
Geometric Trajectory Diffusion Models

Jiaqi Han, Minkai Xu, Aaron Lou, Haotian Ye, Stefano Ermon
Stanford University

Abstract

Generative models have shown great promise in generating 3D geometric systems, which is a fundamental problem in many natural science domains such as molecule and protein design. However, existing approaches only operate on static structures, neglecting the fact that physical systems are always dynamic in nature. In this work, we propose geometric trajectory diffusion models (GeoTDM), the first diffusion model for modeling the temporal distribution of 3D geometric trajectories. Modeling such distribution is challenging as it requires capturing both the complex spatial interactions with physical symmetries and temporal correspondence encapsulated in the dynamics. We theoretically justify that diffusion models with equivariant temporal kernels can lead to density with desired symmetry, and develop a novel transition kernel leveraging SE(3)-equivariant spatial convolution and temporal attention. Furthermore, to induce an expressive trajectory distribution for conditional generation, we introduce a generalized learnable geometric prior into the forward diffusion process to enhance temporal conditioning. We conduct extensive experiments on both unconditional and conditional generation in various scenarios, including physical simulation, molecular dynamics, and pedestrian motion. Empirical results on a wide suite of metrics demonstrate that GeoTDM can generate realistic geometric trajectories with significantly higher quality.¹

1 Introduction

Machine learning for geometric structures is a fundamental task in many natural science problems ranging from particle systems driven by physical laws [1, 26, 43, 2, 15] to molecular dynamics in biochemistry [22, 16, 45, 10]. Modeling such geometric data is challenging due to the physical symmetry constraint [56, 43], making it fundamentally different from common scalar non-geometric data such as images and text. With the recent progress of generative models, many works have been proposed in generating 3D geometric structures like small molecules [66, 42, 21, 64] and proteins [59, 23], showing great promise in solving the equilibrium states of complex systems.

Despite this success, these existing methods are limited to synthesizing static structures and neglect the fact that important real-world processes evolve through time. For example, molecules and proteins are not static but always varying with molecular dynamics, which plays a vital role in analyzing possible binding activities [8, 20]. In this paper, we aim to study the generative modeling of geometric trajectories with the additional temporal dimension. While this problem is more practical and important, it is highly non-trivial with several significant challenges. First, geometric dynamics in 3D ubiquitously preserve physical symmetry. With global translation or rotation applied to a trajectory of molecular dynamics, the entire trajectory still describes the same dynamics and the generative model should estimate the same likelihood. Second, trajectories inherently contain the correspondence between frames in different timesteps, requiring generative models to hold a high capacity for capturing the temporal correlations. Last, moving from a single structure to a trajectory

¹Correspondence to Jiaqi Han: jiaqihan@stanford.edu. Code is available at <https://github.com/hanjq17/GeoTDM>.

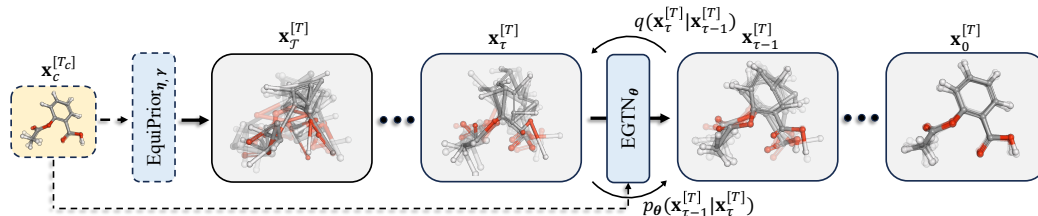


Figure 1: Overview of GeoTDM. The forward diffusion q gradually perturbs the input while the reverse process p_{θ} , parameterized by EGTN, denoises samples from the prior. The condition $\mathbf{x}_c^{[T_c]}$, if available, is leveraged to construct the equivariant prior and as a conditioning signal in EGTN.

composed of multiple ones, the distribution we are interested in becomes much higher-dimensional and more diverse, considering both the initial conditions as well as potential uncertainties injected along the evolution of dynamics.

To this end, we propose geometric trajectory diffusion models (GeoTDM), a principled method for modeling the temporal distribution of geometric trajectories through diffusion models [49, 52, 53, 18], the state-of-the-art generative model on various domains such as images [6], videos [19], and molecules [64]. Our key innovation lies in designing an equivariant temporal diffusion over geometric trajectories, with the reverse process parameterized by equivariant transition kernels, ensuring the desired physical symmetry of the generated trajectory. To better excavate the complex spatial interactions and temporal correlations, we develop a novel temporal denoising network, where we stack equivariant spatial convolution and temporal attention. Our developments not only guarantee the desirable physical symmetry of the trajectories, but also capture the complex spatial and temporal correspondence encapsulated in the dynamics of geometric systems. Moreover, by leveraging generative modeling, GeoTDM enjoys high versatility in generating diverse yet high-quality geometric trajectories from scratch, performing interpolation and extrapolation, and optimizing noisy trajectory, all under the proposed diffusion framework.

In summary, we make the following contributions: **1.** We present GeoTDM, a novel temporal diffusion model for generating geometric trajectories. We design the diffusion process to meet the critical equivariance in modeling both unconditional and conditional distributions over geometric trajectories. Notably, we also propose a conditional learnable equivariant prior for enhanced flexibility in temporal conditioning. **2.** To fulfill the equivariance of the denoising network, we introduce EGTN, a graph neural network that operates on geometric trajectories, which also permits conditioning upon a given trajectory using equivariant cross-attention, making it suitable to serve as the backbone for GeoTDM. **3.** We evaluate our GeoTDM on both unconditional and conditional trajectory generation tasks including particle simulation, molecular dynamics, and pedestrian trajectory prediction. GeoTDM can consistently outperform existing approaches on various metrics, with up to 56.7% lower prediction score for unconditional generation and 16.8% lower forecasting error for conditional generation on molecular dynamics simulation. We also show GeoTDM successfully performs several additional applications, such as temporal interpolation and trajectory optimization.

2 Related Work

Trajectory modeling for geometric systems. Modeling the dynamics of geometric data is challenging since one must capture the interactions between multiple objects. Graph neural networks [11] have emerged as a natural tool to tackle this complexity [26, 41]. Subsequent works [43, 7, 2, 63] discovered equivariance as a critical factor for promoting model generalization. Among these efforts, Radial Fields [27] and EGNN [43] work with equivariant operations between scalars and vectors, while TFN [56] and SE(3)-Transformer [9] generalize to high-order spherical tensors. While considerable progress has been made, they only conduct (time) frame-to-frame prediction, which is subject to error accumulation when performing roll-out inference. Recently, EqMotion [62] approached the problem by learning to predict trajectories. By comparison, our GeoTDM leverages a generative modeling framework, which enables a wider range of tasks such as generation and interpolation.

Generative models in geometric domain. There is growing interest in developing generative models for geometric data, *e.g.* molecule generation [42, 66, 21, 64], protein generation [59, 24, 69], and

antibody design [30]. Recently, diffusion-based models [21, 64] have been shown to yield superior performance compared to flow-based [42] and VAE-based [65] approaches in many of these tasks. Despite these fruitful achievements, most existing works only produce a snapshot of the geometric system, *e.g.*, a molecule in 3D space, whereas our GeoTDM generalizes to generating a trajectory with multiple frames, *e.g.*, an MD trajectory in 3D. DiffMD [60] specifically tackles MD modeling using Markovian assumption, while GeoTDM directly captures the joint distribution of all frames along the entire trajectory.

Temporal diffusion models. Diffusion models have been recently adapted to handle the natural temporality of data in tasks such as video generation [19, 58, 17], time series forecasting [37, 54], PDE simulation [38], human motion synthesis [55, 71] and pedestrian trajectory forecasting [12]. Distinct from these works, GeoTDM models the temporal evolution of geometric data represented as a geometric graph and maintains the aforementioned vital equivariance constraint.

3 Preliminaries

Diffusion models. Diffusion models [49, 18, 52, 53] are a type of latent variable generative model that feature a Markovian forward diffusion process and reverse denoising process. The forward process progressively perturbs the input \mathbf{x}_0 (*e.g.*, image pixels or molecule coordinates) over \mathcal{T} steps using a Gaussian transition kernel $q(\mathbf{x}_\tau|\mathbf{x}_{\tau-1}) = \mathcal{N}(\mathbf{x}_\tau; \sqrt{1-\beta_\tau}\mathbf{x}_{\tau-1}, \beta_\tau\mathbf{I})$. Here, $\{\mathbf{x}_\tau\}_{\tau=1}^{\mathcal{T}}$ are latent variables with the same dimension as the input and β_τ are predefined using the noise schedule such that $\mathbf{x}_\mathcal{T}$ is close to being distributed as $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The reverse process maps back from the prior distribution with $p(\mathbf{x}_\mathcal{T}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ using the kernel $p_\theta(\mathbf{x}_{\tau-1}|\mathbf{x}_\tau) = \mathcal{N}(\mathbf{x}_{\tau-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_\tau, \tau), \sigma_\tau^2\mathbf{I})$, where the variances σ_τ^2 are usually fixed and the mean $\boldsymbol{\mu}_\theta$ is parameterized by a neural network with parameters θ . The model is trained by optimizing the variational lower bound, defined as $\mathcal{L}_{\text{vlb}} = -\log p_\theta(\mathbf{x}_0|\mathbf{x}_1) + D_{\text{KL}}(q(\mathbf{x}_\mathcal{T}|\mathbf{x}_0)||p(\mathbf{x}_\mathcal{T})) + \sum_{\tau=2}^{\mathcal{T}-1} D_{\text{KL}}(q(\mathbf{x}_{\tau-1}|\mathbf{x}_\tau, \mathbf{x}_0)||p_\theta(\mathbf{x}_{\tau-1}|\mathbf{x}_\tau))$. For training stability, [52, 18] suggest the noise-prediction objective:

$$\mathcal{L}_{\text{simple}} := \mathbb{E}_{\mathbf{x}_0, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tau} \lambda(\tau) [\|\epsilon - \epsilon_\theta(\mathbf{x}_\tau, \tau)\|^2], \quad (1)$$

where $\mathbf{x}_0 \sim p_{\text{data}}$, $\tau \sim \text{Unif}(1, \mathcal{T})$, the weighting factors $\lambda(\tau)$ are typically set to 1 to promote sample quality, $\mathbf{x}_\tau = \sqrt{\bar{\alpha}_\tau}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_\tau}\epsilon$ with $\bar{\alpha}_\tau := \prod_{s=1}^{\tau} \alpha_s = \prod_{s=1}^{\tau} (1-\beta_s)$, and ϵ_θ is a specific parameterization of the mean satisfying $\boldsymbol{\mu}_\theta(\mathbf{x}_\tau, \tau) = \frac{1}{\sqrt{\alpha_\tau}}(\mathbf{x}_\tau - \frac{\beta_\tau}{\sqrt{1-\bar{\alpha}_\tau}}\epsilon_\theta(\mathbf{x}_\tau, \tau))$.

Equivariance. Functions. A function f is equivariant *w.r.t* a group G if $f(g \cdot \mathbf{x}) = g \cdot f(\mathbf{x}), \forall g \in G$. Furthermore, f is invariant if $f(g \cdot \mathbf{x}) = f(\mathbf{x}), \forall g \in G$ [46]. Here we focus on the group $\text{SE}(3)$ consisting of all 3D rotations and translations². Each group element $g \in \text{SE}(3)$ can be represented by a rotation matrix \mathbf{R} and a translation $\mathbf{r} \in \mathbb{R}^3$. For geometric graph with node features \mathbf{h} and coordinates \mathbf{x} , if $\mathbf{h}', \mathbf{x}' = f(\mathbf{h}, \mathbf{x})$, we expect $\mathbf{h}', \mathbf{R}\mathbf{x}' + \mathbf{r} = f(\mathbf{h}, \mathbf{R}\mathbf{x} + \mathbf{r})$ ³, *i.e.*, the output node features are invariant while the updated coordinates are equivariant. **Distributions.** We call a density $p(\mathbf{x})$ invariant *w.r.t* a group G if $p(g \cdot \mathbf{x}) = p(\mathbf{x}), \forall g \in G$. Intuitively, geometries that are rotationally and translationally equivalent should share the same density, since they all refer to the same structure. A conditional distribution $p(\mathbf{x}|\mathbf{y})$ is equivariant if $p(g \cdot \mathbf{x}|g \cdot \mathbf{y}) = p(\mathbf{x}|\mathbf{y}), \forall g \in G$. Such a property is important in cases where the target distribution is conditioned on some given structures: if the observed geometry is rotated/translated, the target distribution should also rotate/translate accordingly.

Geometric trajectories and the distributions. We represent a geometric trajectory as $(\mathbf{x}^{[T]}, \mathbf{h}, \mathcal{E})$, where $\mathbf{x}^{[T]} := [\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(T-1)}] \in \mathbb{R}^{T \times N \times D_x}$ is the sequence of temporal geometric coordinates, $\mathbf{h} \in \mathbb{R}^{N \times D_h}$ is the node feature, and \mathcal{E} is the set of edges representing the connectivity of the geometric graph. T is the number of time steps and D_x, D_h refers to the dimension of the coordinate and node feature respectively, with D_x normally being 2 or 3 depending on the input data. In this work, we are interested in modeling the distribution of geometric trajectories given the configuration of the geometric graph, *i.e.*, $p(\mathbf{x}^{[T]}|\mathbf{h}, \mathcal{E})$.

Conditioning. Some applications like trajectory forecasting can be viewed as conditional generative tasks, where we seek to model the distribution of trajectories conditioning on certain observed timesteps, *i.e.*, $p(\mathbf{x}^{[T]}|\mathbf{x}_c^{[T_c]}, \mathbf{h}, \mathcal{E})$ where $\mathbf{x}_c^{[T_c]} \in \mathbb{R}^{T_c \times N \times D_x}$ is the provided trajectory in length T_c .

²The analyses in this paper also hold for the general nD case, *e.g.*, 2D.

³Following convention we use the notation $\mathbf{R}\mathbf{x}$ to denote $\mathbf{x}\mathbf{R}^\top$.

Equivariance for geometric trajectories. Since the dynamics must be invariant to rotation or translation, the distribution of the geometric trajectories should also preserve such symmetry. This is formalized by the following invariance constraint:

$$p(\mathbf{x}^{[T]} \mid \mathbf{h}, \mathcal{E}) = p(g \cdot \mathbf{x}^{[T]} \mid \mathbf{h}, \mathcal{E}), \forall g \in \text{SE}(3). \quad (2)$$

where $g \cdot \mathbf{x}^{[T]} := [\mathbf{R}\mathbf{x}^{(0)} + \mathbf{r}, \dots, \mathbf{R}\mathbf{x}^{(T-1)} + \mathbf{r}]$. The conditional case should instead preserve:

$$p(\mathbf{x}^{[T]} \mid \mathbf{x}_c^{[T_c]}, \mathbf{h}, \mathcal{E}) = p(g \cdot \mathbf{x}^{[T]} \mid g \cdot \mathbf{x}_c^{[T_c]}, \mathbf{h}, \mathcal{E}), \quad (3)$$

for all $g \in \text{SE}(3)$ ⁴. Intuitively, if the given trajectory is rotated and/or translated, the distribution of the future trajectory should also rotate and/or translate by exactly the same amount. For simplicity, we omit writing the conditions \mathbf{h} and \mathcal{E} henceforth when describing the distributions of trajectories.

4 Geometric Trajectory Diffusion Models

In this section, we introduce the machinery of GeoTDM. We first present Equivariant Geometric Trajectory Network (EGTN) in § 4.1, a general purpose backbone operating on geometric trajectories while ensuring equivariance. We then present GeoTDM in § 4.2 for both unconditional and conditional generation using EGTN as the denoising network.

4.1 Equivariant Geometric Trajectory Network

Our proposed Equivariant Geometric Trajectory Network (EGTN) is constructed by stacking equivariant spatial aggregation layers and temporal attention layers in an alternated manner, drawing inspirations from spatio-temporal GNNs [70, 61]. In particular, spatial layers characterize the structural interactions within the system and temporal layers model the temporal dependencies along the trajectory. For spatial aggregation, we employ the Equivariant Graph Convolution Layer (EGCL) [43],

$$\mathbf{x}^{(t)}, \mathbf{h}^{(t)} = \text{EGCL}(\mathbf{x}^{(t)}, \mathbf{h}^{(t)}, \mathcal{E}), \forall t \in [T]. \quad (4)$$

The equivariant message passing is conducted independently for each frame $t \in [T] := \{0, 1, \dots, T-1\}$, with the goal of passing and fusing the geometric information based on the structure of the graph for each time step. Following such layer, we further develop a temporal layer equipped by self-attention, which has exhibited great promise for sequence modeling [57], to capture the temporal correlations encapsulated in the dynamics. We first compute Eqs. 5-6, where $\mathbf{q}^{(t)}$, $\mathbf{k}^{(t,s)}$, $\mathbf{v}^{(t,s)}$ are the query, key, and value, respectively. In detail, $\mathbf{q}^{(t)} = \varphi_{\mathbf{q}}(\mathbf{h}^{(t)})$, $\mathbf{k}^{(t,s)} = \varphi_{\mathbf{k}}(\mathbf{h}^{(s)}) + \psi(t-s)$, and $\mathbf{v}^{(t,s)} = \varphi_{\mathbf{v}}(\mathbf{h}^{(s)}) + \psi(t-s)$, with $\psi(t-s)$ being the sinusoidal encoding [57] of the temporal displacement $t-s$, akin to the relative positional encoding [47]. Incorporating such information is crucial since the model is supposed to distinguish different time spans between two frames on the trajectory. Moreover, compared with directly encoding the absolute time step, our design is beneficial in that it ensures the temporal shift invariance of physical processes. The update of coordinates reuse the attention coefficients $\mathbf{a}^{(t,s)}$ and the values $\mathbf{v}^{(t,s)}$,

$$\mathbf{a}^{(t,s)} = \frac{\exp(\mathbf{q}^{(t)\top} \mathbf{k}^{(t,s)})}{\sum_{u \in [T]} \exp(\mathbf{q}^{(t)\top} \mathbf{k}^{(t,u)})}, \quad (5)$$

$$\mathbf{h}^{(t)} = \mathbf{h}^{(t)} + \sum_{s \in [T]} \mathbf{a}^{(t,s)} \mathbf{v}^{(t,s)}, \quad (6)$$

$$\mathbf{x}'^{(t)} = \mathbf{x}^{(t)} + \sum_{s \in [T]} \mathbf{a}^{(t,s)} \varphi_{\mathbf{x}}(\mathbf{v}^{(t,s)})(\mathbf{x}^{(t)} - \mathbf{x}^{(s)}), \quad (7)$$

where $\varphi_{\mathbf{x}}$ is an MLP that outputs a scalar to preserve rotation equivariance. The entire network f_{EGTN} , with schematic depicted in Fig. 4, is constructed by alternating spatial and temporal layers, enjoying equivariance as desired (proof in Appendix A.4):

Theorem 4.1 (SE(3)-equivariance). *Let $\mathbf{x}^{[T]}, \mathbf{h}^{[T]} = f_{\text{EGTN}}(\mathbf{x}^{[T]}, \mathbf{h}^{[T]}, \mathcal{E})$. Then we have $g \cdot \mathbf{x}^{[T]}, \mathbf{h}^{[T]} = f_{\text{EGTN}}(g \cdot \mathbf{x}^{[T]}, \mathbf{h}^{[T]}, \mathcal{E}), \forall g \in \text{SE}(3)$.*

Geometric conditioning. In certain tasks like trajectory forecasting, we are additionally provided with some partially observed trajectories as side input. In order to leverage their geometric information,

⁴Technically, such a condition is impossible since SE(3) is noncompact, but we show that zero-centering the trajectories and enforcing SO(3)-invariance is equivalent.

we augment the unconditional EGTN with *equivariant cross-attention*, a conditioning technique tailored for geometric trajectories, and more importantly, guaranteeing the crucial equivariance in Theorem 4.1. In principle, our equivariant cross-attention resembles Eqs. 5-7, but instead computes the attention between the conditioning trajectory $\mathbf{x}_c^{[T_c]}$ and the target $\mathbf{x}^{[T]}$. In detail, the attention coefficients are recomputed as $\mathbf{a}^{(t,s)} = \frac{\exp(\mathbf{q}^{(t)\top} \mathbf{k}^{(t,s)})}{\sum_{u \in [T_c] \cup [T]} \exp(\mathbf{q}^{(t)\top} \mathbf{k}^{(t,u)})}$. The updated node feature $\mathbf{h}^{(t)}$ and coordinate $\mathbf{x}^{(t)}$ in Eqs. 6-7 are further renewed by the cross-attention terms, yielding $\mathbf{h}''^{(t)} = \mathbf{h}^{(t)} + \sum_{s \in [T_c]} \mathbf{a}^{(t,s)} \mathbf{v}^{(t,s)}$ and $\mathbf{x}''^{(t)} = \mathbf{x}^{(t)} + \sum_{s \in [T_c]} \mathbf{a}^{(t,s)} \varphi_{\mathbf{x}}(\mathbf{v}^{(t,s)})(\mathbf{x}^{(t)} - \mathbf{x}_c^{(s)})$.

4.2 Geometric Trajectory Diffusion Models

4.2.1 Unconditional Generation

For unconditional generation, we seek to model the trajectory distribution subject to the SE(3)-invariance (Eq. 2). To design a diffusion with the reverse marginal conforming to the invariance, we impose certain constraints to the prior and transition kernel, as depicted in the following theorem.

Theorem 4.2. *If the prior $p_{\mathcal{T}}(\mathbf{x}_{\mathcal{T}}^{[T]})$ is SE(3)-invariant, the transition kernels $p_{\tau-1}(\mathbf{x}_{\tau-1}^{[T]} | \mathbf{x}_{\tau}^{[T]})$, $\forall \tau \in \{1, \dots, \mathcal{T}\}$ are SE(3)-equivariant, then the marginal $p_{\tau}(\mathbf{x}_{\tau}^{[T]})$ at any step $\tau \in \{0, \dots, \mathcal{T}\}$ is also SE(3)-invariant.*

Prior in the translation-invariant subspace. Unfortunately, there is no properly normalized distribution *w.r.t.* Lebesgue measure on the ambient space $\mathcal{X} := \mathbb{R}^{T \times N \times D}$ that permits translation-invariance [43]. We instead build the prior on a translation-invariant subspace $\mathcal{X}_{\mathbf{P}} \subset \mathcal{X}$ induced by a linear transformation $\mathbf{P} \in \mathcal{X} \times \mathcal{X}$ with $\text{rank}(\mathbf{P}) = (TN - 1)D$ [36]. Specifically, we choose the prior to be the projection of the Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$ in \mathcal{X} to $\mathcal{X}_{\mathbf{P}}$ by $\mathbf{P} = \mathbf{I}_D \otimes (\mathbf{I}_{TN} - \frac{1}{TN} \mathbf{1}_{TN} \mathbf{1}_{TN}^{\top})$, which corresponds to the function $P(\mathbf{x}^{[T]}) = \mathbf{x}^{[T]} - \frac{1}{T} \sum_{t=0}^{T-1} \text{CoM}(\mathbf{x}^{(t)})$, with $\text{CoM}(\mathbf{x}^{(t)}) = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i^{(t)}$ being the center-of-mass (CoM) of the system at time t . We denote $\tilde{\mathbf{x}} := P(\mathbf{x})$. Then the resulting distribution is a restricted Gaussian (denoted $\tilde{\mathcal{N}}(\mathbf{0}, \mathbf{I})$) with the variables supported only on the subspace (see App. A.1), and more importantly, is still isotropic and thus SO(3)-invariant. To sample from the prior, one can alternatively sample $\mathbf{x}^{[T]} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \in \mathcal{X}$ and then project it to the subspace to obtain the final sample $\tilde{\mathbf{x}}^{[T]} = P(\mathbf{x}^{[T]}) \in \mathcal{X}_{\mathbf{P}}$.

Transition kernel. To be consistent with the prior, we also parameterize the transition kernel in the subspace $\mathcal{X}_{\mathbf{P}}$, given by $p_{\theta}(\tilde{\mathbf{x}}_{\tau-1}^{[T]} | \tilde{\mathbf{x}}_{\tau}^{[T]}) = \tilde{\mathcal{N}}(\tilde{\boldsymbol{\mu}}_{\theta}(\tilde{\mathbf{x}}_{\tau}^{[T]}, \tau), \sigma_{\tau}^2 \mathbf{I})$. In this way, if the mean function $\tilde{\boldsymbol{\mu}}_{\theta}(\cdot)$ is SO(3)-equivariant, then the transition kernel is also guaranteed SO(3)-equivariance. As suggested by [18], we re-parameterize $\tilde{\boldsymbol{\mu}}_{\theta}(\tilde{\mathbf{x}}_{\tau}^{[T]}, \tau) = \frac{1}{\sqrt{\alpha_{\tau}}} \left(\tilde{\mathbf{x}}_{\tau}^{[T]} - \frac{\beta_{\tau}}{\sqrt{1-\alpha_{\tau}}} \tilde{\boldsymbol{\epsilon}}_{\theta}(\mathbf{x}_{\tau}^{[T]}, \tau) \right)$, where $\tilde{\boldsymbol{\epsilon}}_{\theta} = P \circ \mathbf{f}_{\theta}$, with \mathbf{f}_{θ} being an SO(3)-equivariant adaptation of our proposed EGTN, fulfilled by subtracting the input coordinates from the output for translation invariance. The diffusion step τ is transformed via time embedding and concatenated to the invariant node features $\mathbf{h}^{[T]}$ in the input.

Training and inference. We optimize the VLB for training, which, interestingly, still has a surrogate in the noise-prediction form when specifying the factors $\lambda(\tau)$ as 1 (proof in App. A.1):

$$\mathcal{L}_{\text{uncond}} := \mathbb{E}_{\mathbf{x}_0^{[T]}, \tilde{\boldsymbol{\epsilon}} \sim \tilde{\mathcal{N}}(\mathbf{0}, \mathbf{I}), \tau \sim \text{Unif}(1, \mathcal{T})} \left[\|\tilde{\boldsymbol{\epsilon}} - \tilde{\boldsymbol{\epsilon}}_{\theta}(\tilde{\mathbf{x}}_{\tau}^{[T]}, \tau)\|^2 \right]. \quad (8)$$

The inference process is similar to [18] but with additional applications of P in intermediate steps to keep all samples in the subspace $\mathcal{X}_{\mathbf{P}}$. Details are in Alg. 1 and 2.

4.2.2 Conditional Generation

Distinct from the unconditional generation, in the conditional scenario the target distribution should instead be SE(3)-equivariant *w.r.t.* the given frames, as elucidated in Eq. 3. The following theorem describes the constraints to consider when designing the prior and transition kernel.

Theorem 4.3. *If the prior $p_{\mathcal{T}}(\mathbf{x}_{\mathcal{T}}^{[T]} | \mathbf{x}_c^{[T_c]})$ is SE(3)-equivariant, the transition kernels $p_{\tau-1}(\mathbf{x}_{\tau-1}^{[T]} | \mathbf{x}_{\tau}^{[T]}, \mathbf{x}_c^{[T_c]})$, $\forall \tau \in \{1, \dots, \mathcal{T}\}$ are SE(3)-equivariant, the marginal⁵ $p_{\tau}(\mathbf{x}_{\tau}^{[T]} | \mathbf{x}_c^{[T_c]})$, $\forall \tau \in \{0, \dots, \mathcal{T}\}$ is SE(3)-equivariant.*

⁵Marginal refers to marginalizing the intermediate states in reverse process while still conditioning on $\mathbf{x}_c^{[T_c]}$.

Flexible equivariant prior. There are in general many valid choices for the prior while satisfying SE(3)-equivariance. We provide a guidance on distinguishing feasible designs when using Gaussian-based prior in the proposition below.

Proposition 4.4. $\mathcal{N}(\boldsymbol{\mu}(\mathbf{x}_c^{[T_c]}), \mathbf{I})$ is SE(3)-equivariant w.r.t. $\mathbf{x}_c^{[T_c]}$ if $\boldsymbol{\mu}(\mathbf{x}_c^{[T_c]})$ is SE(3)-equivariant.

Proof is in App. A.2. Notably, the mean of the prior $\mathbf{x}_r^{[T]} := \boldsymbol{\mu}(\mathbf{x}_c^{[T_c]})$ naturally serves as an anchor to transit the geometric information in the provided trajectory to the target distribution we seek to model. For instance, one can choose it as a linear combination of the CoMs of the given frames, i.e., $\mathbf{x}_r^{[T]} = \mathbf{1}_{T \times N} \otimes \sum_{s \in [T_c]} w^{(s)} \hat{\mathbf{x}}_c^{(s)}$, where $\sum_{s \in [T_c]} w^{(s)} = 1$ are fixed parameters determined *a priori* [21, 13]. However, this choice does not leverage temporal consistency of the trajectory and incurs extra effort in optimization, since the model needs to learn to reconstruct the complex structures from points all located at the CoM. In contrast, we propose the following instantiation:

$$\mathbf{x}_r^{(t)} = \sum_{s \in [T_c]} \mathbf{w}^{(t,s)} \hat{\mathbf{x}}_c^{(s)}, \quad \text{s.t.} \quad \sum_{s \in [T_c]} \mathbf{w}^{(t,s)} = \mathbf{1}, \quad (9)$$

for all $t \in [T]$, where each $\mathbf{x}_r^{(t)}$ is a point-wise linear combination of $\hat{\mathbf{x}}_c^{(s)}$, an SE(3)-equivariant transformation of the conditioning frames, with $\mathbf{w}^{(t,s)} \in \mathbb{R}^N$ being the weights. We first obtain $\hat{\mathbf{x}}_c^{[T_c]}, \hat{\mathbf{h}}_c^{[T_c]} = \mathbf{f}_\eta(\mathbf{x}_c^{[T_c]}, \mathbf{h}_c^{[T_c]})$ where \mathbf{f}_η is a lightweight two layer EGTN that aims to synthesize the conditional information. The $\mathbf{w}^{(t,s)}$ is then derived as,

$$\mathbf{W}_{t,s} = [\gamma \otimes \hat{\mathbf{h}}_c^{[T_c]}]_{t,s} \in \mathbb{R}^N, \quad (10)$$

$$\mathbf{w}^{(t,s)} = \begin{cases} \mathbf{W}_{t,s} & s < T_c - 1, \\ \mathbf{1}_N - \sum_{s=0}^{T_c-2} \mathbf{W}_{t,s} & s = T_c - 1. \end{cases} \quad (11)$$

Here $\gamma \in \mathbb{R}^T$ are learnable parameters, and $\mathbf{w}^{(t,s)}$ is parameterized such that it has a sum of $\mathbf{1}_N$ when s goes through $[T_c]$ to satisfy the constraint in Eq. 9 for translation equivariance. Interestingly, as we formally illustrated in Theorem A.4, our parameterization of the prior theoretically *subsumes* the CoM-based priors [21, 13] and the fixed point-wise priors when $\gamma, \hat{\mathbf{h}}_c^{[T_c]}$, and $\hat{\mathbf{x}}_c^{[T_c]}$ reduce to specific values. Such theoretical result underscores the benefit of our design since it permits the model to dynamically update the prior, leading to better optimization. The parameters η and γ are updated during training with gradients coming from optimizing the variational lower bound.

Transition kernel. We need to modify the forward and reverse process such that they both match the proposed prior. The forward process is modified as $q(\mathbf{x}_r^{[T]} | \mathbf{x}_{r-1}^{[T]}, \mathbf{x}_c^{[T_c]}) := \mathcal{N}(\mathbf{x}_r^{[T]}; \mathbf{x}_r + \sqrt{1 - \beta_\tau}(\mathbf{x}_{r-1}^{[T]} - \mathbf{x}_r), \beta_\tau \mathbf{I})$, which ensures $q(\mathbf{x}_r^{[T]} | \mathbf{x}_c^{[T_c]})$ matches the equivariant prior \mathbf{x}_r (proof in App. A.2). The reverse transition kernel is given by $p_{r-1}(\mathbf{x}_{r-1}^{[T]} | \mathbf{x}_r^{[T]}, \mathbf{x}_c^{[T_c]}) = \mathcal{N}(\boldsymbol{\mu}_\theta(\mathbf{x}_r^{[T]}, \tau, \mathbf{x}_c^{[T_c]}), \sigma_\tau^2 \mathbf{I})$. Similar to the unconditional case, we also adopt the noise prediction objective by rewriting $\boldsymbol{\mu}_\theta(\mathbf{x}_r^{[T]}, \mathbf{x}_c^{[T_c]}, \tau) = \mathbf{x}_r^{[T]} + \frac{1}{\sqrt{\alpha_\tau}} \left(\mathbf{x}_r^{[T]} - \mathbf{x}_r^{[T]} - \frac{\beta_\tau}{\sqrt{1 - \alpha_\tau}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_r^{[T]}, \mathbf{x}_c^{[T_c]}, \tau) \right)$. The denoising network $\boldsymbol{\epsilon}_\theta$ is implemented as an EGTN but with its output subtracted by the input for translation invariance, hence the translation equivariance of $\boldsymbol{\mu}_\theta$.

Training and inference. Optimizing the VLB of our diffusion yields the following objective:

$$\mathcal{L}_{\text{cond}} := \mathbb{E}_{\mathbf{x}_0^{[T]}, \mathbf{x}_c^{[T_c]}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tau \sim \text{Unif}(1, T)} \left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_r^{[T]}, \mathbf{x}_c^{[T_c]}, \tau)\|^2 \right], \quad (12)$$

after simplification (proof in App. A.2). The training and inference procedures are in Alg. 3 and 4.

5 Experiments

We evaluate GeoTDM on N-body physical simulation, molecular dynamics, and pedestrian trajectory forecasting, in both conditional (§ 5.1) and unconditional generation (§ 5.2) scenarios. We ablate our core design choices and demonstrate additional use cases in § 5.3.

Table 1: Conditional generation on N-body. Table 2: Pedestrian trajectory forecasting on ETH-Results averaged over 5 runs, std in App. C.4. UCY. Best in **bold** and second best underlined.

	Particle		Spring		Gravity		ETH	Hotel	Univ	Zara1	Zara2	Average	
	ADE	FDE	ADE	FDE	ADE	FDE							
RF [27]	0.479	1.050	0.0145	0.0389	0.791	1.630	Linear	1.07/2.28	0.31/0.61	0.52/1.16	0.42/0.95	0.32/0.72	0.53/1.14
TFN [56]	0.330	0.754	0.1013	0.2364	0.327	0.761	SGAN [14]	0.64/1.09	0.46/0.98	0.56/1.18	0.33/0.67	0.31/0.64	0.46/0.91
SE(3)-Tr [9]	0.395	0.936	0.0865	0.2043	0.338	0.830	SoPhie [39]	0.70/1.43	0.76/1.67	0.54/1.24	0.30/0.63	0.38/0.78	0.54/1.15
EGNN [43]	0.186	0.426	<u>0.0101</u>	0.0231	0.310	0.709	PECNet [32]	0.54/0.87	0.18/0.24	0.35/0.60	0.22/0.39	0.17/0.30	0.29/0.48
EqMotion [62]	0.141	<u>0.310</u>	0.0134	0.0358	<u>0.302</u>	<u>0.671</u>	Traj++ [40]	0.54/0.94	0.16/0.28	0.28/0.55	<u>0.21/0.42</u>	<u>0.16/0.32</u>	0.27/0.50
SVAE [67]	0.378	0.732	0.0120	<u>0.0209</u>	0.582	1.101	BiTraP [68]	0.56/0.98	0.17/0.28	<u>0.25/0.47</u>	0.23/0.45	<u>0.16/0.33</u>	0.27/0.50
GeoTDM	0.110	0.258	0.0030	0.0079	0.256	0.613	MID [12]	0.50/0.76	0.16/0.24	0.28/0.49	0.25/0.41	0.19/0.35	0.27/0.45
							SVAE [67]	<u>0.47/0.76</u>	<u>0.14/0.22</u>	<u>0.25/0.47</u>	0.20/0.37	0.14/0.28	<u>0.24/0.42</u>
							GeoTDM	0.46/0.64	0.13/0.21	0.24/0.45	0.21/0.39	<u>0.16/0.30</u>	0.23/0.40

Table 3: Conditional trajectory generation on MD17. Results averaged over 5 runs (std in App. C.4).

	Aspirin		Benzene		Ethanol		Malonaldehyde		Naphthalene		Salicylic		Toluene		Uracil	
	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE
RF [27]	0.303	0.442	0.120	0.194	0.374	0.515	0.297	0.454	0.168	0.185	0.261	0.343	0.199	0.249	0.239	0.272
TFN [56]	<u>0.133</u>	0.268	<u>0.024</u>	0.049	0.201	0.414	0.184	0.386	0.072	0.098	0.115	0.223	<u>0.090</u>	0.150	0.090	0.159
SE(3)-Tr [9]	0.294	0.556	<u>0.027</u>	0.056	0.188	0.359	0.214	0.456	<u>0.069</u>	0.103	0.189	0.312	0.108	0.184	0.107	0.196
EGNN [43]	0.267	0.564	<u>0.024</u>	<u>0.042</u>	0.268	0.401	0.393	0.958	<u>0.095</u>	0.133	0.159	0.348	0.207	0.294	0.154	0.282
EqMotion [62]	0.185	<u>0.246</u>	0.029	0.043	<u>0.152</u>	<u>0.247</u>	<u>0.155</u>	0.249	0.073	<u>0.092</u>	<u>0.110</u>	<u>0.151</u>	0.097	<u>0.129</u>	<u>0.088</u>	<u>0.116</u>
SVAE [67]	0.301	0.428	0.114	0.133	0.387	0.505	0.287	0.430	0.124	0.135	0.122	0.142	0.145	0.171	0.145	0.156
GeoTDM	0.107	0.193	0.023	0.039	0.115	0.209	0.107	0.176	0.064	0.087	0.083	0.120	0.083	0.121	0.074	0.099

5.1 Conditional Case

5.1.1 N-body

Experimental setup. We adopt three scenarios in the collection of N-body simulation datasets, including **1.** Charged Particles [26, 43], where $N = 5$ particles with charges randomly chosen between $+1/-1$ are moving under Coulomb force; **2.** Spring Dynamics [26], where $N = 5$ particles with random mass are connected by springs with a probability of 0.5 between each pairs, and force on the spring follows Hooke’s law; **3.** Gravity System [2], where $N = 10$ particles with random mass and initial velocity moves driven by gravitational force. For all three datasets, we use 3000 trajectories for training, 2000 for validation, and 2000 for testing. For each trajectory, we use 10 frames as the condition and predict the trajectory for the next 20 frames.

Baselines. We involve baselines from three families. Frame-to-frame prediction models: Radial Field [27], Tensor Field Network [56], SE(3)-Transformer [9], and EGNN [43]; Deterministic trajectory model: EqMotion [62]; Probabilistic trajectory model: SVAE [67]. Details in App. B.3.

Metrics. We employ Average Discrepancy Error (ADE) and Final Discrepancy Error (FDE), which are widely adopted for trajectory forecasting [67, 62], given by $ADE(\mathbf{x}^{[T]}, \mathbf{y}^{[T]}) = \frac{1}{TN} \sum_{t=0}^{T-1} \sum_{i=0}^{N-1} \|\mathbf{x}_i^{(t)} - \mathbf{y}_i^{(t)}\|_2$, and $FDE(\mathbf{x}^{[T]}, \mathbf{y}^{[T]}) = \frac{1}{N} \sum_{i=0}^{N-1} \|\mathbf{x}_i^{(T-1)} - \mathbf{y}_i^{(T-1)}\|_2$. For probabilistic models, we report average ADE and FDE derived from $K = 5$ samples.

Implementation. The input data are processed as geometric graphs. For example, on Charged Particles, the node feature is the charge, and the graph is specified as fully connected without self-loops. We use 6 layers in EGNN with hidden dimension of 128. We use $\mathcal{T} = 1000$ and the linear noise schedule [18]. More details in App. B.2.

Results. We present the results in Table 1, with the following observations. **1.** Trajectory models generally yield lower error than frame to frame prediction models since they mitigate errors accumulated in iterative roll-out. **2.** The equivariant methods, *e.g.*, EGNN, EqMotion, and our GeoTDM significantly improves over the non-equivariant model SVAE, demonstrating the importance of injecting physical symmetry into the modeling of geometric trajectories. **3.** By directly modeling the distribution of geometric trajectories with equivariance, GeoTDM achieves the lowest ADE and FDE on all three tasks, showcasing the superiority of the proposed approach.

5.1.2 Molecular Dynamics

Experimental setup. We employ the MD17 [5] dataset, which contains the DFT-simulated molecular dynamics (MD) trajectories of 8 small molecules, with the number of atoms for each molecule

Table 4: MD Trajectory generation results on MD17. Marg, Class, and Pred refer to Marginal score, Classification score, and Prediction score respectively. GeoTDM performs the best on all 8 molecules.

	Aspirin			Benzene			Ethanol			Malonaldehyde		
	Marg ↓	Class ↑	Pred ↓	Marg ↓	Class ↑	Pred ↓	Marg ↓	Class ↑	Pred ↓	Marg ↓	Class ↑	Pred ↓
SVAE [67]	3.628	6.80×10^{-5}	0.0949	4.755	2.81×10^{-6}	0.0181	2.735	2.39×10^{-5}	0.0929	2.808	5.57×10^{-3}	0.0346
EGVAE [43]	2.650	1.31×10^{-4}	0.0386	3.677	1.50×10^{-4}	0.0104	2.617	5.86×10^{-6}	0.1131	2.767	1.73×10^{-6}	0.0664
GeoTDM	0.726	3.48×10^{-2}	0.0212	0.597	1.62×10^{-1}	0.0019	0.314	4.63×10^{-1}	0.0235	0.403	3.35×10^{-1}	0.0146
	Naphthalene			Salicylic			Toluene			Uracil		
	Marg ↓	Class ↑	Pred ↓	Marg ↓	Class ↑	Pred ↓	Marg ↓	Class ↑	Pred ↓	Marg ↓	Class ↑	Pred ↓
SVAE [67]	3.150	2.50×10^{-2}	0.2123	2.941	3.54×10^{-6}	0.1312	3.083	8.29×10^{-5}	0.2580	2.736	3.73×10^{-5}	0.604
EGVAE [43]	3.007	3.17×10^{-4}	0.0136	3.314	3.76×10^{-6}	0.0221	2.054	2.77×10^{-5}	0.0457	3.570	2.02×10^{-5}	0.0212
GeoTDM	0.770	1.17×10^{-1}	0.0093	0.559	1.82×10^{-1}	0.0135	0.539	1.12×10^{-1}	0.0118	0.954	2.02×10^{-1}	0.0116

ranging from 9 (Ethanol and Malonaldehyde) to 21 (Aspirin). For each molecule, we construct a training set of 5000 trajectories, and 1000/1000 for validation and testing, uniformly sampled along the time dimension. Different from [62], we explicitly involve the hydrogen atoms which contribute most to the vibrations of the trajectory, leading to a more challenging task. The node feature is the one-hot encodings of atomic number [44] and edges are connected between atoms within three hops measured in atomic bonds [48]. We adopt the same set of baselines as the N-body experiments.

Results. As depicted in Table 3, GeoTDM achieves the best performance on all eight molecule MD trajectories, outperforming previous state-of-the-art approach EqMotion. In particular, GeoTDM obtains an improvement of 23.1%/15.3% on average in terms of ADE/FDE, compared with the previous state-of-the-art approach EqMotion, thanks to the probabilistic modeling which is advantageous in capturing the stochasticity of MD simulations.

5.1.3 Pedestrian Trajectory Forecasting

Experimental setup. We apply our model to ETH-UCY [35, 28] dataset, a challenging and large-scale benchmark for pedestrian trajectory forecasting. There are five scenes in total: ETH, Hotel, Univ, Zara1, and Zara2. Following standard setup [14, 67], we use 8 frames (3.2 seconds) as input to predict the next 12 frames (4.8 seconds). The pedestrians are viewed as nodes and their 2D coordinates are extracted from the scenes. Edges are connected for nodes within a preset distance measured from the final frame in the given trajectory. The metrics are minADE/minFDE computed from 20 samples. For baselines, we compare with existing generative models that have been specifically designed for pedestrian trajectory prediction, including GANs: SGAN [14], SoPhie [39]; VAEs: PECNet [32], Traj++ [40], BiTraP [68], SVAE [67]; and diffusion: MID [12]. Baseline results are taken from [67].

Results. From Table 2, we observe that our GeoTDM obtains the best predictions on 3 out of the 5 scenarios while achieving the lowest average ADE and FDE. It is remarkable since compared with these baselines specifically tailored for the task of pedestrian trajectory forecasting, GeoTDM does not involve special data preprocessing of the trajectories through rotations or translations, does not involve extra auxiliary losses to optimize during training, and does not require task-specific backbones, demonstrating its general effectiveness across different geometric domains.

5.2 Unconditional Generation

Experimental setup. For generation we reuse the Charged Particle dataset and the MD17 dataset. We follow the same setup as the conditional case, except that we generate trajectories with length 20 from scratch. We compare with SGAN [14], SVAE [67] (slightly modified to enable generation from scratch), and a VAE-modified version of EGNN [43], dubbed EGVAE (see App. B). The results of SGAN on MD17 is omitted due to mode collapse during training.

Metrics. We adopt three metrics adapted from time series generation to quantify the generation quality of the geometric trajectories: *Marginal* scores [34] measure the distance between the empirical probability density functions of the generated samples and the ground truths; *Classification* scores [25] are computed as the cross-entropy loss given by a trajectory classification model, trained on the task of distinguish whether the trajectory is generated or real; *Prediction* scores [72] are the MSEs of a train-on-synthetic-test-on-real trajectory prediction model (a 1-layer EqMotion) that takes as input the first half of the trajectories to predict the other half.

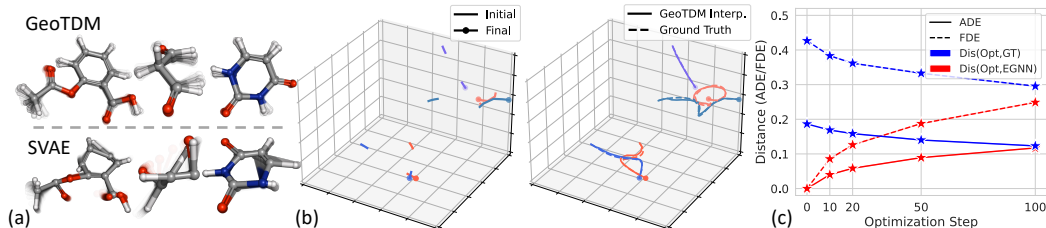


Figure 2: (a) Unconditional generation samples on MD17. GeoTDM generates MD trajectories with much higher quality (see more in App. D). (b) Interpolation. *Left*: the given initial and final 5 frames. *Right*: GeoTDM interpolation and GT. (c) Optimization by GeoTDM on predictions of EGNN. $\text{Dis}(\text{Opt}, \text{GT})/\text{Dis}(\text{Opt}, \text{EGNN})$ is the distance between optimized trajectories and GT/EGNN.

Results. Quantitative results are displayed in Table 5 and 4 for N-body and MD17. Notably, GeoTDM delivers samples with much higher quality than the baselines. On Charged Particles, GeoTDM achieves a classification score of 0.556, indicating its generated samples are generally indistinguishable with the ground truths. We observe similar patterns on MD17, where GeoTDM obtains remarkably lower marginal scores, higher classification scores, and lower prediction scores, showcasing its strong capability to model complex distributions of geometric trajectories on various geometric data. Visualizations are in Fig. 5 and more in App. D.

Table 5: Unconditional generation results on N-body Charged Particle.

	Marg ↓	Class ↑	Pred ↓
SGAN [14]	0.1448	3.98×10^{-7}	0.172
SVAE [67]	0.0668	1.38×10^{-6}	0.282
EGVAE [43]	0.1141	4.22×10^{-2}	0.0467
GeoTDM	0.0055	5.56×10^{-1}	0.00978

5.3 Ablation Studies and Additional Use Cases

Ablations on diffusion prior. We investigate different priors, including non-equivariant $\mathcal{N}(\mathbf{0}, \mathbf{I})$ (*i.e.*, DDPM [18]), equivariant but fixed CoM prior $\mathcal{N}(\text{CoM}(\mathbf{x}_c^{(T_c-1)}), \mathbf{I})$ and point-wise equivariant prior $\mathcal{N}(\mathbf{x}_c^{(T_c-1)}, \mathbf{I})$. In Table 6 we see that non-equivariant prior leads to significantly worse performance. The CoM prior, though equivariant, is still inferior due to extra overhead in denoising the nodes initialized around the CoM to the original geometry. GeoTDM yields the lowest error due to the flexible learnable prior.

Table 6: Ablation studies. The numbers refer to ADE/FDE.

	Charge	Aspirin
GeoTDM $\mathcal{N}(\mathbf{x}_r^{[T]}, \mathbf{I})$	0.110/0.258	0.107/0.193
Fixed $\mathcal{N}(\mathbf{0}, \mathbf{I})$	0.220/0.485	0.235/0.393
$\mathcal{N}(\text{CoM}(\mathbf{x}_c^{(T_c-1)}), \mathbf{I})$	0.135/0.298	0.119/0.212
$\mathcal{N}(\mathbf{x}_c^{(T_c-1)}, \mathbf{I})$	0.123/0.282	0.110/0.204
w/o Equivariance	0.251/0.542	0.252/0.440
w/o Attention	0.133/0.312	0.114/0.208
w/o Shift invariance	0.139/0.330	0.112/0.212

Ablations on EGTM. We further ablate the designs of the denoising model. **1. Equivariance.** We replace all EGCL layers into non-equivariant MPNN [11] layers with same hidden dimension, leading to non-equivariant transition kernels. The performance becomes much worse, verifying the necessity of equivariance. **2. Attention.** We substitute the attentions in temporal layers by equivariant convolutions (see App. B). Compared with this variant, GeoTDM enjoys larger capacity with attention and yields lower prediction error especially on Charged Particle where the particles generally move faster. **3. Temporal shift invariance.** We employ relative temporal embeddings in attention, which enhances the generalization. Notably, the FDE improves from 0.330 to 0.258 on Charged Particle compared with the absolute temporal embedding.

Temporal interpolation. GeoTDM is able to perform interpolation as a special case of the conditional case. We demonstrate such capability on Charged Particle. The model is provided with the first 5 and last 5 frames, and the task is to generate the intermediate 20 frames as interpolation. GeoTDM reports an ADE of 0.055 on the test set, while a linear interpolation baseline reports an ADE of 0.171. From the qualitative visualizations in Fig. 2, we clearly see that GeoTDM can capture the complex dynamics and yield high-quality non-linear interpolations between the given initial and final frames.

Optimization. We further illustrate that GeoTDM can conduct optimization [31, 33] on given trajectories (*e.g.*, those simulated by an EGNN) by simulating K steps through the forward diffusion and then performing the reverse denoising. From Fig. 2 we see the distance between the optimized trajectory and GT gradually decreases as the optimization step grows. This reveals GeoTDM can effectively optimize the given trajectory towards the ground truth distribution.

6 Discussion

Limitations. Akin to other diffusion models, GeoTDM resorts to multi-step sampling which may require more compute. We present empirical runtime benchmarks and more discussions in App. C.3.

Conclusion. We present GeoTDM, a diffusion model built over distribution of geometric trajectories. It is designed to preserve the symmetry of geometric systems, achieved by using EGTN, a novel SE(3)-equivariant geometric trajectory model, as the denoising network. We evaluate GeoTDM on various datasets for unconditional generation, interpolation, extrapolation and optimization, showing that it consistently outperforms the baselines. Future works include streamlining GeoTDM and extending it to more tasks such as protein MD, robot manipulation, and motion synthesis.

Acknowledgments and Disclosure of Funding

We thank the anonymous reviewers for the helpful feedback on improving the manuscript. This work was supported by ARO (W911NF-21-1-0125), ONR (N00014-23-1-2159), and the CZ Biohub.

References

- [1] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. *Advances in neural information processing systems*, 29, 2016. 1
- [2] Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J Bekkers, and Max Welling. Geometric and physical quantities improve $\epsilon(3)$ equivariant message passing. In *International Conference on Learning Representations*, 2022. 1, 2, 7
- [3] Lowik Chanussot, Abhishek Das, Siddharth Goyal, Thibaut Lavril, Muhammed Shuaibi, Morgane Riviere, Kevin Tran, Javier Heras-Domingo, Caleb Ho, Weihua Hu, et al. Open catalyst 2020 (oc20) dataset and community challenges. *Acs Catalysis*, 11(10):6059–6072, 2021. 24
- [4] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018. 22
- [5] Stefan Chmiela, Alexandre Tkatchenko, Huziel E Sauceda, Igor Poltavsky, Kristof T Schütt, and Klaus-Robert Müller. Machine learning of accurate energy-conserving molecular force fields. *Science advances*, 3(5):e1603015, 2017. 7
- [6] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 2
- [7] Weitao Du, He Zhang, Yuanqi Du, Qi Meng, Wei Chen, Nanning Zheng, Bin Shao, and Tie-Yan Liu. Se (3) equivariant graph neural networks with complete local frames. In *International Conference on Machine Learning*, pages 5583–5608. PMLR, 2022. 2
- [8] Jacob D Durrant and J Andrew McCammon. Molecular dynamics simulations and drug discovery. *BMC biology*, 9(1):1–9, 2011. 1
- [9] Fabian B Fuchs, Daniel E Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *arXiv preprint arXiv:2006.10503*, 2020. 2, 7, 22
- [10] Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, 2020. 1
- [11] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017. 2, 9, 22
- [12] Tianpei Gu, Guangyi Chen, Junlong Li, Chunze Lin, Yongming Rao, Jie Zhou, and Jiwen Lu. Stochastic trajectory prediction via motion indeterminacy diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17113–17122, 2022. 3, 7, 8, 25, 26
- [13] Jiaqi Guan, Wesley Wei Qian, Xingang Peng, Yufeng Su, Jian Peng, and Jianzhu Ma. 3d equivariant diffusion for target-aware molecule generation and affinity prediction. In *The Eleventh International Conference on Learning Representations*, 2023. 6, 20

- [14] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2255–2264, 2018. [7](#), [8](#), [9](#), [22](#)
- [15] Jiaqi Han, Wenbing Huang, Hengbo Ma, Jiachen Li, Joshua B. Tenenbaum, and Chuang Gan. Learning physical dynamics with subequivariant graph neural networks. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. [1](#)
- [16] Jiaqi Han, Wenbing Huang, Tingyang Xu, and Yu Rong. Equivariant graph hierarchy-based neural networks. *Advances in Neural Information Processing Systems*, 35:9176–9187, 2022. [1](#), [22](#)
- [17] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. *Advances in Neural Information Processing Systems*, 35:27953–27965, 2022. [3](#)
- [18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. [2](#), [3](#), [5](#), [7](#), [9](#), [18](#), [21](#)
- [19] Jonathan Ho, Tim Salimans, Alexey A. Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. [2](#), [3](#)
- [20] Scott A Hollingsworth and Ron O Dror. Molecular dynamics simulation for all. *Neuron*, 99(6):1129–1143, 2018. [1](#)
- [21] Emiel Hoogeboom, Victor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pages 8867–8887. PMLR, 2022. [1](#), [2](#), [3](#), [6](#), [20](#)
- [22] Wenbing Huang, Jiaqi Han, Yu Rong, Tingyang Xu, Fuchun Sun, and Junzhou Huang. Equivariant graph mechanics networks with constraints. In *International Conference on Learning Representations*, 2022. [1](#)
- [23] John B Ingraham, Max Baranov, Zak Costello, Karl W Barber, Wujie Wang, Ahmed Ismail, Vincent Frappier, Dana M Lord, Christopher Ng-Thow-Hing, Erik R Van Vlack, et al. Illuminating protein space with a programmable generative model. *Nature*, pages 1–9, 2023. [1](#)
- [24] Bowen Jing, Ezra Erives, Peter Pao-Huang, Gabriele Corso, Bonnie Berger, and Tommi Jaakkola. Eigenfold: Generative protein structure prediction with diffusion models, 2023. [2](#)
- [25] Patrick Kidger, James Foster, Xuechen Li, and Terry J Lyons. Neural sdes as infinite-dimensional gans. In *International conference on machine learning*, pages 5453–5463. PMLR, 2021. [8](#)
- [26] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. *arXiv preprint arXiv:1802.04687*, 2018. [1](#), [2](#), [7](#)
- [27] Jonas Köhler, Leon Klein, and Frank Noé. Equivariant flows: sampling configurations for multi-body systems with symmetric energies. *arXiv preprint arXiv:1910.00753*, 2019. [2](#), [7](#), [22](#)
- [28] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer graphics forum*. Wiley Online Library, 2007. [8](#)
- [29] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022. [24](#)
- [30] Shitong Luo, Yufeng Su, Xingang Peng, Sheng Wang, Jian Peng, and Jianzhu Ma. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. [3](#)
- [31] Shitong Luo, Yufeng Su, Xingang Peng, Sheng Wang, Jian Peng, and Jianzhu Ma. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. *Advances in Neural Information Processing Systems*, 35:9754–9767, 2022. [9](#)
- [32] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 759–776. Springer, 2020. [7](#), [8](#)

- [33] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. 9
- [34] Hao Ni, Lukasz Szpruch, Magnus Wiese, Shujian Liao, and Baoren Xiao. Conditional sig-wasserstein gans for time series generation. *arXiv preprint arXiv:2006.05421*, 2020. 8
- [35] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You’ll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th international conference on computer vision*, pages 261–268. IEEE, 2009. 8
- [36] Calyampudi Radhakrishna Rao. *Linear statistical inference and its applications*, volume 2. Wiley New York, 1973. 5
- [37] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*, pages 8857–8868. PMLR, 2021. 3
- [38] Salva Rühling Cachay, Bo Zhao, Hailey Joren, and Rose Yu. DYffusion: a dynamics-informed diffusion model for spatiotemporal forecasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 3
- [39] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1349–1358, 2019. 7, 8
- [40] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 683–700. Springer, 2020. 7, 8, 25
- [41] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International Conference on Machine Learning*, pages 8459–8468. PMLR, 2020. 2
- [42] Victor Garcia Satorras, Emiel Hoogeboom, Fabian Bernd Fuchs, Ingmar Posner, and Max Welling. E(n) equivariant normalizing flows. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. 1, 2, 3
- [43] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. *arXiv preprint arXiv:2102.09844*, 2021. 1, 2, 4, 5, 7, 8, 9, 20, 22
- [44] Kristof Schütt, Oliver Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *International Conference on Machine Learning*, pages 9377–9388. PMLR, 2021. 8
- [45] Kristof T Schütt, Huziel E Sauceda, P-J Kindermans, Alexandre Tkatchenko, and K-R Müller. SchNet—a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24):241722, 2018. 1
- [46] Jean-Pierre Serre et al. *Linear representations of finite groups*, volume 42. Springer, 1977. 3
- [47] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, 2018. 4
- [48] Chence Shi, Shitong Luo, Minkai Xu, and Jian Tang. Learning gradient fields for molecular conformation generation. In *International conference on machine learning*, pages 9558–9568. PMLR, 2021. 8
- [49] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 2, 3
- [50] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 24

- [51] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023. [24](#)
- [52] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019. [2](#), [3](#)
- [53] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020. [2](#), [3](#)
- [54] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csd: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34:24804–24816, 2021. [3](#)
- [55] Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *The Eleventh International Conference on Learning Representations*, 2023. [3](#)
- [56] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018. [1](#), [2](#), [7](#), [22](#)
- [57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. [4](#)
- [58] Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. MCVD - masked conditional video diffusion for prediction, generation, and interpolation. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. [3](#)
- [59] Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023. [1](#), [2](#)
- [60] Fang Wu and Stan Z Li. Diffmd: a geometric diffusion model for molecular dynamics simulations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023. [3](#)
- [61] Liming Wu, Zhichao Hou, Jirui Yuan, Yu Rong, and Wenbing Huang. Equivariant spatio-temporal attentive graph networks to simulate physical dynamics. *Advances in Neural Information Processing Systems*, 36, 2024. [4](#)
- [62] Chenxin Xu, Robby T Tan, Yuhong Tan, Siheng Chen, Yu Guang Wang, Xinchao Wang, and Yanfeng Wang. Eqmotion: Equivariant multi-agent motion prediction with invariant interaction reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1410–1420, 2023. [2](#), [7](#), [8](#), [22](#)
- [63] Minkai Xu, Jiaqi Han, Aaron Lou, Jean Kossaifi, Arvind Ramanathan, Kamyar Azizzadenesheli, Jure Leskovec, Stefano Ermon, and Anima Anandkumar. Equivariant graph neural operator for modeling 3d dynamics. In *Forty-first International Conference on Machine Learning*, 2024. [2](#)
- [64] Minkai Xu, Alexander Powers, Ron Dror, Stefano Ermon, and Jure Leskovec. Geometric latent diffusion models for 3d molecule generation. In *International Conference on Machine Learning*. PMLR, 2023. [1](#), [2](#), [3](#), [24](#), [25](#), [26](#)
- [65] Minkai Xu, Wujie Wang, Shitong Luo, Chence Shi, Yoshua Bengio, Rafael Gomez-Bombarelli, and Jian Tang. An end-to-end framework for molecular conformation generation via bilevel programming. In *International Conference on Machine Learning*, pages 11537–11547. PMLR, 2021. [3](#)
- [66] 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. [1](#), [2](#), [25](#), [26](#)
- [67] Pei Xu, Jean-Bernard Hayet, and Ioannis Karamouzas. Socialvae: Human trajectory prediction using timewise latents. In *European Conference on Computer Vision*, pages 511–528. Springer, 2022. [7](#), [8](#), [9](#), [22](#)
- [68] Yu Yao, Ella Atkins, Matthew Johnson-Roberson, Ram Vasudevan, and Xiaoxiao Du. Bitrap: Bi-directional pedestrian trajectory prediction with multi-modal goal estimation. *IEEE Robotics and Automation Letters*, 6(2):1463–1470, 2021. [7](#), [8](#)

- [69] 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 *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 40001–40039. PMLR, 23–29 Jul 2023. 2
- [70] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 2018. 4
- [71] Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. Physdiff: Physics-guided human motion diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16010–16021, 2023. 3
- [72] Linqi Zhou, Michael Poli, Winnie Xu, Stefano Massaroli, and Stefano Ermon. Deep latent state space models for time-series generation. In *International Conference on Machine Learning*, pages 42625–42643. PMLR, 2023. 8

Appendix

Table of Contents

A Proofs	15
A.1 Unconditional Case	15
A.2 Conditional Case	17
A.3 Optimizable Equivariant Prior	19
A.4 Proof of Theorem 4.1	20
B More Details on Experiments	21
B.1 Compute Resources	21
B.2 Hyper-parameters	21
B.3 Baselines	22
B.4 Model	22
B.5 Evaluation Metrics in the Unconditional Case	22
C More Experiments and Discussions	23
C.1 Model Composition for Longer Trajectory	23
C.2 Number of Diffusion Steps	24
C.3 Sampling Time	24
C.4 Standard Deviations	24
C.5 More Discussions with Existing Works	25
D More Visualizations	25

A Proofs

A.1 Unconditional Case

We note that, naïvely, a distribution $p(\mathbf{x}^{[T]})$ can not be translation invariant. In particular, this would imply that $p(\mathbf{x}^{[T]}) = p(\mathbf{x}^{[T]} + \mathbf{r})$ for all $\mathbf{r} \in \mathbb{R}^D$, but this would imply that $p(\mathbf{x}^{[T]}) = 0$ uniformly, a contradiction.

Instead, we derive an equivalent invariance condition by restricting $\text{SE}(D)$ to its maximally compact subgroup. In particular, we note that it is possible to define $\text{SO}(D)$ -invariant distributions (as this group is compact), and $\text{SE}(D)/\text{SO}(D) \cong \mathbb{T}$, the translation group. The natural way to quotient out our base space $\mathbb{R}^{T \times N \times D} / \mathbb{T} \cong \mathbb{R}^{(T \times N - 1) \times D}$ is to zero-center our data (along each dimension).

However, for practical purposes, we will refer to our construction as $\text{SE}(D)$ -invariant. In particular, since all inputs $\mathbf{x} \in \mathbb{R}^{T \times N \times D}$ are first zero-centered to be projected to $\mathbb{R}^{T \times N \times D} / \mathbb{T}$, the “lifted” unnormalized measure is $\text{SE}(D)$ invariant.

We will define P as our zero-centering operation $P(\mathbf{x}^{[T]}) = \mathbf{x}^{[T]} - \frac{1}{T} \sum_{t=0}^{T-1} \text{CoM}(\mathbf{x}^{(t)})$, with $\text{CoM}(\mathbf{x}^{(t)}) = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i^{(t)}$ and our restricted $(T \times N - 1) \times D$ Gaussian as $\tilde{\mathcal{N}}(\mathbf{y}|\mathbf{x}, \Sigma)$, which can be represented in the ambient space as a degenerated Gaussian variable

$$\tilde{\mathcal{N}}(\mathbf{y}|\mathbf{x}, \Sigma) = \frac{1}{(2\pi)^{2/((T \times N - 1) \times D)} \det^*(\Sigma_{\mathbf{P}})^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{x})^\top \Sigma_{\mathbf{P}}^+(\mathbf{y} - \mathbf{x})\right) \quad (13)$$

where $\Sigma_{\mathbf{P}} = \mathbf{P}\Sigma\mathbf{P}^\top$ and $\Sigma_{\mathbf{P}}^+$ is the pseudo-inverse (and \det^* is the determinant restricted to the subspace). Note that \mathbf{P} is symmetric and idempotent. Then specifically when $\Sigma = \mathbf{I}$, we have $\Sigma_{\mathbf{P}} = \mathbf{P}\mathbf{P}^\top$, then $\Sigma_{\mathbf{P}}^+ = \mathbf{P}$ as $(\mathbf{P}\mathbf{P}^\top)\mathbf{P}(\mathbf{P}\mathbf{P}^\top) = \mathbf{P}$, since $\mathbf{P}\mathbf{P} = \mathbf{P}$ and $\mathbf{P} = \mathbf{P}^\top$.

Base distribution. We require the base distribution to be $\text{SO}(3)$ -invariant. In practice, we let $p_{\mathcal{T}}(\tilde{\mathbf{x}}_{\mathcal{T}}^{[T]}) = \tilde{\mathcal{N}}(\mathbf{0}, \mathbf{I})$ to be the Gaussian distribution in the translation-invariant subspace

Transition kernel. For the transition kernel, we specify it as $p_{\theta}(\tilde{\mathbf{x}}_{\tau-1}^{[T]} | \tilde{\mathbf{x}}_{\tau}^{[T]}) = \tilde{\mathcal{N}}(\tilde{\boldsymbol{\mu}}_{\theta}(\tilde{\mathbf{x}}_{\tau}^{[T]}, \tau), \sigma_{\tau}^2 \mathbf{I})$. In order to ensure $p_{\theta}(\mathbf{R}\tilde{\mathbf{x}}_{\tau-1}^{[T]} | \mathbf{R}\tilde{\mathbf{x}}_{\tau}^{[T]}) = p_{\theta}(\tilde{\mathbf{x}}_{\tau-1}^{[T]} | \tilde{\mathbf{x}}_{\tau}^{[T]})$, it suffices to make $\tilde{\boldsymbol{\mu}}_{\theta}(\tilde{\mathbf{x}}_{\tau}^{[T]}, \tau)$ an $\text{SO}(3)$ -equivariant function. In this way,

$$p_{\theta}(\mathbf{R}\tilde{\mathbf{x}}_{\tau-1}^{[T]} | \mathbf{R}\tilde{\mathbf{x}}_{\tau}^{[T]}) = \tilde{\mathcal{N}}(\mathbf{R}\tilde{\mathbf{x}}_{\tau-1}^{[T]}; \tilde{\boldsymbol{\mu}}_{\theta}(\mathbf{R}\tilde{\mathbf{x}}_{\tau}^{[T]}), \sigma_{\tau}^2 \mathbf{I}), \quad (14)$$

$$= \tilde{\mathcal{N}}(\mathbf{R}\tilde{\mathbf{x}}_{\tau-1}^{[T]} - \tilde{\boldsymbol{\mu}}_{\theta}(\mathbf{R}\tilde{\mathbf{x}}_{\tau}^{[T]}); \mathbf{0}, \sigma_{\tau}^2 \mathbf{I}), \quad (15)$$

$$= \tilde{\mathcal{N}}(\mathbf{R}\tilde{\mathbf{x}}_{\tau-1}^{[T]} - \mathbf{R}\tilde{\boldsymbol{\mu}}_{\theta}(\tilde{\mathbf{x}}_{\tau}^{[T]}); \mathbf{0}, \sigma_{\tau}^2 \mathbf{I}), \quad (16)$$

$$= \tilde{\mathcal{N}}(\mathbf{R}(\tilde{\mathbf{x}}_{\tau-1}^{[T]} - \tilde{\boldsymbol{\mu}}_{\theta}(\tilde{\mathbf{x}}_{\tau}^{[T]})); \mathbf{0}, \sigma_{\tau}^2 \mathbf{I}), \quad (17)$$

$$= \tilde{\mathcal{N}}(\tilde{\mathbf{x}}_{\tau-1}^{[T]} - \tilde{\boldsymbol{\mu}}_{\theta}(\tilde{\mathbf{x}}_{\tau}^{[T]}); \mathbf{0}, \sigma_{\tau}^2 \mathbf{I}), \quad (18)$$

$$= \tilde{\mathcal{N}}(\tilde{\mathbf{x}}_{\tau-1}^{[T]}; \tilde{\boldsymbol{\mu}}_{\theta}(\tilde{\mathbf{x}}_{\tau}^{[T]}), \sigma_{\tau}^2 \mathbf{I}), \quad (19)$$

$$= p_{\theta}(\tilde{\mathbf{x}}_{\tau-1}^{[T]} | \tilde{\mathbf{x}}_{\tau}^{[T]}), \quad (20)$$

which permits the $\text{SO}(3)$ -equivariance of the transition kernel. In our implementation, we further re-parameterize $\tilde{\boldsymbol{\mu}}_{\theta}(\tilde{\mathbf{x}}_{\tau}^{[T]}, \tau)$ as,

$$\tilde{\boldsymbol{\mu}}_{\theta}(\tilde{\mathbf{x}}_{\tau}^{[T]}, \tau) = \frac{1}{\sqrt{\alpha_{\tau}}} \left(\tilde{\mathbf{x}}_{\tau}^{[T]} - \frac{\beta_{\tau}}{\sqrt{1 - \alpha_{\tau}}} \tilde{\boldsymbol{\epsilon}}_{\theta}(\tilde{\mathbf{x}}_{\tau}^{[T]}, \tau) \right), \quad (21)$$

where we instead ensure $\tilde{\boldsymbol{\epsilon}}_{\theta}(\tilde{\mathbf{x}}_{\tau}^{[T]}, \tau)$ to be $\text{SO}(3)$ -equivariant and its output should lie in the subspace $\mathcal{X}_{\mathbf{P}}$.

We now prove the following proposition, which states that if the base distribution is $\text{SO}(3)$ -invariant and the transition kernel is $\text{SO}(3)$ -equivariant, then the marginal at any diffusion time step is also $\text{SO}(3)$ -invariant.

Proposition A.1. *If the prior $p_{\mathcal{T}}(\tilde{\mathbf{x}}_{\mathcal{T}}^{[T]})$ is $\text{SO}(3)$ -invariant, the transition kernels $p_{\tau-1}(\tilde{\mathbf{x}}_{\tau-1}^{[T]} | \tilde{\mathbf{x}}_{\tau}^{[T]})$, $\forall \tau \in \{1, \dots, \mathcal{T}\}$ are $\text{SO}(3)$ -equivariant, then the marginal $p_{\tau}(\tilde{\mathbf{x}}_{\tau}^{[T]})$ at any time step $\tau \in \{0, \dots, \mathcal{T}\}$ is also $\text{SO}(3)$ -invariant.*

Proof. The proof is given by induction.

Induction base. When $\tau = \mathcal{T}$, we have the marginal being the prior $p_{\mathcal{T}}(\tilde{\mathbf{x}}_{\mathcal{T}}^{[T]})$, which is $\text{SO}(3)$ -invariant.

Induction step. Suppose the marginal at diffusion time step τ is $\text{SO}(3)$ -invariant, i.e., $p_{\tau}(\tilde{\mathbf{x}}_{\tau}^{[T]}) = p_{\tau}(\mathbf{R}\tilde{\mathbf{x}}_{\tau}^{[T]})$, then we have the following derivation for the marginal at time step $\tau - 1$:

$$p_{\tau-1}(\mathbf{R}\tilde{\mathbf{x}}_{\tau-1}^{[T]}) = \int p_{\tau-1}(\mathbf{R}\tilde{\mathbf{x}}_{\tau-1}^{[T]} | \tilde{\mathbf{x}}_{\tau}^{[T]}) p_{\tau}(\tilde{\mathbf{x}}_{\tau}^{[T]}) d\tilde{\mathbf{x}}_{\tau}^{[T]}, \quad (22)$$

$$= \int p_{\tau-1}(\mathbf{R}\tilde{\mathbf{x}}_{\tau-1}^{[T]} | \mathbf{R}\mathbf{R}^{-1}\tilde{\mathbf{x}}_{\tau}^{[T]}) p_{\tau}(\mathbf{R}\mathbf{R}^{-1}\tilde{\mathbf{x}}_{\tau}^{[T]}) d\tilde{\mathbf{x}}_{\tau}^{[T]}, \quad (23)$$

$$= \int p_{\tau-1}(\tilde{\mathbf{x}}_{\tau-1}^{[T]} | \mathbf{R}^{-1}\tilde{\mathbf{x}}_{\tau}^{[T]}) p_{\tau}(\mathbf{R}^{-1}\tilde{\mathbf{x}}_{\tau}^{[T]}) d\tilde{\mathbf{x}}_{\tau}^{[T]}, \quad (24)$$

$$= \int p_{\tau-1}(\tilde{\mathbf{x}}_{\tau-1}^{[T]} | \tilde{\mathbf{y}}_{\tau}^{[T]}) p_{\tau}(\tilde{\mathbf{y}}_{\tau}^{[T]}) \det(\mathbf{R}) d\tilde{\mathbf{y}}_{\tau}^{[T]}, \quad (25)$$

$$= p_{\tau-1}(\tilde{\mathbf{x}}_{\tau-1}^{[T]}). \quad (26)$$

□

Notably, for the final step at $\tau = 0$, the marginal $p_0(\mathbf{R}\tilde{\mathbf{x}}_0^{[T]})$ is also $\text{SO}(3)$ -invariant, indicating the final sample from the entire geometric trajectory diffusion process resides in an $\text{SO}(3)$ -invariant distribution, hence the physical symmetry being well preserved.

Algorithm 1 Training Procedure of GeoTDM-uncond

- 1: **repeat**
 - 2: Sample $\tilde{\epsilon}^{[T]} \sim \tilde{\mathcal{N}}(\mathbf{0}, \mathbf{I})^{[T]}$, $\tau \in \text{Unif}(\{1, \dots, \mathcal{T}\})$,
 $\tilde{\mathbf{x}}^{[T]} \sim \tilde{\mathcal{D}}_{\text{data}}$
 - 3: $\hat{\mathbf{x}}_\tau^{[T]} \leftarrow \sqrt{\bar{\alpha}_\tau} \tilde{\mathbf{x}}^{[T]} + \sqrt{1 - \bar{\alpha}_\tau} \tilde{\epsilon}^{[T]}$
 - 4: Take gradient descent step on
 $\nabla_\theta \|\tilde{\epsilon}^{[T]} - \tilde{\epsilon}_\theta(\hat{\mathbf{x}}_\tau^{[T]}, \tau)\|_2^2$
 - 5: **until** converged
-

Algorithm 2 Sampling Procedure of GeoTDM-uncond

- 1: Sample $\tilde{\mathbf{x}}_\tau^{[T]} \sim \tilde{\mathcal{N}}(\mathbf{0}, \mathbf{I})^{[T]}$
 - 2: **for** $\tau \leftarrow \mathcal{T}, \dots, 1$ **do**
 - 3: Sample $\tilde{\mathbf{z}}_\tau^{[T]} \sim \tilde{\mathcal{N}}(\mathbf{0}, \mathbf{I})^{[T]}$ if $\tau > 1$ else $\tilde{\mathbf{z}}_\tau^{[T]} = \mathbf{0}$
 - 4: $\tilde{\mathbf{x}}_{\tau-1}^{[T]} \leftarrow \frac{1}{\sqrt{\alpha_\tau}} \left(\tilde{\mathbf{x}}_\tau^{[T]} - \frac{1 - \alpha_\tau}{\sqrt{1 - \bar{\alpha}_\tau}} \tilde{\epsilon}_\theta(\tilde{\mathbf{x}}_\tau^{[T]}, \tau) \right) + \sigma_\tau \tilde{\mathbf{z}}_\tau^{[T]}$
 - 5: **end for**
 - 6: **return** $\mathbf{x}_0^{[T]}$
-

Algorithm 3 Training Procedure of GeoTDM-cond

- 1: **repeat**
 - 2: $\mathbf{x}_r^{[T]} \leftarrow \text{EquiPrior}_{\eta, \gamma}(\mathbf{x}_c^{[T_c]})$ {Eq. 9}
 - 3: Sample $\epsilon^{[T]} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\tau \in \text{Unif}(\{1, \dots, \mathcal{T}\})$,
 $(\mathbf{x}^{[T]}, \mathbf{x}_c^{[T_c]}) \sim p_{\text{data}}$
 - 4: $\mathbf{x}_\tau^{[T]} \leftarrow \sqrt{\bar{\alpha}_\tau} (\mathbf{x}^{[T]} - \mathbf{x}_r^{[T]}) + \mathbf{x}_r^{[T]} + \sqrt{1 - \bar{\alpha}_\tau} \epsilon^{[T]}$
 - 5: Take gradient descent step on
 $\nabla_{\theta, \eta, \gamma} \|\epsilon^{[T]} - \epsilon_\theta(\mathbf{x}_\tau^{[T]}, \mathbf{x}_c^{[T_c]}, \tau)\|_2^2$
 - 6: **until** converged
-

Algorithm 4 Sampling Procedure of GeoTDM-cond

- 1: $\mathbf{x}_r^{[T]} \leftarrow \text{EquiPrior}_{\eta, \gamma}(\mathbf{x}_c^{[T_c]})$ {Eq. 9}
 - 2: Sample $\mathbf{x}_\tau^{[T]} \sim \mathcal{N}(\mathbf{x}_r^{[T]}, \mathbf{I})$, $\mathbf{x}_c^{[T_c]} \sim \mathcal{D}_{\text{data}}$
 - 3: **for** $\tau \leftarrow \mathcal{T}, \dots, 1$ **do**
 - 4: Sample $\mathbf{z}_\tau^{[T]} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})^{[T]}$ if $\tau > 1$ else $\mathbf{z}_\tau^{[T]} = \mathbf{0}$
 - 5: $\mathbf{x}_{\tau-1}^{[T]} \leftarrow \frac{1}{\sqrt{\alpha_\tau}} \left(\mathbf{x}_\tau^{[T]} - \mathbf{x}_r^{[T]} - \frac{1 - \alpha_\tau}{\sqrt{1 - \bar{\alpha}_\tau}} \epsilon_\theta(\mathbf{x}_\tau^{[T]}, \mathbf{x}_c^{[T_c]}, \tau) \right) + \mathbf{x}_r^{[T]} + \sigma_\tau \mathbf{z}_\tau^{[T]}$
 - 6: **end for**
 - 7: **return** $\mathbf{x}_0^{[T]}$
-

A.2 Conditional Case

In the conditional case, we target on modeling the conditional distribution $p(\mathbf{x}^{[T]} | \mathbf{x}_c^{[T_c]})$. The desired constraint is the following equivariance condition: $p(\mathbf{x}^{[T]} | \mathbf{x}_c^{[T_c]}) = p(g \cdot \mathbf{x}^{[T]} | g \cdot \mathbf{x}_c^{[T_c]})$, for all $g \in \text{SE}(3)$.

Construction of the equivariant prior. The prior is constructed through Eq. 9. Here we formally show that this guarantees SE(3)-equivariance of the prior. For convenience we repeat Eq. 9 below.

$$\mathbf{x}_r^{(t)} = \sum_{s \in [T_c]} \mathbf{w}^{(t,s)} \hat{\mathbf{x}}_c^{(s)}, \quad \text{s.t.} \quad \sum_{s \in [T_c]} \mathbf{w}^{(t,s)} = \mathbf{1}, \quad (27)$$

Then, we have

$$\mathbf{x}_r^{(t)} = \sum_{s \in [T_c]} \mathbf{w}^{(t,s)} \hat{\mathbf{x}}_c^{(s)}, \quad (28)$$

$$= \sum_{s \in [T_c]} \mathbf{w}^{(t,s)} (\mathbf{R} \hat{\mathbf{x}}_c^{(s)} + \mathbf{r}), \quad (29)$$

$$= \sum_{s \in [T_c]} \mathbf{w}^{(t,s)} (\mathbf{R} \hat{\mathbf{x}}_c^{(s)}) + \sum_{s \in [T_c]} \mathbf{w}^{(t,s)} \mathbf{r}, \quad (30)$$

$$= \mathbf{R} \sum_{s \in [T_c]} \mathbf{w}^{(t,s)} \hat{\mathbf{x}}_c^{(s)} + \mathbf{r}, \quad (31)$$

$$= \mathbf{R} \mathbf{x}_r^{(t)} + \mathbf{r}, \quad (32)$$

\forall rotation matrix \mathbf{R} and $\mathbf{r} \in \mathbb{R}^3$, which completes the proof.

Base distribution. We propose to leverage the following base distribution.

$$p_{\mathcal{T}}(\mathbf{x}_{\mathcal{T}}^{[T]} | \mathbf{x}_c^{[T_c]}) = \mathcal{N}(\mathbf{x}_{\mathcal{T}}^{[T]}; \mathbf{x}_r^{[T]}, \mathbf{I}), \quad (33)$$

where $\mathbf{x}_r^{[T]} = \text{EquiPrior}(\mathbf{x}_c^{[T_c]})$ is SE(3)-equivariant with respect to the condition $\mathbf{x}_c^{[T_c]}$. With such choice, the base distribution above is SE(3)-equivariant, since

$$p_{\mathcal{T}}(\mathbf{R} \mathbf{x}_{\mathcal{T}}^{[T]} + \mathbf{r} | \mathbf{R} \mathbf{x}_c^{[T_c]} + \mathbf{r}) = \mathcal{N}(\mathbf{R} \mathbf{x}_{\mathcal{T}}^{[T]} + \mathbf{r}; \mathbf{R} \mathbf{x}_r^{[T]} + \mathbf{r}, \mathbf{I}), \quad (34)$$

$$= \mathcal{N}(\mathbf{R} \mathbf{x}_{\mathcal{T}}^{[T]}; \mathbf{R} \mathbf{x}_r^{[T]}, \mathbf{I}), \quad (35)$$

$$= \mathcal{N}(\mathbf{x}_{\mathcal{T}}^{[T]}; \mathbf{x}_r^{[T]}, \mathbf{I}), \quad (36)$$

where the last equation is due to $\det(\mathbf{R}^T \mathbf{R}) = \mathbf{I}$, $\|\mathbf{x}_{\mathcal{T}}^{[T]} - \mathbf{x}_r^{[T]}\|^2 = \|\mathbf{R} \mathbf{x}_{\mathcal{T}}^{[T]} - \mathbf{R} \mathbf{x}_r^{[T]}\|^2$, which also gives the proof for Theorem 4.4 by a mild substitution of the notations.

Transition kernel. The transition kernel is given by

$$p_{\theta}(\mathbf{x}_{\tau-1}^{[T]} | \mathbf{x}_{\tau}^{[T]}, \mathbf{x}_c^{[T_c]}) = \mathcal{N}(\mathbf{x}_{\tau-1}^{[T]}; \boldsymbol{\mu}_{\theta}(\mathbf{x}_{\tau}^{[T]}, \mathbf{x}_c^{[T_c]}, \tau), \sigma_{\tau}^2 \mathbf{I}), \quad (37)$$

where $\boldsymbol{\mu}_{\theta}(\mathbf{x}_{\tau}^{[T]}, \mathbf{x}_c^{[T_c]}, \tau)$ parameterized to be SE(3)-equivariant with respect to its input $\mathbf{x}_{\tau}^{[T]}, \mathbf{x}_c^{[T_c]}$. In practice, we re-parameterize it as,

$$\boldsymbol{\mu}_{\theta}(\mathbf{x}_{\tau}^{[T]}, \mathbf{x}_c^{[T_c]}, \tau) = \mathbf{x}_r^{[T]} + \frac{1}{\sqrt{\alpha_{\tau}}} \left(\mathbf{x}_{\tau}^{[T]} - \mathbf{x}_r^{[T]} - \frac{\beta_{\tau}}{\sqrt{1 - \bar{\alpha}_{\tau}}} \boldsymbol{\epsilon}_{\theta}(\mathbf{x}_{\tau}^{[T]}, \mathbf{x}_c^{[T_c]}, \tau) \right), \quad (38)$$

where $\boldsymbol{\epsilon}_{\theta}(\mathbf{x}_{\tau}^{[T]}, \mathbf{x}_c^{[T_c]}, \tau)$ is an SO(3)-equivariant but translation-invariant function. It is then easy to see that $\boldsymbol{\mu}_{\theta}(\mathbf{x}_{\tau}^{[T]}, \mathbf{x}_c^{[T_c]}, \tau)$ meets the SE(3)-equivariance as desired.

Proposition A.2. *With such parameterization, optimizing the variational lower bound is equivalent to optimizing the following objective, up to certain re-weighting:*

$$\mathcal{L} = \|\boldsymbol{\epsilon}_{\theta}(\mathbf{x}_{\tau}^{[T]}, \mathbf{x}_c^{[T_c]}, \tau) - \boldsymbol{\epsilon}\|_2^2. \quad (39)$$

Proof. We define $q(\mathbf{x}_{\tau}^{[T]} | \mathbf{x}_{\tau-1}^{[T]}) := \mathcal{N}(\mathbf{x}_{\tau}^{[T]}; \mathbf{x}_r + \sqrt{1 - \beta_{\tau}}(\mathbf{x}_{\tau-1}^{[T]} - \mathbf{x}_r), \beta_{\tau} \mathbf{I})$, which yields $q(\mathbf{x}_{\tau}^{[T]} | \mathbf{x}_0^{[T]}) = \mathcal{N}(\mathbf{x}_{\tau}^{[T]}; \mathbf{x}_r + \sqrt{\bar{\alpha}_{\tau}}(\mathbf{x}_0^{[T]} - \mathbf{x}_r), (1 - \bar{\alpha}_{\tau}) \mathbf{I})$. The proof then generally follows [18] but with all latent variables in [18] being replaced by $\mathbf{x}_{\tau}^{[T]} - \mathbf{x}_r^{[T]}$. Then the terms in the VLB are

given by,

$$\mathcal{L}_{\tau-1} = D_{\text{KL}}(q(\mathbf{x}_{\tau-1}^{[T]} | \mathbf{x}_r^{[T]}, \mathbf{x}_0^{[T]}) \| p_{\theta}(\mathbf{x}_{\tau-1}^{[T]} | \mathbf{x}_r^{[T]})), \quad (40)$$

$$= \mathbb{E}_{\mathbf{x}_0^{[T]}, \epsilon} \left[\frac{1}{2\sigma_r^2} \left\| \mathbf{x}_r^{[T]} + \frac{1}{\sqrt{\alpha_r}} \left(\mathbf{x}_r^{[T]}(\mathbf{x}_0^{[T]}, \epsilon, \mathbf{x}_c^{[T_c]}) - \mathbf{x}_r^{[T]} - \frac{\beta_r}{\sqrt{1-\alpha_r}} \epsilon \right) - \boldsymbol{\mu}_{\theta}(\mathbf{x}_r^{[T]}, \tau) \right\|^2 \right], \quad (41)$$

$$= \mathbb{E}_{\mathbf{x}_0^{[T]}, \epsilon} \left[\frac{1}{2\sigma_r^2} \left\| \mathbf{x}_r^{[T]} + \frac{1}{\sqrt{\alpha_r}} \left(\mathbf{x}_r^{[T]} - \mathbf{x}_r^{[T]} - \frac{\beta_r}{\sqrt{1-\alpha_r}} \epsilon \right) - \mathbf{x}_r^{[T]} - \frac{1}{\sqrt{\alpha_r}} \left(\mathbf{x}_r^{[T]} - \mathbf{x}_r^{[T]} - \frac{\beta_r}{1-\alpha_r} \epsilon_{\theta}(\mathbf{x}_r^{[T]}, \mathbf{x}_c^{[T_c]}, \tau) \right) \right\|^2 \right], \quad (42)$$

$$= \mathbb{E}_{\mathbf{x}_0^{[T]}, \epsilon} \left[\frac{\beta_r^2}{2\sigma_r^2 \alpha_r (1-\alpha_r)} \|\epsilon - \epsilon_{\theta}(\mathbf{x}_r^{[T]}, \mathbf{x}_c^{[T_c]}, \tau)\|^2 \right], \quad (43)$$

which is equivalent to Eq. 39 up to certain re-weighting factors. For $\mathcal{L}_{\tau} = D_{\text{KL}}(q(\mathbf{x}_{\tau}^{[T]} | \mathbf{x}_0^{[T]}) \| p(\mathbf{x}_{\tau}^{[T]}))$, it does not contribute to the gradient since it is irrelevant to $\boldsymbol{\theta}$, and \mathbf{x}_r is also cancelled out in computing the KL, thus stopping the gradient from passing to $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$. \square

Analogous to the unconditional case, we have the following proposition, indicating that if the base distribution is SE(3)-equivariant and the transition kernel is SE(3)-equivariant, then the marginal is also SE(3)-equivariant.

Proposition A.3. *If the base distribution $p_{\mathcal{T}}(\mathbf{x}_{\mathcal{T}}^{[T]} | \mathbf{x}_c^{[T_c]})$ is SE(3)-equivariant and the transition kernels $p_{\tau-1}(\mathbf{x}_{\tau-1}^{[T]} | \mathbf{x}_{\tau}^{[T]}, \mathbf{x}_c^{[T_c]})$ of all diffusion steps $\tau \in \{1, \dots, \mathcal{T}\}$ are SE(3)-equivariant, then the marginal⁶ $p_{\tau}(\mathbf{x}_{\tau}^{[T]} | \mathbf{x}_c^{[T_c]})$ at any diffusion step $\tau \in \{0, \dots, \mathcal{T}\}$ is SE(3)-equivariant.*

Proof. The proof is similarly given by induction.

Induction base. When $\tau = \mathcal{T}$, the distribution is the base distribution $p_{\mathcal{T}}(\mathbf{x}_{\mathcal{T}}^{[T]} | \mathbf{x}_c^{[T_c]})$ is SE(3)-equivariant, as it is designed.

Induction step. Suppose the marginal at diffusion step τ , i.e., $p_{\tau}(\mathbf{x}_{\tau}^{[T]} | \mathbf{x}_c^{[T_c]})$, is SE(3)-equivariant, then we have

$$p_{\tau-1}(\mathbf{R}\mathbf{x}_{\tau-1}^{[T]} + \mathbf{r} | \mathbf{R}\mathbf{x}_c^{[T_c]} + \mathbf{r}) \quad (44)$$

$$= \int p_{\tau-1}(\mathbf{R}\mathbf{x}_{\tau-1}^{[T]} + \mathbf{r} | \mathbf{x}_{\tau}^{[T]}, \mathbf{R}\mathbf{x}_c^{[T_c]} + \mathbf{r}) p_{\tau}(\mathbf{x}_{\tau}^{[T]} | \mathbf{R}\mathbf{x}_c^{[T_c]} + \mathbf{r}) d\mathbf{x}_{\tau}^{[T]}, \quad (45)$$

$$= \int p_{\tau-1}(\mathbf{R}\mathbf{x}_{\tau-1}^{[T]} + \mathbf{r} | \mathbf{R}(\mathbf{R}^{-1}(\mathbf{x}_{\tau}^{[T]} - \mathbf{r})) + \mathbf{r}, \mathbf{R}\mathbf{x}_c^{[T_c]} + \mathbf{r}) p_{\tau}(\mathbf{R}(\mathbf{R}^{-1}(\mathbf{x}_{\tau}^{[T]} - \mathbf{r})) + \mathbf{r} | \mathbf{R}\mathbf{x}_c^{[T_c]} + \mathbf{r}) d\mathbf{x}_{\tau}^{[T]}, \quad (46)$$

$$= \int p_{\tau-1}(\mathbf{x}_{\tau-1}^{[T]} | \mathbf{R}^{-1}(\mathbf{x}_{\tau}^{[T]} - \mathbf{r}), \mathbf{x}_c^{[T_c]}) p_{\tau}(\mathbf{R}^{-1}(\mathbf{x}_{\tau}^{[T]} - \mathbf{r}) | \mathbf{x}_c^{[T_c]}) d\mathbf{x}_{\tau}^{[T]}, \quad (47)$$

$$= \int p_{\tau-1}(\mathbf{x}_{\tau-1}^{[T]} | \mathbf{y}_{\tau}^{[T]}, \mathbf{x}_c^{[T_c]}) p_{\tau}(\mathbf{y}_{\tau}^{[T]} | \mathbf{x}_c^{[T_c]}) \det(\mathbf{R}) d\mathbf{y}_{\tau}^{[T]}, \quad (48)$$

$$= p_{\tau-1}(\mathbf{x}_{\tau-1}^{[T]} | \mathbf{x}_c^{[T_c]}), \quad (49)$$

which concludes the proof. \square

A.3 Optimizable Equivariant Prior

Theorem A.4. *The prior implemented by the parameterization in Eq. 9, 10, and 11 subsumes CoM-based priors and fixed point-wise priors.*

⁶Here the marginal refers to marginalizing the intermediate states in previous diffusion steps, while still being conditional on the input condition $\mathbf{x}_c^{[T_c]}$.

Proof. We repeat the parameterizations specified by Eq. 9, 10, and 11 below for better readability.

$$\mathbf{x}_r^{(t)} = \sum_{s \in [T_c]} \mathbf{w}^{(t,s)} \hat{\mathbf{x}}_c^{(s)}, \quad \text{s.t.} \quad \sum_{s \in [T_c]} \mathbf{w}^{(t,s)} = \mathbf{1}_N, \quad (50)$$

$$\mathbf{W}_{t,s} = [\boldsymbol{\gamma} \otimes \hat{\mathbf{h}}_c^{[T_c]}]_{t,s} \in \mathbb{R}^N, \quad (51)$$

$$\mathbf{w}^{(t,s)} = \begin{cases} \mathbf{W}_{t,s} & s < T_c - 1, \\ \mathbf{1}_N - \sum_{s=0}^{T_c-2} \mathbf{W}_{t,s} & s = T_c - 1. \end{cases} \quad (52)$$

We first show $\mathbf{x}_r^{[T]}$ can reduce to the CoM-based priors. Let $\hat{\mathbf{x}}_c^{(s)} = \text{CoM}(\mathbf{x}_c^{(s)})$, $\hat{\mathbf{h}}_c^{(s)} = \frac{1}{T_c} \mathbf{1}_N$, $\gamma^{(t)} = 1$. In this case,

$$\mathbf{x}_r^{(t)} = \sum_{s \in [T_c]} \mathbf{w}^{(t,s)} \hat{\mathbf{x}}_c^{(s)}, \quad (53)$$

$$= \sum_{s \in [T_c-1]} \gamma^{(t)} \frac{1}{T_c} \mathbf{1}_N \text{CoM}(\mathbf{x}_c^{(s)}) + (\mathbf{1}_N - \sum_{s \in [T_c-1]} \gamma^{(t)} \frac{1}{T_c} \mathbf{1}_N) \text{CoM}(\mathbf{x}_c^{(T_c-1)}), \quad (54)$$

$$= \sum_{s \in [T_c-1]} \frac{1}{T_c} \mathbf{1}_N \text{CoM}(\mathbf{x}_c^{(s)}) + (\mathbf{1}_N - \frac{T_c-1}{T_c} \mathbf{1}_N) \text{CoM}(\mathbf{x}_c^{(T_c-1)}), \quad (55)$$

$$= \sum_{s \in [T_c-1]} \frac{1}{T_c} \mathbf{1}_N \text{CoM}(\mathbf{x}_c^{(s)}) + \frac{1}{T_c} \mathbf{1}_N \text{CoM}(\mathbf{x}_c^{(T_c-1)}), \quad (56)$$

$$= \frac{1}{T_c} \sum_{s \in [T_c]} \text{CoM}(\mathbf{x}_c^{(s)}), \quad (57)$$

where $\frac{1}{T_c} \sum_{s \in [T_c]} \text{CoM}(\mathbf{x}_c^{(s)})$ is the generalization of the CoM-based priors [21, 13] in the multiple frame conditioning scenario, which reduces to $\text{CoM}(\mathbf{x}_c^{(0)})$ when $T_c = 1$.

To show $\mathbf{x}_r^{[T]}$ can reduce to fixed point-wise priors is straightforward. Let $\hat{\mathbf{x}}_c^{(s)} = \mathbf{x}_c^{(s)}$, $\hat{\mathbf{h}}^{[T_c]} = \text{Onehot}(s^*) \mathbf{1}_{T_c \times N}$ and $\gamma^{(t)} = 1, \forall t$. Then $\mathbf{w}^{(t,s)} = \text{Onehot}(s^*) \mathbf{1}_{T_c \times N}$. Therefore,

$$\mathbf{x}_r^{(t)} = \sum_{s \in [T_c]} \mathbf{w}^{(t,s)} \mathbf{x}_c^{(s)}, \quad (58)$$

$$= \sum_{s \in [T_c]} \text{Onehot}(s^*) \mathbf{1}_{T_c \times N} \mathbf{x}_c^{(s)}, \quad (59)$$

$$= \mathbf{x}_c^{(s^*)}, \quad (60)$$

where $\mathbf{x}_c^{(s^*)}$ is the point-wise equivariant prior, and $s^* \in [T_c]$ is the frame index in the conditioning trajectory for this specific prior. \square

We also provide an illustrative comparison of these equivariant priors in Fig. 3.

A.4 Proof of Theorem 4.1

Theorem 4.1 (SE(3)-equivariance of EGTN). *Let $\mathbf{x}'^{[T]}, \mathbf{h}'^{[T]} = f_{\text{EGTN}}(\mathbf{x}^{[T]}, \mathbf{h}^{[T]}, \mathcal{E})$. Then we have $g \cdot \mathbf{x}'^{[T]}, \mathbf{h}'^{[T]} = f_{\text{EGTN}}(g \cdot \mathbf{x}^{[T]}, \mathbf{h}^{[T]}, \mathcal{E}), \forall g \in \text{SE}(3)$.*

Proof. f_{EGTN} is a stack of L EGNN and temporal attention layer in alternated fashion, formally written as $f_{\text{EGTN}} = \underbrace{f_{\text{attn}} \circ f_{\text{EGNN}} \circ \dots \circ f_{\text{attn}} \circ f_{\text{EGNN}}}_{L \times (f_{\text{attn}} \circ f_{\text{EGNN}})}$. Since the chain of SE(3)-equivariant

function is also SE(3)-equivariant, it suffices to prove f_{attn} is SE(3)-equivariant, in that the SE(3)-equivariance of EGNN directly follows [43].

It is directly verified that the attention coefficients $\mathbf{a}^{(t,s)} \in \mathbb{R}$ in Eq. 5 and the query $\mathbf{q}^{[T]}$, key $\mathbf{k}^{[T]}$, and value $\mathbf{v}^{[T]}$ are all SE(3)-invariant, since they are derived based on the SE(3)-invariant input

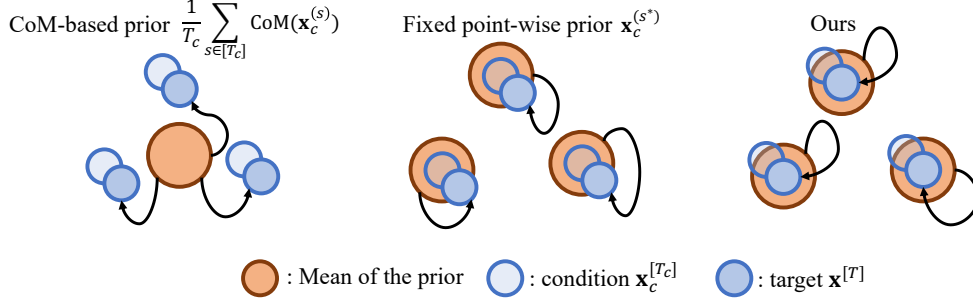


Figure 3: An illustration of different equivariant priors. For simplicity in the chart here we only illustrate the case when $N = 3$ and $T_c = 1$, $T = 1$.

$\mathbf{h}^{[T]}$. This directly leads to the $SE(3)$ -invariance of the updated node feature $\mathbf{h}^{[T]}$. For the updated coordinates,

$$\mathbf{x}'_{\text{tr}}(t) = \mathbf{x}_{\text{tr}}(t) + \sum_{s \in [T]} \mathbf{a}_{\text{tr}}^{(t,s)} \varphi_{\mathbf{x}}(\mathbf{v}_{\text{tr}}^{(t,s)})(\mathbf{x}_{\text{tr}}(t) - \mathbf{x}_{\text{tr}}(s)), \quad (61)$$

$$= \mathbf{R}\mathbf{x}^{(t)} + \mathbf{r} + \sum_{s \in [T]} \mathbf{a}^{(t,s)} \varphi_{\mathbf{x}}(\mathbf{v}^{(t,s)})(\mathbf{R}\mathbf{x}^{(t)} + \mathbf{r} - \mathbf{R}\mathbf{x}^{(s)} - \mathbf{r}), \quad (62)$$

$$= \mathbf{R}\mathbf{x}^{(t)} + \mathbf{r} + \mathbf{R} \left(\sum_{s \in [T]} \mathbf{a}^{(t,s)} \varphi_{\mathbf{x}}(\mathbf{v}^{(t,s)})(\mathbf{x}^{(t)} - \mathbf{x}^{(s)}) \right), \quad (63)$$

$$= \mathbf{R} \left(\mathbf{x}^{(t)} + \sum_{s \in [T]} \mathbf{a}^{(t,s)} \varphi_{\mathbf{x}}(\mathbf{v}^{(t,s)})(\mathbf{x}^{(t)} - \mathbf{x}^{(s)}) \right) + \mathbf{r}, \quad (64)$$

$$= \mathbf{R}\mathbf{x}'^{(t)} + \mathbf{r}, \quad (65)$$

where the variables with subscript tr refers to their transformed counterparts when the input $\mathbf{x}^{[T]}$ is transformed into $\mathbf{R}\mathbf{x}^{[T]} + \mathbf{r}$. Thus it completes the proof of $SE(3)$ -equivariance of the temporal attention f_{attn} and hence the entire f_{EGTN} . \square

B More Details on Experiments

B.1 Compute Resources

We use Distributed Data Parallel on 4 Nvidia A6000 GPUs to train all the models. The training on NBody and ETH-UCY take around 12 hours while each MD17 training phase takes about a day. Our CPUs were standard intel CPUs.

B.2 Hyper-parameters

We provide the detailed hyper-parameters of GeoTDM in Table 7. We adopt Adam optimizer with betas (0.9, 0.999) and $\epsilon = 10^{-8}$. For all experiments, we use the linear noise schedule per [18] with $\beta_{\text{start}} = 0.02$ and $\beta_{\text{end}} = 0.0001$.

Table 7: Hyper-parameters of GeoTDM in the experiments.

	n_layer	hidden	time_emb_dim	\mathcal{T}	batch_size	learning_rate
N-body	6	128	32	1000	128	0.0001
MD	6	128	32	1000	128	0.0001
ETH	4	64	32	100	100	0.0005

B.3 Baselines

For the frame-to-frame prediction models, including RF [27], EGNN [43], TFN [56], and SE(3)-Transformer [9], we adopt the implementation in the codebase maintained by [43]. To yield a strong comparison, instead of taking one frame as input to directly predict the final frame, we employ a discretized NeuralODE [4]-style training and inference procedure. In particular, we train the models with position $\mathbf{x}^{(t)}$ and velocity (computed as the difference of the current and previous frame, *i.e.*, $\mathbf{v}^{(t)} = \mathbf{x}^{(t)} - \mathbf{x}^{(t-1)}$) as input to predict the next velocity $\hat{\mathbf{v}}^{(t+1)}$. The position for the next step is integrated as $\hat{\mathbf{x}}^{(t+1)} = \mathbf{x}^{(t)} + \hat{\mathbf{v}}^{(t+1)}$. The training loss is computed as the Mean Squared Error (MSE) between the predicted position $\hat{\mathbf{x}}^{(t+1)}$ and the ground truth position $\mathbf{x}_{\text{gt}}^{(t+1)}$. In inference time, a roll-out prediction is conducted, which iteratively predict the next step by feeding the predicted position and velocity at the current step, for a total of T steps. We follow the hyper-parameter tuning guideline for these baselines by [16] which conduct a random search over the space spanned by the number of layers in $\{2, 4, 6, 8\}$, the hidden dimension $\{32, 64, 128\}$, learning rate $\{5e-3, 1e-3, 5e-4, 1e-4\}$, and batch size 32, 64, 128, 256, and select the model with best performance. All models are trained towards convergence with an early-stopping counter of 5, with validation performed every 20 epochs.

For EqMotion, we directly adopt the code by [62] and their suggested hyper-parameters for the N-body datasets and MD17 datasets.

For SVAE [67] and SGAN [14], these methods are originally developed for the pedestrian trajectory forecasting task. The backbone model that processes the input trajectory consists of social pooling operation and GRU or LSTM blocks for temporal processing. In order to make them favorable in tackling geometric systems which additionally include node features and edge features, we replace the social pooling operations by MPNNs [11] in both encoder (or discriminator) and decoder (or generator) to synthesize the information on the geometric graph. The temporal module is still kept as GRU for SVAE and LSTM for SGAN, following their original implementations. We also search over the best hyper-parameters which additionally involve the KL-divergence weight in $\{1, 0.1, 0.01, 0.001\}$ for SVAE according to the validation ELBO. For EGVAE, we replace the MPNNs in SVAE by EGNN [43], and restructured the latent space of the prior with both equivariant and invariant latent features. By this means, EGVAE is also guaranteed to model an equivariant distribution in the conditional case and an invariant distribution in the unconditional case.

B.4 Model

In detail, the EGCL layer [43] is given by:

$$\mathbf{m}_{ij} = \varphi_{\mathbf{m}}(\mathbf{h}_i, \mathbf{h}_j, \|\mathbf{x}_i - \mathbf{x}_j\|, \mathbf{e}_{ij}), \quad (66)$$

$$\mathbf{h}'_i = \varphi_{\mathbf{h}}\left(\mathbf{h}_i, \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij}\right), \quad (67)$$

$$\mathbf{x}'_i = \mathbf{x}_i + \sum_{j \in \mathcal{N}(i)} \varphi_{\mathbf{x}}(\mathbf{m}_{ij})(\mathbf{x}_i - \mathbf{x}_j), \quad (68)$$

where $\varphi_{\mathbf{m}}$, $\varphi_{\mathbf{h}}$, and $\varphi_{\mathbf{x}}$ are all MLPs. We also provide a schematic of our proposed EGTN in Fig. 4 for better illustration.

B.5 Evaluation Metrics in the Unconditional Case

All these metrics are evaluated on a set of model samples with the same size as the testing set.

Marginal score is computed as the absolute difference of two empirical probability density functions. Practically, we collect the x, y, z coordinates at each time step marginalized over all nodes in all systems in the predictions and the ground truth (testing set). Then we split the collection into 50 bins and compute the MAE in each bin, finally averaged across all time steps to obtain the score. Note that on MD17, instead of computing the pdf on coordinates, we compute the pdf on the length of the chemical bonds, which is a clearer signal that correlates to the validity of the generated MD trajectory, since during MD simulation the bond lengths are usually stable with very small vibrations. Marginal score gives a broad statistical measurement how each dimension of the generated samples align with the original data.

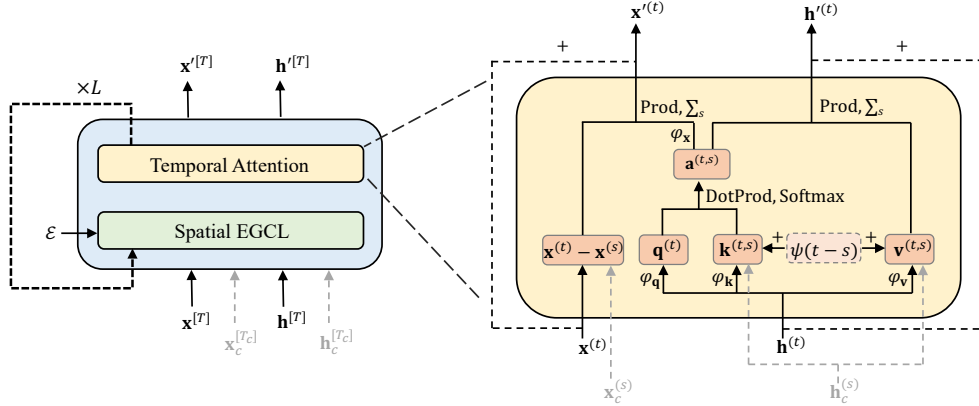


Figure 4: Schematic of the proposed EGTN, which alternates the EGCL layer for extracting spatial interactions and the temporal attention layer for modeling temporal sequence. Additional conditional information $\mathbf{x}_c^{[T_c]}$ and $\mathbf{h}_c^{[T_c]}$ can also be processed using cross-attention. The relative temporal embedding $\psi(t-s)$ is added to the key and value. DotProd refers to dot product and Softmax is performed over indexes of s .

Classification score is computed as the cross-entropy loss of a sequence classification model that aims to distinguish whether the trajectory is generated by the model or from the testing set. To be specific, we construct a dataset mixed by the generated samples and the testing set, and randomly split it into 80% and 20% subsets for training and testing. Then the model is trained on the training set and the classification score is computed as the cross-entropy on the testing set. We use a 1-layer EqMotion with a classification head as the model. The classification score provided intuition on how difficult it is to distinguish the generated samples and the original data.

Prediction score is computed as the MSE loss of a train-on-synthetic-test-on-real sequence to sequence model. In detail, we train a 1-layer EqMotion on the sampled dataset with the task of predicting the second half of the trajectory given the first half. We then evaluate the model on the testing set and report the MSE as the prediction score. Prediction score provides intuition on the capability of the generative model on generating synthetic data that well aligns with the ground truth.

C More Experiments and Discussions

C.1 Model Composition for Longer Trajectory

Since attention is utilized to extract temporal information, the time complexity scales quadratically with the length of the input trajectory, both during training and inference. In practice, we can instead train models on shorter trajectories and compose them during inference for longer trajectories, in both unconditional and conditional cases. For target trajectories with length T , we can first decompose it into K several equal-length⁷ non-overlapping intervals with time span ΔT . Then, for the unconditional case, we have

$$p(\mathbf{x}^{[T]}) = p(\mathbf{x}^{[\Delta T]}) \prod_{k=1}^{K-1} p(\mathbf{x}^{k\Delta T + [\Delta T]} | \mathbf{x}^{(k-1)\Delta T + [\Delta T]}), \quad (69)$$

by assuming mild conditional independence, where $p(\mathbf{x}^{[\Delta T]})$ is an unconditional model for trajectory with length ΔT , and $p(\mathbf{x}^{k\Delta T + [\Delta T]} | \mathbf{x}^{(k-1)\Delta T + [\Delta T]})$ can be learned by a conditional model for short trajectories. The conditional case directly follows by factorizing into products of conditional distribution over shorter trajectories.

We provide a demonstration of such technique as gifs in the **supplementary file**.

⁷In fact they do not necessarily need to be equal-length. Here we make such assumption for conciseness of the presentation.

Table 8: The effect of diffusion steps in the unconditional generation setting (top) and conditional forecasting setting (bottom).

	Marginal ↓	Aspirin Classification ↑	Prediction ↓	Marginal ↓	Charged Particle Classification ↑	Prediction ↓
$\mathcal{T} = 100$	0.808	0.0242	0.0243	0.0065	0.170	0.0118
$\mathcal{T} = 1000$	0.726	0.0348	0.0212	0.0055	0.556	0.00978

	ADE ↓	Aspirin FDE ↓	NLL ↓	ADE ↓	Charged Particle FDE ↓	NLL ↓
$\mathcal{T} = 100$	0.110	0.198	-2125.7	0.120	0.280	-547.5
$\mathcal{T} = 1000$	0.107	0.193	-3461.4	0.110	0.258	-982.7

C.2 Number of Diffusion Steps

We provide results in the unconditional generation setting for $\mathcal{T} = 100$. The results are in Table 8. Compared with the conditional setting, the unconditional generation is more challenging in that it needs to generate trajectories without any given reference geometries. We observe a drop in performance when \mathcal{T} is decreased from 1000 to 100. However, the performance with only 100 diffusion steps is still significantly better than SVAE.

C.3 Sampling Time

In the table below we display the generation metrics and the inference time per batch with batch size 128 on MD17 Aspirin molecule. We compare GeoTDM with EGVAE, an autoregressive VAE-based method with EGNN as the backbone. Here GeoTDM-100 and GeoTDM-1000 refer to GeoTDM using 100 and 1000 diffusion steps, respectively.

Table 9: Sampling runtime comparison on MD17 Aspirin molecule.

	Marginal	Classification	Prediction	Time per batch
EGVAE	2.650	1.31×10^{-4}	0.0386	0.6 ± 0.1
GeoTDM-100	0.808	2.42×10^{-2}	0.0243	7.9 ± 0.8
GeoTDM-1000	0.726	3.48×10^{-2}	0.0212	74.2 ± 2.1

We observe that GeoTDM-100 is approximately 10 times slower than EGVAE, since the model requires 100 calls of the denoising network to generate one batch, while EGVAE consumes the same number of calls as the length of the trajectory (20 in this case) due to autoregressive modeling. Although GeoTDM is slower, the gain in performance is significant and the quality of the generated trajectory is remarkably better than that of EGVAE. When further increasing the number of diffusion steps to 1000, the performance becomes better while requiring much more compute.

However, it is worth noticing that all these deep learning-based methods are significantly faster than traditional methods like DFT, which typically requires hours to even several days to converge depending on the scale of the system, according to OCP [3]. Therefore, although GeoTDM becomes slower than VAEs when using larger number of diffusion steps, it is still much faster than DFT, which indicates its practical value in generating geometric trajectories like molecular dynamics simulation.

The computation overhead of diffusion models compared with VAEs or GANs has been a well-known issue. We recognize enhancing the efficiency of GeoTDM as an interesting direction of future work, potentially through adopting faster solvers like DDIM [50] or DPMSolver [29], performing consistency distillation [51], or developing latent diffusion models [64] that take advantage of a more compact representation of the spatio-temporal geometric space.

C.4 Standard Deviations

We provide the standard deviations in Table 10 and 11.

Table 10: Conditional generation results of GeoTDM on N-body charged particle, spring, and gravity. Results (mean \pm standard deviation) are computed from 5 samples.

	Particle		Spring		Gravity	
	ADE	FDE	ADE	FDE	ADE	FDE
GeoTDM	0.110 \pm 0.014	0.258 \pm 0.032	0.0030 \pm 0.0004	0.0079 \pm 0.0010	0.256 \pm 0.015	0.613 \pm 0.034
SVAE	0.378 \pm 0.005	0.732 \pm 0.005	0.0120 \pm 0.0003	0.0209 \pm 0.0004	0.582 \pm 0.007	1.101 \pm 0.015

Table 11: Conditional generation results of GeoTDM on MD17. Results (mean \pm standard deviation) are computed from 5 samples.

Aspirin		Benzene		Ethanol		Malonaldehyde	
ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE
0.107 \pm 0.005	0.193 \pm 0.016	0.023 \pm 0.001	0.039 \pm 0.004	0.115 \pm 0.012	0.209 \pm 0.035	0.107 \pm 0.010	0.176 \pm 0.025
Naphthalene		Salicylic		Toluene		Uracil	
ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE
0.064 \pm 0.002	0.087 \pm 0.007	0.083 \pm 0.004	0.120 \pm 0.012	0.083 \pm 0.004	0.121 \pm 0.011	0.074 \pm 0.003	0.099 \pm 0.009

C.5 More Discussions with Existing Works

Below we discuss the unique challenges for designing GeoTDM compared with MID [12] and geometric diffusion models like GeoDiff [66] and GeoLDM [64], and how we tackle these challenges.

Modeling geometric trajectories. Although MID can model trajectories, it leverages Trajec-tron++ [40] backbone which takes as input the position vectors through a Transformer network. It requires non-trivial effort to incorporate additional node features and edge features into MID, while for GeoTDM, we design a general backbone EGTN that can process geometric trajectories while preserving equivariance. Existing geometric diffusion models (*e.g.*, GeoDiff and GeoLDM) never consider modeling the temporal dynamics and their backbone can only work on static (single-frame) geometric structures.

Incorporating equivariance into temporal diffusion. While geometric diffusion models have discussed proper ways to inject equivariance into diffusion models, it is unclear how to preserve equivariance when each hidden variable in the diffusion process has an additional dimension of time. In this work, we formally define equivariance constraint we want to impose on the marginal distribution, and how to design the prior and transition kernel in order to fulfill the constraint, in the context where all hidden variables are geometric trajectories. This is technically very different from existing works (*e.g.*, GeoDiff and GeoLDM) since the dimension of the data is fundamentally different, which leads to different analyses.

Consideration of both conditional and unconditional generation scenarios. MID is only designed and evaluated in the conditional setting where the task is to forecast the future trajectory given initial frames. GeoDiff and GeoLDM only operate in the unconditional setting where the task is to generate the structure without any initial 3D structure information. In this work, we systematically discuss both unconditional and conditional generation for geometric trajectories, and elaborate on how to design the prior and transition kernel to meet the equivariance constraint.

Parameterization of the learnable equivariant prior. In the conditional case, we propose to parameterize the equivariant prior with a lightweight EGTN. Such approach offers more flexibility in the equivariant prior, enabling optimizing it during training, which is also proved to be able to subsume existing center-of-mass (CoM) based parameterization (see Theorem A.4 in Appendix). Experiments in ablation studies also verify the superiority of such design.

We summarize the points above in Table 12.

D More Visualizations

We provide more visualizations in Fig. 5, 6, 8, 7, 9, and 10. Please refer to their captions for the detailed descriptions.

Table 12: Technical differences between GeoTDM and existing works.

	Trajectory	Equivariance	Conditional	Unconditional	Learnable Prior
MID [12]	✓		✓		
GeoDiff [66], GeoLDM [64]		✓		✓	
Our GeoTDM	✓	✓	✓	✓	✓

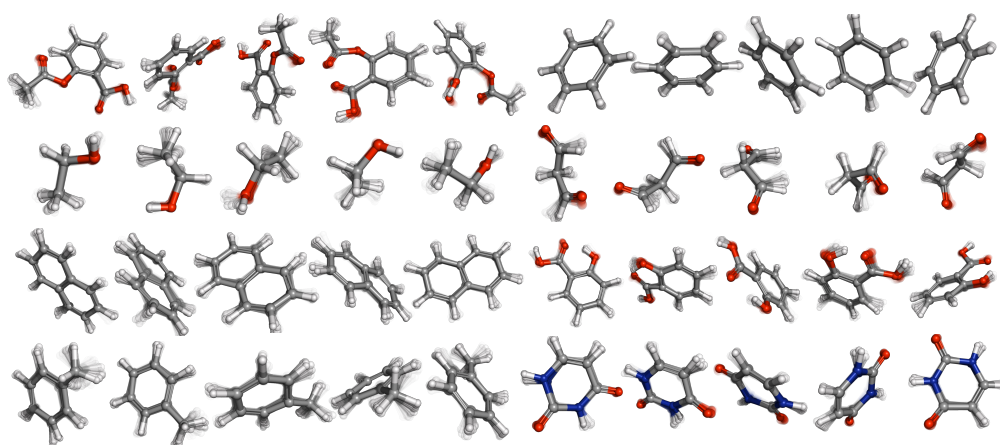


Figure 5: Uncurated samples of GeoTDM on MD17 dataset in the unconditional generation setup. From top-left to bottom-right are trajectories of the eight molecules: Aspirin, Benzene, Ethanol, Malonaldehyde, Naphthalene, Salicylic, Toluene, and Uracil. Five samples are displayed for each molecule. GeoTDM generates high quality samples. It well captures the vibrations and rotating behavior of the methyl groups in Aspirin and Ethanol. The bonds on the benzene ring are also more stable, aligning with findings in chemistry.

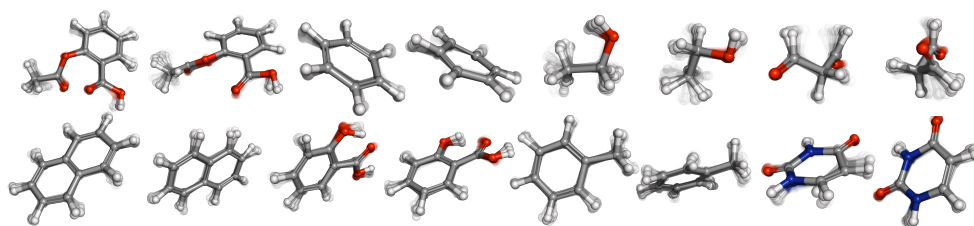


Figure 6: Samples from MD17 dataset.

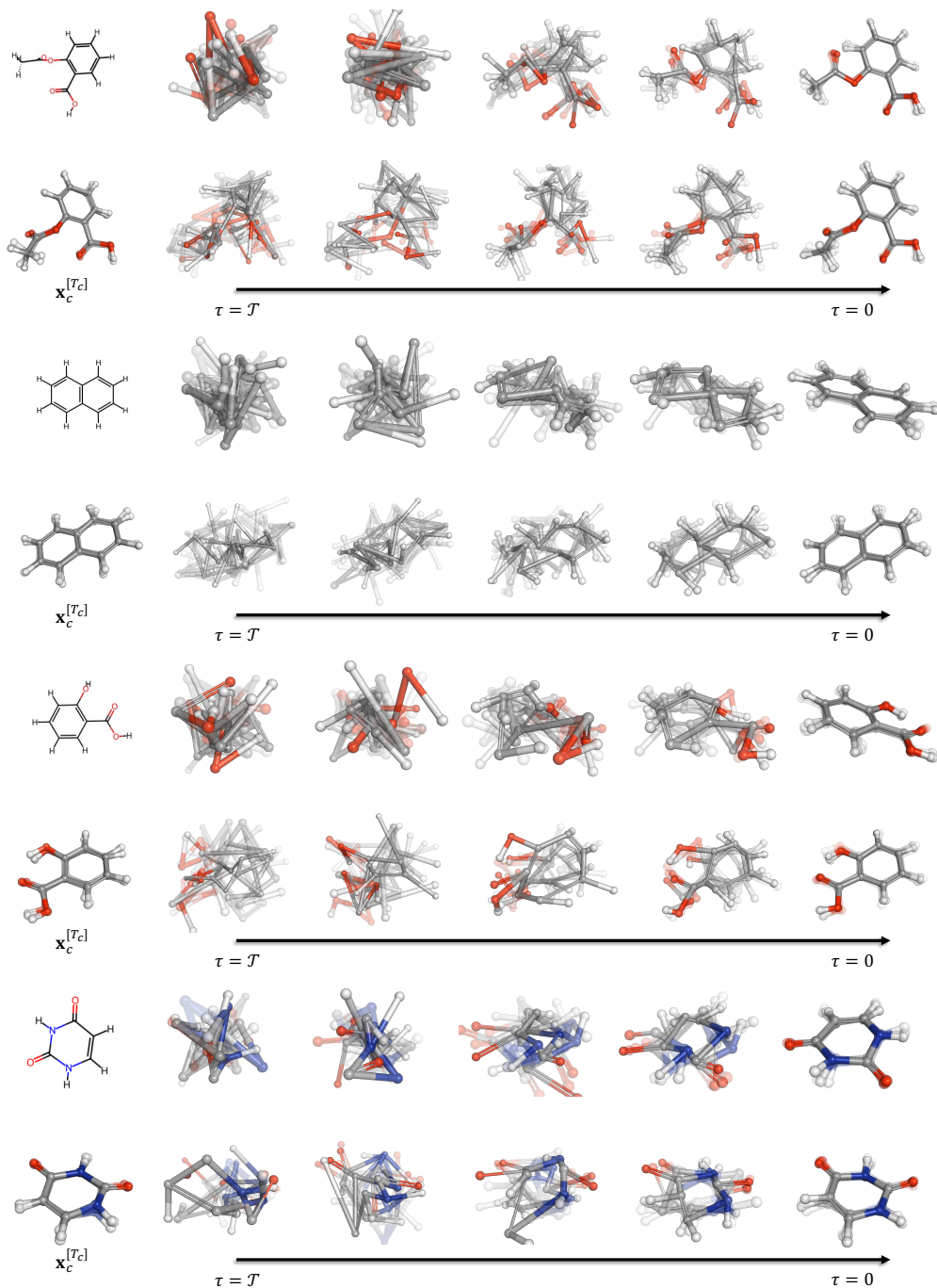


Figure 7: Visualization of the diffusion trajectory at different diffusion steps. From top to bottom: *Aspirin*, *Naphthalene*, *Salicylic*, *Uracil*. For each molecule, the first row shows the unconditional generation process, where the model generates the trajectory from the **invariant prior** purely from the molecule graph without any conditioning structure. The second row refers to the conditional generation, where the model generates from the **equivariant prior**, conditioning on some given frames $\mathbf{x}_c^{[T_c]}$. Notably, the equivariant prior (see samples at $\tau = T$ in each second row) preserves some structural information encapsulated in $\mathbf{x}_c^{[T_c]}$, thanks to our flexible parameterization.

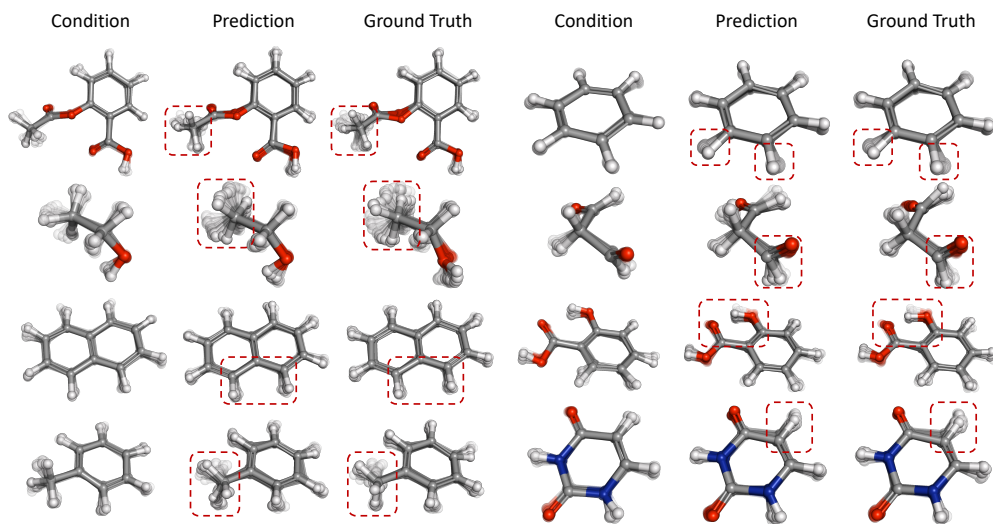


Figure 8: Uncurated samples of GeoTDM on MD17 dataset in the conditional forecasting setting. We highlight some regions of interest in red dashed boxes. GeoTDM delivers samples with very high accuracy while also capturing some stochasticity of the molecular dynamics.

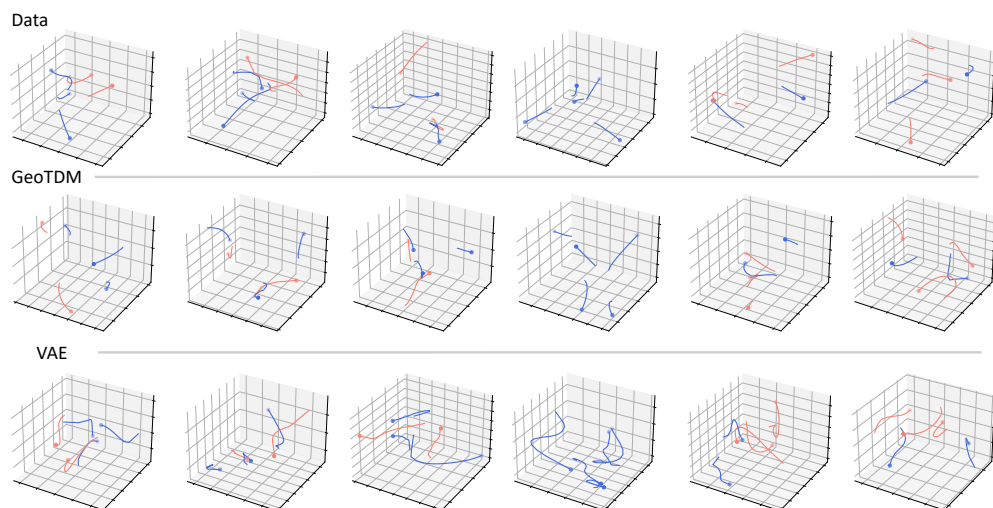


Figure 9: Visualization of data samples and generated samples by GeoTDM and SVAE in the unconditional setting on Charged Particles dataset. Nodes with color red and blue have the charge of +1/-1, respectively. Best viewed by zooming in.

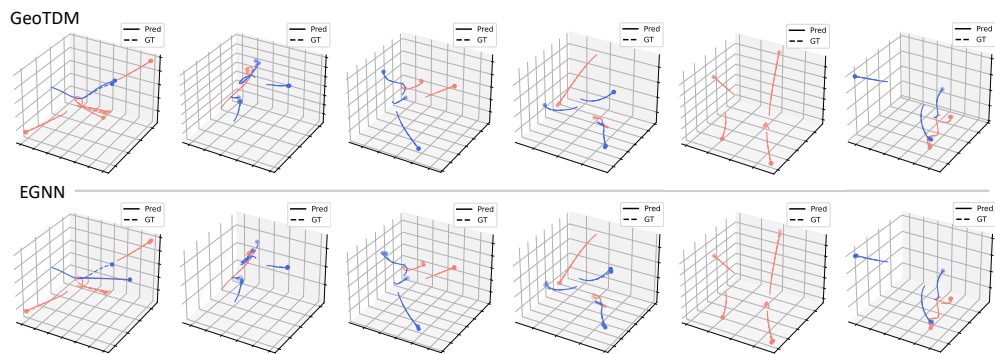


Figure 10: Visualization of predictions by GeoTDM and EGNN in the conditional setting on Charged Particles dataset. Nodes with color red and blue have the charge of $+1/-1$, respectively. Best viewed by zooming in.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: We have provided theoretical and empirical results showing that our proposed GeoTDM is able to work with geometric symmetries through multiple time steps.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We have discussed some limitations of GeoTDM, including the slow sampling speed of diffusion models (see Sec. C.3).

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Yes, we have listed out all of our assumptions and have explicated each step for our proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Yes, we have included all of the relevant details for reproducing our experiments. See App. B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We upload the code in supplementary file.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: See our code and experiment details in App. B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: See Sec. C.4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We have listed out this type of information in App. B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We have reviewed and followed the NeurIPS code of ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: Our method is primarily focused on scientific discovery, and there is no societal impact of our work beyond this scope.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our method does not have a high risk of misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have credited all code and dataset sources.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: We do not introduce any new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This work does not involve human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This work does not involve human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.