(1)},\ \mathbf{A}\mathbf{x}^{(2)},\ \cdots,\ \mathbf{A}\mathbf{x}^{(N)}) \mid \mathbf{A} \in \mathfrak{so}(3)\}.$$

The horizontal space, which is the orthogonal complement of the vertical space, is given by

$$\mathcal{H}_{\mathbf{x}} = \Big\{\mathbf{v} = (\mathbf{v}^{(1)}, \cdots, \mathbf{v}^{(N)}) \in \mathbb{R}^{3N} : \quad \sum\nolimits_{i=1}^{N} \mathbf{x}^{(i)} \times \mathbf{v}^{(i)} = 0\Big\},$$

where "$\times$" denotes the cross product on $\mathbb{R}^3$.

For any tangent vector $\mathbf{u} \in T_{[\mathbf{x}]}\mathcal{Q}$, there is a unique horizontal lift of it given by $\hat{\mathbf{u}} = \pi_{*\mathbf{x}}^{-1}|_{\mathcal{H}}(\mathbf{u})$. Then we can define the Riemannian metric on the quotient space using the inner product on $\mathbb{R}^{3n}$ and the horizontal lift: $\langle \mathbf{u}, \mathbf{v} \rangle_{[\mathbf{x}]}^{\mathcal{Q}} := \langle \hat{\mathbf{u}}, \hat{\mathbf{v}} \rangle_{\mathbf{x}}^{\mathbb{R}^{3n}}, \forall \mathbf{u}, \mathbf{v} \in T_{[\mathbf{x}]}\mathcal{Q}$.

Let $\mathbf{x}_t$ be a diffusion process on $\mathcal{M} = \bar{\mathcal{M}} \backslash \mathcal{D}$, we can still view it as a process on the Euclidean space $\mathbb{R}^{3n}$ with a slight modification. We can define a stopping time $\tau_{\mathcal{D}} = \inf\{t \in [0, \infty] \mid \mathbf{x}_t \in \mathcal{D}\}$, which is the first time point that $\mathbf{x}_t$ hit the "boundary". Define

$$\mathbf{x}_\tau = \begin{cases} \mathbf{x}_t, & \text{if } t \leq \tau, \\ 0, & \text{if } t \geq \tau. \end{cases}$$

Then $\mathbf{x}_\tau$ is also a diffusion process with the same generator, however, $\mathbf{x}_\tau$ stops at the boundary.

D PROOFS

D.1 PROOF OF THEOREM 1

Theorem 35. *Assume $\{\mathbf{x}_t\}_{t\in[0,T]}$ is a diffusion process on $\mathcal{M}$, specified by the following SDE:*

$$\mathrm{d}\mathbf{x}_t = \mathbf{b}_t(\mathbf{x}_t)\,\mathrm{d}t + \sigma_t\,\mathrm{d}\mathbf{w}_t, \quad \mathbf{x}_0 \sim p_{\text{prior}}, \tag{13}$$

where $\mathbf{b}_t$ is a $\mathcal{G}$-equivariant time-dependent vector field on $\mathcal{M}$, $\mathbf{w}_t$ is the Wiener process on $\mathcal{M}$ that is also $\mathcal{G}$-invariant, and p_{prior} is a $\mathcal{G}$-invariant distribution. Then the projected process $\{\mathbf{y}_t := \pi(\mathbf{x}_t)\}_{t\in[0,T]}$ onto the quotient space $\mathcal{Q} := \mathcal{M}/\mathcal{G}$ is the solution to the following SDE:

$$\mathrm{d}\mathbf{y}_t = \left((\pi_*\mathbf{b}_t)(\mathbf{y}_t) - \frac{\sigma_t^2}{2}\mathbf{h}(\mathbf{y}_t)\right)\mathrm{d}t + \sigma_t\,\mathrm{d}\boldsymbol{\omega}_t, \quad \mathbf{y}_0 \sim \pi_\# p_{\text{prior}}, \tag{14}$$

*where $\pi_*\mathbf{b}_t$ is the pushed-forward vector field of $\mathbf{b}_t$ induced by π, $\mathbf{h}(\mathbf{y}_t) := \pi_*(\sum_{i=M-G+1}^{M} \nabla_{\mathbf{e}_i}\mathbf{e}_i)$ is the mean curvature vector at $\mathbf{y}_t$ ($\{e_i\}$ is a local orthonormal basis of $T_{\mathbf{x}_t}\mathcal{M}$ and $\mathcal{V}_{\mathbf{x}} = \text{span}\{\mathbf{e}_{M-G+1}, \cdots, \mathbf{e}_M\}$), $\boldsymbol{\omega}_t$ is the Wiener process on $\mathcal{Q}$, and $\pi_\# p_{prior}$ is the pushed-forward distribution of p_{prior} (*i.e.*, $\mathbf{y}_0 = \pi(\mathbf{x}_0)$ where $\mathbf{x}_0 \sim p_{\text{prior}}$).*

Proof. As $\mathbf{x}_t$ is a diffusion process on $\mathcal{M}$ given by the the SDE $\mathrm{d}\mathbf{x}_t = \mathbf{b}_t(\mathbf{x}_t)\,\mathrm{d}t + \sigma_t\,\mathrm{d}\mathbf{w}_t$, by Prop. 27, $\mathbf{x}_t$ is a $\mathcal{L}_t$-diffusion and the generator is

$$\mathcal{L}_t = \mathbf{b}_t + \frac{\sigma_t^2}{2}\Delta^{\mathcal{M}}.$$

Assume $\mathbf{e}_1, ..., \mathbf{e}_M$ is a local orthonormal basis of $\mathcal{M}$ and $\mathcal{H}_{\mathbf{x}} = \text{span}\{\mathbf{e}_1, \cdots, \mathbf{e}_{M-G}\}$, $\mathcal{V}_{\mathbf{x}} = \text{span}\{\mathbf{e}_{M-G+1}, \cdots, \mathbf{e}_M\}$. Then by the Riemannian submersion construction of $\pi : \mathcal{M} \to \mathcal{Q}$ (see Appx. C), $\{\tilde{\mathbf{e}}_i := \pi_*\mathbf{e}_i\}_{i=1,2,...,M-G}$ is a local orthonormal basis of $\mathcal{Q}$. Let $\nabla^{\mathcal{M}}$ and $\nabla^{\mathcal{Q}}$ be the Levi-Civita connection on $\mathcal{M}, \mathcal{Q}$, respectively. Using the local expression of the Laplace-Beltrami operator (Def. 18), the generator is given by

$$\begin{aligned}
\mathcal{L}_t &= \mathbf{b}_t + \frac{\sigma_t^2}{2}\Delta^{\mathcal{M}} \\
&= \mathbf{b}_t + \frac{\sigma_t^2}{2}\sum_{i=1}^{M}(\mathbf{e}_i(\mathbf{e}_i) - \nabla^{\mathcal{M}}_{\mathbf{e}_i}\mathbf{e}_i) \\
&= \left(\mathbf{b}_t - \frac{\sigma_t^2}{2}\sum_{i=1}^{M}\nabla^{\mathcal{M}}_{\mathbf{e}_i}\mathbf{e}_i\right) + \frac{\sigma_t^2}{2}\sum_{i=1}^{M}\mathbf{e}_i^2.
\end{aligned}$$

Then the process is the solution of the Stratonovitch SDE

$$\mathrm{d}\mathbf{x}_t = \mathbf{v}_0(\mathbf{x}_t, t)\mathrm{d}t + \sum_{i=1}^{M}\mathbf{v}_i(\mathbf{x}_t, t) \circ \mathrm{d}\mathbf{w}_i, \quad \text{where } \mathbf{v}_0 = \mathbf{b}_t - \frac{\sigma_t^2}{2}\sum_{i=1}^{M}\nabla^{\mathcal{M}}_{\mathbf{e}_i}\mathbf{e}_i,\ \mathbf{v}_i = \sigma_t\mathbf{e}_i.$$

By Def. 26, for all $f \in C^\infty(\mathcal{M})$,

$$f(\mathbf{x}_t) = f(\mathbf{x}_0) + \int_0^t \left(\mathbf{v}_0(f)(\mathbf{x}_s, s)\mathrm{d}s + \sum_{i=1}^{d}\mathbf{v}_i(f)(\mathbf{x}_s, s) \circ \mathrm{d}\mathbf{x}_i\right).$$

Let $\tilde{f} \in C^\infty(\mathcal{Q})$, then $f = \tilde{f} \circ \pi \in C^\infty(\mathcal{M})$, then

$$\begin{aligned} f(\mathbf{x}_t) &= \tilde{f}(\pi(\mathbf{x}_t)) \\ &= \tilde{f}(\pi(\mathbf{x}_0)) + \int_0^t \left(\mathbf{v}_0(\tilde{f} \circ \pi)(\mathbf{x}_s, s)\mathrm{d}s + \sum_{i=1}^d \mathbf{v}_i(\tilde{f} \circ \pi)(\mathbf{x}_s, s) \circ \mathrm{d}\mathbf{x}_i \right) \\ &= \tilde{f}(\pi(\mathbf{x}_0)) + \int_0^t \left((\pi_* \mathbf{v}_0)(\tilde{f})(\pi(\mathbf{x}_s), s)\mathrm{d}s + \sum_{i=1}^d (\pi_* \mathbf{v}_i)(\tilde{f})(\pi(\mathbf{x}_s), s) \circ \mathrm{d}\mathbf{x}_i \right), \text{ by Def. 9.} \end{aligned}$$

Since $\tilde{f}$ is arbitrary, by Def. 26, $\mathbf{y}_t := \pi(\mathbf{x}_t)$ is the solution of

$$\mathrm{d}\mathbf{y}_t = \pi_* \mathbf{v}_0(\mathbf{y}_t, t)\mathrm{d}t + \sum_{i=1}^M \pi_* \mathbf{v}_i(\mathbf{y}_t, t) \circ \mathrm{d}\mathbf{w}_i.$$

We first need to check that the projected vector field is well defined. In fact, we only need to check that $\pi_* \mathbf{b}$ is well defined. Since $\mathbf{b}$ is $\mathcal{G}$-equivariant, then for any $g \in \mathcal{G}$, $g_* \mathbf{b}_t(\mathbf{x}) = \mathbf{b}_t(g \cdot \mathbf{x})$. Then $\pi_*(\mathbf{b}_t(g \cdot \mathbf{x})) = \pi_*(g_* \mathbf{b}_t(\mathbf{x})) = (\pi \circ g)_*(\mathbf{b}_t(\mathbf{x}))$, where we use the chain rule of differential calculus in the last step. Thus $\pi_*(\mathbf{b}_t(\mathbf{x}))$ is the same in the fiber $\mathbf{x} \in \pi^{-1}(\pi(\mathbf{x}))$, which implies that the projected vector field is well defined.

Next, we calculate the expression of the projected vector field. Since $\mathcal{H}_\mathbf{x} = \text{span}\{\mathbf{e}_1, \cdots, \mathbf{e}_{M-G}\}$, $\mathcal{V}_\mathbf{x} = \text{span}\{\mathbf{e}_{M-G+1}, \cdots, \mathbf{e}_M\}$, we have

$$\pi_* \mathbf{e}_i = \begin{cases} \tilde{\mathbf{e}}_i, & \text{if } i \leq M - G, \\ 0, & \text{if } i \geq M - G + 1, \end{cases}$$

and $\pi_* \mathbf{v}_i = \sigma_t \pi_* \mathbf{e}_i$. For the drift term, using Prop. 34, we have

$$\begin{aligned} \pi_* \mathbf{v}_0(\mathbf{x}, t) &= \pi_* \mathbf{b}_t(\mathbf{x}) - \frac{\sigma_t^2}{2} \sum_{i=1}^M \pi_*(\nabla^\mathcal{M}_{\mathbf{e}_i} \mathbf{e}_i) \\ &= \pi_* \mathbf{b}_t(\mathbf{x}) - \frac{\sigma_t^2}{2} \sum_{i=1}^{M-G} \pi_*(\nabla^\mathcal{M}_{\mathbf{e}_i} \mathbf{e}_i) - \frac{\sigma_t^2}{2} \sum_{i=M-G+1}^M \pi_*(\nabla^\mathcal{M}_{\mathbf{e}_i} \mathbf{e}_i) \\ &= \pi_* \mathbf{b}_t(\mathbf{x}) - \frac{\sigma_t^2}{2} \sum_{i=1}^{M-G} \nabla^\mathcal{Q}_{\tilde{\mathbf{e}}_i} \tilde{\mathbf{e}}_i - \frac{\sigma_t^2}{2} \sum_{i=M-G+1}^M \pi_*(\nabla^\mathcal{M}_{\mathbf{e}_i} \mathbf{e}_i) \\ &= \pi_* \mathbf{b}_t(\mathbf{x}) - \frac{\sigma_t^2}{2} \sum_{i=1}^{M-G} \nabla^\mathcal{Q}_{\tilde{\mathbf{e}}_i} \tilde{\mathbf{e}}_i - \frac{\sigma_t^2}{2} \mathbf{h}(\mathbf{x}). \end{aligned}$$

So the generator of the process $\mathbf{y}_t$ is

$$\begin{aligned} \tilde{\mathcal{L}}_s &= \pi_* \mathbf{b}_t - \frac{\sigma_t^2}{2} \sum_{i=1}^{M-G} \nabla^\mathcal{Q}_{\tilde{\mathbf{e}}_i} \tilde{\mathbf{e}}_i - \frac{\sigma_t^2}{2} \mathbf{h} + \frac{\sigma_t^2}{2} \sum_{i=1}^{M-G} \tilde{\mathbf{e}}_i^2 \\ &= \left(\pi_* \mathbf{b}_t - \frac{\sigma_t^2}{2} \mathbf{h} \right) + \frac{\sigma_t^2}{2} \left(\sum_{i=1}^{M-G} \tilde{\mathbf{e}}_i^2 - \sum_{i=1}^{M-G} \nabla^\mathcal{Q}_{\tilde{\mathbf{e}}_i} \tilde{\mathbf{e}}_i \right) \\ &= \left(\pi_* \mathbf{b}_t - \frac{\sigma_t^2}{2} \mathbf{h} \right) + \frac{\sigma_t^2}{2} \Delta^\mathcal{Q}. \end{aligned}$$

Then we can conclude that the projected process $\mathbf{y}_t := \pi(\mathbf{x}_t)$ is the solution of the following SDE

$$\mathrm{d}\mathbf{y}_t = \left((\pi_* \mathbf{b}_t)(\mathbf{y}_t) - \frac{\sigma_t^2}{2} \mathbf{h}(\mathbf{y}_t) \right) \mathrm{d}t + \sigma_t \, \mathrm{d}\boldsymbol{\omega}_t,$$

where $\pi_* \mathbf{b}_t$ is the push-forward vector field, $\mathbf{h}(\mathbf{y}_t)$ is the mean curvature vector at $\mathbf{y}_t$ and $\boldsymbol{\omega}_t$ is the standard Wiener process on the quotient space $\mathcal{Q}$.

□

D.2 Proof of Theorem 2

In Def. 32, we define the horizontal lift of a vector field that generates a deterministic flow. In fact, for a stochastic process on $\mathcal{Q}$, we can define the horizontal lift for it similarly. First, we need to define the stochastic line integral, which is the integration of a one-form along the trajectory of a stochastic process.

Definition 36. (Hsu, 2002, Prop. 2.4.2) Let Θ be a 1-form (Def. 11) on $\mathcal{M}$ and $\mathbf{x}_t$ the solution of the equation

$$\mathrm{d}\mathbf{x}_t = \mathbf{v}_0(\mathbf{x}_t, t)\mathrm{d}t + \sum_{i=1}^{d} \mathbf{v}_i(\mathbf{x}_t, t) \circ \mathrm{d}\mathbf{w}_i(t).$$

Then

$$\int_{\mathbf{x}_{[0,t]}} \Theta = \int_0^t \sum_{i=0}^{d} \Theta(\mathbf{v}_i)(\mathbf{x}_s) \circ \mathrm{d}\mathbf{w}_i(s).$$

Definition 37. (Baudoin et al., 2024, Def. 3.1.9) A semimartingale $\mathbf{x}_t$ on $\mathcal{M}$ is called horizontal if for every 1-form Θ on $\mathcal{M}$ whose kernel contains the horizontal space $\mathcal{H}$, one has $\int_{\mathbf{x}_{[0,t]}} \Theta = 0,\ \forall t \geq 0$. Let $\mathbf{y}_t$ be a semimartingale on $\mathcal{Q}$ such that $\mathbf{y}_0$ is a point of $\mathcal{Q}$. Let $\mathbf{x}_0 \in \pi^{-1}(\mathbf{y}_0)$. Then there exists a unique horizontal semimartingale $\mathbf{x}_t$ on $\mathcal{M}$ such that $\mathbf{x}_t$ starts from $\mathbf{x}_0$ and $\pi(\mathbf{x}_t) = \mathbf{y}_t, \forall t \geq 0$. $\mathbf{x}_t$ is called the horizontal lift of $\mathbf{y}_t$ at $\mathbf{x}_0$.

Theorem 38. *The horizontal lift of Eq. (14) has the following explicit expression:*

$$\mathrm{d}\tilde{\mathbf{x}}_t = \left(P_{\tilde{\mathbf{x}}_t}(\mathbf{b}_t(\tilde{\mathbf{x}}_t)) - \frac{\sigma_t^2}{2}\tilde{\mathbf{h}}(\tilde{\mathbf{x}}_t) \right) \mathrm{d}t + \sigma_t \mathrm{d}\tilde{\mathbf{w}}_t, \quad \tilde{\mathbf{x}}_0 \sim p_{\text{prior}}, \tag{15}$$

where $P_{\mathbf{x}}(\mathbf{v}) = \mathbf{v}^{\mathcal{H}}$ is the horizontal projection on the tangent space of $\mathcal{M}$, $\tilde{\mathbf{h}}$ is the horizontal lift of the mean curvature vector, $\tilde{\mathbf{w}}_t$ is the horizontal lift of the Wiener process on $\mathcal{Q}$.

Proof. We only need to check the definition of the horizontal lift (Def. 37). Again, assume $\mathbf{e}_1, ..., \mathbf{e}_M$ is a local orthonormal basis of $\mathcal{M}$ and $\mathcal{H}_{\mathbf{x}} = \text{span}\{\mathbf{e}_1, \cdots, \mathbf{e}_{M-G}\}$, $\mathcal{V}_{\mathbf{x}} = \text{span}\{\mathbf{e}_{M-G+1}, \cdots, \mathbf{e}_M\}$. Then by the Riemannian submersion construction of $\pi : \mathcal{M} \to \mathcal{Q}$ (see Appx. C), $\{\tilde{\mathbf{e}}_i := \pi_* \mathbf{e}_i\}_{i=1,2,...,M-G}$ is a local orthonormal basis of $\mathcal{Q}$. Let $\nabla^{\mathcal{M}}$ and $\nabla^{\mathcal{Q}}$ be the Levi-Civita connection on $\mathcal{M}, \mathcal{Q}$, respectively.

Now we calculate the generator of the SDE in Eq. (15):

$$\begin{aligned} \tilde{\mathcal{L}}_t &= \left(P\mathbf{b}_t - \frac{\sigma_t^2}{2}\tilde{\mathbf{h}} \right) + \frac{\sigma_t^2}{2} \sum_{i=1}^{M} P(e_i)^2 - P\nabla^{\mathcal{M}}_{e_i} e_i \\ &= \left(\mathbf{b}_t^{\mathcal{H}} - \frac{\sigma_t^2}{2}\tilde{\mathbf{h}} \right) + \frac{\sigma_t^2}{2} \sum_{i=1}^{M-G} e_i^2 - (\nabla^{\mathcal{M}}_{e_i} e_i)^{\mathcal{H}}. \end{aligned} \tag{16}$$

Its projection under π_* is given by

$$\mathcal{L}_t = \left(\pi_* \mathbf{b}_t - \frac{\sigma_t^2}{2}\mathbf{h} \right) + \sum_{i=1}^{M-G} \tilde{e}_i^2 - (\nabla^{\mathcal{M}}_{\tilde{e}_i} \tilde{e}_i)^{\mathcal{H}},$$

which is the generator of Eq. (14). So we have $\pi(\tilde{\mathbf{x}}_t) = \mathbf{y}_t$ defined in Eq. (14).

For an 1-form Θ on $\mathcal{M}$ whose kernel contains the horizontal space $\mathcal{H}$. From Eq. (16), $\tilde{\mathbf{x}}_t$ is the following SDE

$$\mathrm{d}\mathbf{x}_t = \mathbf{v}_0(\mathbf{x}_t, t)\mathrm{d}t + \sum_{i=1}^{M} \mathbf{v}_i(\mathbf{x}_t, t) \circ \mathrm{d}\mathbf{w}_i,$$

$$\text{where } \mathbf{v}_0 = \left(\mathbf{b}_t^{\mathcal{H}} - \frac{\sigma_t^2}{2}\tilde{\mathbf{h}} \right) - \frac{\sigma_t^2}{2} \sum_{i=1}^{M-G} (\nabla^{\mathcal{M}}_{e_i} e_i)^{\mathcal{H}}, \quad \mathbf{v}_i = \sigma_t \mathbf{e}_i.$$

Then the line integral

$$\int_{\tilde{\mathbf{x}}_{[0,t]}} \Theta = \int_0^t \sum_{i=0}^{M} \Theta(\mathbf{v}_i)(\tilde{\mathbf{x}}_s) \circ \mathrm{d}\mathbf{w}_i(s) = 0,$$

since $\mathbf{v}_i \in \mathcal{H},\ \Theta(\mathbf{v}_i) = 0$. So we can conclude that $\tilde{\mathbf{x}}_t$ is the horizontal lift of $\mathbf{y}_t$. □

Corollary 39. *$\tilde{\mathbf{x}}_1$ (defined by Eq. (8)) has the same distribution on $\mathcal{Q}$ with $\mathbf{x}_1$ (defined by Eq. (13)). When $\sigma_t = 0, \forall \mathbf{x}_0 \in \mathcal{M}$, Eq. (8) has shorter trajectory length than Eq. (13):*

$$\int_0^1 \langle P_{\tilde{\mathbf{x}}_t}(\mathbf{b}_t(\tilde{\mathbf{x}}_t)), P_{\tilde{\mathbf{x}}_t}(\mathbf{b}_t(\tilde{\mathbf{x}}_t))\rangle^{\mathcal{M}} \mathrm{d}t \le \int_0^1 \langle \mathbf{b}_t(\mathbf{x}_t), \mathbf{b}_t(\mathbf{x}_t))\rangle^{\mathcal{M}} \mathrm{d}t.$$

Proof. By definition of horizontal lift, $\pi(\tilde{\mathbf{x}}_t) = \mathbf{y}_t = \pi(\mathbf{x}_t), \forall t \in [0,1]$, then $\tilde{\mathbf{x}}_1$ (defined by Eq. (8)) has the same distribution on $\mathcal{Q}$ with $\mathbf{x}_1$ (defined by Eq. (13)). Since $\pi(\tilde{\mathbf{x}}_t) = \pi(\mathbf{x}_t)$, then $\mathbf{x}_t = g_t \tilde{\mathbf{x}}_t, g_t \in \mathcal{G}$. Then by the $\mathcal{G}$-equivariant property of $\mathbf{b}$, we have

$$\begin{aligned}
\int_0^1 \langle \mathbf{b}_t(\mathbf{x}_t), \mathbf{b}_t(\mathbf{x}_t))\rangle^{\mathcal{M}} \mathrm{d}t &= \int_0^1 \langle \mathbf{b}_t(g_t\tilde{\mathbf{x}}_t), \mathbf{b}_t(g_t\tilde{\mathbf{x}}_t))\rangle^{\mathcal{M}} \mathrm{d}t \\
&= \int_0^1 \langle g_{t*}\mathbf{b}_t(\tilde{\mathbf{x}}_t), \mathbf{b}_t(g_{t*}\tilde{\mathbf{x}}_t))\rangle^{\mathcal{M}} \mathrm{d}t \\
&= \int_0^1 \langle \mathbf{b}_t(\tilde{\mathbf{x}}_t), \mathbf{b}_t(\tilde{\mathbf{x}}_t))\rangle^{\mathcal{M}} \mathrm{d}t \\
&= \int_0^1 \left(\langle \mathbf{b}_t(\tilde{\mathbf{x}}_t)^{\mathcal{H}}, \mathbf{b}_t(\tilde{\mathbf{x}}_t)^{\mathcal{H}})\rangle^{\mathcal{M}} + \langle \mathbf{b}_t(\tilde{\mathbf{x}}_t)^{\mathcal{V}}, \mathbf{b}_t(\tilde{\mathbf{x}}_t)^{\mathcal{V}})\rangle^{\mathcal{M}} \right) \mathrm{d}t \\
&\ge \int_0^1 \langle \mathbf{b}_t(\tilde{\mathbf{x}}_t)^{\mathcal{H}}, \mathbf{b}_t(\tilde{\mathbf{x}}_t)^{\mathcal{H}})\rangle^{\mathcal{M}} \mathrm{d}t \\
&= \int_0^1 \langle P_{\tilde{\mathbf{x}}_t}(\mathbf{b}_t(\tilde{\mathbf{x}}_t)), P_{\tilde{\mathbf{x}}_t}(\mathbf{b}_t(\tilde{\mathbf{x}}_t))\rangle^{\mathcal{M}} \mathrm{d}t.
\end{aligned}$$

□

D.3 Proof of Theorem 3

For the calculation of the mean curvature vector, we can embed the fiber $\pi^{-1}(\mathbf{y})$ into the total space where $\mathbf{y} \in \mathcal{Q}$. Thus, we can define the embedding $\Phi^{\mathbf{x}} : \mathcal{G} \to \mathcal{M}$ by $\Phi^{\mathbf{x}}(g) = g \cdot \mathbf{x}$. For $\mathbf{x} \in \pi^{-1}(\mathbf{y})$ the horizontal lift of mean curvature vector is defined by $\tilde{\mathbf{h}}(\mathbf{x}) := (\sum_{i=M-G+1}^{M} \nabla_{e_i} e_i)^{\mathcal{H}}$, where $\{e_i\}$ is a local orthonormal basis of $T_{\mathbf{x}}\mathcal{M}$ and $\mathcal{V}_{\mathbf{x}} = \mathrm{span}\{\mathbf{e}_{M-G+1}, \cdots, \mathbf{e}_M\}$. The mean curvature vector has a nice geometric relation to the volume of the fiber that helps us to calculate it.

Definition 40. Let $\Phi : \mathcal{G} \to \mathcal{M}$ be an immersion. A smooth variation of Φ is a smooth mapping $F : \mathcal{P} \times (-\epsilon, \epsilon) \to \mathcal{M}$ satisfying:

- For any $t \in (-\epsilon, \epsilon)$, $\Phi_t = F(\cdot, t)$ is an immersion;
- $\Phi_0 = F(\cdot, 0) = \Phi$;
- $\Phi_t \mid_{\partial\mathcal{G}} = \Phi \mid_{\partial\mathcal{G}}, \forall t \in (-\epsilon, \epsilon)$, where $\partial\mathcal{G}$ is the boundary of $\mathcal{G}$.

Proposition 41. *(First variation of volume (Chavel, 1995, Exercise. III.14)) The mean curvature vector $\tilde{\mathbf{h}}(\mathbf{x})$ satisfies the following formula:*

$$\frac{\mathrm{d}}{\mathrm{d}t} \mid_{t=0} Vol(\mathcal{G}) = -\int_{\mathcal{G}} \left\langle \tilde{\mathbf{h}}, \mathbf{v} \right\rangle \mathrm{d}Vol(\mathcal{G}),$$

where $\mathbf{v} = F_*(\frac{\partial}{\partial t})$.

In local orthonormal frame $\{\bar{e}_i\}$ of $\mathcal{G}$, the volume of $\mathcal{G}$ is defined by

$$\mathrm{Vol}(\mathcal{G}) := \int_{\mathcal{G}} \sqrt{\det(\mathbf{G})} \mathrm{d}w^1 \wedge \cdots \wedge w^G,$$

where $\mathbf{G}_{ij} = \langle \Phi_* \bar{e}_i, \Phi_* \bar{e}_j \rangle^{\mathcal{M}}$, w^i is the dual form of e_i, i.e. $w^i(\bar{e}_j) = 1$ if $i = j$ and $w^i(\bar{e}_j) \neq 1$ if $i \neq j$.

Theorem 42. *Assume $\mathbf{x}_t$ is a diffusion process in the COM subspace $\mathcal{M} \subset \mathbb{R}^{3n}$, given by the following SDE:*

$$\mathrm{d}\mathbf{x}_t = \mathbf{b}_t(\mathbf{x}_t)\mathrm{d}t + \sigma_t \mathrm{d}\mathbf{w}_t,$$

where $\mathbf{b}_t(\mathbf{x}_t)$ is a $\mathrm{SO}(3)$-equivariant vector field $\forall t \in [0, T]$, $\mathbf{w}_t$ is the standard Wiener process on COM. The horizontal lift of the process $\pi(\mathbf{x}_t)$ is given by the following SDE:

$$\mathrm{d}\tilde{\mathbf{x}}_t = \left(P_{\tilde{\mathbf{x}}_t}(\mathbf{b}_t(\tilde{\mathbf{x}}_t)) - \frac{\sigma_t^2}{2}\tilde{\mathbf{h}}(\tilde{\mathbf{x}}_t) \right) \mathrm{d}t + \sigma_t P_{\tilde{\mathbf{x}}_t} \mathrm{d}\mathbf{w}_t,$$

where the $P_{\mathbf{x}}$ is the horizontal projection operator at $\mathbf{x}$ and $\tilde{\mathbf{h}}(\mathbf{x})$ is the horizontal lift of mean curvature vector. The explicit expressions of P and $\tilde{\mathbf{h}}$ are shown as follows:

$$P_{\mathbf{x}}\mathbf{v} = \mathbf{v} - \mathcal{I}^{-1}\left(\frac{1}{N}\sum_{i=1}^{N} \mathbf{x}^{(i)} \times \mathbf{v}^{(i)} \right) \times \mathbf{x}, \forall \mathbf{v} \in T_{\mathbf{x}}\mathcal{M}$$

$$\tilde{\mathbf{h}}(\mathbf{x}) = -(\mathrm{tr}(\mathcal{I}^{-1})I - \mathcal{I}^{-1})\mathbf{x}, \quad \textit{where} \quad \mathcal{I} = \left(\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}^{(i)}\|^2 \mathbf{I} - \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}^{(i)}\mathbf{x}^{(i)\top} \right).$$

Proof. Again, assume $\mathbf{e}_1, ..., \mathbf{e}_M$ is a local orthonormal basis of $\mathcal{M}$ and $\mathcal{H}_{\mathbf{x}} = \mathrm{span}\{\mathbf{e}_1, \cdots, \mathbf{e}_{M-G}\}$, $\mathcal{V}_{\mathbf{x}} = \mathrm{span}\{\mathbf{e}_{M-G+1}, \cdots, \mathbf{e}_M\}$. Then by the Riemannian submersion construction of $\pi : \mathcal{M} \to \mathcal{Q}$ (see Appx. C), $\{\tilde{\mathbf{e}}_i := \pi_* \mathbf{e}_i\}_{i=1,2,...,M-G}$ is a local orthonormal basis of $\mathcal{Q}$. Let $\nabla^{\mathcal{M}}$ and $\nabla^{\mathcal{Q}}$ be the Levi-Civita connection on $\mathcal{M}, \mathcal{Q}$, respectively. As shown in the Appx. D.2, the horizontal lift of Eq. (8) has the generator

$$\mathcal{L}_t = \left(\mathbf{b}_t^{\mathcal{H}} - \frac{\sigma_t^2}{2}\tilde{\mathbf{h}} \right) + \frac{\sigma_t^2}{2}\sum_{i=1}^{M-G} e_i^2 - (\nabla^{\mathcal{M}}_{e_i} e_i)^{\mathcal{H}}.$$

By Prop. 34, $\sum_{i=1}^{M-G} (\nabla^{\mathcal{M}}_{e_i} e_i)^{\mathcal{V}} = 0$, then

$$\mathcal{L}_t = \left(\mathbf{b}_t^{\mathcal{H}} - \frac{\sigma_t^2}{2}\tilde{\mathbf{h}} \right) + \frac{\sigma_t^2}{2}\sum_{i=1}^{M-G} e_i^2 - (\nabla^{\mathcal{M}}_{e_i} e_i).$$

Since $\mathcal{M}$ is a Euclidean space, then $\nabla^{\mathcal{M}}_{e_i} e_i = \sum_{j=1}^{M} e_i(e_i^j)\partial_j$, where e_i^j is the j-th component of e_i and $\partial_j = \partial/\partial x_j$. Since $\mathbf{b}_t^{\mathcal{H}}(\mathbf{x}) = P_{\mathbf{x}}\mathbf{b}_t(\mathbf{x})$, then the generator becomes

$$\begin{aligned}
\mathcal{L}_t &= \left(\mathbf{b}_t^{\mathcal{H}} - \frac{\sigma_t^2}{2}\tilde{\mathbf{h}} \right) + \frac{\sigma_t^2}{2}\sum_{i=1}^{M-G} e_i^2 - (\nabla^{\mathcal{M}}_{e_i} e_i) \\
&= \left(P\mathbf{b}_t - \frac{\sigma_t^2}{2}\tilde{\mathbf{h}} \right) + \frac{\sigma_t^2}{2}\sum_{i=1}^{M-G}\sum_{j,k=1}^{M} e_i^j(\partial_j e_i^k)\partial_k + e_i^j e_i^k \partial_j\partial_k - e_i^j(\partial_j e_i^k)\partial_k \\
&= \left(P\mathbf{b}_t - \frac{\sigma_t^2}{2}\tilde{\mathbf{h}} \right) + \frac{\sigma_t^2}{2}\sum_{i=1}^{M-G}\sum_{j,k=1}^{M} e_i^j e_i^k \partial_j\partial_k \\
&= \left(P\mathbf{b}_t - \frac{\sigma_t^2}{2}\tilde{\mathbf{h}} \right) + \frac{\sigma_t^2}{2}\sum_{i=1}^{M-G}\sum_{j,k=1}^{M} (P)_{jk}\partial_j\partial_k \\
&= \left(P\mathbf{b}_t - \frac{\sigma_t^2}{2}\tilde{\mathbf{h}} \right) + \frac{\sigma_t^2}{2}\sum_{i=1}^{M-G}\sum_{j,k=1}^{M} (PP^T)_{jk}\partial_j\partial_k,
\end{aligned}$$

where we use $P_{\mathbf{x}} = \sum_{i=1}^{M-G} e_i e_i^T$ is a projection operator. Then $\mathcal{L}_t$ is the generator of

$$\mathrm{d}\tilde{\mathbf{x}}_t = \left(P_{\tilde{\mathbf{x}}_t}(\mathbf{b}_t(\tilde{\mathbf{x}}_t)) - \frac{\sigma_t^2}{2}\tilde{\mathbf{h}}(\tilde{\mathbf{x}}_t) \right) \mathrm{d}t + \sigma_t P_{\tilde{\mathbf{x}}_t} \mathrm{d}\mathbf{w}_t.$$

For the explicit calculation, recall that in this case, the tangent space $T_{\mathbf{x}}\mathcal{M}$ of $\mathcal{M}$ at $\mathbf{x}$ has the following decomposition:

- The vertical tangent space $\mathcal{V}_{\mathbf{x}}$:
$$\mathcal{V}_{\mathbf{x}} = \{(\mathbf{l}\times\mathbf{x}^{(1)},\ \mathbf{l}\times\mathbf{x}^{(2)},\ \cdots,\ \mathbf{l}\times\mathbf{x}^{(N)}) \mid \mathbf{l}\times\in\mathbb{R}^3\}.$$
- The horizontal space $\mathcal{H}_{\mathbf{x}}$, which is the orthogonal complement of the vertical space:
$$\mathcal{H}_{\mathbf{x}} = \Big\{\mathbf{v}=(\mathbf{v}^{(1)},\cdots,\mathbf{v}^{(N)})\in\mathbb{R}^{3N}:\quad \sum_{i=1}^{N}\mathbf{x}^{(i)}\times\mathbf{v}^{(i)}=0\Big\}.$$

The horizontal projection mapping is defined by $P_{\mathbf{x}}(\mathbf{v}) = \mathbf{v}^{\mathcal{H}} = \mathbf{v}-\mathbf{v}^{\mathcal{V}}, \forall \mathbf{v}\in T_{\mathbf{x}}\mathcal{M}$, and we can find an explicit form of it. By definition, $\sum_{i=1}^{N}\mathbf{x}^{(i)}\times\mathbf{v}^{\mathcal{H}(i)}=0$, then

$$\sum_{i=1}^{N}\mathbf{x}^{(i)}\times\mathbf{v}^{(i)} = \sum_{i=1}^{N}\mathbf{x}^{(i)}\times\mathbf{v}^{\mathcal{V}(i)}.$$

Assume $\mathbf{v}^{\mathcal{V}} = (\mathbf{l}\times\mathbf{x}^{(1)},\ \mathbf{l}\times\mathbf{x}^{(2)},\ \cdots,\ \mathbf{l}\times\mathbf{x}^{(N)})$, then

$$\begin{aligned}
\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}^{(i)}\times\mathbf{v}^{(i)} &= \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}^{(i)}\times\mathbf{v}^{\mathcal{V}(i)}\\
&= \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}^{(i)}\times(\mathbf{l}\times\mathbf{x}^{(i)})\\
&= \frac{1}{N}\sum_{i=1}^{N}\Big\langle\mathbf{x}^{(i)},\mathbf{x}^{(i)}\Big\rangle\mathbf{l}-\Big\langle\mathbf{x}^{(i)},\mathbf{l}\Big\rangle\mathbf{x}^{(i)}\\
&= \left(\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{x}^{(i)}\|^2\mathbf{I}-\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}^{(i)}{\mathbf{x}^{(i)}}^{\top}\right)\mathbf{l},
\end{aligned}$$

where we use the identity $\mathbf{x}^{(i)}\times(\mathbf{l}\times\mathbf{x}^{(i)}) = \big\langle\mathbf{x}^{(i)},\mathbf{x}^{(i)}\big\rangle\mathbf{l}-\big\langle\mathbf{x}^{(i)},\mathbf{l}\big\rangle\mathbf{x}^{(i)}$. Denote

$$\mathcal{I} := \left(\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{x}^{(i)}\|^2\mathbf{I}-\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}^{(i)}{\mathbf{x}^{(i)}}^{\top}\right).$$

And we have $\mathbf{l} = \mathcal{I}^{-1}(\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}^{(i)}\times\mathbf{v}^{(i)})$, and

$$\begin{aligned}
\mathbf{v}^{\mathcal{V}} &= (\mathbf{l}\times\mathbf{x}^{(1)},\ \mathbf{l}\times\mathbf{x}^{(2)},\ \cdots,\ \mathbf{l}\times\mathbf{x}^{(N)})\\
&= \mathcal{I}^{-1}\left(\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}^{(i)}\times\mathbf{v}^{(i)}\right)\times\mathbf{x}\,.
\end{aligned}$$

Then

$$P_{\mathbf{x}}\mathbf{v} = \mathbf{v}^{\mathcal{H}} = \mathbf{v}-\mathcal{I}^{-1}\left(\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}^{(i)}\times\mathbf{v}^{(i)}\right)\times\mathbf{x}, \forall\mathbf{v}\in T_{\mathbf{x}}\mathcal{M}.$$

For the calculations of the mean curvature vector $\tilde{\mathbf{h}}$, we can use Prop. 41. As $\mathcal{G}=\mathrm{SO}(3)$, its local frame (the norm of each vector us $\sqrt{2}$) is given by the following matrices:

$$\tilde{e}_1 = \begin{bmatrix}0&0&0\\0&0&-1\\0&1&0\end{bmatrix},\quad \tilde{e}_2 = \begin{bmatrix}0&0&1\\0&0&0\\-1&0&0\end{bmatrix},\quad \tilde{e}_3 = \begin{bmatrix}0&-1&0\\1&0&0\\0&0&0\end{bmatrix}.$$

Then the Gram matrix $\mathbf{G}$ is defined by $\mathbf{G}_{ij} := \langle\tilde{e}_i\mathbf{x},\tilde{e}_j\mathbf{x}\rangle$. By direct calculations, we have $\mathbf{G} = 2N\mathcal{I}$. Then by Prop. 41,

$$\tilde{\mathbf{h}}(\mathbf{x}) = -\nabla\log\sqrt{\det\mathbf{G}}.$$

Using Jacobi's formula in matrix calculus, $\mathrm{d}\log\det\mathbf{G} = \mathrm{tr}(\mathcal{I}^{-1}\mathrm{d}\mathcal{I})$. Then by

$$\mathcal{I} := \frac{1}{N}\sum_{i=1}^{N}\|\mathbf{x}^{(i)}\|^2\mathbf{I} - \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}^{(i)}\mathbf{x}^{(i)\top}, \quad \frac{\partial\mathcal{I}}{\partial\mathbf{x}_j^{(i)}} = \left(\frac{1}{N}\sum_{i=1}^{N}2\mathbf{x}_j^{(i)}\mathbf{I} - \frac{1}{N}\sum_{i=1}^{N}(\delta_j\mathbf{x}^{(i)\top} + \mathbf{x}^{(i)}\delta_j^\top)\right),$$

where $\delta_j \in \mathbb{R}^3$ is a one-hot vector at j. Then

$$\mathrm{tr}(\mathcal{I}^{-1}\frac{\partial\mathcal{I}}{\partial\mathbf{x}_j^{(i)}}) = 2\,\mathrm{tr}(\mathcal{I}^{-1})\mathbf{x}_j^{(i)} - 2\delta_j^\top\mathcal{I}^{-1}\mathbf{x}_j^{(i)}.$$

Then we have

$$\tilde{\mathbf{h}}(\mathbf{x}) = -\frac{1}{2}\nabla\log\det\mathbf{G} = -(\mathrm{tr}(\mathcal{I}^{-1})I - \mathcal{I}^{-1})\mathbf{x}.$$

□

E Training and Sampling Method in General Case

Training Objective The diffusion model on the total space $\mathcal{M}$ is trained by the denoising score matching objective. Since the vertical components of the velocity are not strictly needed, we propose to supervise the model only on the horizontal components and allow arbitrary vertical output of the model. Recall that the horizontal projection operator $P_{\mathbf{x}}$ projects a vector to its horizontal component, i.e. $P_{\mathbf{x}}(\mathbf{v}) = \mathbf{v}^{\mathcal{H}}$. Thus the improved training objective is given by

$$\mathcal{L}^{(}\theta) := \mathbb{E}_{p(t)}w(t)\mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_1)\sim p_{\text{joint}},\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})}\|P_{\mathbf{x}_t}\left(\mathbf{v}_\theta(\mathbf{x}_t,t) - (\alpha_t'\mathbf{x}_0 + \beta_t'\mathbf{x}_1 + \gamma_t'\boldsymbol{\epsilon})\right)\|^2.$$

ODE Sampler After the training stage, $P_{\mathbf{x}_t}(\mathbf{v}_\theta(\mathbf{x}_t,t))$ is an approximation of the ground truth vector field in the horizontal subspace. For the deterministic sampler, we need to simulate the horizontal lift of the projected ODE, which is given by

$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = P_{\mathbf{x}_t}\mathbf{v}(\mathbf{x}_t,t)\mathrm{d}t.$$

In practice, the ODE process is approximated by numerical solvers, e.g. the Euler method and Runge-Kutta methods.

SDE Sampler For the stochastic sampler, we need to simulate the horizontal lift of the projected original SDE in Eq. (3). According to Thm. 1 and Thm. 4, the lifted process is given by

$$\mathrm{d}\mathbf{x}_t = P_{\mathbf{x}_t}\left(\mathbf{v}_\theta(\mathbf{x}_t,t) + g_t\mathbf{s}_\theta(\mathbf{x}_t,t)\right)\mathrm{d}t + \gamma\eta_t\mathbf{h}(\mathbf{x}_t)\mathrm{d}t + \sqrt{2\gamma\eta_t}P_{\mathbf{x}_t}\mathrm{d}\mathbf{w}_t,$$

where we introduce the hyperparameter γ for protein generation following Geffner et al. (2025). The training and sampling algorithm is summarized in Algorithm 2 and 3.

Algorithm 1 Training for $p_{\text{prior}} = \mathcal{N}(0,\mathbf{I})$

1: **repeat**
2: $(\mathbf{x}_0,\mathbf{x}_1) \sim p_{\text{joint}}$
3: $t \sim p_t$
4: $\mathbf{x}_t = \hat{\alpha}_t\mathbf{x}_0 + \beta_t\mathbf{x}_1$
5: Take a gradient descent step on
$\nabla_\theta\, w(t)\,\|P_{\mathbf{x}_t}\left(\mathbf{D}_\theta(\mathbf{x}_t,t) - \mathbf{x}_1\right)\|^2$
6: **until** converged

Algorithm 2 Training for general p_{prior}

1: **repeat**
2: $(\mathbf{x}_0,\mathbf{x}_1) \sim p_{\text{joint}}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$
3: $t \sim p_t$
4: $\mathbf{x}_t = \alpha_t\mathbf{x}_0 + \beta_t\mathbf{x}_1 + \gamma_t\boldsymbol{\epsilon}$
5: $\mathbf{v}_t = \alpha_t'\mathbf{x}_0 + \beta_t'\mathbf{x}_1 + \gamma_t'\boldsymbol{\epsilon}$
6: Take a gradient descent step on
$\nabla_\theta\, w(t)\,\|P_{\mathbf{x}_t}\left(\mathbf{v}_\theta(\mathbf{x}_t,t) - \mathbf{v}_t\right)\|^2$
7: **until** converged

Algorithm 3 Sampling

$\mathbf{x}_0 \sim p_{\text{prior}}$
for $i = 0$ **to** $K - 1$ **do**
  $\Delta t_i = t_{i+1} - t_i$
  if ODE sampling **then**
    $\mathbf{x}_{t_{i+1}} = \mathbf{x}_{t_i} + P_{\mathbf{x}_{t_i}} \mathbf{v}_\theta(\mathbf{x}_{t_i}, t_i)\Delta t_i$
  end if
  if SDE sampling **then**
    $\mathbf{d}_i = P_{\mathbf{x}_{t_i}} (\mathbf{v}_\theta(\mathbf{x}_{t_i}, t_i) + \eta_{t_i} \mathbf{s}_\theta(\mathbf{x}_{t_i}, t_i)) + \gamma g_{t_i} \mathbf{h}(\mathbf{x}_{t_i})$
    $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathbf{x}_{t_{i+1}} = \mathbf{x}_{t_i} + \mathbf{d}_i \Delta t_i + \sqrt{2\gamma\eta_{t_i}\Delta t_i} P_{\mathbf{x}_{t_i}} \boldsymbol{\epsilon}$
  end if
end for

F Additional Experimental Results

F.1 Efficiency and Complexity Analysis

Complexity analysis. In this subsection, we give a detailed discussion on the computational cost of our method. As mentioned in Thm. 4, we need to compute the inversion of the matrix $\mathcal{I}$ and the cross product for the horizontal projection operator $P_{\mathbf{x}}$ and the mean curvature vector $\tilde{\mathbf{h}}(\mathbf{x})$. For the calculation of $\mathcal{I}^{-1}$, notice that $\mathcal{I}$ is always a 3×3 matrix, so construction cost of $\mathcal{I}^{-1}$ is only linear $O(N)$, where N is the number of atoms (linear $O(N)$ cost for constructing $\mathcal{I}$, and constant $O(1)$ cost for inversion). The cross product is conducted atom-wise, so its computational cost is also linear $O(N)$. So we can conclude that the overall computational complexity is $O(N)$ for both $P_{\mathbf{x}}$ and $\tilde{\mathbf{h}}(\mathbf{x})$.

We would like to mention that the alignment operation adopted in the heuristic alignment-based diffusion strategies also has the same complexity. To see this, for aligning $\mathbf{x} \in \mathbb{R}^{3\times N}$ towards $\mathbf{y} \in \mathbb{R}^{3\times N}$, the Kabsch-Umeyama algorithm constructs the optimal rotation matrix as $(\mathbf{H}^T\mathbf{H})^{\frac{1}{2}}\mathbf{H}^{-1}$, where $\mathbf{H} := \mathbf{y}\mathbf{x}^T \in \mathbb{R}^{3\times 3}$ requires a linear $O(N)$ cost. In practice, the $O(N)$ computational cost is negligible compared to the cost of gradient back-propagation through the neural network. A comparison of practical training times is shown in the following table.

Methods	Original diffusion	GeoDiff alignment	Af3 alignment	Quotient-space diffusion
training speed (iters/s)	4.19	4.07	4.08	4.10

All the results are tested on a single Nvidia A100 GPU. From the results, we can see that the additional computational cost brought by the alignment and projection is negligible.

Numerical stability. In our quotient-space diffusion model framework, we need to calculate the matrix inversion of $\mathcal{I}$, which may have numerical issues for near-collinear systems of points. In practice, we add an $\epsilon\mathbf{I}$ term before conducting matrix inversion, that is, we calculate $(\epsilon\mathbf{I} + \mathcal{I})^{-1}$ in practice, where $\mathbf{I}$ is the 3×3 identity matrix. This treatment is widely adopted in algorithms facing similar situations, e.g., the practical implementation of the Kabsch-Umeyama algorithm for alignment. Our typical choice of ϵ is 1e-8, and we found that the training process is stable under this setting. We have shown the training curve of the model on the protein backbone generation task in Fig. 4, which indicates no numerical issues arise during the training process.

F.2 The Implementation of $\mathcal{G}$-equivariant Vector Field

In Thm. 4, we require that the vector field is $\mathrm{SO}(3)$-equivariant. In practice, this can be implemented by using a $\mathrm{SO}(3)$-equivariant network architecture or applying data augmentation. In this subsection, we justify that both of these choices are valid, such that the diffusion model can generate a $\mathrm{SO}(3)$-invariant distribution.

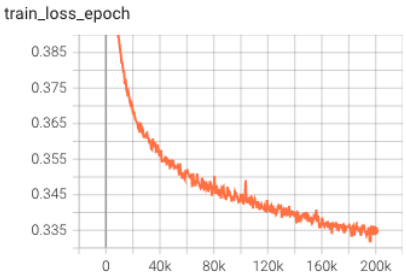

Figure 4: Training loss vs. training epochs. We find that our training is stable in practice.

Diffusion model with data augmentation. The optimal solution of the Euclidean diffusion model is given by $\mathbf{D}^*_\theta(\mathbf{x}_t) = \mathbb{E}[\mathbf{x}_1|\mathbf{x}_t]$(Song et al., 2021; Karras et al., 2022). When the data distribution is augmented by random rotation, the data distribution becomes SO(3)-invariant. Thus, the optimal diffusion model can recover the SO(3)-invariant data distribution. When the transition density $p(\mathbf{x}_t|\mathbf{x}_1)$ is SO(3)-equivariant, i.e. $p(\mathbf{x}_t|\mathbf{x}_1) = p(g\cdot\mathbf{x}_t|g\cdot\mathbf{x}_1), \forall g \in \mathrm{SO}(3)$, the optimal network is SO(3)-equivariant. To see this, let $g \in \mathrm{SO}(3)$ be an arbitrary rotation matrix. Since $\mathbf{D}^*_\theta(g\cdot\mathbf{x}_t) = \mathbb{E}[\mathbf{x}_1|g\cdot\mathbf{x}_t]$, by the Bayes formula,

$$\begin{aligned}
\mathbb{E}[\mathbf{x}_1|g\cdot\mathbf{x}_t] &= \frac{\mathbb{E}_{p_{\text{target}}(\mathbf{x}_1)}[\mathbf{x}_1 p(g\cdot\mathbf{x}_t|\mathbf{x}_1)]}{\mathbb{E}_{p_{\text{target}}(\mathbf{x}_1)}[p(g\cdot\mathbf{x}_t|\mathbf{x}_1)]} \\
&= \frac{\mathbb{E}_{p_{\text{target}}(\mathbf{x}_1)}[\mathbf{x}_1 p(\mathbf{x}_t|g^{-1}\cdot\mathbf{x}_1)]}{\mathbb{E}_{p_{\text{target}}(\mathbf{x}_1)}[p(\mathbf{x}_t|g^{-1}\mathbf{x}_1)]} \\
&= \frac{g\cdot\mathbb{E}_{p_{\text{target}}(g^{-1}\mathbf{x}_1)}[g^{-1}\mathbf{x}_1 p(\mathbf{x}_t|g^{-1}\cdot\mathbf{x}_1)]}{\mathbb{E}_{p_{\text{target}}(g^{-1}\mathbf{x}_1)}[p(\mathbf{x}_t|g^{-1}\mathbf{x}_1)]} \\
&= g\cdot\mathbb{E}[\mathbf{x}_1|\mathbf{x}_t],
\end{aligned}$$

where we use the equivariance property of the transition density to get the second equality and the invariance property of p_{target} to get the third equality. Thus, we can conclude that the optimal solution under these conditions is SO(3)-equivariant. Geffner et al. (2025) also gives an empirical validation that a well-trained neural network becomes nearly equivariant even if its architecture is not equivariant.

Equivariant architecture. When the model is required to be SO(3)-equivariant, the optimal solution of the diffusion model is not $\mathbb{E}[\mathbf{x}_1|\mathbf{x}_t]$. To figure out the optimal solution, we consider the training loss at time t. The loss function at t is given by

$$\begin{aligned}
\mathcal{L}_t(\theta) &= \mathbb{E}\|\mathbf{D}_\theta(\mathbf{x}_t, t) - \mathbf{x}_1\|^2 \\
&= \int \mathrm{d}^{3N}\mathbf{x}_1 \int \mathrm{d}^{3N}\mathbf{x}_t \quad p(\mathbf{x}_1, \mathbf{x}_t)\left(\|\mathbf{D}_\theta(\mathbf{x}_t, t)\|^2 + \|\mathbf{x}_1\|^2 - 2\langle\mathbf{D}_\theta(\mathbf{x}_t, t), \mathbf{x}_1\rangle\right).
\end{aligned}$$

The optimal solution satisfies

$$\mathbf{D}^*_\theta(\mathbf{x}_t, t) = \operatorname*{argmin}_{\mathbf{D}_\theta \text{ is SO(3) equivariant}} \mathcal{L}_t(\theta).$$

The training loss can be simplified using the equivariant constraint:

$$\begin{aligned}
\mathcal{L}_t(\theta) &= \int \mathrm{d}^{3N}\mathbf{x}_1 \int \mathrm{d}^{3N}\mathbf{x}_t\, p(\mathbf{x}_1, \mathbf{x}_t)\left(\|\mathbf{D}_\theta(\mathbf{x}_t)\|^2 + \|\mathbf{x}_1\|^2 - 2\langle\mathbf{D}_\theta(\mathbf{x}_t), \mathbf{x}_1\rangle\right) \\
&= \int_{\mathbb{R}^{3N}/\mathrm{SO}(3)} \mathrm{d}\mathbf{r}_t \int_{\mathrm{SO}(3)} \mathrm{d}g \int \mathrm{d}^{3N}\mathbf{x}_1\, p(\mathbf{x}_1, g\cdot\mathbf{r}_t)\left(\|\mathbf{D}_\theta(g\cdot\mathbf{r}_t)\|^2 + \|\mathbf{x}_1\|^2 - 2\langle\mathbf{D}_\theta(g\cdot\mathbf{r}_t), \mathbf{x}_1\rangle\right).
\end{aligned}$$

Since $\mathbf{D}_\theta$ is SO(3)-equivariant, $\mathbf{D}_\theta(g\cdot\mathbf{r}_t) = g\cdot\mathbf{D}_\theta(r_t)$, then we have

$$\mathcal{L}_t(\theta) = \int_{\mathbb{R}^{3N}/\mathrm{SO}(3)} \mathrm{d}\mathbf{r}_t \int_{\mathrm{SO}(3)} \mathrm{d}g \int \mathrm{d}^{3N}\mathbf{x}_1\, p(\mathbf{x}_1, g\cdot\mathbf{r}_t)\left(\|\mathbf{D}_\theta(\mathbf{r}_t)\|^2 + \|\mathbf{x}_1\|^2 - 2\langle g\cdot\mathbf{D}_{1\theta}(\mathbf{r}_t), \mathbf{x}_1\rangle\right).$$

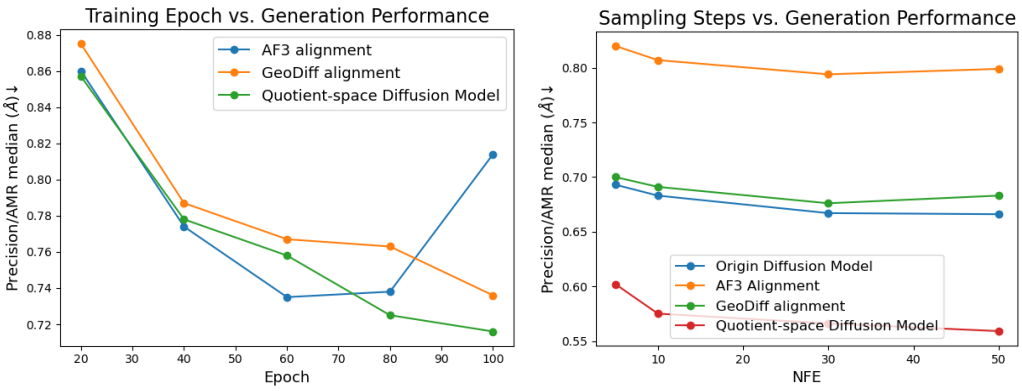

Figure 5: Training and sampling convergence speed comparison on GEOM-DRUGS. **(Left)** The relationship between training epochs and generation performance measured by the precision AMR median metric. **(Right)** The relationship between the number of function evaluations (NFE) for sampling and generation performance measured by the precision AMR median metric.

Define $p(\mathbf{r}_t) = \int_{\mathrm{SO}(3)} \mathrm{d}g \int \mathrm{d}^{3N}\mathbf{x}_1 \; p(\mathbf{x}_1, g \cdot \mathbf{r}_t)$, and $p(\mathbf{x}_1, g \mid \mathbf{r}_t) = \frac{p(\mathbf{x}_1, g \cdot \mathbf{r}_t)}{p(\mathbf{r}_t)}$. Then we have

$$\mathcal{L}_t(\theta) = \int_{\mathbb{R}^{3N}/\mathrm{SO}(3)} \mathrm{d}\mathbf{r}_t \left[p(\mathbf{r}_t) \|\mathbf{D}_\theta(\mathbf{r}_t)\|^2 - 2\langle \mathbf{D}_\theta(\mathbf{r}_t), \int_{\mathrm{SO}(3)} \mathrm{d}g \int \mathrm{d}^{3N}\mathbf{x}_1 \; p(\mathbf{x}_1, g \cdot \mathbf{r}_t) g^{-1} \cdot \mathbf{x}_1 \rangle \right]$$
$$+ \int_{\mathbb{R}^{3N}/\mathrm{SO}(3)} \mathrm{d}\mathbf{r}_t \int_{\mathrm{SO}(3)} \mathrm{d}g \int \mathrm{d}^{3N}\mathbf{x}_1 \; p(\mathbf{x}_1, g\mathbf{r}_t) \|\mathbf{x}_1\|^2.$$

So we can conclude that

$$\mathbf{D}^*_\theta(\mathbf{r}_t, t) = \int_{\mathrm{SO}(3)} \mathrm{d}g \int \mathrm{d}^{3N}\mathbf{x}_1 \;\; p(\mathbf{x}_1, g \mid \mathbf{r}_t) \; g^{-1} \cdot \mathbf{x}_1,$$
$$\mathbf{D}^*_\theta(g' \cdot \mathbf{r}_t) = \int_{\mathrm{SO}(3)} \mathrm{d}g \int \mathrm{d}^{3N}\mathbf{x}_1 \;\; p(\mathbf{x}_1, g \mid \mathbf{r}_t) \; g' \cdot g^{-1} \cdot \mathbf{x}_1, \forall g \in \mathrm{SO}(3).$$

Notice that

$$\begin{aligned}
\mathbf{D}^*_\theta(\mathbf{r}_t) &= \int_{\mathrm{SO}(3)} \mathrm{d}g \int \mathrm{d}^{3N}\mathbf{x}_1 \;\; p(\mathbf{x}_1, g \mid \mathbf{r}_t) \; g^{-1} \cdot \mathbf{x}_1 \\
&= \frac{\int_{\mathrm{SO}(3)} \mathrm{d}g \int \mathrm{d}^{3N}\mathbf{x}_1 \;\; p(g \cdot \mathbf{x}_1) p(g \cdot \mathbf{r}_t \mid g \cdot \mathbf{x}_1) \mathbf{x}_1}{\int_{\mathrm{SO}(3)} \mathrm{d}g \int \mathrm{d}^{3N}\mathbf{x}_1 \;\; p(g \cdot \mathbf{x}_1) p(g \cdot \mathbf{r}_t \mid g \cdot \mathbf{x}_1)} \\
&= \frac{\int_{\mathrm{SO}(3)} \mathrm{d}g \int \mathrm{d}^{3N}\mathbf{x}_1 \;\; p(g \cdot \mathbf{x}_1) p(\mathbf{r}_t \mid \mathbf{x}_1) \mathbf{x}_1}{\int_{\mathrm{SO}(3)} \mathrm{d}g \int \mathrm{d}^{3N}\mathbf{x}_1 \;\; p(g \cdot \mathbf{x}_1) p(\mathbf{r}_t \mid \mathbf{x}_1)},
\end{aligned}$$

which is equivalent to the case $p_{\text{data}} = \int_{\mathrm{SO}(3)} \mathrm{d}g \;\; p(g \cdot \mathbf{x}_1)$, i.e. using the augmentation by random $\mathrm{SO}(3)$ rotation.

F.3 Training and Sampling Acceleration

In this subsection, we study the training and sampling convergence speed of different methods. For the training convergence speed comparison, we plot the generation performance measured by the precision AMR median metric with respect to the training epochs for previous heuristic alignment methods and our quotient-space diffusion model in Fig. 5(Left). We only focus on the first 100 epochs for all the methods. These models are trained with the same architecture ET-Flow($\mathrm{SO}(3)$) and training configurations on the GEOM-DRUGS dataset. The results indicate that our method achieves a similar convergence speed to the AF3 heuristic method, because both methods reduce the learning difficulty of the model, as shown in Table 1. This theoretical benefit leads to faster convergence than the GeoDiff alignment method. We also notice that the AF3 alignment method starts to get worse generation performance after 80 training epochs. This happens due to the incompatibility

between the training loss and the generation performance metric, as the AF3 method is originally designed for the protein structure prediction task, which is not evaluated by distributional metrics.

For the sampling convergence speed comparison, we plot the generation performance measured by the precision AMR median metric with respect to the number of function evaluations (NFE) for the sampling process in Fig. 5(Right). For all these methods trained on the GEOM-DRUGS dataset, we use the Flow Matching ODE sampler (Lipman et al., 2023) with Euler discretization. From the results, we can observe that models trained with different strategies exhibit similar convergence trends (performance gradually degrades as NFE decreases), our quotient-space diffusion framework consistently outperforms all baselines across every NFE setting.

F.4 Quotient Space beyond $\mathbb{R}^{3N}/\mathrm{SE}(3)$

Our framework can generalize to quotient spaces generated by symmetry groups beyond the special Euclidean group $\mathrm{SE}(3)$. Possible examples include the $\mathrm{U}(1)$ symmetry in quantum wavefunctions, the $\mathrm{SU}(2)$ symmetry in particle physics, and the $\mathrm{SO}(3)$ symmetry in higher (> 3) representation spaces for tasks including the mean-field electron Hamiltonian matrix prediction. In this work, we focus on the $\mathrm{SE}(3)$ case for its significant relevance to scientific research (Abramson et al., 2024). Applications of our framework on the mentioned more diverse systems above are left as future work.

G Experiments

G.1 Molecule Generation

This appendix summarizes our experimental setup, which strictly follows that of Etflow (Hassan et al., 2024). We detail the datasets, model architecture, training, sampling, and evaluation. For a more comprehensive discussion of each component, we refer the reader to the appendices of their original paper.

Dataset. First, we evaluate our framework on the molecule structure generation task. In this scenario, our goal is to generate the 3D coordinates of a molecule given the graph structure of the molecule. We conduct the experiments on the GEOM datasets (Axelrod & Gomez-Bombarelli, 2022), which provide structure ensembles generated by metadynamics in CREST (Pracht et al., 2024), and we focus on the GEOM-QM9 and GEOM-DRUGS datasets. Following the data processing and splits from (Hassan et al., 2024), we use the random splits with train/validation/test of 243473/30433/1000 for GEOM-DRUGS and 106586/13323/1000 for GEOM-QM9. In addition, data with disconnected molecule graphs are removed for GEOM-DRUGS (Hassan et al., 2024). Our reproduction is based on the modified data-processing pipeline following the released configs thus different from the results reported in the original paper.

Settings. We primarily follow the setting in (Hassan et al., 2024). We set the Gaussian distribution as the prior distribution on GEOM-QM9 and use the harmonic prior for GEOM-DRUGS (Volk et al., 2023). Following (Jing et al., 2022; Xu et al., 2022), we report the RMSD-based metrics, e.g. Coverage and Average Minimum RMSD (AMR) between generated and ground truth structure ensembles. We parameterize $\mathbf{v}_\theta$ by using equivariant graph transformer architectures from ET-Flow (Hassan et al., 2024), including the O(3) and SO(3) equivariant variants, which also serves as a verification that our framework is compatible with different backbone models. For training, we use AdamW as the optimizer, and set the hyper-parameter ϵ to 1e-8 and (β_1, β_2) to (0.9,0.999). We use the dynamic gradient clipping as (Hassan et al., 2024; Hoogeboom et al., 2022b). The peak learning rate is set to 5e-4 for GEOM-DRUGS and 7e-4 for GEOM-QM9. The batch size is set to 48 for GEOM-DRUGS and 128 for GEOM-QM9. The weight decay is set to 1e-8. The model is trained for 1000 epochs for both datasets. The noise scale σ is set to 0.1. We also use 50 time steps with the Euler solver for sampling. All models are trained on 8 NVIDIA A100 GPUs.

Baselines. Following (Hassan et al., 2024), we choose strong baselines trained on GEOM-DRUGS and GEOM-QM9 for a challenging comparison. We report the performance of GeoMol (Ganea et al., 2021), GeoDiff (Xu et al., 2022), Torsional Diffusion (Jing et al., 2022), and MCF (Wang et al., 2023).

G.2 PROTEIN

This appendix summarizes our experimental setup, which strictly follows that of Proteína (Geffner et al., 2025). We detail the datasets, model architecture, training, sampling, and evaluation. For a more comprehensive discussion of each component, we refer the reader to the appendices of their original paper.

G.2.1 DATASET

For training, we utilize the Foldseek AFDB clusters (D_{FS}) dataset as curated and described in the Proteína. This dataset is a high-quality, non-redundant subset of the AlphaFold Database (AFDB), containing 588,318 cluster-representative protein structures with lengths between 32 and 256 residues. The dataset is annotated with hierarchical CATH labels, which are leveraged during training. Our data processing and handling strictly follow the pipeline detailed in Appendix M of (Geffner et al., 2025).

G.2.2 MODEL ARCHITECTURE AND TRAINING

Our model architecture is the same as the efficient, non-equivariant transformer proposed by (Geffner et al., 2025). Specifically, we adopt the variant that forgoes the use of computationally expensive triangle update layers. The model is trained using the conditional flow matching (CFM) objective. Key aspects of the training protocol from Proteína are preserved, including their novel Beta-Uniform mixture for the time-sampling distribution $p(t)$, the use of self-conditioning, and data augmentation with random rotations. All model and training hyperparameters, such as embedding dimensions, number of layers, attention heads, and optimizer settings, are kept consistent with hyperparameters saved in their released checkpoint $\mathcal{M}_{\text{FS}}^{\text{small}}$. The hyperparameters for the $\mathcal{M}_{\text{FS}}^{\text{small}}$ model are detailed in Table 5, in comparison with the larger models from the original Proteína paper.

Table 5: Hyperparameters for Proteína model.

Hyperparameter	$\mathcal{M}_{\text{FS}}$	$\mathcal{M}_{\text{FS}}^{\text{no-tri}}$	$\mathcal{M}_{\text{FS}}^{\text{small}}$
Proteína Architecture			
sequence repr dim	768	768	512
# registers	10	10	10
sequence cond dim	512	512	128
t sinusoidal enc dim	256	256	196
idx. sinusoidal enc dim	128	128	196
fold emb dim	256	256	196
pair repr dim	512	512	196
seq separation dim	128	128	128
pair distances dim (x_t)	64	64	64
pair distances dim ($\tilde{x}(x_t)$)	128	128	128
pair distances min (Å)	1	1	1
pair distances max (Å)	30	30	30
# attention heads	12	12	12
# tranformer layers	15	15	12
# triangle layers	5	—	—
# trainable parameters	200M	200M	60M
Proteína Training			
# steps	200K	360K	150K
batch size per GPU	4	10	5
# GPUs	128	96	16
# grad. acc. steps	1	1	1

G.2.3 SAMPLING

To facilitate a direct comparison with the publicly available Proteína checkpoints, we trained our model with an identical hierarchical fold class conditioning mechanism. However, to ensure a fair assessment of foundational generative capabilities, all experiments reported in our main text were

performed in a strictly unconditional setting. We applied the same sampling protocol across all models, using 400 sampling steps and enabling self-conditioning, which consistently improved performance. No other guidance techniques, such as autoguidance, were utilized. We use deterministic ODE sampling to assess distributional fidelity and SDE sampling to explore the designability-diversity trade-off. We adapt the SDE formulation and its Euler-Maruyama numerical scheme, detailed in Appendix I of (Geffner et al., 2025), for our quotient space framework, while retaining all other configurations, such as the sampling scheduler and $g(t)$, from the original paper.

G.2.4 Evaluation

We evaluate our models rigorously adheres to the metrics established and validated in the Proteína paper. We assess model performance using the standard suite of metrics in protein design:

- **Designability.** Quantified by the self-consistency RMSD (scRMSD) protocol, using ProteinMPNN for inverse folding and ESMFold for structure prediction, with a success threshold of scRMSD less than 2Å.
- **Diversity.** Measured in two ways: by the average pairwise TM-score among designable samples, and by the number of distinct structural clusters identified by Foldseek at a TM-score threshold of 0.5.
- **Novelty.** Assessed by calculating the maximum TM-score of each designable sample against reference structures in the PDB and AFDB databases.

We also adopt the novel probabilistic metrics introduced by (Geffner et al., 2025), to measure how well our model captures the true distribution of protein structures:

- **FPSD.** Measured the distributional similarity between generated and reference structures in the feature space of a pre-trained fold class predictor.
- **fS.** Evaluated both the quality and diversity of samples based on the confidence and entropy of fold class predictions.
- **fJSD.** Quantified the similarity between the categorical fold class distributions of generated and reference sets.

It is noteworthy that we have omitted the Diversity and Novelty metrics from our main text to avoid comparisons with potentially inaccurate results in the literature. This decision is based on a bug recently identified in the alntmscore output of FoldSeek versions prior to v10 (release 10-941cd33), which renders many previously reported TM-based metrics incorrect (also found in (Daras et al., 2025)). To provide a controlled and accurate benchmark, we conducted our own analysis using the FoldSeek v10 (release 10-941cd33). We limited this re-evaluation to the released small Proteína model and our corresponding model trained in the quotient space. The full results of this comparison are summarized in Table 6.

Table 6: Complete performance comparison of the released Proteína checkpoints against our version in the quotient space. Best results are marked in **bold**.

Model	Designability (%)	Diversity		Novelty vs.		FPSD vs.		fS	fJSD vs.	
		Cluster↑	TM-Sc.↓	PDB↓	AFDB↓	PDB↓	AFDB↓	(C/A/T)↑	PDB↓	AFDB↓
SDE Sampling										
$\mathcal{M}_{\text{FS}}^{\text{small}}, \gamma = 0.35$	96.0	0.44 (209)	0.50	0.86	0.91	386.5	378.2	1.77/4.97/17.78	2.17	1.73
$\mathcal{M}_{\text{FS}}^{\text{small}}, \gamma = 0.35$ + ours	**97.6**	0.40 (197)	0.48	0.86	0.91	274.7	277.1	2.24/6.69/20.99	1.68	1.55
$\mathcal{M}_{\text{FS}}^{\text{small}}, \gamma = 0.45$	92.2	0.55 (253)	0.49	0.84	0.90	332.9	320.4	1.83/5.01/20.22	1.93	1.49
$\mathcal{M}_{\text{FS}}^{\text{small}}, \gamma = 0.45$ + ours	92.6	0.51 (253)	0.47	0.85	0.90	244.5	246.3	2.24/6.68/23.47	1.43	1.28
$\mathcal{M}_{\text{FS}}^{\text{small}}, \gamma = 0.50$	89.2	0.57 (255)	0.48	0.83	0.89	306.2	290.8	1.86/4.92/21.15	1.81	1.36
$\mathcal{M}_{\text{FS}}^{\text{small}}, \gamma = 0.50$ + ours	90.2	0.51 (231)	0.47	0.84	0.90	228.0	228.7	2.25/6.59/25.24	1.32	1.17
ODE Sampling										
$\mathcal{M}_{\text{FS}}^{\text{small}}$	13.8	0.90 (62)	0.43	0.80	0.87	83.18	21.93	2.45/5.63/31.76	0.58	0.12
$\mathcal{M}_{\text{FS}}^{\text{small}}$ + ours	15.6	0.87 (68)	0.43	0.80	0.86	**69.94**	**17.56**	**2.57/6.40/32.14**	**0.41**	0.11