Align Your Structures: Generating Trajectories with Structure Pretraining for Molecular Dynamics

Aniketh Iyengar [*1] **Jiaqi Han** [*1] **Pengwei Sun** [*1] **Mingjian Jiang** [1] **Jianwen Xie** [2] **Stefano Ermon** [1]

Abstract

Generating molecular dynamics (MD) trajectories using deep generative models has attracted increasing attention, yet remains inherently challenging due to the limited availability of MD data and the complexities involved in modeling high-dimensional MD distributions. To overcome these challenges, we propose a novel framework that leverages structure pretraining for MD trajectory generation. Specifically, we first train a diffusion-based structure generation model on a large-scale conformer dataset, on top of which we introduce an interpolator module trained on MD trajectory data, designed to enforce temporal consistency among generated structures. Our approach effectively harnesses abundant conformer data to mitigate the scarcity of MD trajectory data and effectively decomposes the intricate MD modeling task into two manageable subproblems: structural generation and temporal alignment. We comprehensively evaluate our method on QM9 and DRUGS datasets across various tasks, including unconditional generation, forward simulation, and interpolation. Experimental results confirm that our approach excels in generating chemically realistic MD trajectories, as evidenced by remarkable improvements of accuracy in measurements such as bond length, bond angle, and torsion angle distributions.

1. Introduction

Molecular Dynamics (MD) is a computational simulation method used to model the physical movements of atoms and molecules over time (Alder & Wainwright, 1959; Verlet, 1967). By numerically integrating Newton's equations of motion, MD simulates the temporal evolution of molecular systems at atomic resolution. It has become a vital and widely adopted tool for addressing complex problems in biology (McCammon et al., 1977), chemistry (Rahman, 1964), and materials science (Antalík et al., 2024). Despite its utility, MD is computationally intensive, often requiring long simulation times and a large number of small integration steps to ensure numerical precision. This high cost has motivated significant efforts to accelerate MD and improve its sampling efficiency (Shaw et al., 2009; Darden et al., 1993; Laio & Parrinello, 2002). In this context, recent growing interest has been towards deep generative models, especially diffusion models (Noé et al., 2019; Jing et al., 2024a; Klein et al., 2023), as efficient surrogates for capturing the complex and diverse distributions observed in MD simulations.

Despite their promise, we identify a significant factor that poses remarkable limitations on their utility. The MD generative models are typically optimized on a single or a group of limited number of molecular systems (Noé et al., 2019; Han et al., 2024; Jing et al., 2024c), making it a fundamental challenge for them to generalize across different molecules. This is primarily due to two reasons. *Data scarcity*: Curating large-scale MD dataset over diverse molecular systems is extremely prohibitive due to the remarkably high computation demand for performing MD simulation at scale, leading to insufficiency in the amount of data for the models to well capture the underlying MD distribution. *Modeling complexity*: MD data is of high-dimensionality by extending the molecular structure space with an additional axis, the temporal dimension, which further contributes to modeling difficulty.

In this work, we propose a novel approach named EGINTERPOLATOR that addresses the challenges through *structure pretraining*. Specifically, we decompose the MD modeling problem into two sequential subtasks. First, we train a conformer diffusion model to generate conformers—*i.e.*, the static molecular structures corresponding to individual frames along an MD trajectory—using large-scale conformer datasets. Building on this pretrained structure model, we then initialize additional temporal layers and integrate structural and temporal information through a novel module called the equivariant temporal interpolator. We theoretically show that the temporal interpolator implicitly mod-

[*]Equal contribution [1]Stanford University [2]Lambda, Inc. Correspondence to: Jiaqi Han <jiaqihan@stanford.edu>.

Proceedings of the Workshop on Generative AI for Biology at the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

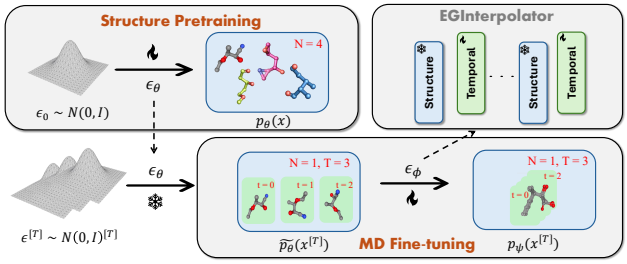

Figure 1. The overall framework of EGINTERPOLATOR.

els a transition from a temporally independent structural distribution to the fully correlated MD distribution. This formulation not only alleviates the optimization difficulty during training by decoupling spatial and temporal learning, but also provides greater flexibility during inference.

Our approach effectively addresses the two problems. First, we address the MD data scarcity issue by leveraging existing large-scale conformer datasets with diverse types of molecular systems, to complement the small scale MD dataset and provide generalization capabilities to the resulting MD diffusion model towards unseen molecular systems. Second, the two-stage pipeline conceptually decomposes the complexity of modeling the high-dimensional MD distribution into two tasks by first modeling the distribution of independent frames and then learning the temporal dependency.

Main contributions. 1. We systematically investigate the challenges that hinder the generalization of MD diffusion models towards new molecular systems and propose to leverage structure pretraining as a solution. **2.** We propose a principled approach for training an MD diffusion model based on structure pretraining, demonstrating on small molecular systems. **3.** We introduce a novel module, the equivariant temporal interpolator, as a vital building block that learns the temporal dependency of the structures at individual frames. **4.** We comprehensively evaluate our approach on a wide suite of tasks including unconditional generation, forward simulation, and interpolation. Extensive experiment results verify the efficacy of the proposed approach towards accurately capturing the complex MD distributions, while also being able to preserve superior conformer generation capability.

2. Related Work

Geometric diffusion models. Generative models for geometric data have garnered increasing attention across multiple domains. In molecular generation, GeoDiff (Xu et al., 2022) pioneered for conformer generation while EDM (Hoogeboom et al., 2022b) operates on both continuous coordinates and categorical atom types. Subsequent works(Xu et al., 2023; 2024a) introduced structured latent spaces to enhance scalability and controllability. For larger molecules, GCDM (Morehead & Cheng, 2024) incorporated geometry-complete local frames and chirality-sensitive features into SE(3)-equivariant networks. EBD (Park & Shen, 2024) performs hierarchically by first sampling scaffolds before refining atom positions through blurring-based denoising. Yet, they only model static structures while in this work we study the problem of their temporal correlation in MD.

ML-based Molecular Dynamics. Modeling the dynamics of geometric data presents significant challenges due to complex multi-object interactions, data scarcity, and high-dimensional spaces. Prior works such as EGNN (Satorras et al., 2021b) and SE(3)-Transformer (Fuchs et al., 2020) enhance model generalization by incorporating equivariance principles into their architectural designs (Brandstetter et al., 2022; Xu et al., 2024b). Timewarp (Klein et al., 2023) adopts an autoregressive approach to learn dynamic transition and emulate MD trajectories through simulation rollouts. These frame-to-frame prediction methods, despite their progress, suffer from compounding errors, while diffusion-based generative models prevent such accumulation by modeling entire trajectories holistically. GeoTDM (Han et al., 2024) directly models trajectories by introducing diffusion processes and architectures with equivalence guarantees. EquiJump (dos Santos Costa et al., 2024) employs a two-sided stochastic interpolant framework with an SO(3)-equivariant model to bridge all-atom proteins time steps directly, effectively capturing long-range temporal correlations in protein dynamics. While MDGen (Jing et al., 2024b) also introduces a end-to-end full trajectory modelling paradigm through a flow-based model, such a framework was designed specially for modeling torsions in peptides conditioned on at least one key frame, whereas we seek to design an approach that generalizes across diverse molecular systems.

3. Preliminaries

Geometric representation of molecular dynamics. In this work, we represent each molecular dynamics trajectory as a collection of *static structures*, or equivalently *conformers* that evolve through time. Each frame of conformer at timestep t is viewed as a geometric graph $\mathcal{G}^{(t)} := (\mathbf{h}, \mathbf{x}^{(t)}, \mathcal{E})$ where each row $\mathbf{h}_i \in \mathbb{R}^H$ is the node feature of atom i such as its atomic number, $\mathbf{x}_i^{(t)} \in \mathbb{R}^3$ is the Euclidean coordinate of atom i at timestep t, and $\mathcal{E}$ is the set of edges induced by the chemical bonds between atoms. The trajectory with length T is correspondingly represented as $\mathbf{x}^{[T]} := \mathbf{x}^{(0:T-1)} \in \mathbb{R}^{T\times N\times 3}$.

Geometric diffusion model for static structure generation. Geometric diffusion models (Xu et al., 2022; Hoogeboom et al., 2022a; Xu et al., 2023) are a family of diffusion-based generative models (Sohl-Dickstein et al., 2015; Ho et al., 2020a; Song & Ermon, 2019; Song et al., 2021) dedicated to capture the distribution of static

conformer structures $p(\mathbf{x}|\mathbf{h}, \mathcal{E})$, given the configuration of the molecular graph specified by the node feature $\mathbf{h}$ and edge connectivity $\mathcal{E}$. Inheriting the framework of diffusion models, they feature a Markovian forward noising process that gradually perturbs $\mathbf{x}_0$ toward $\mathbf{x}_\mathcal{T}$ through $\mathcal{T}$ diffusion steps, with the Gaussian transition kernel $q(\mathbf{x}_\tau|\mathbf{x}_{\tau-1}) = \mathcal{N}(\mathbf{x}_\tau; \sqrt{1-\beta_\tau}\mathbf{x}_{\tau-1}, \beta_\tau\mathbf{I})$, where β_τ is the noise schedule such that $\mathbf{x}_\mathcal{T}$ is close to the Gaussian prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The reverse process denoises toward the clean data using $p_\theta(\mathbf{x}_{\tau-1}|\mathbf{x}_\tau) = \mathcal{N}(\mathbf{x}_{\tau-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_\tau, \tau), \sigma_\tau^2\mathbf{I})$. The model is optimized via (Ho et al., 2020a)

$$\mathcal{L}_{\text{struct}} = \mathbb{E}_{\mathbf{x}_0\sim\mathcal{D}_{\text{struct}},\tau,\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_\tau, \tau)\|_2^2, \quad (1)$$

where $\mathcal{D}_{\text{struct}}$ is the conformer dataset, $\tau \sim \text{Unif}(1, \mathcal{T})$, $\mathbf{x}_\tau = \sqrt{\bar{\alpha}_\tau}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_\tau}\boldsymbol{\epsilon}$ with $\bar{\alpha}_\tau$ being certain noise schedule, and $\boldsymbol{\epsilon}_\theta$ is a parameterization of the mean satisfying $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_\tau, \tau) = \frac{1}{\sqrt{\alpha_\tau}}(\mathbf{x}_\tau - \frac{\beta_\tau}{\sqrt{1-\bar{\alpha}_\tau}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_\tau, \tau))$. A critical property of geometric diffusion models lies in the SE(3)-invariance of their marginal[1], *i.e.*, $p_\theta(\mathbf{x}_0) = g \cdot p_\theta(\mathbf{x}_0), \quad g \in \text{SE}(3)$, where g is an arbitrary group action in SE(3) that consists of all 3D rotations and translations, and $p_\theta(\mathbf{x}_0) = p(\mathbf{x}_\mathcal{T})\prod_{\tau=1}^{\mathcal{T}} p_\theta(\mathbf{x}_{\tau-1}|\mathbf{x}_\tau)$. This is achieved by parameterizing $\boldsymbol{\epsilon}_\theta$ with an equivariant graph neural network (Satorras et al., 2021b;a; Xu et al., 2022) such that $g \cdot \boldsymbol{\epsilon}_\theta(\mathbf{x}_\tau, \tau) = \boldsymbol{\epsilon}_\theta(g \cdot \mathbf{x}_\tau, \tau)$ which guarantees the SE(3)-equivariance of the transition kernel $p_\theta(\mathbf{x}_{\tau-1}|\mathbf{x}_\tau)$ at each step τ.

Problem definition. In this work, we seek to design a diffusion model that captures the distribution of molecular dynamics $p(\mathbf{x}^{[T]})$ given node features $\mathbf{h}$ and edges $\mathcal{E}$. Based on this goal, we are additionally interested in two relevant subtasks, namely *forward simulation*, which models the conditional distribution $p(\mathbf{x}^{(1:T-1)}|\mathbf{x}^{(0)})$ given the initial structure $\mathbf{x}^{(0)}$, and *interpolation*, which models $p(\mathbf{x}^{(1:T-2)}|\mathbf{x}^{(0)}, \mathbf{x}^{(T-1)})$ given both the initial frame $\mathbf{x}^{(0)}$ and final frame $\mathbf{x}^{(T-1)}$.

4. Method

In this section, we detail our approach that learns to generate MD trajectories by aligning the structural distributions temporally along the MD trajectory. In § 4.1, we introduce the overall framework of conformer pretraining and temporal alignment for MD generation. In § 4.2, we propose a temporal interpolator module that effectively couples the conformer model and temporal layers through an interpolation operation. In § 4.3, we discuss the key implementations of EGINTERPOLATOR.

[1]For conciseness we henceforth omit the conditions $\mathbf{h}, \mathcal{E}$ in $p(\mathbf{x}_0|\mathbf{h}, \mathcal{E})$ unless otherwise specified.

4.1. Trajectory Generation by Aligning Structure Model

Motivation. While substantial research has advanced the modeling of static structure distributions $p(\mathbf{x})$, generalizing this paradigm to molecular dynamics trajectories remains inherently challenging for two primary reasons. **1.** *Data scarcity.* Unlike conformer modeling, which benefits from extensive datasets such as GEOM-QM9 (Ramakrishnan et al., 2014) and GEOM-Drugs (Axelrod & Gomez-Bombarelli, 2022), molecular dynamics simulations incur prohibitive computational costs. Consequently, existing MD datasets (Chmiela et al., 2017; Meersche et al., 2024) are typically constrained to specific molecular systems or limited molecular classes, significantly restricting generalizability across diverse molecular types. **2.** *Modeling complexity.* MD trajectories inhabit high-dimensional spaces with an additional temporal dimension. The inherent complexity of the joint distribution $p(\mathbf{x}^{[T]})$ is further exacerbated by data scarcity, as insufficient training samples create greater sparsity in the high-dimensional data support, thereby complicating accurate density estimation.

Our solution. We propose to leverage a pretrained static structure (conformer) diffusion model and transforming it into an MD generation model, by stacking additional trainable temporal layers to enforce temporal consistency along each MD trajectory. Formally, given a pretrained conformer diffusion model $\boldsymbol{\epsilon}_\theta$ inducing the marginal $p_\theta(\mathbf{x})$, we devise $\boldsymbol{\epsilon}'_\psi$ for modeling the MD distribution $p_\psi(\mathbf{x}^{[T]})$, where $\psi = \{\theta, \phi\}$ with ϕ representing parameters in the additional temporal layers, indicating that the MD generative model with parameter set ψ is partially initialized from the pretrained structure model θ. The MD diffusion model is then optimized on the MD trajectory dataset with the loss

$$\mathcal{L}_{\text{traj}} = \mathbb{E}_{\mathbf{x}_0^{[T]}\sim\mathcal{D}_{\text{traj}},\tau,\boldsymbol{\epsilon}^{[T]}}\|\boldsymbol{\epsilon}^{[T]} - \boldsymbol{\epsilon}'_\psi(\mathbf{x}_\tau^{[T]}, \tau)\|_2^2, \quad (2)$$

where $\mathbf{x}_\tau^{[T]} = \sqrt{\bar{\alpha}_\tau}\mathbf{x}_0^{[T]} + \sqrt{1-\bar{\alpha}_\tau}\boldsymbol{\epsilon}^{[T]}$, $\tau \sim \text{Unif}(1, \mathcal{T})$, and $\boldsymbol{\epsilon}^{[T]} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the i.i.d. Gaussian noise.

Our proposal effectively addresses the core challenges. We mitigate MD data scarcity by initializing with a conformer model trained on large-scale datasets, transferring generalization capability to unseen molecules. Furthermore, our two-stage pipeline decomposes the complex modeling of $p(\mathbf{x}^{[T]})$ into manageable subproblems: conformer pretraining first models each frame independently, yielding $\tilde{p}_\theta(\mathbf{x}^{[T]}) = \prod_{t=0}^{T-1} p_\theta(\mathbf{x}^{(t)})$ without temporal correlation. The second stage incorporates additional parameters ϕ to capture the temporal dependency across different frames, leading to the joint distribution $p_\psi(\mathbf{x}^{[T]})$ with $\psi = \{\theta, \phi\}$. This approach efficiently offloads the complexity by using $\tilde{p}_\theta(\mathbf{x}^{[T]})$ as an anchor. The flowchart of our proposed framework is depicted in Fig. 1.

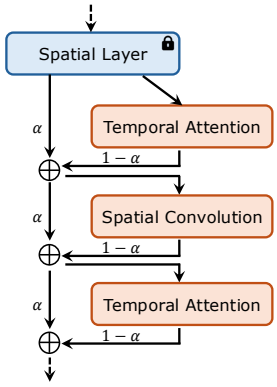

Figure 2. Flowchart of cascaded temporal interpolator.

4.2. Temporal Interpolator

With the proposed framework, it is still yet unrevealed how to allocate the additional parameters ϕ to capture the temporal dependency across frames for aligning the structures into an MD trajectory. To this end, we introduce a novel temporal interpolator module that entangles the pretrained structure denoiser $\boldsymbol{\epsilon}_\theta$ with the additional temporal network $\boldsymbol{\epsilon}_\phi$ through a linear interpolation:

$$\begin{aligned} \boldsymbol{\epsilon}'_\psi(\mathbf{x}_\tau^{[T]}, \tau) =& \alpha \tilde{\mathbf{x}}_\tau^{[T]} + (1-\alpha)\boldsymbol{\epsilon}_\phi(\tilde{\mathbf{x}}_\tau^{[T]}, \tau), \\ \text{s.t. } \tilde{\mathbf{x}}_\tau^{[T]} =& [\boldsymbol{\epsilon}_\theta(\mathbf{x}_\tau^{(t)}, \tau)]_{t=0}^{T-1}, \end{aligned} \tag{3}$$

where $\alpha \in \mathbb{R}$ is the interpolation coefficient, and $[\boldsymbol{\epsilon}_\theta(\mathbf{x}_\tau^{(t)}, \tau)]_{t=0}^{T-1}$ is the concatenation along the temporal axis for the outputs $\boldsymbol{\epsilon}_\theta(\mathbf{x}_\tau^{(t)})$ at frames $0 \le t \le T-1$.

Intuitively, Eq. 3 mixes the output from the structure model $\boldsymbol{\epsilon}_\theta$ together with the the temporal model $\boldsymbol{\epsilon}_\phi$ as the final output $\boldsymbol{\epsilon}'_\psi$, making it both structural and temporal-aware. Notably, compared with other mixing strategies, our design has several unique benefits, evidenced both in training and inference time.

We start by showing that the interpolation mechanism in Eq. 3 implicitly induces an intermediate distribution for the temporal network to learn. We reveal such insight in the following theorem.

Theorem 4.1 (Informal)**.** *Suppose $\boldsymbol{\epsilon}_\theta$ perfectly models $p(\mathbf{x})$ and $\boldsymbol{\epsilon}'_\psi$ perfectly models $p(\mathbf{x}^{[T]})$, then the interpolation in Eq. 3 implicitly induces the distribution $\hat{p}(\mathbf{x}^{[T]}) \propto p(\mathbf{x}^{[T]})^\beta \tilde{p}(\mathbf{x}^{[T]})^{1-\beta}$ for $\boldsymbol{\epsilon}_\phi$, where $\beta = \frac{1}{1-\alpha}$.*

Temporal interpolator reduces training overhead. Instead of directly matching the highly complex MD distribution $p(\mathbf{x}^{[T]})$, the temporal network is now expected to model an intermediate transition between the frame-independent distribution $\tilde{p}(\mathbf{x}^{[T]})$ obtained from the structure model and the target MD distribution $p(\mathbf{x}^{[T]})$, with $\beta = \frac{1}{1-\alpha}$ defining the weight. By this means, we relieve from the optimization difficulty for learning the MD distribution by leveraging the interpolation $\hat{p}(\mathbf{x}^{[T]})$ as the stepping stone, while also effectively taking advantage from the conformer pretraining by incorporating $\tilde{p}(\mathbf{x}^{[T]})$. Besides, another core design lies in that we inherit the output from the structure model $\tilde{\mathbf{x}}_\tau^{[T]}$ as the input to the temporal model, instead of feeding in the original noised trajectory $\mathbf{x}_\tau^{[T]}$. This is beneficial in terms of facilitates the optimization for $\boldsymbol{\epsilon}_\phi$. Consider the extreme case that the frame-independent distribution is close to the MD distribution, *i.e.*, $\tilde{p}(\mathbf{x}^{[T]}) \approx p(\mathbf{x})$. According to Theorem 4.1, we have that the implicit distribution for the temporal model to approach would be $\hat{p}(\mathbf{x}^{[T]}) \approx \tilde{p}(\mathbf{x}^{[T]})$. Therefore, equivalently the temporal model only needs to satisfy $\boldsymbol{\epsilon}_\theta(\tilde{\mathbf{x}}_\tau^{[T]}, \tau) \approx \tilde{\mathbf{x}}_\tau^{[T]}$, which can be simply realized by the residual connection and thus negligible optimization effort is required for $\boldsymbol{\epsilon}_\phi$. Empirically, we adopt the parameterization of $\alpha = \sigma(k)$ where $\sigma(\cdot)$ is the Sigmoid function to ensure a smooth interpolation, where k is a learnable parameter optimized during training.

Temporal interpolator enables flexible inference. Furthermore, our design unlocks intriguing potentials at inference time. First, by manually setting $\alpha = 1$, we completely blocks the output from the temporal network and the output from $\boldsymbol{\epsilon}'_\psi$ exactly equals $[\boldsymbol{\epsilon}_\theta(\mathbf{x}_\tau^{(t)}, \tau)]_{t=0}^{T-1}$, which is equivalent to performing the original structure generation with T being the batch size. Such property implies that our model always preserves the original conformer generation capability when setting $\alpha = 1$. Furthermore, by progressively decreasing α, the trajectory samples obtained by the interpolator will exhibit increasing temporal consistency since the temporal network implicitly aligns the structures temporally with the strength $1 - \alpha$.

Temporal interpolator preserves equivariance. Importantly, the linear interpolation rule for our temporal interpolator preserves the SE(3)-equivariance, given the SE(3)-equivariance of both the structure and the temporal models. This property is vital in terms of ensuring the SE(3)-invariance of the marginal, a critical inductive bias to promote data efficiency.

Cascaded temporal interpolator. Given the justifications for the interpolator, we further extend such operation in a block-wise manner, enabling more expressive information fusion between the pretrained structure model and the additional temporal module. Specifically, we perform the interpolation for the output from the structure and temporal model at the l-th block with $\alpha^{(l)} \in \mathbb{R}$ being the coefficient. Furthermore, we also incorporate the interpolation between each layer in the temporal block and the output from the structure block. Detailed flowchart can be found in Fig. 2. Such design inherits the benefits of the interpolator while permitting a much denser information flow between the network that evidently improves optimization.

4.3. Instantiation of EGINTERPOLATOR

Based on the dedicated design of the temporal interpolator in § 4.2, we describe the overall instantiation of our framework following the paradigm depicted in § 4.1.

Conformer pretrainings stage. The first stage of our pipeline is the structure pretraining using the large scale conformer dataset $\mathcal{D}_{\text{struct}}$. For the structure model $\boldsymbol{\epsilon}_\theta$, we resort to Equivariant Graph Convolution Layer (EGCL) proposed by (Satorras et al., 2021b) as the basic building block, whose update is denoted by

$$\mathbf{x}', \mathbf{h}' = f_{\text{ES}}(\mathbf{x}, \mathbf{h}, \mathcal{E}), \tag{4}$$

where ES is shorthand for Equivariant Structure layer. The denoiser $\boldsymbol{\epsilon}_\theta$ consists of L layers of f_{ES} stacked sequentially, and is optimized using the loss in Eq. 1 for structure pretraining.

MD training stage. With the pretrained conformer model $\boldsymbol{\epsilon}_\theta$, we conduct the second stage, the MD training stage with the limited-size MD dataset $\mathcal{D}_{\text{traj}}$, with the additionally initialized temporal network $\boldsymbol{\epsilon}_\phi$. For the temporal network, we utilize the Equivariant Temporal Attention Layer introduced in Han et al. (2024) to capture the temporal dependency across different frames using attention. In form, we have

$$\mathbf{x}'^{[T]}, \mathbf{h}'^{[T]} = f_{\text{ET}}(\mathbf{x}^{[T]}, \mathbf{h}^{[T]}, \mathcal{E}), \tag{5}$$

where ET refers to Equivariant Temporal layer. In particular, we design each temporal block as a stack of three layers, with one ET layer on the top, one on the bottom, and an ES layer in the middle. Such design is demonstrated to be favorable in practice due to the dense entanglement of both the structure and temporal layers. For each ES layer in the pretrained model, we initialize one temporal block, which, by putting together, constitutes one temporal interpolator block, leading to L temporal interpolator blocks in total. The model is then optimized using the trajectory denoising loss in Eq. 2 with the pretrained ES layers freezed. By this means, the final model is not only a performant MD generative model, but also yields exactly no performance degradation on the pretraining task of conformer generation, an interesting property that is never assured in previous works.

Forward simulation and interpolation. Our model can naturally support structure-conditioned MD generation such as the forward simulation, which conditions on the first frame $\mathbf{x}^{(0)}$, and interpolation, which conditions on the first and last frames, namely $\mathbf{x}^{(0)}$ and $\mathbf{x}^{(T-1)}$. Such purpose can be fulfilled by treating the conditioning frames as additional control signal, which is preserved without adding any noise. The conditioning frames are then passed along with the noisy frames into the interpolator and finally removed from the output to ensure the loss is only computed over the noisy frames.

5. Experiments

We refer to our framework as EGINTERPOLATOR, which leverages pretrained spatial layers from BASICES, our lightweight structure learning model. In this section, we evaluate EGINTERPOLATOR on its ability to generate realistic molecular dynamics (MD) trajectories for unseen organic molecules under practical data constraints—specifically, limited MD simulations of training molecules supplemented by diverse static structural data.

5.1. Conformer Pretraining

Datasets. We use GEOM-QM9 (Ramakrishnan et al., 2014) and GEOM-Drugs (Axelrod & Gomez-Bombarelli, 2022) following prior work in conformer generation (Xu et al., 2022; Ganea et al., 2021). Our spatial model is pretrained separately on each dataset, using the same train/validation splits as Xu et al. (2022) and a preprocessing pipeline similar to Ganea et al. (2021) (details in Appendix). This results in 37.7K/4.7K training/validation molecules with 188.6K/23.7K conformers for QM9 and 38.0K/4.8K training/validation molecules with 190.0K/23.7K conformers for Drugs. Unlike Xu et al. (2022); Shi et al. (2021a), who use a subsample of 200 molecules for testing, we evaluate on the full test set from the (Xu et al., 2021) codebase: 964 QM9 molecules (117.9K conformers) and 958 Drugs molecules (68.9K conformers).

Experimental Setup. We train our base BASICES model on this conformer generation task up to 800K steps for both QM9 and Drugs, learning 1000 denoising steps over only heavy atom coordinates. (Further details in Appendix).

Baselines: We compare the performance of our pretrained spatial models to that reported in Xu et al. (2022), namely GEODIFF-A as well as CONFGF (Shi et al., 2021a). Due to the different test benchmark, we compare metrics solely as a check of effective model learning over structure and conformer distributions.

Metrics. Per prior work in the space, we utilize the **Cov**erage and **Mat**ching metrics (Ganea et al., 2021; Xu et al., 2022) (Details in Appendix). We report both the Recall (R) to measure diversity of generated conformers and Precision (P) to measure accuracy of the samples. We utilize the default threshold δ values for coverage metrics, 0.5Å for QM9 and 1.25Å for Drugs.

Results & Discussion. Results are summarized in Figure 3. Our pretrained BASICES model performs competitively with prior SOTA methods. Notably, our training framework yields significant gains on Drugs, likely due to the emphasis on heavy atoms. For QM9, we prioritize precision-based metrics relevant to MD pretraining, which leads to slightly lower COV/MAT-R scores but superior fidelity in conformer bond angle and bond length distributions (see Appendix).

A. Coverage and Matching Results on QM9 and GEOM-Drugs

	Method	COV-R (%) ↑		MAT-R (Å) ↓		COV-P (%) ↑		MAT-P (Å) ↓	
		Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.
QM9	CONFGF	88.49	94.31	0.2673	0.2685	46.43	43.41	0.5224	0.5124
	GEODIFF-A	90.54	94.61	0.2104	0.2021	52.35	50.10	0.4539	0.4399
	BASICES	80.38	82.12	0.2819	0.2941	58.83	55.13	0.4298	0.4230
Drugs	CONFGF	62.15	70.93	1.1629	1.1596	23.42	15.52	1.7219	1.6863
	GEODIFF-A	88.36	96.09	0.8704	0.8628	60.14	61.25	1.1864	1.1391
	BASICES	93.15	100.00	0.7932	0.7812	69.68	76.35	1.0837	1.0381

B. Generated Conformers

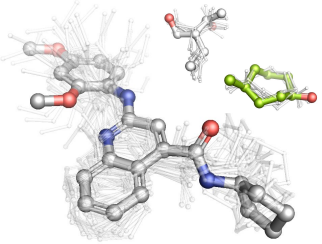

Figure 3. (**A**) reports the performance of BASICES along with borrowed numbers from (Xu et al., 2022) on SOTA baselines; (**B**) Highlights example conformers sampled from BASICES on both QM9 and Drugs.

5.2. Molecular Dynamics Finetuning

To generate MD data for diverse organic and drug-like molecules, we subsample QM9 and Drugs datasets, resulting in 1109/1018/240 train/validation/test splits for QM9 and 1137/1044/100 for Drugs. We then perform five, all-atom (including hydrogens), explicit-solvent simulations of 5 ns per molecule. In the test set, four trajectories are used as reference data and the fifth serves as an oracle baseline (MD ORACLE). Full simulation details and force field parameters are provided in the Appendix.

Experimental setup. Unless otherwise noted, all models are trained with trajectory time-steps $\Delta t = 5.2$ ps. We also continue to learn across heavy atoms and use 1000 denoising steps throughout all MD generation experiments.

Baselines. We compare the performance of our EGINTERPOLATOR framework against that of GEOTDM (Han et al., 2024), a recent all-atom trajectory diffusion model. In addition, we implement Markovian, autoregressive baselines using EGNN (Hoogeboom et al., 2022a) and the Equivariant Transformer (Thölke & Fabritiis, 2022) as push-forward networks, representing a mainstream alternative to diffusion-based approaches. These are denoted as AR + EGNN and AR + ET, respectively.

5.3. Unconditional Generation

In the *unconditional generation* setting, we train models to generate 2.6 ns trajectories with no reliance on a reference frame. For evaluation, we sample ten unconditional generations per molecule, resulting in 26 ns of generated trajectories. We focus on QM9 for this setting given the smaller memory footprint of these molecules.

Distributional Results. We assess the similarity between generated and reference trajectories using average Jensen–Shannon divergence (JSD) across key collective variable distributions, including bond lengths and bond angles—features tightly constrained by molecular energetics—as well as torsion angles and the top components from *time-lagged independent component analysis* (TICA), which captures slow dynamical modes. As shown in Table 1, EGINTERPOLATOR outperforms baselines on all metrics, exhibiting substantially closer alignment with ground-truth distributions. These results are further illustrated in Figure 4A and Figure 4B in a direct comparison to GeoTDM (Han et al., 2024) on an example molecule.

5.4. Forward Simulation

In the *forward simulation* setting, models are trained to generate 1.3 ns trajectories conditioned on a reference frame. We then extend these to 5.2 ns using successive block diffusion roll-outs, sampling five such trajectories per molecule. This setting focuses on GEOM-Drugs to target larger and more complex atomic systems (see Appendix).

Distributional Results. Once again, across all metrics in Table 1, EGINTERPOLATOR outperforms baselines and generates samples that more closely align with ground-truth distributions. More importantly, we highlight that EGINTERPOLATOR in fact approaches the distributional fidelity of the replicate MD ORACLE with respect to the torsion and TICA component variables.

Dynamical Results. We moreover evaluate torsional dynamics via decorrelation time and find that EGINTERPOLATOR better captures distinct relaxation behaviors within molecules compared to GeoTDM (Fig. 4E,F,G). Furthermore, by constructing Markov State Models (MSMs) from torsion angles and clustering into 10 metastates, we observe strong agreement in metastate occupancy between generated and reference trajectories (Fig. 4C). Our model even surpasses MD oracle baselines in capturing coarse-grained dynamical distributions (Fig. 4D).

5.5. Interpolation

In the *interpolation* (or *transition path sampling*) setting, models are trained to generate 0.52 ns trajectories conditioned on both start and end reference frames. Here we discuss results for Drugs, with QM9 details in the Ap-

Table 1. Performance Comparison on QM9 Unconditional Generation and Drugs Forward Simulation

	Method	JSD (Mean — Median) (↓)									
		Bond Angle		Bond Length		Torsion		TICA_0		TICA_0,1	
QM9	MD ORACLE	0.042	0.028	0.032	0.031	0.192	0.134	0.318	0.291	0.413	0.394
	AR + EGNN	0.702	0.677	0.770	0.780	0.702	0.761	0.770	0.788	0.820	0.824
	AR + ET	0.705	0.746	0.680	0.721	0.553	0.586	0.568	0.562	0.783	0.786
	GEOTDM	0.691	0.690	0.676	0.670	0.489	0.527	0.449	0.453	0.691	0.694
	EGINTERPOLATOR	**0.305**	**0.292**	**0.210**	**0.188**	**0.363**	**0.380**	**0.417**	**0.406**	**0.636**	**0.642**
Drugs	MD ORACLE	0.036	0.023	0.030	0.028	0.215	0.131	0.484	0.494	0.610	0.630
	AR + EGNN	0.663	0.655	0.748	0.784	0.723	0.741	0.716	0.731	0.806	0.821
	AR + ET	0.765	0.766	0.733	0.745	0.526	0.533	0.565	0.558	0.791	0.795
	GEOTDM	0.640	0.645	0.643	0.645	0.498	0.503	0.531	0.550	0.712	0.720
	EGINTERPOLATOR	**0.173**	**0.153**	**0.1419**	**0.112**	**0.377**	**0.388**	**0.454**	**0.441**	**0.650**	**0.644**

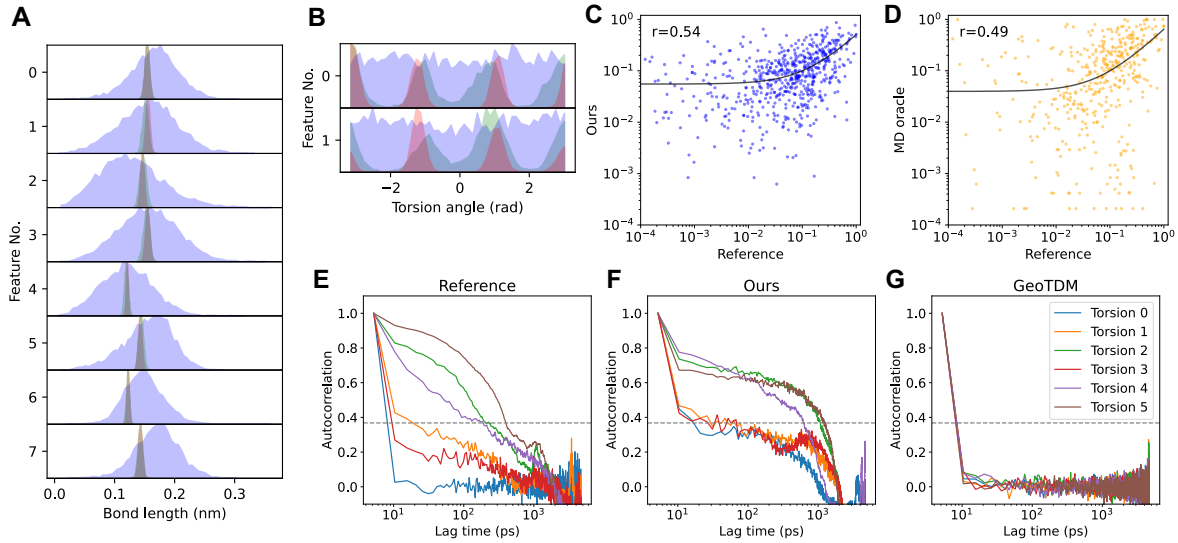

Figure 4. (**A**) Bond length distributions and (**B**) torsion angle distributions of reference trajectories (red), our generations (green), and GeoTDM generations (blue). MSM occupancies computed from reference trajectories versus our generations (**C**) or MD oracles (**D**). Autocorrelations of each torsion angle in an example molecule from reference trajectories (**E**), our generations (**F**), and GeoTDM (**G**). The gray dashline denotes the decorrelation threshold of 1/e.

pendix. Moreover, as this task requires conditioning on both endpoints, we compare only against the machine learning baseline GeoTDM (Han et al., 2024). We however adopt the MSM pipeline from (Jing et al., 2024c) to benchmark against MD oracle trajectories of varying lengths.

Evaluation. Following (Jing et al., 2024c), we evaluate interpolation as a transition path sampling task. After constructing an MSM from reference trajectories, we identify two distant metastates as start and end states and sample 900 frame pairs accordingly. We then generate 900 interpolation trajectories with our model and compare them to reference and MD oracle trajectories using JSD over metastate occupancies. Given the high barrier and rare transition between states, we also report valid path rate, average path probability, and valid path probability (details in Appendix).

Results. As shown in Fig. 5D, our 0.52 ns trajectories achieve the lowest JSD and highest average path probability, outperforming MD oracles of the same length and even matching longer trajectories in terms of path quality. While long MD oracles have higher valid path rates, our model excels at generating high-probability valid transitions. Fig. 5A, B illustrates the reference free energy surface (FES) and metastate assignments for a representative molecule, with a generated trajectory successfully traversing key states and reaching end states—highlighting the model's capacity for efficient and meaningful transition path sampling.

5.6. Ablations

Structural Pretraining. We ablate structural pretraining by evaluating a variant of our framework, EGINTERPOLATOR-

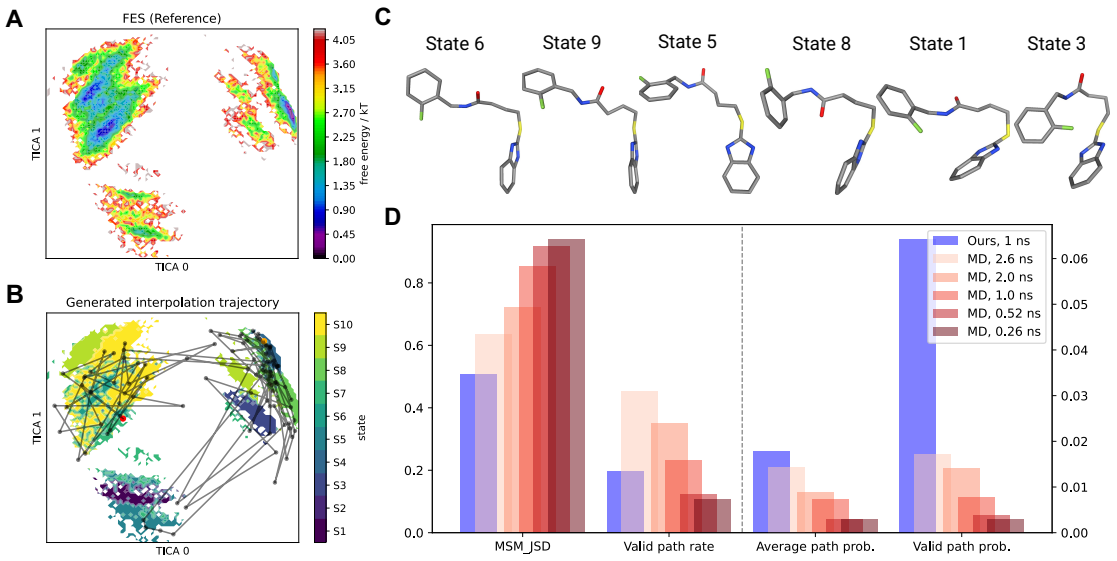

Figure 5. (**A**) Reference free energy surface along the top two TICA components. (**B**) Generated interpolation trajectory projected on the reference surface. Red point is the start frame, Orange point is the end frame. Reference surface is colored by the metastate assignment. (**C**) Key frames from intermediate metastates in the generated trajectory. (**D**) Statistics evaluating the JSD with the reference trajectories, valid path rate, average path probability, and valid path probability of our generated trajectories and replicate MD oracles.

Table 2. Ablation study on QM9 and Drugs.

	Method	JSD (Mean — Median) (↓)							
		Bond Angle		Bond Length		Torsion		Decorrelation	
		Mean	Median	Mean	Median	Mean	Median	Mean	Median
QM9	OURS-N	0.538	0.538	0.583	0.580	0.441	0.494	0.619	0.718
	OURS	**0.305**	**0.292**	**0.210**	**0.188**	**0.363**	**0.380**	0.607	0.727
Drugs	OURS-N	0.332	0.332	0.386	0.383	0.455	0.466	0.720	0.833
	OURS	**0.173**	**0.153**	**0.142**	**0.112**	**0.377**	**0.388**	**0.670**	**0.794**

Naive, trained directly on trajectories without any conformer pretraining. On both QM9 (unconditional generation) and Drugs (forward simulation), we observe degraded fidelity in bond length, angle, and torsion distributions, along with diminished de-correlation behavior (see table 2 and Figure in Appendix). These results highlight that structural pretraining enriches limited dynamic data and facilitates learning of accurate spatiotemporal distributions.

Frozen BASICES Layers. To assess the benefit of finetuning the frozen spatial encoder, we train a fully end-to-end version of EGINTERPOLATOR on the Drugs forward simulation task. As shown in Figure 6, performance remains largely unchanged across metrics, indicating that the pretrained spatial model generalizes well without task-specific tuning, while the temporal layers effectively capture the necessary dynamic information.

The values of α. We present the values of α after convergence in Sec. F. Interestingly, we observe that the values of α exhibit shared pattern across different tasks on the same dataset while varying across different datasets, indicating that α captures dataset-specific temporal correlation.

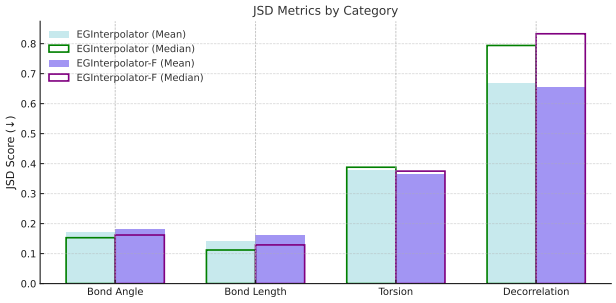

Figure 6. JSD metrics computed for Bond Angles, Bond Lengths, Torsions, and Decorrelation Times. Compared between EGINTERPOLATOR (green) and EGINTERPOLATOR-**F** (purple).

6. Discussion

Limitation. As shown in Table 1, our model, as a deep generative model-based surrogate, may not yield MD simulation that is at exactly the same level of accuracy as the MD oracle. Yet, it still produces MD trajectories of very high quality and is able to generalize to novel molecular systems.

Conclusion. We have introduced a diffusion model for modeling MD distributions by pretraining a structure model on conformer dataset and then finetuning on trajectory dataset. At the core of our approach is an module named EGInterpolator that mixes the output from the pretrained structure model and the temporal model to captures the temporal dependency. Our approach demonstrates strong performance in terms of producing realistic MD trajectories on diverse benchmarks and tasks.

References

Alder, B. J. and Wainwright, T. E. Studies in molecular dynamics. i. general method. *The Journal of Chemical Physics*, 31(2):459–466, August 1959. doi: 10.1063/1.1730376. URL https://doi.org/10.1063/1.1730376.

Antalík, A., Levy, A., Kvedaravičiūtė, S., Johnson, S. K., Carrasco-Busturia, D., Raghavan, B., Mouvet, F., Acocella, A., Das, S., Gavini, V., Mandelli, D., Ippoliti, E., Meloni, S., Carloni, P., Rothlisberger, U., and Olsen, J. M. H. Mimic: A high-performance framework for multiscale molecular dynamics simulations. *The Journal of Chemical Physics*, 161(2), July 2024. ISSN 1089-7690. doi: 10.1063/5.0211053. URL http://dx.doi.org/10.1063/5.0211053.

Axelrod, S. and Gomez-Bombarelli, R. Geom: Energy-annotated molecular conformations for property prediction and molecular generation, 2022. URL https://arxiv.org/abs/2006.05531.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

Boothroyd, S., Behara, P. K., Madin, O. C., Hahn, D. F., Jang, H., Gapsys, V., Wagner, J. R., Horton, J. T., Dotson, D. L., Thompson, M. W., Maat, J., Gokey, T., Wang, L.-P., Cole, D. J., Gilson, M. K., Chodera, J. D., Bayly, C. I., Shirts, M. R., and Mobley, D. L. Development and benchmarking of open force field 2.0.0: The sage small molecule force field. *Journal of Chemical Theory and Computation*, 19(11):3251–3275, 2023. doi: 10.1021/acs.jctc.3c00039. URL https://doi.org/10.1021/acs.jctc.3c00039. PMID: 37167319.

Brandstetter, J., Hesselink, R., van der Pol, E., Bekkers, E. J., and Welling, M. Geometric and physical quantities improve e(3) equivariant message passing, 2022. URL https://arxiv.org/abs/2110.02905.

Chmiela, S., Tkatchenko, A., Sauceda, H. E., Poltavsky, I., Schütt, K. T., and Müller, K.-R. Machine learning of accurate energy-conserving molecular force fields. *Science advances*, 3(5):e1603015, 2017.

Darden, T., York, D., and Pedersen, L. Particle mesh ewald: An n·log(n) method for ewald sums in large systems. *The Journal of Chemical Physics*, 98(12):10089–10092, 1993. ISSN 0021-9606. doi: 10.1063/1.464397.

dos Santos Costa, A., Mitnikov, I., Pellegrini, F., Daigavane, A., Geiger, M., Cao, Z., Kreis, K., Smidt, T., Kucukbenli, E., and Jacobson, J. Equijump: Protein dynamics simulation via so(3)-equivariant stochastic interpolants, 2024. URL https://arxiv.org/abs/2410.09667.

Eastman, P., Swails, J., Chodera, J. D., McGibbon, R. T., Zhao, Y., Beauchamp, K. A., Wang, L.-P., Simmonett, A. C., Harrigan, M. P., Stern, C. D., Wiewiora, R. P., Brooks, B. R., and Pande, V. S. Openmm 7: Rapid development of high performance algorithms for molecular dynamics. *PLOS Computational Biology*, 13(7):1–17, 07 2017. doi: 10.1371/journal.pcbi.1005659. URL https://doi.org/10.1371/journal.pcbi.1005659.

Fuchs, F. B., Worrall, D. E., Fischer, V., and Welling, M. Se(3)-transformers: 3d roto-translation equivariant attention networks, 2020. URL https://arxiv.org/abs/2006.10503.

Ganea, O.-E., Pattanaik, L., Coley, C. W., Barzilay, R., Jensen, K. F., Green, W. H., and Jaakkola, T. S. Geomol: Torsional geometric generation of molecular 3d conformer ensembles, 2021. URL https://arxiv.org/abs/2106.07802.

Han, J., Xu, M., Lou, A., Ye, H., and Ermon, S. Geometric trajectory diffusion models. *arXiv preprint arXiv:2410.13027*, 2024.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020a.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models, 2020b. URL https://arxiv.org/abs/2006.11239.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pp. 8867–8887. PMLR, 2022a.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant diffusion for molecule generation in 3d, 2022b. URL https://arxiv.org/abs/2203.17003.

Hyvärinen, A. and Dayan, P. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Jing, B., Eismann, S., Suriana, P., Townshend, R. J. L., and Dror, R. Learning from protein structure with geometric vector perceptrons, 2021. URL https://arxiv.org/abs/2009.01411.

Jing, B., Berger, B., and Jaakkola, T. Alphafold meets flow matching for generating protein ensembles, 2024a. URL https://arxiv.org/abs/2402.04845.

Jing, B., Stärk, H., Jaakkola, T., and Berger, B. Generative modeling of molecular dynamics trajectories. *arXiv preprint arXiv:2409.17808*, 2024b.

Jing, B., Stärk, H., Jaakkola, T., and Berger, B. Generative modeling of molecular dynamics trajectories, 2024c. URL https://arxiv.org/abs/2409.17808.

Kabsch, W. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976.

Klein, L., Foong, A. Y. K., Fjelde, T. E., Mlodozeniec, B., Brockschmidt, M., Nowozin, S., Noé, F., and Tomioka, R. Timewarp: Transferable acceleration of molecular dynamics by learning time-coarsened dynamics, 2023. URL https://arxiv.org/abs/2302.01170.

Laio, A. and Parrinello, M. Escaping free-energy minima. *Proceedings of the National Academy of Sciences*, 99(20):12562–12566, 2002. doi: 10.1073/pnas.202427399. URL https://www.pnas.org/doi/abs/10.1073/pnas.202427399.

McCammon, J. A., Gelin, B. R., and Karplus, M. Dynamics of folded proteins. *Nature*, 267(5612):585–590, 1977. doi: 10.1038/267585a0.

McIsaac, A., Behara, P. K., Gokey, T., Cavender, C., Horton, J., Wang, L., Jang, H., Wagner, J., Cole, D., Bayly, C., and Mobley, D. openforcefield/openff-forcefields, January 2024. URL https://doi.org/10.5281/zenodo.10553473.

Meersche, Y. V., Cretin, G., Gheeraert, A., Gelly, J.-C., and Galochkina, T. Atlas: protein flexibility description from atomistic molecular dynamics simulations. *Nucleic Acids Research*, 52(D1):D384–D392, 2024. doi: 10.1093/nar/gkad1084.

Morehead, A. and Cheng, J. Geometry-complete diffusion for 3d molecule generation and optimization, 2024. URL https://arxiv.org/abs/2302.04313.

Noé, F., Olsson, S., Köhler, J., and Wu, H. Boltzmann generators – sampling equilibrium states of many-body systems with deep learning, 2019. URL https://arxiv.org/abs/1812.01729.

Park, J. and Shen, Y. Equivariant blurring diffusion for hierarchical molecular conformer generation, 2024. URL https://arxiv.org/abs/2410.20255.

Rahman, A. Correlations in the motion of atoms in liquid argon. *Phys. Rev.*, 136:A405–A411, Oct 1964. doi: 10.1103/PhysRev.136.A405. URL https://link.aps.org/doi/10.1103/PhysRev.136.A405.

Ramakrishnan, R., Dral, P. O., Rupp, M., and von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1, 2014.

Satorras, V. G., Hoogeboom, E., Fuchs, F. B., Posner, I., and Welling, M. E(n) equivariant normalizing flows. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021a. URL https://openreview.net/forum?id=N5hQI_RowVA.

Satorras, V. G., Hoogeboom, E., and Welling, M. E(n) equivariant graph neural networks. *arXiv preprint arXiv:2102.09844*, 2021b.

Scherer, M. K., Trendelkamp-Schroer, B., Paul, F., Pérez-Hernández, G., Hoffmann, M., Plattner, N., Wehmeyer, C., Prinz, J.-H., and Noé, F. Pyemma 2: A software package for estimation, validation, and analysis of markov models. *Journal of Chemical Theory and Computation*, 11(11):5525–5542, 2015. doi: 10.1021/acs.jctc.5b00743. URL https://doi.org/10.1021/acs.jctc.5b00743. PMID: 26574340.

Shaw, D. E., Dror, R. O., Salmon, J. K., Grossman, J. P., Mackenzie, K. M., Bank, J. A., Young, C., Deneroff, M. M., Batson, B., Bowers, K. J., Chow, E., Eastwood, M. P., Ierardi, D. J., Klepeis, J. L., Kuskin, J. S., Larson, R. H., Lindorff-Larsen, K., Maragakis, P., Moraes, M. A., Piana, S., Shan, Y., and Towles, B. Millisecond-scale molecular dynamics simulations on anton. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, SC '09, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605587448. doi: 10.1145/1654059.1654126. URL https://doi.org/10.1145/1654059.1654126.

Shi, C., Luo, S., Xu, M., and Tang, J. Learning gradient fields for molecular conformation generation. In *International conference on machine learning*, pp. 9558–9568. PMLR, 2021a.

Shi, C., Luo, S., Xu, M., and Tang, J. Learning gradient fields for molecular conformation generation, 2021b. URL https://arxiv.org/abs/2105.03902.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Thölke, P. and Fabritiis, G. D. Torchmd-net: Equivariant transformers for neural network based molecular potentials, 2022. URL https://arxiv.org/abs/2202.02541.

Verlet, L. Computer "experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Phys. Rev.*, 159:98–103, Jul 1967. doi: 10.1103/PhysRev.159.98. URL https://link.aps.org/doi/10.1103/PhysRev.159.98.

Xu, C., Wang, H., Wang, W., Zheng, P., and Chen, H. Geometric-facilitated denoising diffusion model for 3d molecule generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 338–346, 2024a.

Xu, M., Wang, W., Luo, S., Shi, C., Bengio, Y., Gomez-Bombarelli, R., and Tang, J. An end-to-end framework for molecular conformation generation via bilevel programming. In *International Conference on Machine Learning*, pp. 11537–11547. PMLR, 2021.

Xu, M., Yu, L., Song, Y., Shi, C., Ermon, S., and Tang, J. Geodiff: A geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=PzcvxEMzvQC.

Xu, M., Powers, A., Dror, R., Ermon, S., and Leskovec, J. Geometric latent diffusion models for 3d molecule generation. In *International Conference on Machine Learning*. PMLR, 2023.

Xu, M., Han, J., Lou, A., Kossaifi, J., Ramanathan, A., Azizzadenesheli, K., Leskovec, J., Ermon, S., and Anandkumar, A. Equivariant graph neural operator for modeling 3d dynamics. In *Forty-first International Conference on Machine Learning*, 2024b. URL https://openreview.net/forum?id=dccRCYmL5x.

Zhang, Z., Liu, X., Yan, K., Tuckerman, M. E., and Liu, J. Unified efficient thermostat scheme for the canonical ensemble with holonomic or isokinetic constraints via molecular dynamics. *The Journal of Physical Chemistry A*, 123(28):6056–6079, 2019. doi: 10.1021/acs.jpca.9b02771. URL https://doi.org/10.1021/acs.jpca.9b02771. PMID: 31117592.

A. Experiments Continued

A.1. Optimizing for Conformer Precision Metrics

As discussed in Section 5.1, we prioritize precision-based conformer quality metrics when selecting our base structure model. While this may come at the cost of lower COV/MAT-R scores, we observe superior fidelity in bond length, bond angle, and torsion angle distributions—an aspect we consider more critical for a pretrained structure module.

Table 3. Conformer metrics on QM9 compared between two checkpoints.

Checkpoint	COV-R (%) ↑		MAT-R (Å) ↓		COV-P (%) ↑		MAT-P (Å) ↓	
	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.
99	**89.37**	**93.55**	0.2838	0.2932	54.49	51.79	0.4828	0.4741
539	80.38	82.12	0.2819	0.2941	**58.83**	**55.13**	**0.4298**	**0.4230**

We highlight this point using two checkpoints of the BASICES model trained on QM9. In Table 3 we can see that while *539* lacks in COV-R, it does substantially better than *99* in COV/MAT-P metrics. In Figure 7, we then see that *539* reflects high quality bond angle, length, and torsion distributions, as compared to *99*. We select checkpoint *539* for the conformer results reported in Section 5.1 and for training the downstream trajectory models.

A.2. QM9 Interpolation

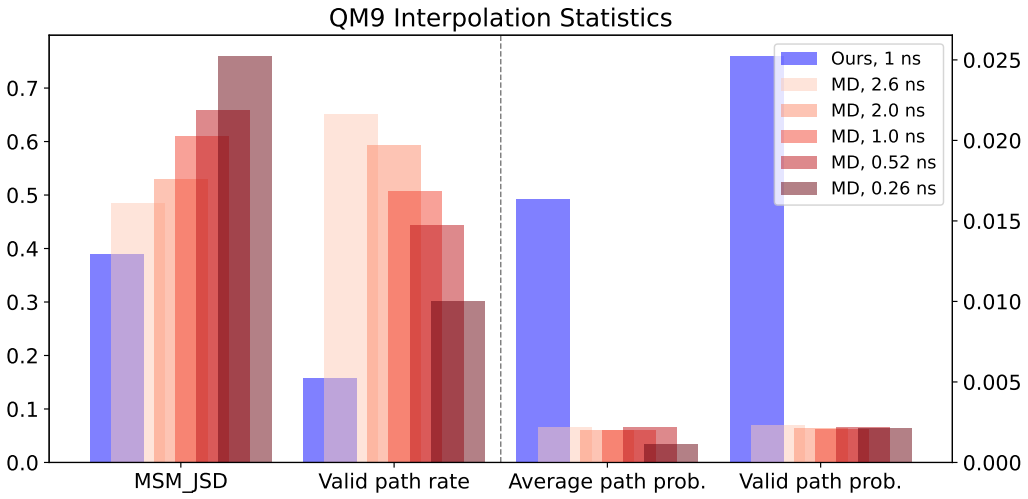

Figure 6. Statistics evaluating the JSD with the reference trajectories, valid path rate, average path probability, and valid path probability of our generated trajectories and replicate MD oracles.

For the interpolation task on QM9 dataset, as shown in Figure 6, our 0.52 ns trajectories consistently achieve the lowest Jensen-Shannon Divergence (JSD) and the highest average path probability, outperforming MD oracles of the same duration. It reveals that our method can samples transition paths between far metastates more efficiently. While the MD oracles exhibit higher valid path rates in this setting, our model still performs competitively in generating high-probability valid transitions.

Figure 11 illustrates several free energy surfaces (FES) and corresponding metastate assignments for representative molecules. We observe that the generated trajectories successfully traverse key intermediate states and reach the appropriate end states, demonstrating the model's ability to perform efficient and meaningful transition path sampling.

A.3. Trajectory Model Ablations

A.3.1. Generalization to an Extended Test Set

To further assess the robustness of our QM9 unconditional generation model, we evaluate performance on an extended test set of 959 molecules, which includes the original test set from Section 5.2. As shown in Table 4, we compare GeoTDM (Han et al., 2024), EGInterpolator-**N** (without structure pretraining), and our full EGInterpolator model. While all models perform comparably on this larger evaluation set, EGInterpolator consistently outperforms the baselines, underscoring its strong generalization and the value of structural pretraining.

Table 4. JSD Metric (↓) for QM9 Unconditional Generation. Top: Trained on **Standard** Train, evaluated on **Enlarged** Test. Bottom: Trained on **Enlarged** Train, evaluated on **Standard** Test.

Train → Test	**Method**	Bond Angle		Bond Length		Torsion		TICA_0		TICA_0,1	
		Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.
Standard → Enlarged	GeoTDM	0.690	0.690	0.674	0.668	0.488	0.529	0.452	0.451	0.695	0.699
	EGInterpolator-N	0.539	0.538	0.584	0.582	0.447	0.492	0.438	0.440	0.678	0.685
	EGInterpolator	**0.307**	**0.293**	**0.214**	**0.194**	**0.361**	**0.385**	**0.416**	**0.409**	**0.633**	**0.639**
Enlarged → Standard	GeoTDM	0.757	0.757	0.782	0.793	0.488	0.533	0.454	0.453	0.697	0.703
	EGInterpolator-N	0.470	0.460	0.540	0.544	0.433	0.481	0.443	0.440	0.681	0.691
	EGInterpolator	**0.296**	**0.286**	**0.261**	**0.247**	**0.370**	**0.388**	**0.405**	**0.394**	**0.636**	**0.638**

A.3.2. Contribution of an Extended Train Set

While our framework is motivated by the scarcity of trajectory data, we also evaluate model performance under increased supervision. We train on an enlarged dataset—4× larger than the original—comprising 4437 molecules, with the original split from Section 5.2 as a subset. As shown in Table 4, while EGInterpolator-**N** and EGInterpolator interestingly do not improve substantially with more data, the latter maintains a clear advantage. This highlights the continued value of structural pretraining even in higher-data regimes.

B. Experimental Details

B.1. Conformer Pretraining

B.1.1. Data Preprocessing

The datasets obtained from the (Xu et al., 2022) codebase are provided as pickle files, each containing a list of PyTorch Geometric data objects representing individual conformers. We apply the following filtering steps to ensure data quality. First, we verify that the saved `RDMol` objects can be successfully sanitized using RDKit. Next, we remove any conformers exhibiting fragmentation in their `RDMol` representations. Following (Ganea et al., 2021), we also account for conformers that may have reacted in the original data generation process. Namely, we compare the canonical SMILES strings derived from both the saved SMILES and the corresponding `RDMol`, and discard any conformers where the two do not match. We also exclude any molecules whose saved SMILES cannot be converted into a valid `RDMol` by RDKit. Lastly, specific to our method, we remove hydrogens from the molecules according to `rdkit.Chem.RemoveHs` [2] and retain heavy atoms. For QM9, this leaves [*C, N, O, F*]. For Drugs, we have [*C, N, O, S, P, F, Cl, Br, I, B, Si*].

B.1.2. Training Details

We train both the QM9 and Drugs conformer models using 4 NVIDIA RTX A4000 GPUs, with an effective batch size of 128 (32 samples per GPU) and a learning rate of 1×10^{-4}. Training is carried out until convergence, typically around 800K steps. As described in Section 5.1, all models are trained using 1000 diffusion steps. We adopt a DDPM framework (Ho et al., 2020b) with a linear noise schedule. Additionally, we employ an equivariant loss function that leverages optimal Kabsch alignment (Kabsch, 1976), with more details in Section C.4.

[2]Note that `RemoveHs` does not eliminate all hydrogen atoms and may retain chemically relevant ones (see the RDKit documentation). Our method explicitly incorporates and models such retained hydrogens.

B.1.3. Evaluation Details

We evaluate the quality of generated conformers using Coverage (COV-P) and Matching (MAT-P), both based on the root mean square deviation (RMSD) computed after Kabsch alignment (Kabsch, 1976).

Let S_g and S_r denote the sets of generated and reference conformers, respectively. The metrics are defined as:

$$\text{COV-P}(S_g, S_r) = \frac{1}{|S_g|} \left| \left\{ \hat{C} \in S_g \;\middle|\; \min_{C \in S_r} \text{RMSD}(\hat{C}, C) \leq \delta \right\} \right|, \tag{6}$$

$$\text{MAT-P}(S_g, S_r) = \frac{1}{|S_g|} \sum_{\hat{C} \in S_g} \min_{C \in S_r} \text{RMSD}(\hat{C}, C), \tag{7}$$

where δ is a predefined threshold. COV-R and MAT-R, inspired by *Recall*, are defined analogously by swapping S_g and S_r.

Following (Xu et al., 2022), we set $|S_g| = 2 \times |S_r|$ per molecule. The results reported in Section 5.1 correspond to the average COV-*/MAT-* scores across all test molecules. COV-P reflects precision by measuring the fraction of generated conformers that are sufficiently close to the reference set (within threshold δ), while MAT-P captures the mean deviation of each generated conformer from its closest reference match. High COV and low MAT scores indicate greater fidelity and precision in conformer generation.

B.2. Molecular Dynamics for Small Molecules

B.2.1. Parameterization

We run all-atom molecular dynamics simulations, including hydrogens, using OpenMM (Eastman et al., 2017) and employ `openmmforcefields` to apply small molecule force field parameterizations developed by the Open Force Field Initiative (OpenFF) (Boothroyd et al., 2023). We follow the setup guidelines provided in the `openmmforcefields` GitHub repository. Specifically, we adopt the `openff-2.2.1` (Sage) (McIsaac et al., 2024) small molecule force field in conjunction with a base `amber/protein.ff14SB.xml` protein force field and a combination of `amber/tip3p_standard.xml` and `amber/tip3p_HFE_multivalent.xml` for explicit solvent and ion parameters. Continuing with standard hyperparameters, we set the nonbonded cutoff to 0.9 nm and the switch distance to 0.8 nm. Hydrogen mass repartitioning (HMR) is applied with a mass of 1.5 amu, along with constraints on all hydrogen bonds. Long-range electrostatic interactions are computed using the Particle Mesh Ewald (PME) method under periodic boundary conditions. A padding of 1.5 nm is used for the explicit solvent box.

B.2.2. Simulation

All molecular dynamics simulations are performed using a friction coefficient of 1 ps^{-1}, a temperature of 300 K, and an integration timestep of 4 fs, employing the `LangevinMiddleIntegrator` (Zhang et al., 2019). As described in Section 5.2, five independent trajectories are generated per molecule, each initialized from a conformer assigned to that molecule in the selected data subset. Each trajectory simulation begins with energy minimization, followed by 5000 steps of equilibration under constant volume and temperature (NVT) conditions. This is followed by a 5 ns production run under constant pressure and temperature (NPT) conditions, comprising a total of 1.25M integration steps. Trajectory simulation is parallelized across 32 NVIDIA RTX A4000 GPUs and saved with a frame rate of 400 fs/0.4 ps.

B.3. Trajectory Finetuning

B.3.1. Dataset Preparation

As mentioned in Section 5.2, we randomly sample a subset of the molecules from the GEOM-QM9 and Drugs conformer data to generate trajectory data from. As this is quite costly, for Drugs we generate simulations for the standard train/validation/test splits mentioned in Section 5.2. For QM9, we generate data for enlarged train/test sets along with the standard validation set. We then subsample 25% of the enlarged splits to be the standard train/test sets. A summary of the dataset splits is provided below:

- **Drugs:**
  - *Standard splits:* `1137/1044/100` train/validation/test molecules

(5682/5209/496 associated trajectories)

- **QM9:**
  - *Standard splits:* 1109/1018/240 train/validation/test molecules (5534/5080/1193 associated trajectories)
  - *Enlarged sets:* 4437/959 train/test molecules (22132/4793 associated trajectories)

As a note, out of the test trajectories, we select 1 out of 5 per molecule to be the MD ORACLE baseline. Moreover, we filter out any molecules over 60 atoms in the Drugs dataset to reduce memory usage variance. Finally, the test set for the interpolation is a subset of the standard test sets mentioned above. We further define this process of selection in Section B.6 and B.3.3.

B.3.2. TRAINING PROTOCOL

While the compute setup and batch size vary across datasets and generation settings, we consistently employ a DDPM framework with a linear noise schedule and train all models using 1000 diffusion steps. A fixed learning rate of 1×10^{-4} is used and training is performed until convergence. Additionally, we adopt an equivariant loss function based on optimal global Kabsch alignment of trajectories, as detailed in Section C.4. Setting-specific training configurations are provided in Sections B.4-B.6.

B.3.3. EVALUATION METRICS

Jensen-Shannon Divergence. We compute the JSD as implemented in scipy, where $m = (p + q)/2$:

$$\sqrt{\frac{D(p \mid\mid m) + D(q \mid\mid m)}{2}} \tag{8}$$

- Torsions: The 1D JSD is computed over a 100-bin histogram discretized across $[-\pi, \pi]$.
- Bond Angles: The 1D JSD is computed over a 100-bin histogram discretized across $[0, \pi]$.
- Bond Lengths: The 1D JSD is computed over a 100-bin histogram discretized across $[100, 220]$ pm.
- Torsion decorrelation: The 1D JSD is computed over 275-bin histogram discretized across $[5, 1380]$ ps, which are corresponding to the minimum and maximum torsion decorrelation time of molecules across the dataset.
- TICA-0 and TICA-0,1: We reduce the dimensionality of the trajectory by time-lagged independent component analysis (TICA). Then 1D, 2D JSDs are computed over 100-bin histograms on the first TICA component (TICA-0) and the first two components (TICA-0,1), respectively. Since different molecules have totally different TICA projections and values, we use the minimum and maximum values from each molecule as its unique discretization range for TICA-0 and TICA-0,1. We use 10.4 ps (2 steps) lag time for QM9 and 20.8 ps (4 steps) for drugs.

Markov State Models. We intensively use Markov State Models (MSM) for interpolation tasks. We featurize reference trajectories with all torsion angles except for those within an aromatic ring. Then TICA is performed on the torsion-based trajectories. After dimensionality reduction, a k-means clustering algorithm is used to discretize the trajectories to 100 clusters. An MSM analysis is performed on the trajectories of 100 states and PCCA+ spectral clustering from PyEMMA package (Scherer et al., 2015) is used to aggregate clusters to 10 coarse metastates. A second MSM analysis is done on the coarse trajectories. We use 52 ps (10 steps) lag time for QM9 and 104 ps (20 steps) for drugs.

To sample the start and end frames used in the interpolation task, we compute the flux matrix over the 10 metastates. To construct a high barrier and rare transition probability, we choose the two states with least flux between them as start and end states. Then we randomly sample 900 start and end frames from the corresponding states, and those frames are used as the conditions in the interpolation inference process. The generated trajectories undergo the same featurization process, and then projected on the TICA components defined by the reference trajectories. They are further discretized according to the reference metastate assignments, and a new MSM is performed on the discretized generation trajectories.

To compare the generation with reference trajectories, we compute the JSD over the metastate occupancy probabilites. To evaluate interpolation sampling quality, we compute the average path probability, valid path rate, and valid path probability as described in (Jing et al., 2024c). The average path probability is the average of all paths' likelihood for transitioning from the start to the end. The valid path rate is the fraction of paths that successfully traverse from the start to the end. The valid path probability is the average of all valid paths' likelihood (excluding zero-probability paths). To fairly compare the generation and MD oracle, we truncate the MD oracle trajectories to varying time length, and sample 900 transition paths based on the MSM constructed from the metastates. With the sampled transition paths, we can compute the JSD over metastates, average path probability, valid path rate, and valid path probability of MD oracles.

B.4. Unconditional Generation Details

Training. Training is conducted by denoising randomly sampled 2.6 ns segments (500 frames) from the training trajectories. For QM9, we utilize 8 NVIDIA RTX A4000 GPUs with an effective batch size of 32 (4 samples per GPU), training the models for 400 epochs.

Evaluation. For each molecule in the test set, we generate ten independent 2.6 ns segments (500 frames each). Distributional histograms are then computed from these generated trajectories and compared against those derived from four reference 5 ns molecular dynamics (MD) trajectories. Results reported for this model setting for QM9 include both the standard test in Section 5.3 and enlarged test set in Section A.3.2-A.3.3.

B.5. Forward Simulation Details

Training. Training is conducted by randomly sampling 251-frame segments at a 5.2 ps frame rate and denoising the subsequent 250 frames (corresponding to 1.3 ns), conditioned on the initial frame-0. For the Drugs dataset, we utilize 4 NVIDIA A100 GPUs with an effective batch size of 32 (8 samples per GPU), training the models for 400 epochs.

Evaluation. For each molecule in the test set, we generate five forward roll-outs of 5.2 ns (1,000 frames total), each conditioned on the first frame of a reference trajectory. Distributional histograms are then computed from the generated trajectories and compared against those obtained from four reference 5 ns molecular dynamics (MD) trajectories. For a fair comparison, we truncate our generation trajectories to the same length as the reference trajectories in evaluation. Results reported for this model setting for Drugs are based on the standard test set in Section 5.4.

B.6. Interpolation Details

Training. Training is conducted by randomly sampling 101-frame segments at a 5.2 ps frame rate and denoising the middle 99 frames (corresponding to $\approx$0.52 ns), conditioned on frame-0 and frame-100. For the QM9 dataset, we utilize 2 NVIDIA A100 GPUs with an effective batch size of 128 (64 samples per GPU), training the models for 300 epochs. For the Drugs dataset, we utilize 4 NVIDIA A100 GPUs with an effective batch size of 32 (8 samples per GPU), training the models for 400 epochs.

Evaluation. For each molecule in the test set, we perform featurization, dimensionality reduction, and clustering on the reference trajectories. We then construct an MSM on the discretized trajectories and retain only those test molecules for which all microstates from clustering are represented in the MSM. After filtering, this yields 124 QM9 and 36 Drug test molecules. Due to computational constraints, we subsample 80 QM9 molecules while using all 36 Drug molecules for inference and evaluation. For each selected test molecule, we generate 900 interpolation trajectories conditioned on 900 sampled start and end states. For each MD oracle length, we also sample 900 transition paths. We report the average results across all molecules successfully modeled by the MSM, as shown in Section 5.5, Figure 5, as well as Section A.2, Figure 6 (see details in Section B.3.3).

C. Method Details

C.1. Molecule Input Representation

Throughout our framework, input molecules are represented as 2D heterogeneous graphs. The bonding network includes both the original bond types present in the molecule and additional higher-order edges that we incorporate. Specifically, we include edges up to third-order for both the QM9 and Drug datasets. Following the approach of (Shi et al., 2021b), this augmentation is designed to facilitate more effective information transfer between atoms involved in bond angle and torsion

angle interactions.

Table 5. Atom and bond embedding specifications.

Embedding Type	Input	Dimension
Atom Embedding	Atomic Number	30
Bond Embedding	No Bond, Bond Type, 2nd/3rd-order edge	4

We defined learned embeddings for atom type as well as bond type. Moreover, we also provide input node features per atom, largely based on (Ganea et al., 2021). Below, we provide a table with these details. These two information sources, the learned embedding and input features, as combined in our embedding module as described in Section C.2.

Table 6. Node feature vector based on atom-level properties.

Atom Features			
Indices	**Description**	**Options**	**Type**
0–1	Aromaticity	true, false	One-hot
2–7	Hybridization	sp, sp^2, sp^3, sp^3d, sp^3d^2, other	One-hot
8	Partial charge	$\mathbb{R}$	Value
9–16	Implicit valence	0, 1, 2, 3, 4, 5, 6, other	One-hot
17–24	Degree	0, 1, 2, 3, 4, 5, 6, other	One-hot
25–28	Formal charge	-1, 0, 1, other	One-hot
29–35	In ring of size x	3, 4, 5, 6, 7, 8, other	k-hot
36–39	Number of rings	0, 1, 2, 3+	One-hot
40–42	Chirality	CHI_TETRAHEDRAL_CW, CHI_TETRAHEDRAL_CCW, unspecified/other	One-hot

C.2. Architectures

Embeddings. Across all of our models—both conformer and trajectory—we use a hidden dimension of 128 and a diffusion timestep embedding dimension of 32. For molecular embeddings, we combine atom type embeddings and atom-level features via a single linear projection: $\mathbb{R}^{\texttt{node\_dim}+\texttt{ft\_dim}} \to \mathbb{R}^{\texttt{node\_dim}}$.

BASICES. As introduced in Section 4.3, our BASICES architecture consists of 6 Equivariant Graph Convolution (EGCL) layers, following the formulation in (Satorras et al., 2021b). To promote interaction between invariant and equivariant representations, we insert a Geometric Vector Perceptron (GVP) (Jing et al., 2021) transition layer after each EGCL block. The full model contains approximately 918K parameters.

EGINTERPOLATOR. As described in Section 4.3, EGINTERPOLATOR extends BASICES by introducing temporal attention to model dependencies across trajectory frames. Specifically, we incorporate the Equivariant Temporal Attention Layer (ETLayer) from (Han et al., 2024) to capture temporal structure through attention mechanisms. The architecture is constructed by stacking an additional sequence of ETLayer + EGCL + ETLayer on top of each pretrained EGCL layer from BASICES, as illustrated in Figure 2. We retain the use of GVP-based transition layers and introduce LayerNorm (Ba et al., 2016) at key interpolation steps to improve numerical stability. The resulting model comprises 6 layers and contains 3.3M parameters in total, with 2.3M trained during trajectory finetuning in the EGINTERPOLATOR framework.

C.3. Conditional Generation

We control the conditional generation by setting appropriate entries of a conditioning mask $\mathbf{m}$ to either 1 or 0. Let $\mathbf{m}[t, a]$ denote the conditioning status for frame t and atom a. We define the mask as follows:

- Forward simulation:

$$\mathbf{m}[t, :] = \begin{cases} 1 & t = 0 \\ 0 & \text{otherwise} \end{cases}$$

- Interpolation:

$$\mathbf{m}[t,:] = \begin{cases} 1 & t \in \{0, M\} \\ 0 & \text{otherwise} \end{cases},$$

where M is the index of the final frame.

In the unconditional setting, we default to $\mathbf{m}[:,:] = 0$. To incorporate this conditioning information, we use a condition state embedding added to the invariant node features, with the same hidden dimension as the main model. The conditioning mask is also used to restrict the denoising process and loss computation to frames where $\mathbf{m}[t',:] = 0$.

C.4. Kabsch Alignment

Inspired by (Xu et al., 2022), we propose to use trajectory-level Kabsch alignment to find the optimal rotation and translation between the noisy trajectory $\mathbf{x}_\tau^{[T]}$ and the input trajectory $\mathbf{x}_0^{[T]}$ at diffusion step τ. This corresponds to the following optimization problem:

$$\mathbf{R}^*, \mathbf{t}^* = \arg\min_{\mathbf{R},\mathbf{t}} \|\mathbf{R}\mathbf{x}_\tau^{[T]} + \mathbf{t} - \mathbf{x}_0^{[T]}\|_2. \tag{9}$$

In practice, this can be realized by extending the original Kabsch algorithm (Kabsch, 1976) on the set of points with the temporal dimension T combined into the number of points dimension N, that forms a point cloud with effective number of points $T \times N$. Afterwards, we re-compute the target noise $\bar{\boldsymbol{\epsilon}}$ based on the aligned $\bar{\mathbf{x}}_\tau^{[T]} = \mathbf{R}^*\mathbf{x}_\tau^{[T]} + \mathbf{t}^*$ and the clean data $\mathbf{x}_0^{[T]}$ by the forward diffusion process, and then match the output of EGINTERPOLATOR towards re-computed noise $\bar{\boldsymbol{\epsilon}}$ after alignment.

C.5. Baselines

Autoregressive Models. In the autoregressive baseline configuration, molecular dynamics trajectories are modeled autoregressively under the Markov assumption, whereby the model—either EGNN (Satorras et al., 2021b) or Equivariant Transformer (Thölke & Fabritiis, 2022)—learns the transition distribution $p(x_{t+1}|x_t)$. To ensure fair comparison, we maintain consistent timestep intervals and frame counts across all datasets during both training and inference phases, matching the parameters used in our proposed methods. For both architectures, we employ identical model configurations consisting of six stacked layers of EGCL or Equivariant Transformer blocks, respectively, to maintain experimental consistency.

GEOTDM. The training setup and embedding configurations for our implementation of GEOTDM are aligned with those used in our proposed framework. Following the architecture described in (Han et al., 2024), the model consists of 6 stacked layers of EGCL and ETLayer blocks, resulting in a total of 1.4M parameters.

D. Proofs

D.1. Proof of Theorem 4.1

For better readability we restate Theorem 4.1 below.

Theorem 4.1. *Suppose $\boldsymbol{\epsilon}_\theta$ perfectly models $p(\mathbf{x})$ and $\boldsymbol{\epsilon}'_\psi$ perfectly models $p(\mathbf{x}^{[T]})$, then the interpolation in Eq. 3, namely,*

$$\boldsymbol{\epsilon}'_\psi(\mathbf{x}_\tau^{[T]}, \tau) = \alpha\tilde{\mathbf{x}}_\tau^{[T]} + (1-\alpha)\boldsymbol{\epsilon}_\phi(\tilde{\mathbf{x}}_\tau^{[T]}, \tau), \quad \textit{s.t. } \tilde{\mathbf{x}}_\tau^{[T]} = [\boldsymbol{\epsilon}_\theta(\mathbf{x}_\tau^{(t)}, \tau)]_{t=0}^{T-1},$$

implicitly induces the distribution $\hat{p}(\mathbf{x}^{[T]}) \propto p(\mathbf{x}^{[T]})^\beta \tilde{p}(\mathbf{x}^{[T]})^{1-\beta}$ for $\boldsymbol{\epsilon}_\phi$, where $\beta = \frac{1}{1-\alpha}$.

Proof. Upon perfect optimization, we have the connection between the denoiser and the score of the underlying distribution (Song & Ermon, 2019; Song et al., 2021):

$$\boldsymbol{\epsilon}_\theta(\mathbf{x}_\tau^{(t)}, \tau) = -\sqrt{1-\bar{\alpha}_\tau}\nabla \log p(\mathbf{x}^{(t)}), \quad \forall 0 \le t \le T-1, 0 \le \tau \le \mathcal{T}, \tag{10}$$

and similarly,

$$\boldsymbol{\epsilon}'_\psi(\mathbf{x}_\tau^{[T]}, \tau) = -\sqrt{1-\bar{\alpha}_\tau}\nabla \log p(\mathbf{x}^{[T]}), \quad \forall 0 \le \tau \le \mathcal{T}. \tag{11}$$

By leveraging Eq 10 for all frames $0 \leq t \leq T-1$, we have

$$\tilde{\mathbf{x}}_\tau^{[T]} = [\boldsymbol{\epsilon}_\theta(\mathbf{x}_\tau^{(t)}, \tau)]_{t=0}^{T-1} = -\sqrt{1-\bar{\alpha}_\tau}\nabla \log \tilde{p}(\mathbf{x}^{[T]}), \tag{12}$$

where $\tilde{p}(\mathbf{x}^{[T]})$ is the joint of i.i.d. framewise distributions $p(\mathbf{x})$. Combining with the interpolation rule in Eq. 3, we have

$$\begin{aligned}
\boldsymbol{\epsilon}_\phi &= \frac{1}{1-\alpha}\boldsymbol{\epsilon}'_\psi - \frac{\alpha}{1-\alpha}\tilde{\mathbf{x}}_\tau^{[T]}, &(13)\\
&= (-\sqrt{1-\bar{\alpha}_\tau})\left(\frac{1}{1-\alpha}\nabla \log p(\mathbf{x}^{[T]}) - \frac{\alpha}{1-\alpha}\nabla \log \tilde{p}(\mathbf{x}^{[T]})\right), &(14)\\
&= (-\sqrt{1-\bar{\alpha}_\tau})\left(\beta\nabla \log p(\mathbf{x}^{[T]}) + (1-\beta)\nabla \log \tilde{p}(\mathbf{x}^{[T]})\right), &(15)
\end{aligned}$$

where $\beta = \frac{1}{1-\alpha}$. Now, consider the distribution $\hat{p}(\mathbf{x}^{[T]}) \propto p(\mathbf{x}^{[T]})^\beta \tilde{p}(\mathbf{x}^{[T]})^{1-\beta}$, we have

$$\nabla \log \hat{p}(\mathbf{x}^{[T]}) = \beta\nabla \log p(\mathbf{x}^{[T]}) + (1-\beta)\nabla \log \tilde{p}(\mathbf{x}^{[T]}). \tag{16}$$

Therefore, $\boldsymbol{\epsilon}_\phi = -\sqrt{1-\bar{\alpha}_\tau}\nabla \log \hat{p}(\mathbf{x}^{[T]})$. This verifies that the interpolation rule implicitly induces the distribution $\hat{p}(\mathbf{x}^{[T]})$ with $\boldsymbol{\epsilon}_\phi$ as its score network. Furthermore, the induction is unique, since for any distribution $q(\mathbf{x}^{[T]})$ satisfying $\boldsymbol{\epsilon}_\phi = -\sqrt{1-\bar{\alpha}_\tau}\nabla \log q(\mathbf{x}^{[T]})$, we have that $\nabla \log \hat{p}(\mathbf{x}^{[T]}) = \nabla \log q(\mathbf{x}^{[T]})$, which gives us $q(\mathbf{x}^{[T]}) = \hat{p}(\mathbf{x}^{[T]})$ due to the property of Stein score as demonstrated in (Hyvärinen & Dayan, 2005; Song & Ermon, 2019).

□

D.2. Proof of Equivariance

Theorem D.2. EGINTERPOLATOR *is SE(3)-equivariant. Namely,* $g \cdot f_{\text{EGI}}(\mathbf{x}^{[T]}) = f_{\text{EGI}}(g \cdot \mathbf{x}^{[T]})$*, for all* $g \in SE(3)$ *where* f_{EGI} *is the mapping defined per* EGINTERPOLATOR.

Proof.

$$\boldsymbol{\epsilon}'_\psi(\mathbf{x}_\tau^{[T]}, \tau) = \alpha\tilde{\mathbf{x}}_\tau^{[T]} + (1-\alpha)\boldsymbol{\epsilon}_\phi(\tilde{\mathbf{x}}_\tau^{[T]}, \tau), \quad \text{s.t. } \tilde{\mathbf{x}}_\tau^{[T]} = [\boldsymbol{\epsilon}_\theta(\mathbf{x}_\tau^{(t)}, \tau)]_{t=0}^{T-1},$$

It suffices to show that the temporal interpolator in Eq. 3 is SE(3)-equivariant, since the SE(3)-equivariance of the structure model $\boldsymbol{\epsilon}_\theta$ and $\boldsymbol{\phi}$ directly follows the original works of (Satorras et al., 2021b) and (Han et al., 2024), respectively. For any $g \coloneqq (\mathbf{R}, \mathbf{t}) \in \text{SE}(3)$, we have $[\boldsymbol{\epsilon}_\theta(\mathbf{R}\mathbf{x}_\tau^{(t)} + \mathbf{t}, \tau)]_{t=0}^{T-1} = \mathbf{R}[\boldsymbol{\epsilon}_\theta(\mathbf{x}_\tau^{(t)}, \tau)]_{t=0}^{T-1} + \mathbf{t} = \mathbf{R}\tilde{\mathbf{x}}_\tau^{[T]} + \mathbf{t}$. Therefore, we have

$$\begin{aligned}
\boldsymbol{\epsilon}'\psi(\mathbf{R}\mathbf{x}_\tau^{[T]} + \mathbf{t}, \tau) &= \alpha(\mathbf{R}\tilde{\mathbf{x}}_\tau^{[T]} + \mathbf{t}) + (1-\alpha)\boldsymbol{\epsilon}_\phi(\mathbf{R}\tilde{\mathbf{x}}_\tau^{[T]} + \mathbf{t}, \tau) &(17)\\
&= \alpha\mathbf{R}\tilde{\mathbf{x}}_\tau^{[T]} + (1-\alpha)\mathbf{R}\boldsymbol{\epsilon}_\phi(\tilde{\mathbf{x}}_\tau^{[T]}, \tau) + \alpha\mathbf{t} + (1-\alpha)\mathbf{t}, &(18)\\
&= \mathbf{R}\left(\alpha\tilde{\mathbf{x}}_\tau^{[T]} + (1-\alpha)\boldsymbol{\epsilon}_\phi(\tilde{\mathbf{x}}_\tau^{[T]}, \tau)\right) + \mathbf{t}, &(19)\\
&= \mathbf{R}\boldsymbol{\epsilon}'_\psi(\mathbf{x}_\tau^{[T]}, \tau) + \mathbf{t}, &(20)
\end{aligned}$$

which concludes the proof. □

E. Additional Results

E.1. Conformer Pretraining: QM9

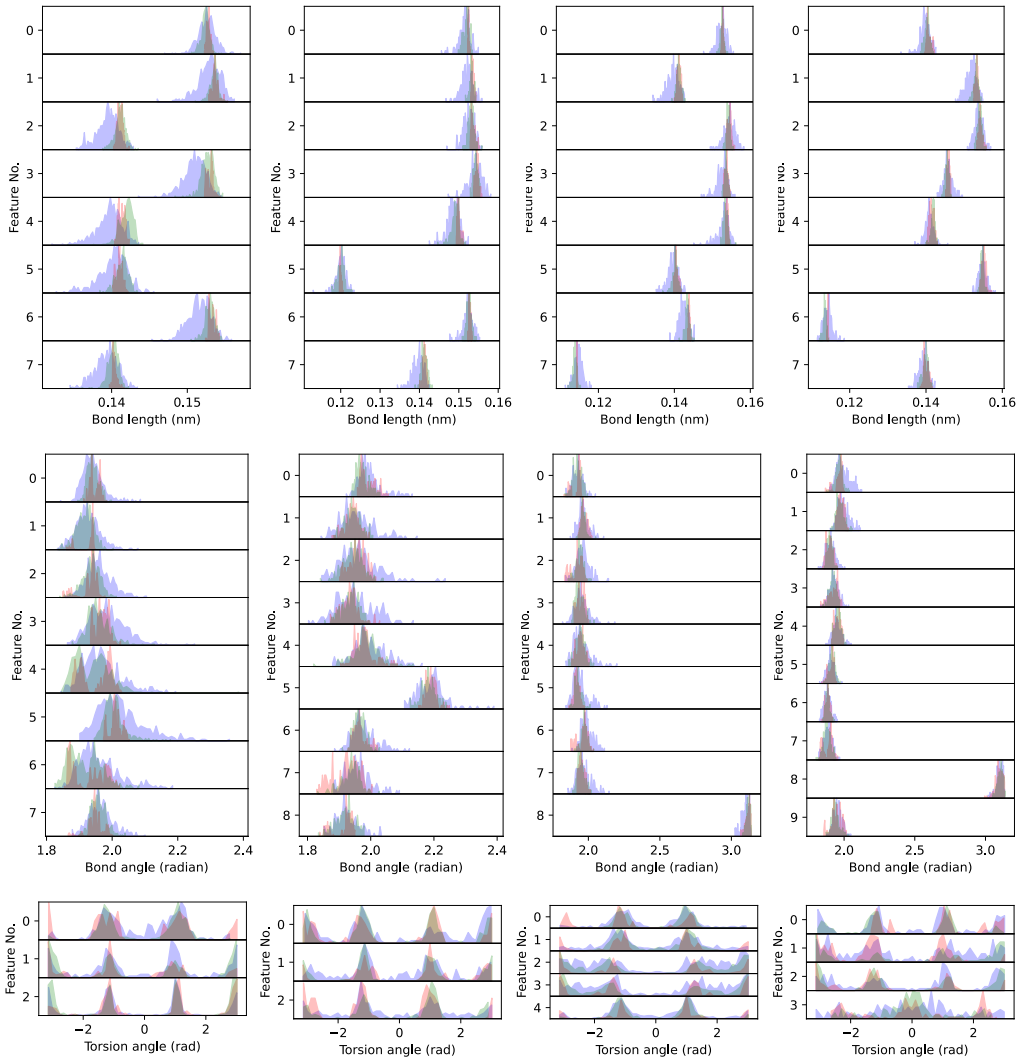

Figure 7. Distributions computed from reference conformers shown in red, Checkpoint 539 in green, and Checkpoint 99 in purple. We see that 539 aligns more closely with reference distributions across all collective variables and shows improved discretization of torsional states.

Above we show the additional plot associated with Section 5.1 and A.1. The plots above correspond to the following molecules (left to right):

`N#C[C@](O)(CO)CCO`, `C[C@@H](O)[C@@H](CO)CC#N`,
`C[C@@H](O)CCOCCO`, `CC[C@@H](CC=O)[C@@H](C)O`

E.2. Unconditional Generation: QM9

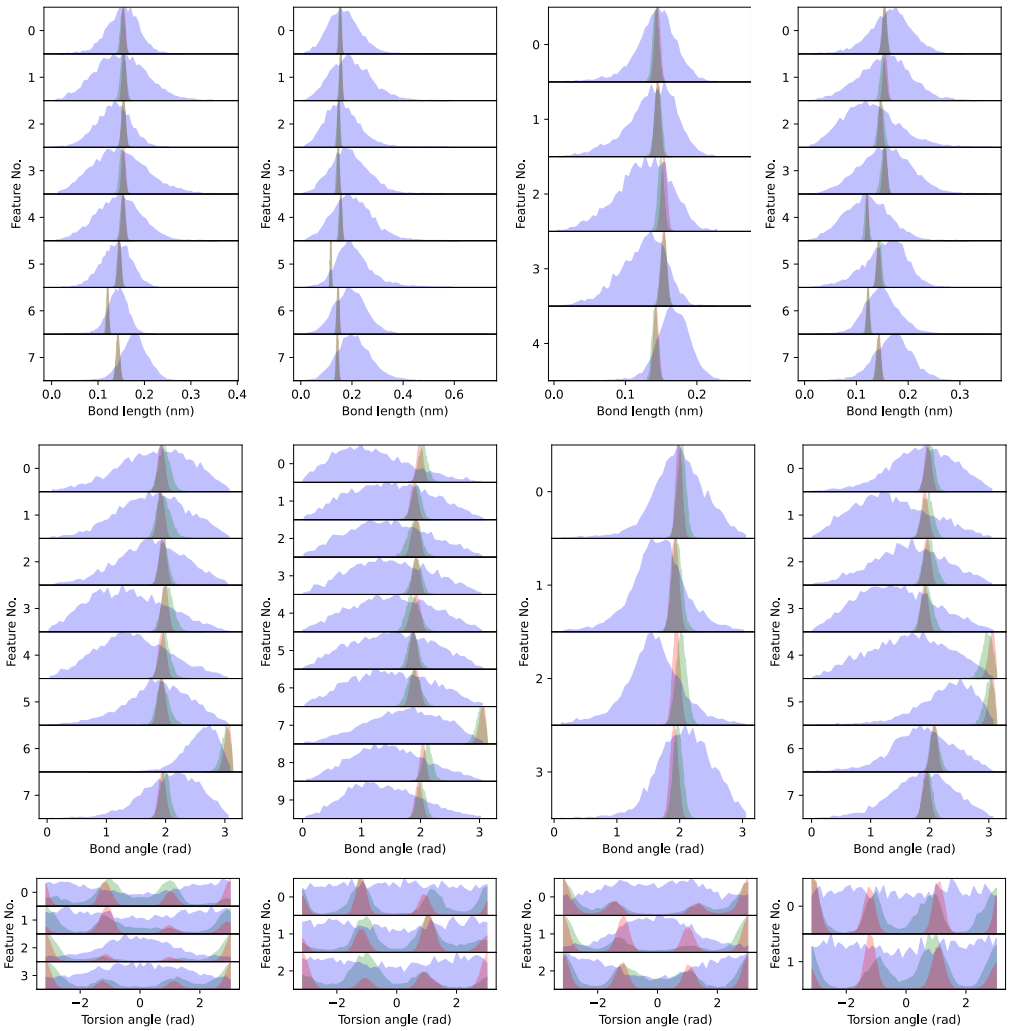

Figure 8. Distributions computed from reference QM9 trajectories (red), EGINTERPOLATOR (green), and GeoTDM (purple). Across all examples, our framework more closely matches the reference distributions across all collective variables and better captures torsional state discretizations than GeoTDM.

The figure above provides additional examples corresponding to the distributional analysis in Section 5.3. The molecule featured in the main paper in Figure 4A and 4B is:

`CC[C@H](C#CC=O)CO`

The plots above correspond to the following molecules (left to right):

`C#CCCC[C@@H](C)CO`, `CC[C@@](C#N)(CO)OC`,
`COCCCO`, `CC[C@H](C#CC=O)CO`

E.3. Forward Simulation: Drugs

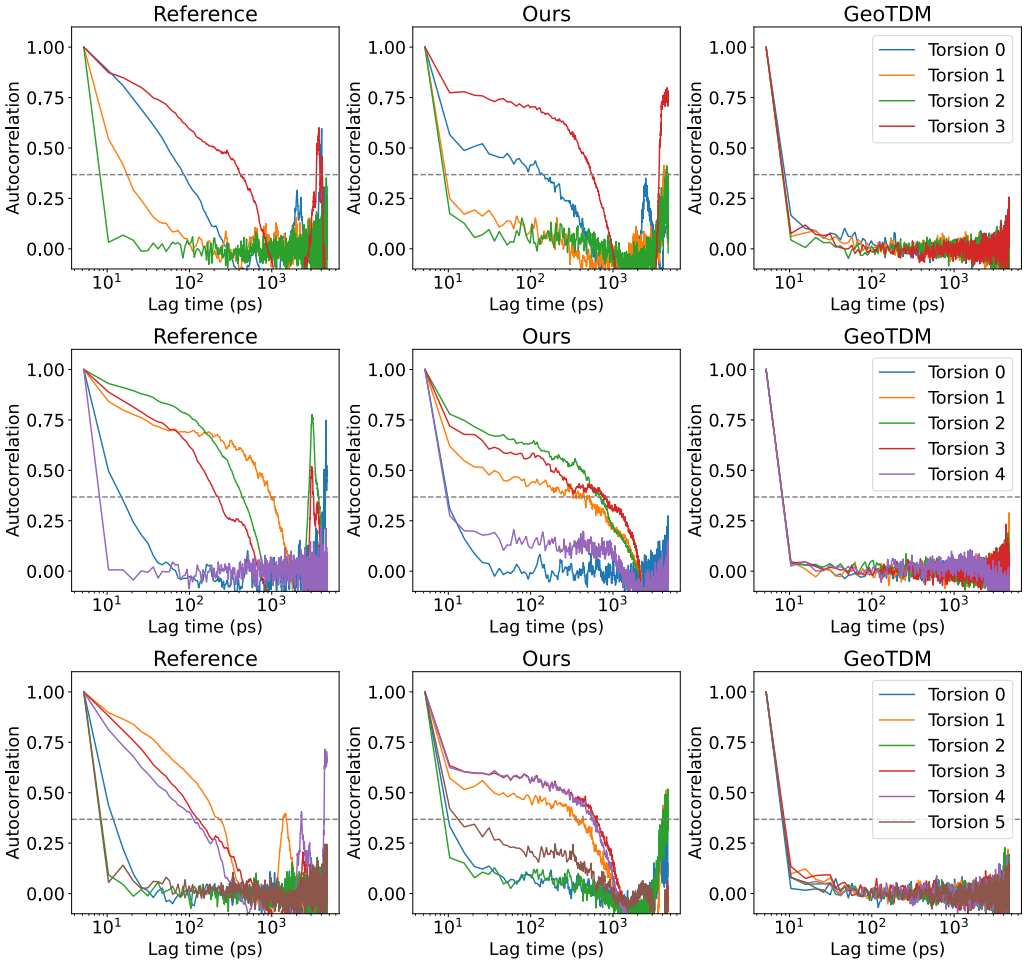

Figure 9. Autocorrelations of individual torsion angles for an example molecule, comparing reference trajectories with generations from EGINTERPOLATOR and GeoTDM. For the challenging task of capturing temporal de-correlation behavior, EGINTERPOLATOR closely follows the reference dynamics, whereas GeoTDM fails to model frame-to-frame correlations effectively.

The figure above provides additional examples corresponding to the dynamical analysis in Section 5.4. The molecule featured in the main paper in Figure 4E-G is:

`O=C(O)c1[nH]c2ccc(Cl)cc2c1CC(=O)N1CCN(c2ccccc2)CC1`

The plots above correspond to the following molecules (left to right):

`Cc1ccc(C)c(CN2C(=O)NC3(CCCCC3)C2=O)c1`,
`COc1ccc(NS(=O)(=O)c2ccc3c(c2)Cc2ccccc2-3)cn1`,
`COc1ccc(S(=O)(=O)Nc2c(C(=O)O)[nH]c3ccccc23)c(OC)c1`

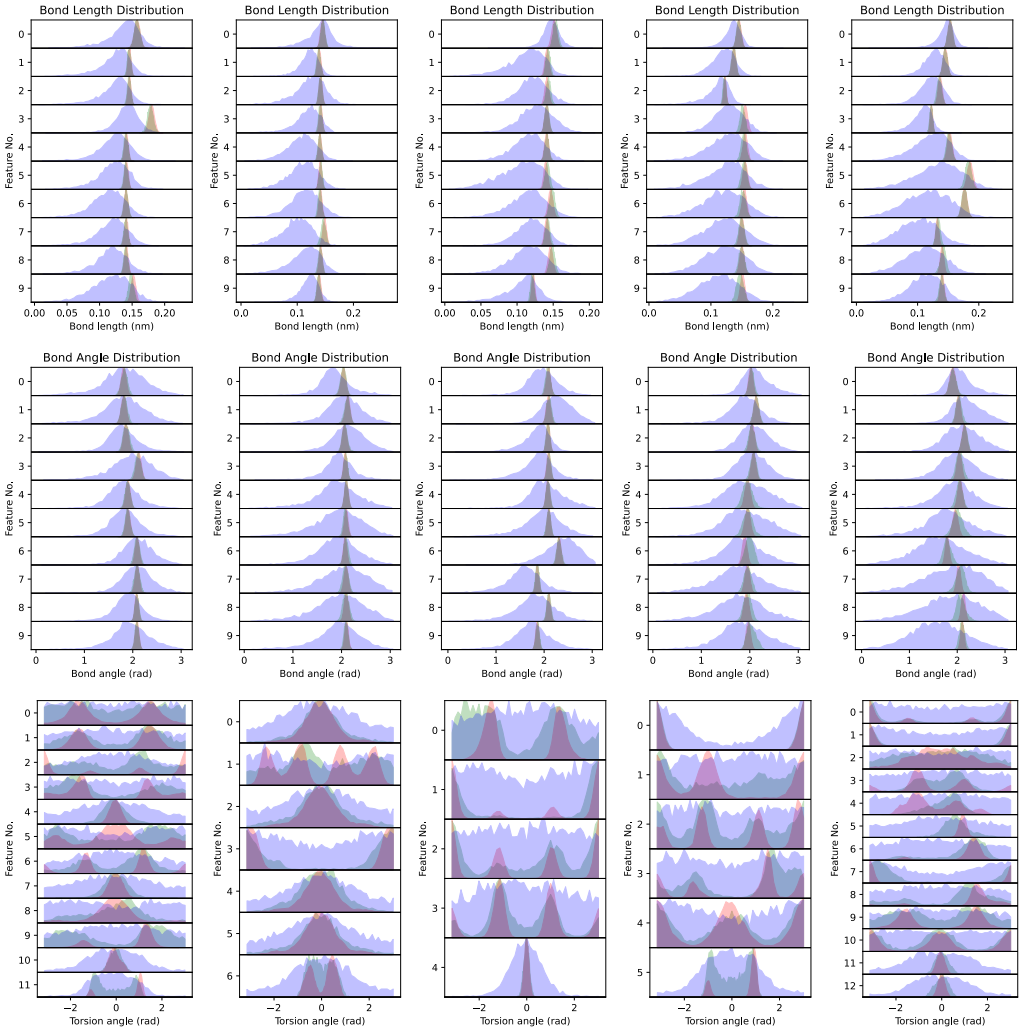

Figure 10. Distributions computed from reference Drugs trajectories (red), EGINTERPOLATOR (green), and GeoTDM (purple). Across all examples, our framework aligns closely with reference distributions across all collective variables and exhibits improved torsional state discretization compared to GeoTDM.

The figure above provides additional examples related to the distributional analysis in Section 5.4.

The plots above correspond to the following molecules (left to right):

`NS(=O)(=O)c1ccc(CCNC(=O)COC(=O)CN2C(=O)[C@H]3CCCC[C@H]3C2=O)cc1`,
`COc1ccc(C(=O)N2CCc3cc(OC)c(OC)cc3C2)cc1OC`,
`Cc1ccc2c(c1)C(=O)N(CCCCO)C2=O`,
`COC(=O)C1CCN(Cc2cc(=O)oc3cc(OC)ccc23)CC1`,
`CCOC(=O)CSC1=Nc2ccccc2C2=N[C@H](CC(=O)NCc3ccc(OC)cc3)C(=O)N12`

E.4. Interpolation: QM9

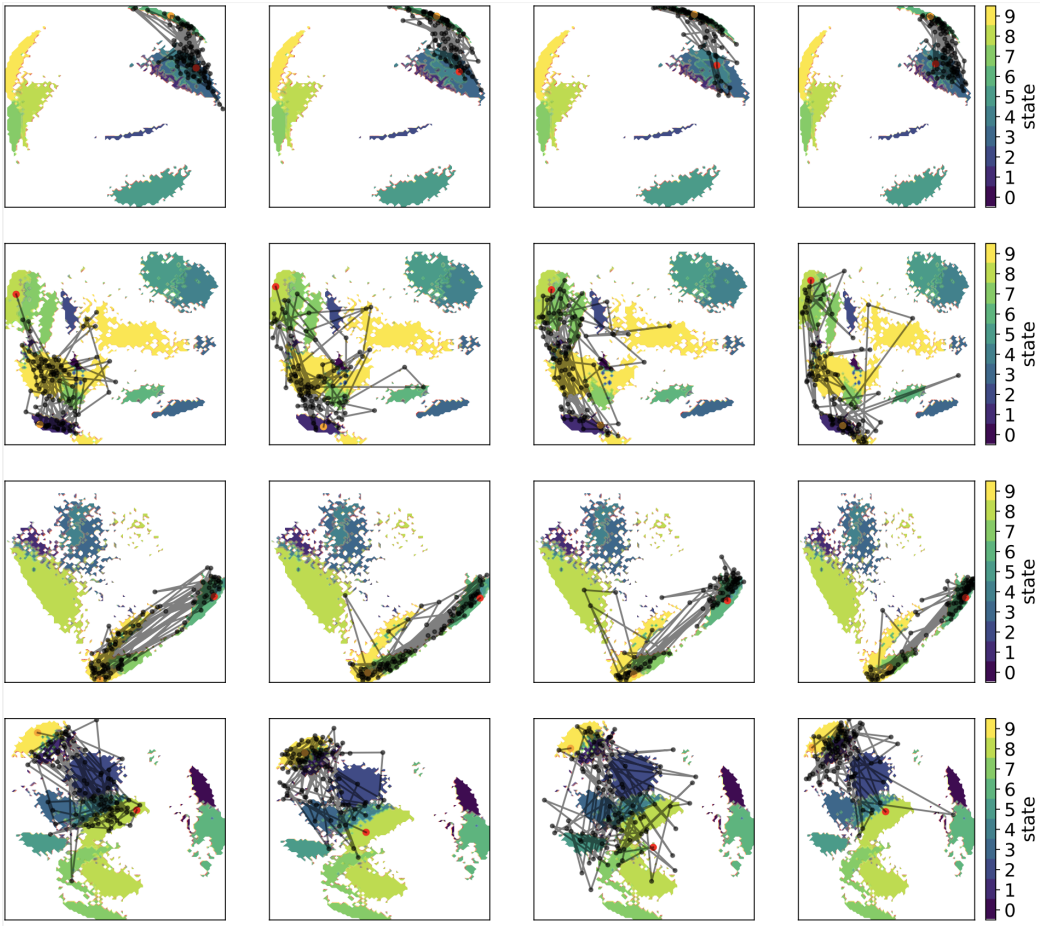

Figure 11. Generated QM9 interpolation trajectories from EGINTERPOLATOR, projected on the reference surface. The red point denotes the start frame, and the orange point denotes the end frame. The reference surface is colored by metastate assignment. Each row corresponds to a different molecule, and each column shows a generated interpolation. These examples illustrate the model's ability to generate efficient and meaningful transition paths.

The figure above provides additional examples related to the analysis in Section A.2.

The trajectories correspond to the following QM9 molecules (top to bottom):

`C#C[C@@](O)(CC)COC`, `N#CC[C@H](O)CCCO`,
`C[C@H](C=O)NCC=O`, `CCC[C@@H](O)CC#N`

E.5. Interpolation: Drugs

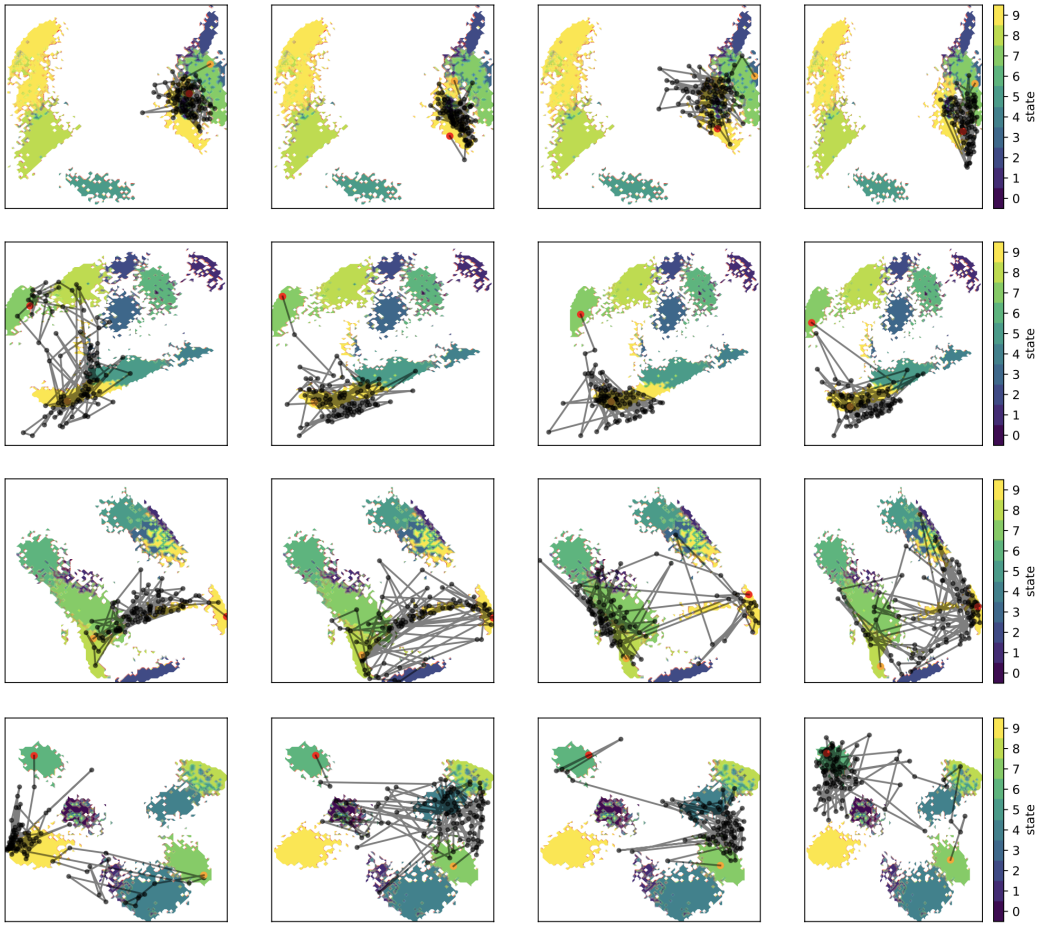

Figure 12. Generated Drug interpolation trajectories from EGINTERPOLATOR, projected onto the reference surface. The red point indicates the start frame, and the orange point indicates the end frame. The reference surface is colored by metastate assignment. Each row corresponds to a different molecule, and each column shows a generated interpolation. These examples highlight the model's ability to generate efficient and meaningful transition paths.

The figure above provides additional examples related to the analysis in Section 5.5. The molecule featured in the main paper in Figure 5B is:

`O=C(CCCSc1nc2ccccc2[nH]1)NCc1ccccc1F`

The trajectories above correspond to the following Drug molecules (top to bottom):

`COc1ccc(S(=O)(=O)Nc2c(C(=O)O)[nH]c3ccccc23)c(OC)c1`,
`Cn1c(C(=O)NCCN2CCOCC2)cc2c(=O)n(C)c3ccccc3c21`,
`O=C(c1ccc(Br)o1)N1CCN(c2ccccc2F)CC1`,
`CCOC(=O)c1c(C)[nH]c(C)c1C(=O)CSc1ncccn1`

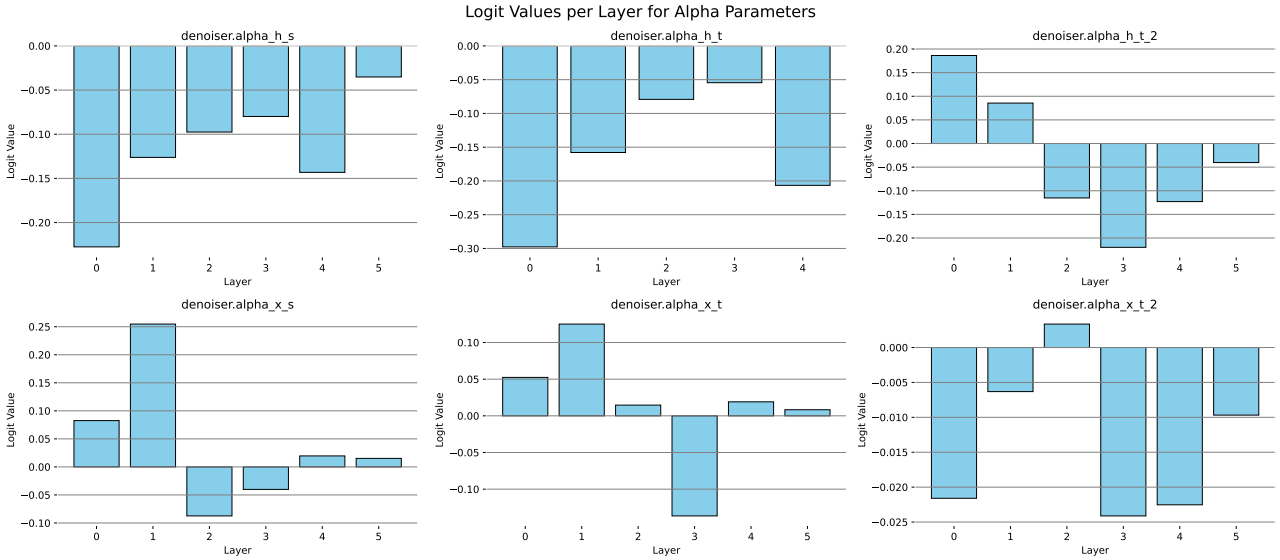

Figure 13. The logits of α for each spatial and temporal layer after convergence on QM9 unconditional generation task.

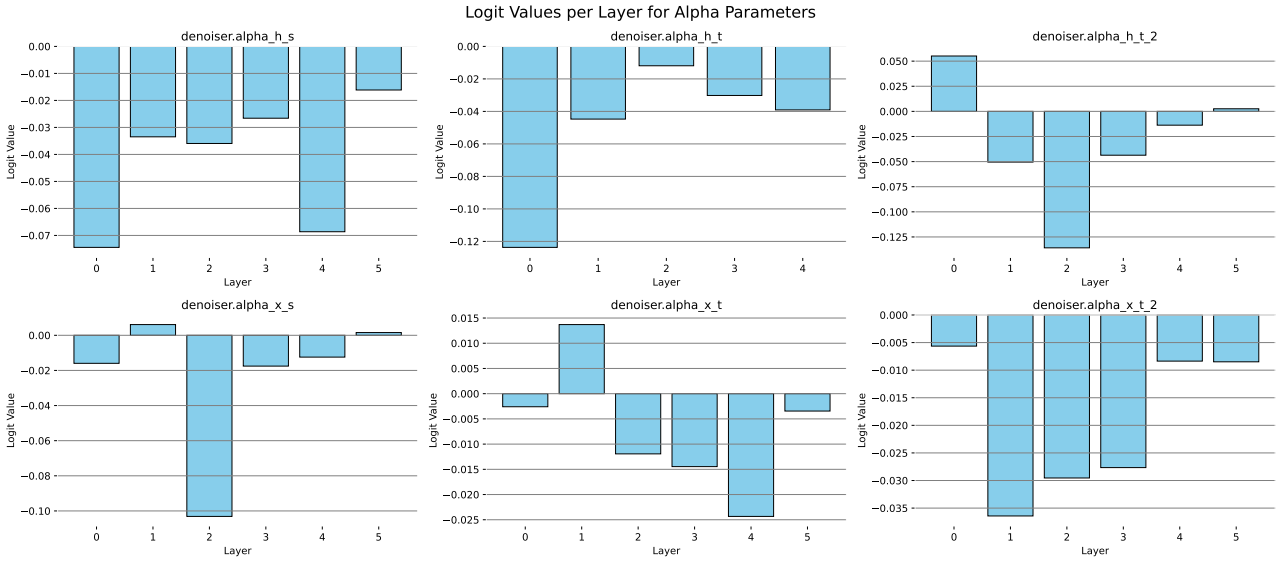

Figure 14. The logits of α for each spatial and temporal layer after convergence on QM9 interpolation task.

F. More Results on α

We present the plots of the logits of α after the training has converged on QM9 unconditional generation task (Fig. 13), QM9 interpolation task (Fig. 14), DRUGS forward simulation task (Fig. 15), and DRUGS interpolation task (Fig. 16), respectively. Interestingly, we observe that the trend of alpha is generally shared across different tasks on the same dataset, while they also exhibit divergent behaviors across different datasets. This indicates that α is able to capture the temporal consistency that is shared across tasks while being dataset specific.

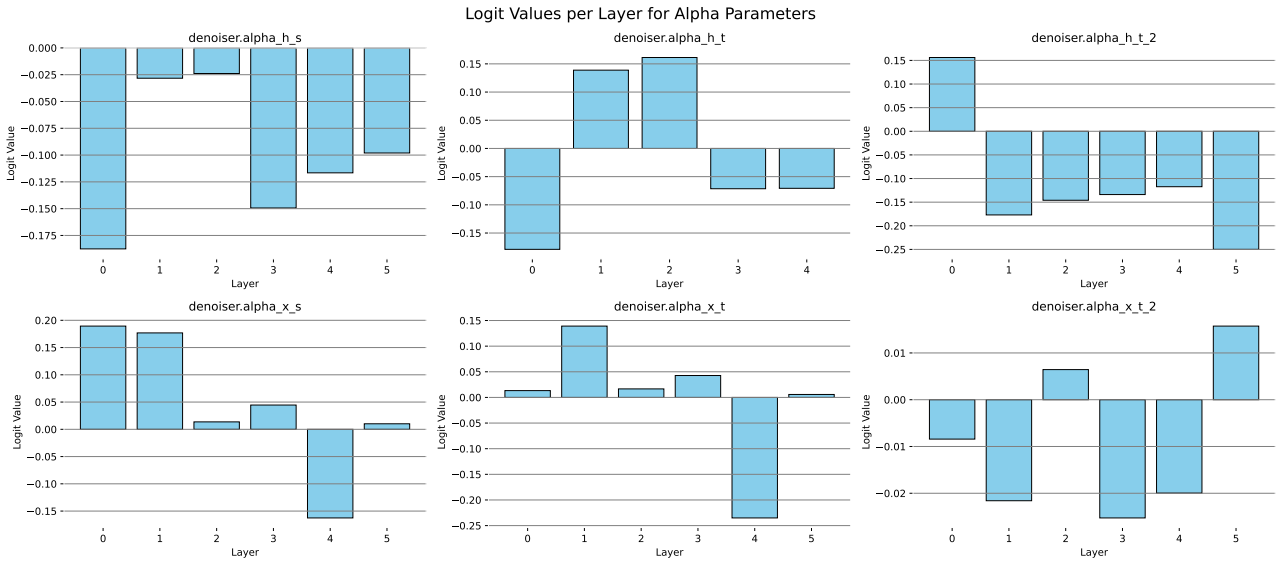

Figure 15. The logits of α for each spatial and temporal layer after convergence on DRUGS forward simulation task.

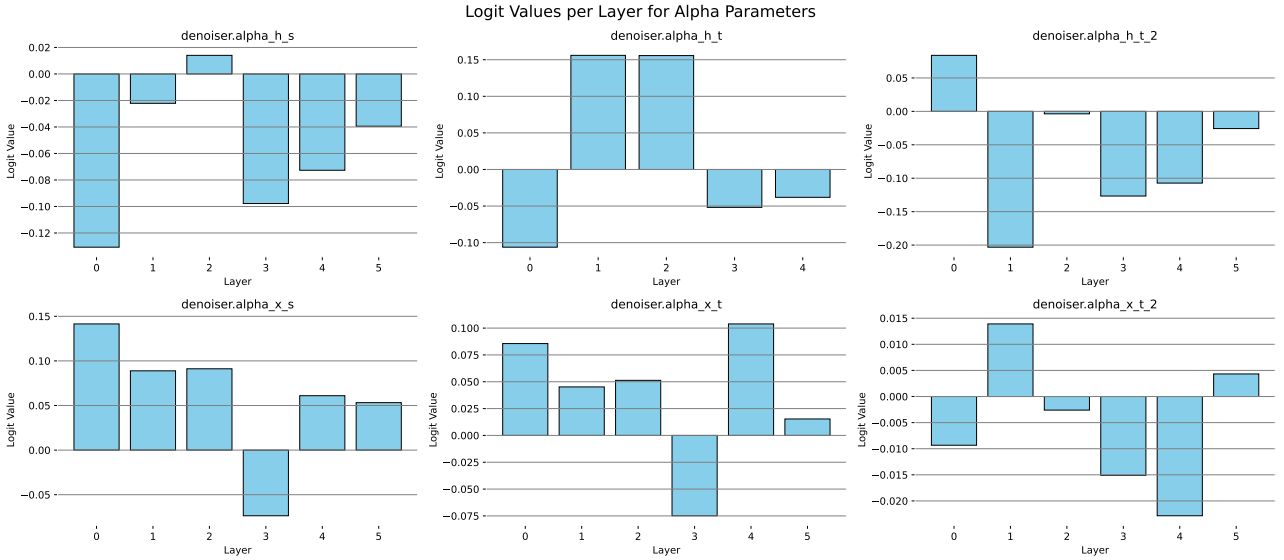

Figure 16. The logits of α for each spatial and temporal layer after convergence on DRUGS interpolation task.

G. Statements and Discussions

G.1. Limitations Cont. and Future Opportunities

Our results demonstrate that structural pretraining significantly enhances all-atom diffusion models for simulating small molecule molecular dynamics trajectories. Nonetheless, our work has limitations that highlight directions for future research. As noted in Section 6, machine learning methods still lag behind ground-truth MD simulations in terms of physical accuracy. Future work may therefore explore improved learning objectives, molecular parameterizations, and the incorporation of physics-based regularization to help bridge this gap.

While our focus is on the challenging domain of organic small molecules, molecular dynamics is broadly applicable to larger N-body systems, such as peptides and protein–ligand complexes. Future work may extend our framework to these more complex settings, leveraging structural pretraining to enable generative modeling of larger biomolecular simulations.

Additionally, although our approach effectively reproduces distributions and dynamics consistent with classical mechanics, it remains subject to the inherent biases of molecular dynamics simulations. Future research may explore aligning both conformer and trajectory generation more closely with Boltzmann-distributed energy landscapes to improve thermodynamic fidelity.

G.2. Ethics and Impacts Statement

This work develops generative models for molecular dynamics to advance efficient, accurate simulation in chemistry and biology. While such models can accelerate scientific discovery, they also raise concerns around AI safety and dual-use risks, particularly in the design of harmful chemical or biological agents.

Our goal is to support beneficial applications in drug discovery, materials science, and molecular understanding through data-efficient and physically grounded modeling. All models are trained on publicly available, non-sensitive data and are released under open licenses to promote transparency and responsible use. We encourage continued dialogue on the safe development and deployment of generative AI in the physical and natural sciences.