GeoAda: Efficiently Finetune Geometric Diffusion Models with Equivariant Adapters

Wanjia Zhao,* **Jiaqi Han,*** **Siyi Gu, Mingjian Jiang, James Zou, Stefano Ermon**
Department of Computer Science
Stanford University

Abstract

Geometric diffusion models have shown remarkable success in molecular dynamics and structure generation. However, efficiently fine-tuning them for downstream tasks with varying geometric controls remains underexplored. In this work, we propose an SE(3)-equivariant adapter framework (GeoAda) that enables flexible and parameter-efficient fine-tuning for controlled generative tasks without modifying the original model architecture. GeoAda introduces a structured adapter design: control signals are first encoded through coupling operators, then processed by a trainable copy of selected pretrained model layers, and finally projected back via decoupling operators followed by an equivariant zero-initialized convolution. By fine-tuning only these lightweight adapter modules, GeoAda preserves the model's geometric consistency while mitigating overfitting and catastrophic forgetting. We theoretically prove that the proposed adapters maintain SE(3)-equivariance, ensuring that the geometric inductive biases of the pretrained diffusion model remain intact during adaptation. We demonstrate the wide applicability of GeoAda across diverse geometric control types, including frame control, global control, subgraph control, and a broad range of application domains such as particle dynamics, molecular dynamics, human motion prediction, and molecule generation. Empirical results show that GeoAda achieves state-of-the-art fine-tuning performance while preserving original task accuracy, whereas other baselines experience significant performance degradation due to overfitting and catastrophic forgetting.

1 Introduction

Diffusion models have emerged as powerful generative frameworks across a wide range of domains, including image synthesis [44, 38], robotics [36, 27], and molecular generation [23, 41, 39, 42]. In particular, geometric diffusion models [11] which incorporate spatial and symmetry-aware inductive biases have shown strong empirical performance in tasks such as particle dynamic prediction [15, 17, 28], molecular generation [3, 14] and protein-ligand binding structure prediction [8]. By modeling data in an equivariant network, these models are able to capture complex geometric relationships essential for physical and chemical systems.

However, despite their strong task-specific performance, existing geometric diffusion models lack the ability to generalize across tasks. In particular, it remains unclear how a model pretrained on one geometric generation task can be effectively adapted to a new task involving additional or different control signals. This limitation is especially pronounced in real-world molecular applications, where the available data across tasks are often highly imbalanced, and collecting labeled pretraining data for every new condition is costly and time-consuming. Without a mechanism for transfer, models must be retrained from scratch for each new task, which is inefficient and often leads to overfitting or loss of previously learned capabilities.

*Equal contribution. Correspondence to wanjiazh@cs.stanford.edu. Code is available here.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

 https://doi.org/10.52202/085713-3172

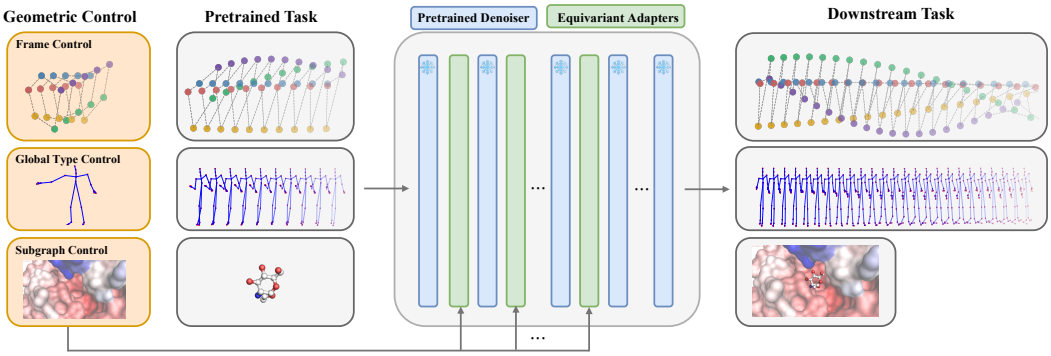

Figure 1: Overall framework of GeoAda. The model integrates diverse control signals, including frame, global type, and subgraph controls through lightweight equivariant adapters inserted into the frozen pretrained denoiser.

To this end, we propose a general and efficient framework(GeoAda) that enables the transfer of geometric diffusion models across diverse downstream tasks with minimal computational overhead. Inspired by the success of ControlNet [44] in conditional image generation, we introduce an equivariant adapter module that augments a pretrained geometric diffusion model with task-specific control capability. The Equivariant Adapter comprises two key components: 1) The equivariant adapter block that operates through a structured sequence—where control signals are encoded via coupling operators, processed by a trainable copy of selected pretrained model layers, and then decoded via decoupling operators. 2) Equivariant zero convolution, which acts as a safeguard for the original score by zeroing out the conditional contribution at initialization without blocking gradient updates. This design preserves the model's SE(3)-equivariance and allows modular, task-specific adaptation without altering the original model architecture. In addition to being lightweight and flexible, the adapter is parameter-efficient and implicitly regularized, thereby mitigating overfitting and preserving the performance of the pretrained model.

In summary, we make the following contributions: **1.** We propose an equivariant adapter framework (GeoAda) for geometric diffusion models that enables efficient task adaptation with minimal overhead. The adapter modules are lightweight and operate as plug-and-play components, allowing flexible conditioning on new control signals without architectural modifications to the pretrained model. **2.** GeoAda is parameter-efficient, introducing minimal overhead for downstream tasks compared to full fine-tuning, which updates the entire model and incurs substantial memory and computational costs. **3.** By freezing the pretrained model and introducing trainable adapters, GeoAda imposes implicit regularization, helping to mitigates overfitting and avoids catastrophic forgetting, thereby preserving performance on the pretraining task. **4.** We carefully design the adapter architecture to be SE(3)-equivariant, ensuring that the adapted model retains the geometric inductive bias and the theoretical benefits of equivariant diffusion models, including SE(3)-invariant marginal distributions during generation. **5.**We evaluate GeoAda across diverse geometric control types, including Frame Fontrol, Global Type Control, Subgraph Control, and a wide range of application domains, such as particle dynamics, molecular dynamics, human motion prediction, and molecule generation. GeoAda consistently matches or outperforms full fine-tuning baselines on downstream task, while avoiding performance degradation on the original pretrain task—a common failure mode of naive tuning and prompt-base approaches.

2 Related Work

Geometric diffusion models. Recent diffusion models have been extended to 3D geometric data, with SE(3) equivariance enabling physically consistent generation for tasks like molecular design and trajectory modeling. One of the earliest efforts, EDM [14] introduced an SE(3)-equivariant framework for 3D molecule generation that significantly improved sample quality. GeoDiff [42] pioneered this by learning stable molecular conformations through SE(3)-invariant diffusion, while GeoLDM [41, 39] advanced scalability and controllability via structured latent spaces. GCDM [23] advanced large molecule generation by incorporating geometry-complete local frames and chirality-sensitive features into SE(3)-equivariant networks. TargetDiff [9] further extended these models to structure-based

drug design by generating molecules conditioned on protein targets through an SE(3)-equivariant processor. Beyond molecular applications, diffusion augmented with geometric inductive bias has been explored in other domains such as 3D shape and scene generation [2] and robotics [36, 27]. Beyond static geometric modeling, GeoTDM [11] and EquiJump [5] address dynamic 3D systems by introducing temporal attention mechanisms. However, existing geometric diffusion models lack cross-task generalization. Our framework enables efficient adaptation to new controls.

Finetuning for (geometric) graphs. Finetuning for geometric GNNs generally falls into two categories: prompt-based and adapter-based methods. Pioneering prompt-based approaches [34, 20] introduce virtual class-prototype nodes with learnable links for edge prediction pre-trained models, but lack generalizability to alternative pre-training strategies. Meanwhile, works like GPF [6] explore universal prompt-based tuning by adding shared learnable vectors to all node features in the graph. Adapter-based methods, exemplified by AdapterGNN [18], insert lightweight modules into GNN layers, achieving parameter-efficient adaptation across diverse graph domains.

Finetuning diffusion models. Recent research has proposed various strategies for fine-tuning diffusion models with improved efficiency, control [44], and alignment [37]. ELEGANT [35] formulates fine-tuning as an entropy-regularized control problem, directly optimizing entropy-enhanced rewards with neural SDEs. ControlNet [44] improves controllability by adding lightweight trainable branches to frozen diffusion backbones. Prompt Diffusion [38] enables training-free in-context learning for image-to-image tasks via example-based conditioning. However, fine-tuning diffusion models in geometric domains (e.g., particles, molecules) remains underexplored. GeoAda addresses this gap by enabling efficient and effective adaptation of diffusion models to geometric tasks.

3 Preliminaries

Geometric graphs and trajectories. We represent a *geometric graph* as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges. In particular, each node i is equipped with certain node feature $\mathbf{h}_i \in \mathbb{R}^H$ representing its type or physical property, and the Euclidean coordinate $\mathbf{x}_i \in \mathbb{R}^3$ representing its spatial position. An edge exists between node i and j if they bear certain connectivity through, *e.g.*, chemical bonds, or spatial proximity with a distance smaller than a cutoff. A *trajectory* is a generalization of geometric graph in the dynamical setting where the coordinates $\mathbf{x}_i^{[T]} \in \mathbb{R}^{3\times T}$ are augmented with an additional temporal dimension, where T is the number of frames.

Geometric diffusion models. Geometric diffusion models are a family of generative models for capturing the distribution of geometric graphs and/or trajectories. Given an input data point $\mathcal{G}_0$, they feature a forward noising process that gradually perturbs the clean data with a transition $q(\mathcal{G}_\tau|\mathcal{G}_0)$ where $\mathcal{G}_\mathcal{T}$ converges to a tractable prior. A neural network $\boldsymbol{\epsilon}_\theta(\mathcal{G}_\tau, \tau)$ (*a.k.a.* the denoiser) is learned to approximate the Stein score [33] through denoising score matching [31, 32, 13], which will be leveraged to derive the transition kernel $p_\theta(\mathcal{G}_{\tau-1}|\mathcal{G}_\tau)$ in the reverse process at sampling time. Notably, a core distinction of *geometric* diffusion models from others is that they enforce an SE(3)-invariant marginal, *i.e.*,

$$p_\theta(\mathcal{G}_0) = p_\theta(g \cdot \mathcal{G}_0), \qquad \forall g \in \mathrm{SE}(3), \tag{1}$$

by parameterizing the denoiser $\boldsymbol{\epsilon}_\theta$ with an SE(3)-equivariant architecture [14, 42], *i.e.*,

$$\boldsymbol{\epsilon}_\theta(g \cdot \mathcal{G}_\tau, \tau) = g \cdot \boldsymbol{\epsilon}_\theta(\mathcal{G}_\tau, \tau), \tag{2}$$

where SE(3) is the Special Euclidean group consisting of all rotations and translations in 3D.

4 Method

In this section, we detail our approach, equivariant adapter for geometric diffusion models. We first specify three types of controls in § 4.1 that are ubiquitously enforced to geometric diffusion models in various downstream tasks. In § 4.2, we propose an architecture-agnostic and principled recipe for encoding such controls that seamlessly enables finetuning on the pretrained denoiser. In § 4.3, we present our design of GeoAda, a plug-in-and-play adapter module tuned for each downstream task that unlocks transferability.

 https://doi.org/10.52202/085713-3172

4.1 Geometric Controls for Geometric Diffusion Models

In this work, we aim to transfer the generation capability of pretrained geometric diffusion models to downstream tasks where additional *geometric controls* present. Specifically, we are concerned with three different types of geometric controls, namely global type control $\mathbb{C}_{\mathrm{G}}$, subgraph control $\mathbb{C}_{\mathrm{S}}$, and frame control $\mathbb{C}_{\mathrm{F}}$, as detailed below.

Global type control. Each global type control $\tilde{\mathbf{c}} \in \mathbb{C}_{\mathrm{G}}$ is a vector in $\mathbb{R}^K$ describing certain global signal enforced on the geometric graph, such as an encoding of the class label, some quantum chemical property of the molecule [14], or even the embedding of some text prompt [22].

Subgraph control. Each subgraph control $\tilde{\mathcal{G}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}}) \in \mathbb{C}_{\mathrm{S}}$ is represented as a geometric graph with the set of nodes $\tilde{\mathcal{V}}$ and edges $\tilde{\mathcal{E}}$. Subgraph control widely exists in scenarios where generating a geometric graph conditioned on another fixed subgraph is of interest. For example, in the task of pocket-conditioned ligand generation [8], the protein pocket is viewed as the fixed subgraph $\tilde{\mathcal{G}}$ while a geometric diffusion model is learned to generate the ligand, conditioned on $\tilde{\mathcal{G}}$.

Frame control. Each frame control in $\mathbb{C}_{\mathrm{F}}$ takes the form of a sequence of additional $\tilde{T}$ frames, namely $\tilde{\mathbf{x}}_i^{[\tilde{T}]} \in \mathbb{R}^{3\times\tilde{T}}$ for each node i. Frame controls are enforced in cases when, *e.g.*, a trajectory has been partially observed and the model is expected to generate the future or missing frames conditioned on the observed frames.

4.2 Encoding Geometric Controls

In this subsection, we propose a simple yet effective approach for incorporating the geometric controls into the denoiser *without* modifying its architecture. Such feature is critical since it enables us to initialize the denoiser fully with the pretrained checkpoint when performing finetuning on downstream tasks, thus significantly alleviating optimization overheads and potential inconsistencies in the parameter space. More importantly, our design is also guaranteed to preserve the equivariance of the denoiser, a fundamental principle that leads to the success of geometric diffusion models.

In form, given the denoiser $\boldsymbol{\epsilon}_\theta(\mathcal{G}_\tau, \tau)$, we seek to devise $\boldsymbol{\epsilon}_\theta(\mathcal{G}_\tau, \tau, \mathcal{C})$ where $\mathcal{C} \in \mathbb{C}_{\mathrm{G}} \cup \mathbb{C}_{\mathrm{S}} \cup \mathbb{C}_{\mathrm{F}}$ is any of the control we specified in § 4.1. Our core observation lies in that each type of the control can be encoded through certain coupling operator $\mathbf{f}(\mathcal{G}_\tau, \mathcal{C})$ of the input noised graph $\mathcal{G}_\tau$ and control $\mathcal{C}$, and a corresponding decoupling operator $\mathbf{g}$ that extracts the scores on the nodes and frames in $\mathcal{G}_\tau$ from the output of $\boldsymbol{\epsilon}_\theta$. We introduce our design of $\mathbf{f}$ and $\mathbf{g}$ with respect to different controls as follows.

Global type control. For global type control $\mathcal{C}_{\mathrm{G}} \coloneqq \tilde{\mathbf{c}} \in \mathbb{R}^K$, we design $\mathbf{f}$ as a node-wise addition of the input node feature and a linear transformation of the control $\tilde{\mathbf{c}}$, *i.e.*, $\mathcal{V}', \mathcal{E}' = \mathbf{f}(\mathcal{V}, \mathcal{E}, \mathcal{C})$, where

$$\mathcal{V}' = (\{\mathbf{x}_i\}, \{\mathbf{h}_i + \sigma(\tilde{\mathbf{c}})\}), \qquad \mathcal{E}' = \mathcal{E}, \tag{3}$$

where $\sigma : \mathbb{R}^K \mapsto \mathbb{R}^H$ is an MLP that lifts the control signal to the node feature space. We use identity function as the decoupling operator $\mathbf{g}$.

Subgraph control. For subgraph control $\mathcal{C}_{\mathrm{S}} \coloneqq \tilde{\mathcal{G}}$, the coupling operator $\mathbf{f}$ is realized by computing the supergraph of the input $\mathcal{G}$ and the control $\tilde{\mathcal{G}}$, *i.e.*, $\mathcal{V}', \mathcal{E}' = \mathbf{f}(\mathcal{V}, \mathcal{E}, \mathcal{C})$, where

$$\mathcal{V}' = \mathcal{V} \cup \tilde{\mathcal{V}}, \qquad \mathcal{E}' = \mathcal{E} \cup \tilde{\mathcal{E}}. \tag{4}$$

The decoupling operator $\mathbf{g}$ is implemented by extracting the features of subgraph that corresponds to the nodes in the input graph $\mathcal{G}$ from the output of $\boldsymbol{\epsilon}_\theta$.

Frame control. For frame control $\mathcal{C}_{\mathrm{F}} \coloneqq \{\tilde{\mathbf{x}}_i^{[\tilde{T}]}\}$, we implement $\mathbf{f}$ as a concatenation of the input frames and the frame control, *i.e.*, $\mathcal{V}', \mathcal{E}' = \mathbf{f}(\mathcal{V}, \mathcal{E}, \mathcal{C})$, where

$$\mathcal{V}' = \left(\{\mathrm{concat}(\mathbf{x}_i^{[T]}, \tilde{\mathbf{x}}_i^{[\tilde{T}]})\}, \{\mathbf{h}_i\}\right), \qquad \mathcal{E}' = \mathcal{E}, \tag{5}$$

and $\mathbf{g}$ performs the reverse operation by discarding the frames corresponding to $[\tilde{T}]$ from the output of $\boldsymbol{\epsilon}_\theta$.

Proposition 4.1 (Equivariance of control encoding)**.** *If the denoiser $\boldsymbol{\epsilon}_\theta$ is SE(3)-equivariant, the composition $\mathbf{g} \circ \boldsymbol{\epsilon}_\theta \circ \mathbf{f}$ is also SE(3)-equivariant, for all controls $\mathcal{C} \in \mathbb{C}$.*

4.3 Equivariant Adapters

With the control encoding in § 4.2, a straightforward approach to leverage a pretrained diffusion model on downstream tasks is to perform supervised finetuning (SFT). However, SFT usually induces suboptimal empirical performance, since **1.** SFT is parameter-inefficient since each gradient update is conducted on all parameters of the pretrained model; **2.** the full-parameter finetuning is prone to overfitting with limited amount of finetuning data; and **3.** the model finetuned after SFT loses performance guarantee on the original task, a phenomenon widely acknowledged as catastrophic forgetting.

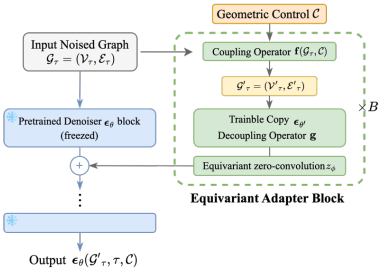

Figure 2: Overall framework of GeoAda. A control signal $\mathcal{C}$ is injected into the noised graph $\mathcal{G}_\tau$ and processed by an equivariant adapter block. The adapter output is added to the frozen denoiser and repeated B times to produce the final output.

To address these challenges, we draw inspiration from the successful application of adapters on image diffusion models, *e.g.*, ControlNet [45], to devise a diffusion adapter for geometric diffusion models. Our approach, dubbed equivariant adapter, is a lightweight tunable module plugged-in on top of the pretrained model, which is optimized for each downstream task.

The equivariant adapter block. In detail, each equivariant adapter block is responsible for processing the control signal and fusing it into the score produced by the pretrained model, whose parameters are always freezed at finetuning stage. Each adapter block consists of, in a sequential manner, the coupling operator $\mathbf{f}$, a *trainable copy* of the corresponding layers in pretrained model $\boldsymbol{\epsilon}_{\theta'}$, the decoupling operator $\mathbf{g}$, followed by an equivariant *zero-convolution* layer.

Specifically, the composition $\mathbf{g} \circ \boldsymbol{\epsilon}_{\theta'} \circ \mathbf{f}$, as depicted in § 4.2, functions altogether as a conditional score network $\boldsymbol{\epsilon}_{\theta'}(\mathcal{G}_\tau, \tau, \mathcal{C})$ that captures the bias of the control signal on the original score $\boldsymbol{\epsilon}_\theta(\mathcal{G}_\tau, \tau)$ while ensuring the SE(3)-equivariance of the conditional score. Moreover, $\boldsymbol{\epsilon}_{\theta'}$ can be initialized as a subset of the layers in the pretrained model $\boldsymbol{\epsilon}_\theta$, thus reducing the total number of tunable parameters compared with SFT. While the selection strategy can be arbitrary, empirically we have found that selecting the first layer for every K consecutive layers from the pretrained model performs more favorably compared with naive choices such as the initial or last several layers, under the same parameter budget (*c.f.*, § 5.4).

Equivariant zero-convolution. While the equivariant adapter block offers an parameter-efficient way of modeling the conditional score, its non-zero initialization introduces additional noise when it is added to the original score $\boldsymbol{\epsilon}_\theta$, leading to instability at the beginning of the finetuning stage. To alleviate such issue, we borrow insight from the *zero-convolution* module proposed in [44] that acts as a safeguard of the original score by zeroing out the conditional score at initialization without blocking the gradient update.

For any $(\{\mathbf{x}_i\}, \{\mathbf{h}_i\})$, equivariant zero-convolution is given by

$$z_\phi(\{\mathbf{x}_i\}, \{\mathbf{h}_i\}) = (\{\phi_\mathbf{x} \cdot (\mathbf{x}_i - \bar{\mathbf{x}})\}, \{\boldsymbol{\phi}_\mathbf{h} \odot \mathbf{h}_i\}), \tag{6}$$

where $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^N \mathbf{x}_i$ is the center-of-mass of the input graph, and $\phi_\mathbf{x} \in \mathbb{R}, \boldsymbol{\phi}_\mathbf{h} \in \mathbb{R}^H$ are learnable parameters initialized all as zero. By such design, we guarantee that each equivariant adapter block, when equipped with equivariant zero-convolution, yields a rotation-equivariant and translation-invariant output, hence the SE(3)-equivariance of the conditional score after adding the output to the original score. Furthermore, the output of equivariant adapter block will always be zero at initialization, which does not affect the original score, thus enabling smooth and noiseless optimization when tuning the adapter.

4.4 Overall Framework

The overall framework of our adapter is depicted in Fig. 2. In general, our adapter is comprised of B equivariant adapter blocks, where each block is a sequential stack of the coupling operator $\mathbf{f}$, the trainable copy of one layer of the denoiser, the decoupling operator $\mathbf{g}$, and a zero-convolution module. At finetuning stage, all parameters in the original denoiser are freezed while the trainable copies and coefficients in zero-convolution are updated through gradient coming from minimizing the denoising loss

$$\mathcal{L}_{\text{finetune}}(\theta', \phi) = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}),(\mathcal{G},\mathcal{C})\sim\mathbb{D},\tau\in\text{Unif}(0,T)} \|\boldsymbol{\epsilon}_\theta(\mathcal{G}_\tau, \tau) + \mathbf{s}_{\theta',\phi}(\mathcal{G}_\tau, \tau, \mathcal{C}) - \boldsymbol{\epsilon}\|_2^2, \tag{7}$$

https://doi.org/10.52202/085713-3172

where $\mathbb{D}$ is the downstream dataset, $\mathcal{G}_\tau$ is the noised graph drawn from $q(\mathcal{G}_\tau|\mathcal{G}_0)$, and $\mathbf{s}_{\theta',\phi}$ refers to our proposed equivariant diffusion adapter. At inference time, we use $\boldsymbol{\epsilon}_\theta(\mathcal{G}_\tau, \tau) + \mathbf{s}_{\theta',\phi}(\mathcal{G}_\tau, \tau, \mathcal{C})$ as the conditional score when computing the reverse transition kernel $p(\mathcal{G}_{\tau-1}|\mathcal{G}_\tau, \mathcal{C})$.

Our GeoAda offers several key advantages over standard supervised fine-tuning (SFT):**1.** The adapter modules are lightweight and operate as plug-and-play components, allowing flexible conditioning on new control signals without architectural modifications to the pretrained model. **2.** Parameter-efficient, as only a subset of trainable adapter modules are introduced. **3.** By freezing the Pretrained model and only optimizing lightweight adapters, the method imposes implicit regularization, helping to prevent overfitting. **4.** Through the careful design of SE(3)-equivariant adapter blocks and zero convolutions, GeoAda guarantees equivariance throughout the tuning process, thereby retaining the theoretical benefits of geometric diffusion models.

5 Experiment

We evaluate GeoAda across three categories of additional fine-tuning controls: (1) Frame control during dynamic prediction (§ 5.1), (2) Global type control in human motion prediction(§ 5.2), and (3) Subgraph control in molecule generation (§ 5.3). We also performed ablation studies on core design choices and present some observations in §5.4.

Baselines. We compare with three types of baselines: (1) *Fine-tuning* methods, including **Full FT** , which fine-tunes the entire model, and **PARTIAL-**k [12, 16, 45], which updates only the last k layers of the pre-trained model; (2) *Prompt-based* methods, including **GPF**, **GPF-plus** [6], which both inject learnable prompt features into the input space. And **Prompt-Template** maps new inputs to pre-training-style inputs using manually designed graph templates, specifically for the conditional case. (3) *Head-only* tuning methods, where **MLP-**k freezes the pre-trained model and uses a k-layer MLP as the prediction head. To preserve equivariance, we replace the MLP with an EGTN block in our implementation. More details can be found in App. 9.2.

Implementation. The input data are processed as geometric graphs. For both trajectory and global control settings, we follow the same experimental setup as GeoTDM [11], adopting EGTN as the backbone model with three GeoAda blocks and a hidden dimension of 128. For subgraph control, the base model follows the configuration of TargetDiff [8]. We use $\mathcal{T} = 1000$ and linear noise schedule [13]. More details in App. 9.1.

5.1 Frame Control

Task setup. For pre-training, we use the first 10 frames as the condition and train the model to predict the trajectory over the following 20 frames. In the downstream task, we adopt a different setup where the model observes 15 conditional frames and predicts the next 20 frames. We evaluate all models on both the original task and the new task to assess their generalization and adaptability across different settings.

Metrics. For conditional trajectory generation, we employ Average Discrepancy Error (ADE) and Final Discrepancy Error (FDE), which are widely adopted for trajectory forecasting [43, 40], given by $\text{ADE}(\mathbf{x}^{[T]}, \mathbf{y}^{[T]}) = \frac{1}{TN}\sum_{t=0}^{T-1}\sum_{i=0}^{N-1}\|\mathbf{x}_i^{(t)} - \mathbf{y}_i^{(t)}\|_2$, and $\text{FDE}(\mathbf{x}^{[T]}, \mathbf{y}^{[T]}) = \frac{1}{N}\sum_{i=0}^{N-1}\|\mathbf{x}_i^{(T-1)} - \mathbf{y}_i^{(T-1)}\|_2$. For probabilistic models, we report average ADE and FDE derived from $K = 5$ samples. For unconditional trajectory generation, we report three complementary scores: The Marginal score measures statistical alignment by computing the mean absolute error (MAE) between binned distributions of model-generated and ground-truth coordinates (or bond lengths for MD17). The Classification score is the cross-entropy of a binary classifier trained to distinguish generated trajectories from real ones, offering insight into sample realism. The Prediction score measures the mean squared error (MSE) of a sequence model trained on generated data and tested on real trajectories, reflecting the utility of generated samples for downstream prediction. For more detailed metric definitions, please refer to Appendix 9.5.

5.1.1 Particle Dynamic

Experimental Setup. We adopt the CHARGED PARTICLES dataset [17, 28] for particle dynamics simulation. In this dataset, $N = 5$ particles with randomly assigned charges of either $+1$ or -1 interact via Coulomb forces, resulting in complex, non-linear trajectories. We use 3000 trajectories for training, 2000 for validation, and 2000 for testing. We explore two settings: (1) Conditional trajectory generation: we use the first 10 frames of each trajectory as input to predict the subsequent 20 frames during pretraining, and 15 frames as input during finetuning to predict the next 20 frames. (2) Unconditional trajectory generation: we generate trajectories of length 20 from scratch during pretraining. During finetuning, we condition on the first 10 frames and predict the next 20 frames.

Table 1: Comparisons on CHARGED PARTICLES dataset.(all results reported by $\times 10^{-1}$).(↑) / (↓) denotes whether a larger / smaller number is preferred. "NaN" denotes generation collapse due to numerical instability, typically observed in baseline models after fine-tuning on original task. "–"indicates that the baseline PromptTem requires explicit conditioning and cannot be applied when no conditioning frame is given.

Setting	Uncondition					Condition			
Task	Downstream		Pretrain			Downstream		Pretrain	
Metric	ADE(↓)	FDE(↓)	Marg(↓)	Class(↑)	Pred(↓)	ADE(↓)	FDE(↓)	ADE(↓)	FDE(↓)
Pretrain	nan	nan	0.079±0.000	**5.149**±0.285	**0.109**±0.004	11.826±0.133	20.395±0.249	1.177±0.018	2.815±0.037
Full FT	**1.093** ±0.014	2.676±0.024	1.025±0.000	nan	nan	1.106±0.007	**2.590**±0.040	5.998 ±0.041	11.75 ±0.107
Prompt-Tem	-	-	-	-	-	1.723±0.014	3.703 ±0.061	nan	nan
PARTIAL-k [12]	1.685 ±0.006	3.594 ±0.040	1.016 ±0.000	nan	nan	1.409 ±0.009	3.330±0.042	9.325 ±0.064	11.94 ±0.149
MLP-k	6.258 ±0.947	9.111±2.628	1.015±0.000	0.00±0.000	5.740 ±2.914	1.503±0.016	3.338±0.039	3924	3950
GPF [6]	1.643 ±0.014	3.671 ±0.029	1.027±0.000	nan	nan	1.575 ±0.017	3.390±0.050	nan	nan
GPF-plus [6]	1.596 ±0.011	3.574±0.024	1.023 ±0.000	nan	nan	1.648±0.009	3.670±0.030	nan	nan
GeoAda	1.119 ±0.019	**2.669** ±0.022	**0.079**±0.000	5.134±0.247	0.111 ±0.006	**1.105**±0.012	2.621±0.033	**1.175**±0.033	**2.806**±0.033

Results. We present the results in Table 1, with the following observations. Under both the unconditional and conditional trajectory generation settings, GeoAda achieves comparable or better performance than Full FT on the downstream task (conditioning on the first 10 frames to predict the next 20), while only tuning half the number of parameters. Furthermore, it consistently outperforms other fine-tuning and prompt-based baselines, achieving 35.69% improvement on ADE and 21.29% on FDE. On the original pretraining task, all baselines exhibit substantial performance degradation, with some failing to generate valid diffusion samples, indicating that these methods suffer from overfitting and catastrophic forgetting due to excessive adaptation to the downstream task."In contrast, by leveraging equivariant zero convolutions, GeoAda retains the pretrained model's performance.

5.1.2 Molecular Dynamics

Experimental setup. We employ the MD17 [3] dataset, which contains the DFT-simulated molecular dynamics trajectories of 8 small molecules, with the number of atoms for each molecule ranging from 9 (Ethanol and Malonaldehyde) to 21 (Aspirin). For each molecule, 5000 trajectories are used for training and 1000/1000 for validation and testing, uniformly sampled along the time dimension. Different from [40], we explicitly involve the hydrogen atoms which contribute most to the vibrations of the trajectory, leading to a more challenging task. The node feature is the one-hot encodings of atomic number [29] and edges are connected between atoms within three hops measured in atomic bonds [30].

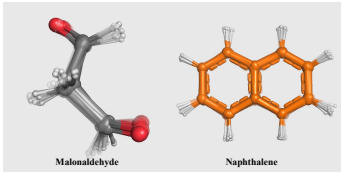

Figure 3: Visualization results of GeoAda on Malonaldehyde and Naphthalene from MD17 dataset.

Results. As shown in Table 2, GeoAda achieves state-of-the-art performance across all five molecular systems in the MD17 dataset, indicating strong transferability in geometric diffusion models. In downstream fine-tuning task, the method consistently matches or exceeds the performance of full fine-tuning and outperforms other prior methods by an average of 18.94% in ADE and 18.22% in FDE. Importantly, when returning to the original pretraining task, it retains performance comparable to the pretrained model, while both fine-tuning and prompt-based methods exhibit significant degradation or collapse due to overfitting. More experiment results on Malonaldehyde and Naphthalene in App. 10.1.

https://doi.org/10.52202/085713-3172

Table 2: Comparisons for Molecular Dynamics prediction on MD17 dataset (all results reported by $\times 10^{-1}$). The best results are highlighted in bold. Results averaged over 5 runs. "NaN" denotes generation collapse due to numerical instability, typically observed in baseline models after fine-tuning on the original task.

Scenarios	Aspirin				Benzene				Ethanol			
Task	Downstream		Pretrain		Downstream		Pretrain		FT		Pretrain	
Metric	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE
Pretrain	3.782±0.010	7.345±0.016	1.062±0.002	1.857±0.013	0.603±0.000	1.325±0.008	0.241±0.000	**0.393**±0.002	3.263±0.010	4.357±0.014	0.999±0.009	**1.856**±0.032
Full FT	0.929±0.002	1.602±0.005	1.323±0.003	2.280±0.016	0.217±0.001	0.360±0.002	nan	nan	0.997±0.007	1.906±0.002	inf	inf
PARTIAL-k [12]	1.071±0.003	1.875±0.009	1.439±0.003	2.407±0.007	0.249±0.005	0.407±0.003	0.318±0.110/nan	0.491±0.001/nan	1.288±0.008	2.161±0.018	6.502±4.574	4.636±2.423
MLP-k	1.132±0.004	1.921±0.004	1.579±0.005	2.641±0.007	0.248±0.012	0.412±0.004	nan	nan	1.301±0.001	2.275±0.021	2.228±0.004	2.716±0.022
Prompt-Tem	1.197±0.008	2.014±0.030	1.571±0.010	2.668±0.024	0.241±0.012	0.409±0.021	nan	nan	1.207±0.018	2.194±0.081	nan	nan
GPF [6]	1.130±0.006	1.909±0.024	3.260±0.015/inf	4.272±0.025/inf	0.246±0.001	0.415±0.004	nan	nan	1.239±0.023	2.233±0.059	2.420±0.156	3.519±0.056
GPF-plus [6]	1.014±0.006	1.962±0.021	2.243±0.009	3.349±0.018	0.234±0.010	0.331±0.057	1.118±0.006	1.083±0.010	1.137±0.025	2.048±0.046	3.240±0.081/inf	4.074 ±0.171/inf
GeoAda	**0.891** ±0.003	**1.533** ±0.008	**1.060**±0.003	**1.852** ±0.012	**0.191** ±0.000	**0.319**±0.002	**0.240**±0.002	0.394±0.005	**0.905**±0.007	**1.745**±0.010	**0.995**±0.005	1.867±0.019

5.2 Global Type Control

Experimental setup. The CMU Mocap dataset is a commonly used dataset for human pose prediction, which includes 8 action categories. A single pose has 38 body joints in the original dataset, among which we choose 25 joints following the configuration of MSR-GCN [4], using 10 frames as input to predict the subsequent 25 frames. For pre-training, we construct a dataset by combining the three most frequent actions: directing traffic, washing windows, and giving basketball signals. The remaining five actions are used as the downstram task fine-tuning dataset.

Results We report short-term and long-term motion prediction results on the CMU Mocap dataset in Tables 3 and 4. More results on jumping and soccer scenarios are in App. 10.2. GeoAda consistently achieves state-of-the-art performance across all action categories and time horizons. In this setting, the pretraining dataset is significantly larger than the downstream task dataset (30k vs. 245–1345 datapoints). As a result, naïve fine-tuning and prompt-based methods are highly prone to overfitting to the limited downstream training data, leading to notably degraded performance. Moreover, they are more likely to fail to generate valid samples on the original pretraining task, indicating a severe loss of pretrained knowledge and catastrophic forgetting. In contrast, GeoAda benefits from the implicit regularization effect of the adapter, which mitigates overfitting and preserves the performance of the pretrained model.

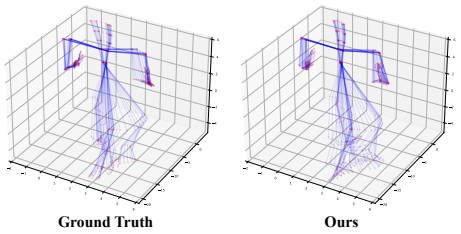

Figure 4: Visualization of Running trajectory

Table 3: Comparisons for short-term prediction on 5 action categories of the CMU Mocap dataset. The best results are highlighted in bold. Results averaged over 5 runs (std in App. 10.2).

scenarios	**running**				**pretrain**				**walking**				**pretrain**				**basketball**				**pretrain**			
millisecond (ms)	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Pretrain	28.74	57.99	126.20	159.37	**7.941**	**16.84**	**39.91**	**52.45**	15.49	30.66	68.71	89.23	7.941	**16.84**	39.91	**52.45**	17.92	35.89	78.48	101.12	7.941	16.84	39.91	**52.45**
Full FT	20.34	35.26	60.58	70.20	nan	nan	nan	nan	10.01	15.12	23.98	28.49	nan	nan	nan	nan	16.95	30.33	58.28	71.93	nan	nan	nan	nan
PARTIAL-k [12]	20.47	36.04	68.15	82.59	nan	nan	nan	nan	10.47	16.97	30.97	37.69	17.26	31.75	66.29	84.27	17.54	32.03	62.96	78.95	nan	nan	nan	nan
MLP-k	23.35	44.44	84.27	101.42	nan	nan	nan	nan	10.86	18.74	35.78	44.86	nan	nan	nan	nan	17.60	34.76	72.90	86.34	nan	nan	nan	nan
GPF [6]	18.48	31.94	58.80	73.56	21.38	35.44	65.23	79.55	10.79	16.79	28.32	33.48	22.04	37.10	71.24	88.23	18.48	31.94	58.80	73.56	21.38	35.44	65.23	79.55
GPF-plus [6]	19.11	31.93	**49.85**	56.67	nan	nan	nan	nan	10.17	16.23	26.29	31.54	nan	nan	nan	nan	17.17	31.86	58.48	72.71	nan	nan	nan	nan
GeoAda	**18.70**	**33.56**	50.26	**55.54**	7.972	16.91	40.06	52.61	**8.92**	**13.82**	**22.99**	**26.68**	**7.932**	16.90	**38.96**	52.54	**16.85**	**29.71**	**57.59**	**71.19**	**7.898**	**16.79**	**39.70**	52.50

Table 4: Comparisons for long-term prediction on 5 action categories of the CMU Mocap dataset. The best results are highlighted in bold. Results averaged over 5 runs (std in App. 10.2).

scenarios	**running**		**pretrain**		**walking**		**pretrain**		**basketball**		**pretrain**	
millisecond (ms)	560	1000	560	1000	560	1000	560	1000	560	1000	560	1000
Pretrain	219.16	314.85	**77.06**	130.51	129.43	212.94	**77.06**	130.51	143.49	223.99	77.06	130.51
Full FT	85.14	97.02	nan	nan	36.92	52.58	nan	nan	94.59	132.34	nan	nan
PARTIAL-k [12]	102.85	108.47	nan	nan	51.36	84.72	118.82	182.88	106.84	146.27	nan	nan
MLP-k	127.67	131.59	nan	nan	62.97	102.34	nan	nan	107.30	149.58	nan	nan
GPF [6]	61.92	71.42	nan	nan	42.37	52.24	119.43	171.74	97.16	128.29	nan	nan
GPF-plus [6]	63.56	71.60	nan	nan	41.31	56.47	nan	nan	104.54	130.76	104.51	155.02
GeoAda	**60.88**	**70.22**	77.22	**130.17**	**34.52**	**50.49**	78.12	**129.97**	**91.03**	**120.35**	**76.94**	**129.81**

5.3 Subgraph Control

Experimental setup. We adopt the QM9 [26] and GEOM-Drugs [1] dataset for pretraining a model for molecule generation, use the CrossDocked2020 dataset [7] for finetuning protein-ligand pair generation. QM9 [26] contains 130k small molecules with atom types (H, C, N, O, F). GEOM-Drugs [1]

Table 5: Summary of binding affinity and molecular properties of reference molecules and molecules generated by GeoAda and baselines. (↑) / (↓) denotes whether a larger / smaller number is preferred.

Methods	Vina Score (↓)		Vina Min (↓)		Vina Dock (↓)		High Affinity(↑)		QED(↑)		SA(↑)		Diversity(↑)	
	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.
liGAN [25]	-	-	-	-	-6.33	-6.20	21.1%	11.1%	0.39	0.39	0.59	0.57	0.66	0.67
GraphBP [19]	-	-	-	-	-4.80	-4.70	14.2%	6.7%	0.43	0.45	0.49	0.48	**0.79**	**0.78**
AR [21]	-5.75	-5.64	-6.18	-5.88	-6.75	-6.62	37.9%	31.0%	0.51	0.50	0.63	0.63	0.70	0.70
Pocket2Mol [24]	-5.14	-4.70	-6.42	-5.82	-7.15	-6.79	48.4%	51.0%	**0.56**	**0.57**	**0.74**	**0.75**	0.69	0.71
TargetDiff [10]	-5.47	-6.30	-6.64	**-6.83**	**-7.80**	**-7.91**	58.1%	59.1%	0.48	0.48	0.58	0.58	0.72	0.71
GeoAda (qm9)	**-5.54**	**-6.31**	-6.64	-6.46	-7.62	-7.64	57.4%	58.2%	0.49	0.51	0.58	0.58	0.74	0.75
GeoAda (Geom)	-5.54	-6.01	**-6.68**	-6.32	-7.64	-7.71	**58.3%**	**59.3%**	0.48	0.50	0.58	0.58	0.76	0.75
Reference	-6.36	-6.41	-6.71	-6.49	-7.45	-7.26	-	-	0.48	0.47	0.73	0.74	-	-

Table 6: Jensen-Shannon divergence of bond distance distributions between reference and generated molecules. (↓)

Bond	liGAN	AR	Pocket2Mol	TargetDiff	GeoAda (qm9)	GeoAda (Geom)
C−C	0.601	0.609	0.496	0.369	**0.243**	0.269
C=C	0.665	0.620	0.561	0.505	**0.377**	0.393
C−N	0.634	0.474	0.416	0.363	**0.363**	0.396
C=N	0.749	0.635	0.629	0.550	0.300	**0.299**
C−O	0.656	0.492	0.454	0.421	**0.418**	0.428
C=O	0.661	0.558	0.516	0.461	0.279	**0.257**
C:C	0.497	0.451	0.416	**0.263**	0.305	0.335
C:N	0.638	0.552	0.487	**0.235**	0.297	0.330

Table 7: Percentage of different ring sizes for reference and model generated molecules.

Ring Size	Ref.	liGAN	AR	Pocket2Mol	TargetDiff	GeoAda (qm9)	GeoAda (geom)
3	1.7%	28.1%	29.9%	0.1%	0.0%	0.0%	0.0%
4	0.0%	15.7%	0.0%	0.0%	2.8%	6.7%	5.8%
5	30.2%	29.8%	16.0%	16.4%	30.8%	47.2%	45.8%
6	67.4%	22.7%	51.2%	80.4%	50.7%	69.1%	78.2%
7	0.7%	2.6%	1.7%	2.6%	12.1%	23.5	21.3%
8	0.0%	0.8%	0.7%	0.3%	2.7%	5.3%	4.7%
9	0.0%	0.3%	0.5%	0.1%	0.9%	3.8%	1.6%

is a large-scale dataset of 430k molecular conformers with heavy atoms, and we keep the lowest energy conformation for each molecule. Following the common setup for CrossDocked2020 [10], we obtain 100k complexes for training and 100 novel complexes for testing. Since CrossDocked2020 has different atom type configuration from QM9 and GEOM-Drugs, we limit the atom type to (H, C, N, O, F, P, S, Cl). Following [10], proteins and ligands are expressed as 3D atom coordinates and one-hot vectors containing the atom types.

Implementation. Following prior work [10], we use the Adam optimizer with a learning rate of 0.001 and β values of (0.95, 0.999). Batch size is set to 4 and gradient clipping set to 8. To balance the atom type and position losses, we scale the atom type loss by a factor of $\lambda = 100$.

Results. We evaluate molecular properties and molecular structures of the proposed model and baselines on target-aware molecule generation in Table 5, 6, and 7. Baseline models are trained on CrossDocked2020 under explicit protein conditioning. GeoAda matches, and in multiple cases surpasses the strongest baselines on all metrics, generating ligand molecules that maintain realistic structures, high binding affinity, comparable drug-likeness and sythetic accessibility. The lightweight adapter can inject subgraph (pocket) control into a broadly pretrained geometric diffusion model, achieving or surpassing task-specific baselines that rely on end-to-end training with protein context.

5.4 Ablations and Observations

Observation of the sudden convergence phenomenon Similar to the phenomenon observed in ControlNet [44], we also observe a *sudden convergence phenomenon* in our training process. As shown in Figure 5, between step 4500 and 4700, both training loss and validation MSE drop abruptly rather than gradually. To investigate this behavior, we conducted inference using the saved checkpoints from steps 3600 and 5600, and observed a notable performance jump between steps 4400 and 4800, which corresponds to significant reductions in ADE and FDE by 68.3% and 73.4%.

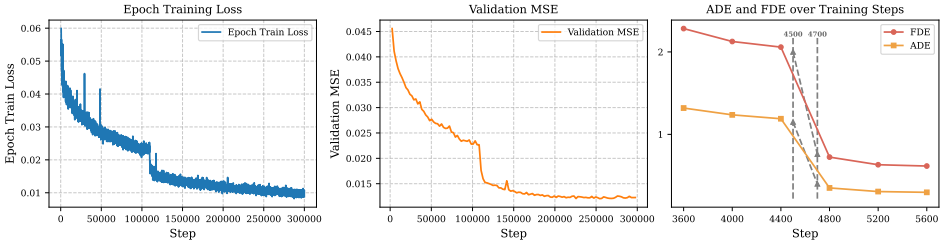

Figure 5: The sudden convergence phenomenon

Parameter efficiency analysis As shown in Appendix 10.3.1, we explore the impact of varying the number of equivariant zero layers. Increasing the number of trainable copy layers generally improves

https://doi.org/10.52202/085713-3172

performance, but introduces more parameters and computational cost, revealing a trade-off between performance and efficiency. We also reported the number of tunable parameters for different tuning strategies in Table 21. Except for full fine-tuning, which is substantially larger, all other methods, including GeoAda, use comparable parameter sizes.

Ablation on Equivariant Zero Convolutions We evaluate two variants to assess the role of equivariant zero convolution: (1) replacing it with Gaussian-initialized standard convolutions, and (2) replacing each trainable copy with a single convolution layer (see App. 10.3.2). Both modifications result in notable performance drops, underscoring the importance of zero initialization and structural design for stable and effective fine-tuning.

6 Conclusion

We present GeoAda, a parameter-efficient and SE(3)-equivariant adapter framework for geometric diffusion models. It enables effective adaptation to diverse geometric control tasks without modifying the pretrained backbone, preserving both performance and geometric consistency.

7 Acknlowlegdements

This project was funded in part by ARO (W911NF-21-1-0125), ONR (N00014-23-1-2159), and the CZ Biohub.

References

[1] Simon Axelrod and Rafael Gomez-Bombarelli. Geom, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9(1):185, 2022. 8

[2] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander Schwing, and Liangyan Gui. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation, 2023. 3

[3] Stefan Chmiela, Alexandre Tkatchenko, Huziel E Sauceda, Igor Poltavsky, Kristof T Schütt, and Klaus-Robert Müller. Machine learning of accurate energy-conserving molecular force fields. *Science advances*, 3(5):e1603015, 2017. 1, 7

[4] Lingwei Dang, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. Msr-gcn: Multi-scale residual graph convolution networks for human motion prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11467–11476, 2021. 8

[5] Allan dos Santos Costa, Ilan Mitnikov, Franco Pellegrini, Ameya Daigavane, Mario Geiger, Zhonglin Cao, Karsten Kreis, Tess Smidt, Emine Kucukbenli, and Joseph Jacobson. Equijump: Protein dynamics simulation via so(3)-equivariant stochastic interpolants, 2024. 3

[6] Taoran Fang, Yunchao Zhang, Yang Yang, Chunping Wang, and Lei Chen. Universal prompt tuning for graph neural networks. *Advances in Neural Information Processing Systems*, 36:52464–52489, 2023. 3, 6, 7, 8, 24, 25, 26

[7] Paul G Francoeur, Tomohide Masuda, Jocelyn Sunseri, Andrew Jia, Richard B Iovanisci, Ian Snyder, and David R Koes. Three-dimensional convolutional neural networks and a cross-docked data set for structure-based drug design. *Journal of chemical information and modeling*, 60(9):4200–4215, 2020. 8

[8] Jiaqi Guan, Wesley Wei Qian, Xingang Peng, Yufeng Su, Jian Peng, and Jianzhu Ma. 3d equivariant diffusion for target-aware molecule generation and affinity prediction. In *The Eleventh International Conference on Learning Representations*, 2023. 1, 4, 6

[9] Jiaqi Guan, Wesley Wei Qian, Xingang Peng, Yufeng Su, Jian Peng, and Jianzhu Ma. 3d equivariant diffusion for target-aware molecule generation and affinity prediction, 2023. 2

[10] Jiaqi Guan, Wesley Wei Qian, Xingang Peng, Yufeng Su, Jian Peng, and Jianzhu Ma. 3d equivariant diffusion for target-aware molecule generation and affinity prediction. *arXiv preprint arXiv:2303.03543*, 2023. 9

[11] Jiaqi Han, Minkai Xu, Aaron Lou, Haotian Ye, and Stefano Ermon. Geometric trajectory diffusion models. *arXiv preprint arXiv:2410.13027*, 2024. 1, 3, 6

[12] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 6, 7, 8, 24, 25, 26

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3, 6, 21

[14] Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pages 8867–8887. PMLR, 2022. 1, 2, 3, 4

[15] Zijie Huang, Wanjia Zhao, Jingdong Gao, Ziniu Hu, Xiao Luo, Yadi Cao, Yuanzhou Chen, Yizhou Sun, and Wei Wang. Physics-informed regularization for domain-agnostic dynamical system modeling. *arXiv preprint arXiv:2410.06366*, 2024. 1

[16] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European conference on computer vision*, pages 709–727. Springer, 2022. 6

[17] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. *arXiv preprint arXiv:1802.04687*, 2018. 1, 7

[18] Shengrui Li, Xueting Han, and Jing Bai. Adaptergnn: Parameter-efficient fine-tuning improves generalization in gnns, 2023. 3

[19] Meng Liu, Youzhi Luo, Kanji Uchino, Koji Maruhashi, and Shuiwang Ji. Generating 3d molecules for target protein binding. In *International Conference on Machine Learning*, 2022. 9

[20] Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. Graphprompt: Unifying pre-training and downstream tasks for graph neural networks, 2023. 3

[21] Shitong Luo, Jiaqi Guan, Jianzhu Ma, and Jian Peng. A 3d generative model for structure-based drug design. *Advances in Neural Information Processing Systems*, 34, 2021. 9

[22] Yanchen Luo, Junfeng Fang, Sihang Li, Zhiyuan Liu, Jiancan Wu, An Zhang, Wenjie Du, and Xiang Wang. Text-guided diffusion model for 3d molecule generation. *arXiv preprint arXiv:2410.03803*, 2024. 4

[23] Alex Morehead and Jianlin Cheng. Geometry-complete diffusion for 3d molecule generation and optimization, 2024. 1, 2

[24] Xingang Peng, Shitong Luo, Jiaqi Guan, Qi Xie, Jian Peng, and Jianzhu Ma. Pocket2mol: Efficient molecular sampling based on 3d protein pockets. *arXiv preprint arXiv:2205.07249*, 2022. 9

[25] Matthew Ragoza, Tomohide Masuda, and David Ryan Koes. Generating 3D molecules conditional on receptor binding sites with deep generative models. *Chem Sci*, 13:2701–2713, Feb 2022. 9

[26] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014. 8

[27] Hyunwoo Ryu, Jiwoo Kim, Hyunseok An, Junwoo Chang, Joohwan Seo, Taehan Kim, Yubin Kim, Chaewon Hwang, Jongeun Choi, and Roberto Horowitz. Diffusion-edfs: Bi-equivariant denoising generative modeling on se(3) for visual robotic manipulation, 2023. 1, 3

 https://doi.org/10.52202/085713-3172

[28] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. *arXiv preprint arXiv:2102.09844*, 2021. 1, 7

[29] Kristof Schütt, Oliver Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *International Conference on Machine Learning*, pages 9377–9388. PMLR, 2021. 7

[30] Chence Shi, Shitong Luo, Minkai Xu, and Jian Tang. Learning gradient fields for molecular conformation generation. In *International conference on machine learning*, pages 9558–9568. PMLR, 2021. 7

[31] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019. 3

[32] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 3

[33] Charles Stein. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proceedings of the sixth Berkeley symposium on mathematical statistics and probability, volume 2: Probability theory*, volume 6, pages 583–603. University of California Press, 1972. 3

[34] Mingchen Sun, Kaixiong Zhou, Xin He, Ying Wang, and Xin Wang. Gppt: Graph pre-training and prompt tuning to generalize graph neural networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 1717–1727, New York, NY, USA, 2022. Association for Computing Machinery. 3

[35] Masatoshi Uehara, Yulai Zhao, Kevin Black, Ehsan Hajiramezanali, Gabriele Scalia, Nathaniel Lee Diamant, Alex M Tseng, Tommaso Biancalani, and Sergey Levine. Fine-tuning of continuous-time diffusion models as entropy-regularized control, 2024. 3

[36] Julen Urain, Niklas Funk, Jan Peters, and Georgia Chalvatzaki. Se(3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion, 2023. 1, 3

[37] Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization, 2023. 3

[38] Zhendong Wang, Yifan Jiang, Yadong Lu, yelong shen, Pengcheng He, Weizhu Chen, Zhangyang Wang, and Mingyuan Zhou. In-context learning unlocked for diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 1, 3

[39] Can Xu, Haosen Wang, Weigang Wang, Pengfei Zheng, and Hongyang Chen. Geometric-facilitated denoising diffusion model for 3d molecule generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 338–346, 2024. 1, 2

[40] Chenxin Xu, Robby T Tan, Yuhong Tan, Siheng Chen, Yu Guang Wang, Xinchao Wang, and Yanfeng Wang. Eqmotion: Equivariant multi-agent motion prediction with invariant interaction reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1410–1420, 2023. 6, 7

[41] Minkai Xu, Alexander Powers, Ron Dror, Stefano Ermon, and Jure Leskovec. Geometric latent diffusion models for 3d molecule generation. In *International Conference on Machine Learning*. PMLR, 2023. 1, 2

[42] Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations*, 2022. 1, 2, 3

[43] Pei Xu, Jean-Bernard Hayet, and Ioannis Karamouzas. Socialvae: Human trajectory prediction using timewise latents. In *European Conference on Computer Vision*, pages 511–528. Springer, 2022. 6

[44] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023. 1, 2, 3, 5, 9

[45] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 649–666. Springer, 2016. 5, 6

 https://doi.org/10.52202/085713-3172

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and Introduction clearly state the main contributions, which include the proposal of GeoAda, an SE(3)-equivariant adapter framework for efficient fine-tuning of geometric diffusion models. The claims cover parameter efficiency, preservation of geometric consistency/equivariance, mitigation of overfitting and catastrophic forgetting, and wide applicability across diverse geometric control types and domains. These claims appear to be supported by the theoretical proofs (discussed in Section 4 and Appendix 7) and experimental results presented (Section 5).

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Discussed in App. 11.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: The paper states in the abstract that it theoretically proves the proposed adapters maintain SE(3)-equivariance. Proposition 4.1 (Section 4.2) addresses the equivariance of control encoding. Section 4 (Method) details the design to ensure SE(3)-equivariance. Appendix 7 is explicitly titled "Proof", and Appendix 8.5.4 and 8.5.5 discuss theoretical aspects like Theorem 8.1 and 8.2 with related assumptions.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Section 5 (Experiment) details the experimental setup, including datasets, task setups for pre-training and fine-tuning, evaluation metrics, and baselines. Section 5 also mentions implementation details. Further specifics are provided in Appendix 8, including Appendix 8.2 (Hyper-parameters), Appendix 8.3 (Baselines), and Appendix 8.4 (Details of the datasets).

   Guidelines:

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

 https://doi.org/10.52202/085713-3172

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

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

   Answer: [No]

   Justification: The paper provides detailed descriptions of the experimental setup, datasets, and implementation details in Section 5 and Appendix 8, which support reproducibility. However, it does not explicitly state that the code and data are open access or provide URLs to repositories.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 5 (Experiment) outlines experimental setups for different tasks, including dataset descriptions and fine-tuning specifics. Implementation details are mentioned within Section 5. Appendix 8 provides further details, with Appendix 8.2 specifying hyperparameters (Table 5), Appendix 8.1 discussing compute resources, Appendix 8.3 detailing baselines, and Appendix 8.4 giving dataset statistics including splits.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The tables in Section 5 (e.g., Table 1, Table 2, Table 3, Table 4) and Appendix 9 (e.g., Table 8, Table 9, Table 10) report mean results along with what appear to be standard deviations (e.g., "11.826 ±0.133" from Table 1). Captions for these tables often state "Results averaged over 5 runs" and some explicitly mention "(std in App. 9.4)".

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Appendix 8.1 ("Compute Resources") specifies the use of "4 Nvidia A6000 GPUs," training times for different datasets ("NBody and ETH-UCY take around 12 hours while each MD17 training phase takes about a day"), and that "CPUs were standard intel CPUs."

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: There is no indication of ethical violations from the research described.

https://doi.org/10.52202/085713-3172

Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Discussed in Appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: The research focuses on geometric diffusion models for scientific applications such as molecular and particle dynamics, and human motion prediction using established datasets. These applications and datasets do not typically fall into the high-risk category for misuse in the same vein as large language models generating text or image generators creating photorealistic fakes of individuals, which would necessitate specific, explicit safeguards mentioned in the paper beyond standard responsible research conduct.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The paper credits the original sources of the datasets used (e.g., MD17, QM9, GEOM-Drugs, CrossDocked2020, CHARGED PARTICLES, CMU Mocap) by citing the relevant publications in Section 5 and the References section. While the specific license for each dataset is not reiterated within this paper's text, citing the original publication is standard academic practice for acknowledging the source and implicitly, adherence to their terms of use.

    Guidelines:

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

13. **New assets**

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

    Answer: [NA]

    Justification: This paper is not about a new dataset or a standalone software/model asset in the sense that requires separate documentation released alongside an asset. The proposed method itself is documented within the paper (Section 4).

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [NA]

https://doi.org/10.52202/085713-3172

Justification: The paper utilizes existing datasets, such as the CMU Mocap dataset for human motion studies (Section 5.2). It does not describe any new crowdsourcing efforts or new research involving direct data collection from human subjects.

Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

Answer: [NA]

Justification: The research relies on pre-existing, publicly available datasets (e.g., CMU Mocap, mentioned in Section 5.2). The paper does not detail new experiments involving human subjects that would necessitate a new IRB approval process to be described within this work.

Guidelines:

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

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core methodology of the paper revolves around geometric diffusion models and equivariant adapters (Section 4). There is no mention or indication that Large Language Models (LLMs) are an important, original, or non-standard component of the research methods presented.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

Appendix

8 Proof

Below is the explanation and proof of Proposition 4.1:

Proof. Since $\boldsymbol{\epsilon}_\theta$ is SE(3)-equivariant by assumption, we have for any $h \in \mathrm{SE}(3)$,

$$\boldsymbol{\epsilon}_\theta(h \cdot \mathcal{G}') = h \cdot \boldsymbol{\epsilon}_\theta(\mathcal{G}'), \quad \text{where } \mathcal{G}' = \mathbf{f}(\mathcal{G}_\tau, \mathcal{C}).$$

We consider each component:

- The coupling operator $\mathbf{f}$ augments $\mathcal{G}_\tau$ with control $\mathcal{C}$ in a way that respects the SE(3) structure: global controls modify features invariantly; subgraph controls are merged geometrically; frame controls concatenate along the temporal axis. Thus, $\mathbf{f}$ is SE(3)-equivariant.
- The decoupling operator $\mathbf{g}$ selects a subset of nodes or frames without altering their coordinates. Therefore, it commutes with SE(3) action: $\mathbf{g}(h \cdot \mathcal{G}'') = h \cdot \mathbf{g}(\mathcal{G}'')$.

Combining the above, we explicitly see that for any $h \in \mathrm{SE}(3)$ defined as $h(\mathbf{x}) = R\mathbf{x} + \mathbf{d}$, we have:

$$\begin{aligned}
&\mathbf{g} \circ \boldsymbol{\epsilon}_\theta \circ \mathbf{f}(h \cdot (\mathcal{G}_\tau, \mathcal{C})) \\
&= \mathbf{g}\left(\boldsymbol{\epsilon}_\theta\left(h \cdot \mathbf{f}(\mathcal{G}_\tau, \mathcal{C})\right)\right) \\
&= \mathbf{g}\left(h \cdot \boldsymbol{\epsilon}_\theta\left(\mathbf{f}(\mathcal{G}_\tau, \mathcal{C})\right)\right) \\
&= h \cdot \mathbf{g}\left(\boldsymbol{\epsilon}_\theta\left(\mathbf{f}(\mathcal{G}_\tau, \mathcal{C})\right)\right) \\
&= R \cdot (\mathbf{g} \circ \boldsymbol{\epsilon}_\theta \circ \mathbf{f}(\mathcal{G}_\tau, \mathcal{C})) + \mathbf{d} \\
&= h \cdot (\mathbf{g} \circ \boldsymbol{\epsilon}_\theta \circ \mathbf{f}(\mathcal{G}_\tau, \mathcal{C}))
\end{aligned}$$

Therefore, the composed function $\mathbf{g} \circ \boldsymbol{\epsilon}_\theta \circ \mathbf{f}$ is SE(3)-equivariant.

□

9 More Details on Experiments

9.1 Hyper-parameters

We provide the detailed hyper-parameters of GeoAda in Table 8. We adopt Adam optimizer with betas $(0.9, 0.999)$ and $\epsilon = 10^{-8}$. For all experiments, we use the linear noise schedule per [13] with $\beta_{\text{start}} = 0.02$ and $\beta_{\text{end}} = 0.0001$.

Table 8: Hyper-parameters of GeoAda in the experiments.

	`n_layer`	`hidden`	`time_emb_dim`	$\mathcal{T}$	`batch_size`	`learning_rate`
N-body	6	128	32	1000	128	0.0001
MD	6	128	32	1000	128	0.0001
CMU Mocap	6	64	32	100	128	0.0001

9.2 Baselines

Full FT.

Full FT fully fine-tunes the pre-trained model f during downstream training. The entire model is updated to fit the target task.

https://doi.org/10.52202/085713-3172

PARTIAL-k.

PARTIAL-k fine-tunes only the last k layers of the model f, while freezing the remaining layers. This method balances adaptability with parameter efficiency by limiting the number of updated layers.

Graph Prompt Feature (GPF).

In **GPF**, the pre-trained encoder f is kept frozen, and a learnable prompt vector p is injected into the input feature space. During training, only the prompt vector p and the prediction head θ are optimized. This method enables task adaptation through a lightweight, task-specific prompt without modifying the backbone model. In our implementation, we replace the original MLP head with a three-layer Equivariant Geometric Trajectory Network (EGTN), which ensures the projection head maintains geometric consistency with the model.

Graph Prompt Feature-Plus (GPF-plus).

Extending GPF, this variant constructs node-wise prompt vectors using k learnable basis vectors $p_1^b, \ldots, p_k^b$ and a set of learnable linear weights $a_1, \ldots, a_k$. These components are used to compute node-specific prompts p_i via a compositional mechanism. The model f remains frozen, while prediction head θ, learnable basis vectors p_i^b, andlearnable linear weights a_i are optimized.

Prompt-Template

We prepend a learnable prompt layer(Equivariant Geometric Trajectory Network) to adapt new inputs to the distribution seen during pretraining, following with prediction head θ.

MLP-k (EGTN-k).

This baseline freezes the entire pre-trained model f and replaces the prediction head with a k-layer multilayer perceptron (MLP). To preserve equivariance in our setting, we replace the MLP with an Equivariant Geometric Trajectory Network (EGTN) block. Only the EGTN-based head is trained during the downstream task.

9.3 Details of the datasets

9.3.1 Global Type Control

Pretrain Dataset The statistics of the pretrained datasets on Global Type Control are presented in Table. 9.

Table 9: Pretrain Dataset statistics by Global Type.

Type	Washwindow	Directing Traffic	Basketball Signal	Pretrain
train	12126	9557	7776	29459
val	1342	2346	1920	5588
test	1342	2346	1920	5588

Downstream datasets The statistics of the downstream datasets utilized for the models pretrained on Global Type Control are presented in Table. 10.

Table 10: Downstream Dataset statistics by Global Type.

Type	Running	Walking	Jumping	Basketball	Soccer
train	245	869	1345	1044	1210
val	47	145	1008	254	264
test	47	145	1008	254	264

9.4 Model

9.4.1 Geometric Trajectory Diffusion Models

Unconditional Generation For unconditional generation, we model the trajectory distribution subject to SE(3)-invariance. The following theorem provides constraints for the prior and transition kernel.

Theorem 9.1. *If the prior $p_{\mathcal{T}}(\mathbf{x}_{\mathcal{T}}^{[T]})$ is SE(3)-invariant, and the transition kernels $p_{\tau-1}(\mathbf{x}_{\tau-1}^{[T]} \mid \mathbf{x}_{\tau}^{[T]}), \forall \tau \in \{1, \cdots, \mathcal{T}\}$ are SE(3)-equivariant, then the marginal $p_{\tau}(\mathbf{x}_{\tau}^{[T]})$ at any step $\tau \in \{0, \cdots, \mathcal{T}\}$ is also SE(3)-invariant.*

Prior in the translation-invariant subspace. The prior is built on a translation-invariant subspace $\mathcal{X}_{\mathbf{P}} \subset \mathcal{X}$, induced by a linear transformation $\mathbf{P}$:

$$\mathbf{P} = \mathbf{I}_D \otimes \left(\mathbf{I}_{TN} - \frac{1}{TN} \mathbf{1}_{TN} \mathbf{1}_{TN}^{\top} \right)$$

which results in a restricted Gaussian distribution supported only on the subspace, denoted $\tilde{\mathcal{N}}(\mathbf{0}, \mathbf{I})$, and is isotropic and SO(3)-invariant. To sample, one samples from $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and projects it onto the subspace.

Transition kernel. The transition kernel is parameterized in the subspace $\mathcal{X}_{\mathbf{P}}$, given by:

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

where the mean function $\tilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}}$ is SO(3)-equivariant. The function is re-parameterized as:

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

where $\tilde{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}} = P \circ \mathbf{f}_{\boldsymbol{\theta}}$ is an SO(3)-equivariant adaptation of the proposed EGTN.

Training and inference. The VLB is optimized for training, with the objective:

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

The inference process involves projecting intermediate samples onto the subspace $\mathcal{X}_{\mathbf{P}}$.

Conditional Generation In conditional generation, the target distribution is SE(3)-equivariant with respect to the given frames. The following theorem provides constraints for the prior and transition kernel.

Theorem 9.2. *If the prior $p_{\mathcal{T}}(\mathbf{x}_{\mathcal{T}}^{[T]} \mid \mathbf{x}_c^{[T_c]})$ is SE(3)-equivariant, and the transition kernels $p_{\tau-1}(\mathbf{x}_{\tau-1}^{[T]} \mid \mathbf{x}_{\tau}^{[T]}, \mathbf{x}_c^{[T_c]})$, $\forall \tau \in \{1, \cdots, \mathcal{T}\}$ are SE(3)-equivariant, the marginal $p_{\tau}(\mathbf{x}_{\tau}^{[T]} \mid \mathbf{x}_c^{[T_c]})$, $\forall \tau \in \{0, \cdots, \mathcal{T}\}$ is SE(3)-equivariant.*

Flexible equivariant prior. We provide a guideline for distinguishing feasible prior designs. The prior $\mathcal{N}(\boldsymbol{\mu}(\mathbf{x}_c^{[T_c]}), \mathbf{I})$ is SE(3)-equivariant if $\boldsymbol{\mu}(\mathbf{x}_c^{[T_c]})$ is SE(3)-equivariant. The mean function $\boldsymbol{\mu}(\mathbf{x}_c^{[T_c]})$ serves as an anchor to transition geometric information from the given frames to the target distribution. For instance, the anchor can be defined as:

$$\mathbf{x}_r^{[T]} = \mathbf{1}_{T \times N} \otimes \sum_{s \in [T_c]} w^{(s)} \bar{\mathbf{x}}_c^{(s)}$$

where the weights satisfy $\sum_{s \in [T_c]} w^{(s)} = 1$.

The weights $\mathbf{w}^{(t,s)}$ are derived as:

$$\mathbf{W}_{t,s} = [\boldsymbol{\gamma} \otimes \hat{\mathbf{h}}_c^{[T_c]}]_{t,s} \in \mathbb{R}^N, \quad \mathbf{w}^{(t,s)} = \begin{cases} \mathbf{W}_{t,s} & \text{if } s < T_c - 1, \\ \mathbf{1}_N - \sum_{s=0}^{T_c-2} \mathbf{W}_{t,s} & \text{if } s = T_c - 1. \end{cases}$$

where $\boldsymbol{\gamma} \in \mathbb{R}^T$ are learnable parameters, ensuring the constraint for translation equivariance.

Transition kernel. To match the proposed prior, we modify both the forward and reverse processes. The forward process is defined as:

$$q(\mathbf{x}_{\tau}^{[T]} | \mathbf{x}_{\tau-1}^{[T]}, \mathbf{x}_c^{[T_c]}) \coloneqq \mathcal{N}(\mathbf{x}_{\tau}^{[T]}; \mathbf{x}_r + \sqrt{1 - \beta_{\tau}}(\mathbf{x}_{\tau-1}^{[T]} - \mathbf{x}_r), \beta_{\tau} \mathbf{I}),$$

which ensures that $q(\mathbf{x}_{\mathcal{T}}^{[T]} | \mathbf{x}_c^{[T_c]})$ matches the equivariant prior $\mathbf{x}_r$ (proof in App. **??**). The reverse transition kernel is:

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

 https://doi.org/10.52202/085713-3172

We adopt the noise prediction objective for the reverse process, rewriting $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ as:

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

where the denoising network $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ is implemented as an EGTN with translation invariance, ensuring the translation equivariance of $\boldsymbol{\mu}_{\boldsymbol{\theta}}$.

Training and inference. Optimizing the VLB of our diffusion model leads to the following objective:

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

9.5 Evaluation Metrics in the Unconditional Case

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

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

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

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

10 More Experiments and Discussions

10.1 Molecular

Additional experimental results on the Malonaldehyde and Naphthalene are shown below:

Table 11: Comparisons for Molecular Dynamics prediction on MD17 dataset (all results reported by $\times 10^{-1}$). The best results are highlighted in bold. Results averaged over 5 runs

Scenarios	Malonaldehyde				Naphthalene			
Task	Downstream		Pretrain		Downstream		Pretrain	
Metric	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE
Pretrain	3.235±0.012	5.189±0.023	**0.962**±0.007	1.584±0.021	1.416±0.003	2.268±0.005	0.714±0.002	0.972±0.006
FT	0.897±0.002	1.511±0.009	1.405±0.006	2.237±0.023	**0.555**±0.001	0.867±0.010	nan	nan
PARTIAL-k [12]	0.981±0.004	1.675±0.015	1.230±0.003	2.110±0.006	0.653±0.002	0.903±0.003	2.083±0.009	1.629±0.007
MLP-k	0.997±0.005	1.694±0.010	1.291±0.004	2.051±0.015	0.718±0.001	0.969±0.005	nan	nan
Prompt-Tem	1.092±0.019	2.003±0.056	2.323±0.024	3.351±0.081	0.972±0.006	1.593±0.021	nan	nan
GPF [6]	1.176±0.012	1.931±0.030	282.1±157.9/inf	24.97±24.43/inf	0.758±0.002	1.005±0.005	nan	nan
GPF-plus [6]	1.018±0.003	1.793±0.010	3.527±0.501	4.719±1.397	0.717±0.003	0.873±0.006	1.891±0.056	2.674±0.156
GeoAda	**0.862** ±0.002	**1.414** ±0.014	**0.963**±0.007	**1.573**±0.018	0.581±0.002	**0.822**±0.004	**0.714**±0.001	**0.969**±0.007

10.2 Human Motion

Additional experimental results on the *jumping* and *soccer* scenarios are presented below. We also report the standard deviations across all experiments.

Table 12: Short-term prediction on **running** from the CMU Mocap dataset.

scenarios	running				pretrain			
millisecond (ms)	80	160	320	400	80	160	320	400
Pretrain	28.74±0.34	57.99±0.33	126.20 ±0.93	159.37±1.15	**7.941**±0.02	**16.84** ±0.04	**39.91**±0.43	**52.45** ±0.07
Full FT	20.34 ±0.32	35.26 ±0.19	60.58±1.14	70.20 ±1.36	nan	nan	nan	nan
PARTIAL-k	20.47±0.32	36.04 ±0.40	68.15±0.70	82.59±0.86	nan	nan	nan	nan
MLP-k	23.35 ±0.39	44.44±0.71	84.27 ±1.58	101.42±2.11	nan	nan	nan	nan
GPF	19.17±0.28	32.85±0.66	52.83±1.03	60.90 ±1.54	21.38 ±0.97	35.44±1.05	65.23±1.76	79.55 ±1.11
GPF-plus	19.11±0.54	31.93±0.81	**49.85** ±1.21	56.67±1.39	nan	nan	nan	nan
GeoAda	**18.70** ±0.37	**33.56**±0.25	50.26±0.42	**55.54**±0.36	7.972±0.02	16.91±0.05	40.06±0.42	52.61±0.07

Table 13: Short-term prediction on **walking** from the CMU Mocap dataset.

scenarios	walking				pretrain			
millisecond (ms)	80	160	320	400	80	160	320	400
Pretrain	15.49±0.07	30.66±0.16	68.71±0.33	89.23 ±0.43	7.941±0.02	**16.84** ±0.04	39.91±0.43	**52.45** ±0.07
Full FT	10.01 ±0.17	15.12 ±0.14	23.98±0.13	28.49 ±0.19	nan	nan	nan	nan
PARTIAL-k	10.47 ±0.81	16.97 ±0.43	30.97±0.60	37.69 ±0.60	17.26±0.23	31.75±0.36	66.29±0.44	84.27±0.79
MLP-k	10.86±1.07	18.74 ±1.37	35.78±1.92	44.86±1.35	nan	nan	nan	nan
GPF	10.79±0.14	16.79 ±0.16	28.32±0.21	33.48±0.22	22.04±0.38	37.10 ±0.40	71.24±0.39	88.23±0.97
GPF-plus	10.17 ±0.16	16.23±0.12	26.29±0.22	31.54 ±0.19	nan	nan	nan	nan
GeoAda	**8.92**±1.02	**13.82**±1.26	**22.99**±1.30	**26.68**±1.31	**7.932**±0.03	16.90±0.04	**38.96**±0.47	52.54±0.10

Table 14: Short-term prediction comparison on the **basketball** action from the CMU Mocap dataset.

scenarios	basketball				pretrain			
millisecond (ms)	80	160	320	400	80	160	320	400
Pretrain	17.92±0.06	35.89±0.12	78.48±0.47	101.12±0.69	7.941±0.02	16.84±0.04	39.91 ±0.43	**52.45**±0.07
FT	16.95±0.11	30.33 ±0.17	58.28±0.41	71.93±0.54	-	-	-	-
PARTIAL-k	17.54±0.12	32.03±0.43	62.96±0.86	78.95±0.98	-	-	-	-
MLP-k	17.60±0.50	34.76±1.26	72.90±2.36	86.34 ±1.53	-	-	-	-
GPF	18.48±0.09	31.94±0.14	58.80±0.28	73.56±0.29	21.38±0.11	35.44±0.27	65.23±0.30	79.55±0.51
GPF-plus	17.17 ±0.07	31.86±0.15	58.48±0.32	72.71 ±0.24	-	-	-	-
GeoAda	**16.85**±0.24	**29.71**±0.41	**57.59**±0.39	**71.19**±0.48	**7.898**±0.05	**16.79** ±0.05	**39.70** ±0.04	**52.50**±0.07

Table 15: Short-term prediction comparison on the **jumping** action from the CMU Mocap dataset.

scenarios	jumping				pretrain			
millisecond (ms)	80	160	320	400	80	160	320	400
Pretrain	-	-	-	-	**7.941** ±0.02	16.84 ±0.04	39.91 ±0.43	**52.45**±0.07
FT	-	-	-	-	26.67±0.10	50.83±0.29	94.13±0.58	112.66±0.71
PARTIAL-k	26.01±0.08	49.19 ±0.09	95.84±0.13	116.24 ±0.23	19.08±0.17	38.53±0.34	79.77±0.52	99.85±1.07
MLP-k	22.63/nan	44.68/nan	88.93/nan	108.43/nan	15.32±0.30	31.92±0.26	66.50±0.42	82.55±0.93
GPF	28.74±0.19	51.97±0.19	97.80±0.34	117.94 ±0.37	±0.08 -	-	-	-
GPF-plus	-	-	-	-	-	-	-	-
GeoAda	**25.91**±0.09	**48.83**±0.83	**91.51**±0.07	**109.24**±0.60	7.956 ±0.03	**16.82** ±0.04	**39.55** ±0.47	52.57 ±0.09

Table 16: Short-term prediction comparison on the **soccer** action from the CMU Mocap dataset.

scenarios	soccer				pretrain			
millisecond (ms)	80	160	320	400	80	160	320	400
Pretrain	-	-	-	-	**7.941**±0.02	**16.84**±0.04	39.91 ±0.43	52.45±0.07
FT	17.65 ±0.17	31.43 ±0.35	59.76±0.44	74.30 ±0.54	-	-	-	-
PARTIAL-k	18.83±0.10	32.86±0.07	64.58±0.37	81.53 ±0.50	14.14±0.41	24.95±0.38	50.89 ±0.40	64.24±0.51
MLP-k	-	-	-	-	-	-	-	-
GPF	19.28±0.10	32.18±0.14	69.58±0.42	73.63±0.63	15.70±0.30	28.31 ±0.28	58.57±0.42	74.28±0.61
GPF-plus	19.11±0.18	32.03±0.31	59.67±0.46	74.25 ±0.65	15.22±0.23	26.22±0.29	52.02±0.47	65.17±0.53
GeoAda	**17.04**±0.14	**30.03**±0.12	**53.51**±0.25	**64.78**±0.42	7.961 ±0.02	16.90±0.03	**39.46** ±0.41	**52.43** ±0.09

Table 17: Long-term prediction on CMU Mocap: **Running** and **Walking**.

scenarios	running		pretrain		walking		pretrain	
millisecond (ms)	560	1000	560	1000	560	1000	560	1000
Pretrain	219.16±2.18	314.85±3.03	**77.06** ±0.47	130.51 ±0.27	129.43 ±0.77	212.94 ±1.90	**77.06** ±0.47	130.51 ±0.27
Full FT	85.14 ±2.36	97.02±1.45	nan	nan	36.92±1.37	52.58±0.62	nan	nan
PARTIAL-k [12]	102.85 ±1.68	108.47±2.03	nan	nan	51.36±1.93	84.72±0.67	118.82±0.58	182.88±0.79
MLP-k	127.67±2.46	131.59±2.90	nan	nan	62.97±1.45	102.34±0.86	nan	nan
GPF [6]	61.92±1.02	71.42±1.27	nan	nan	42.37±0.31	52.24 ±0.38	119.43±0.60	171.74±0.55
GPF-plus [6]	63.56±1.54	71.60±0.95	nan	nan	41.31±0.35	56.47±0.4	nan	nan
GeoAda	**60.88** ±0.82	**70.22**±2.02	77.22 ±0.45	**130.17** ±0.26	**34.52** ±2.26	**50.49** ±0.33	78.12 ±0.49	**129.97** ±0.30

https://doi.org/10.52202/085713-3172

Table 18: Long-term prediction on CMU Mocap: **Jumping** and **Soccer**.

scenarios	jumping		pretrain		soccer		pretrain	
millisecond (ms)	560	1000	560	1000	560	1000	560	1000
Pretrain	–	–	77.06±0.47	**130.51**±0.27	–	–	**77.06**±0.47	130.51±0.27
Full FT	–	–	145.76±1.23	199.34±2.01	101.44 ±0.51	157.11±0.96	–	–
PARTIAL-k [12]	149.06±0.37	181.52±0.81	135.25±0.92	186.16±1.58	113.88 ±0.75	170.50±1.98	89.63 ±1.62	142.41±1.84
MLP-k	140.11/nan	184.77/nan	111.71±0.74	158.55±1.37	–	–	–	–
GPF [6]	151.50±0.65	194.94±0.34	–	–	100.23±0.88	151.84±1.23	104.03±0.81	161.56±1.31
GPF-plus [6]	–	–	–	–	101.15 ±0.86	153.43 ±0.79	90.12 ±1.10	142.30±1.12
GeoAda	139.46 ±0.29	184.01±0.60	**76.98**±0.51	130.72±0.26	**84.91**±0.46	**125.91** ±0.75	77.19 ±0.48	**130.06**±0.31

Table 19: Long-term prediction on CMU Mocap: **Basketball**.

scenarios	basketball		pretrain	
millisecond (ms)	560	1000	560	1000
Pretrain	143.49±1.19	223.99±2.23	77.06±0.47	130.51±0.27
Full FT	94.59±0.58	132.34±1.30	nan	nan
PARTIAL-k [12]	106.84 ±1.00	146.27 ±1.24	nan	nan
MLP-k	107.30±2.76	149.58±2.24	nan	nan
GPF [6]	97.16±0.35	128.29±0.43	nan	nan
GPF-plus [6]	104.54 ±0.26	130.76±1.28	104.51±0.57	155.02 ±0.51
GeoAda	**91.03**±0.33	**120.35**±0.44	**76.94**±0.45	**129.81**±0.30

10.3 Ablations

10.3.1 Parameter efficiency analysis

As shown in Table 20, we explore the impact of varying the number of equivariant adapter blocks. Increasing the number of trainable copy layers generally improves performance, but introduces more parameters and computational cost, revealing a trade-off between performance and efficiency. We have computed the number of tunable parameters for all baselines and GeoAda. The statistics are presented in Table 21. Except for Full Fine-Tuning, all methods have a comparable number of tunable parameters, ensuring a fair comparison.

Table 20: Different numbers of adapter blocks

Number	ADE	FDE
1	1.321	3.088
2	1.291	2.968
3	**1.108**	**2.621**
4	1.104	2.588
5	1.106	2.686

Table 21: The number of tunable parameters for different tuning strategies.

Dataset	Tuning Strategy	Total Parameters	Tunable Parameters
Charged Particle	Full FT	1418252 ~5.41 MB	1418252 ~5.41 MB
	PARTIAL-k	1418252 ~5.41 MB	711302 ~2.71 MB
	MLP-k	2125202 ~8.11 MB	711302 ~2.71 MB
	Prompt-Tem	1418252 ~5.41 MB	711302 ~2.71 MB
	GPF	2125266 ~8.11 MB	711366 ~2.71 MB
	GPF-plus	2125847 ~8.11 MB	711947 ~2.72 MB
	GeoAda	2125691 ~8.11 MB	711791 ~2.72 MB
MD17	Full FT	1424268 ~5.43 MB	1424268 ~5.43 MB
	PARTIAL-k	1424268 ~5.43 MB	716166 ~2.73 MB
	MLP-k	2132370 ~8.13 MB	716166 ~2.73 MB
	Prompt-Tem	1424268 ~5.43 MB	716166 ~2.73 MB
	GPF	2132434 ~8.13 MB	716230 ~2.73 MB
	GPF-plus	2135079 ~8.14 MB	718875 ~2.74 MB
	GeoAda	2132844 ~8.14 MB	716640 ~2.73 MB
CMU Mocap	Full FT	368012 ~1.40 MB	368012~1.40 MB
	PARTIAL-k	368012 ~1.40 MB	185990~0.71 MB
	MLP-k	550034 ~2.10 MB	185990~0.71 MB
	GPF	550098 ~2.10 MB	186054 ~0.71 MB
	GPF-plus	553259 ~2.11 MB	189215 ~0.72 MB
	GeoAda	550075 ~2.10 MB	186031~0.71 MB

10.3.2 Ablative Architectures

We study the following ablative architectures as shown in Figure 6, Figure 7, Figure 8:

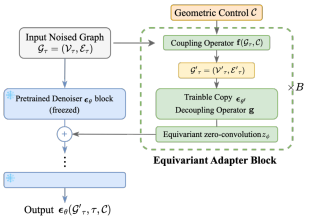

Figure 6: GeoAda

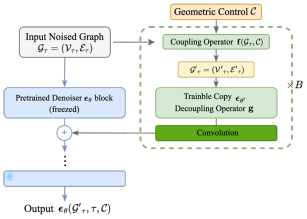

Figure 7: w/o zero convolution

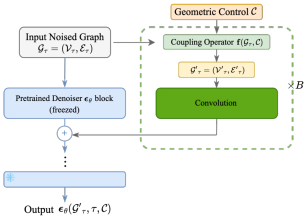

Figure 8: w/o trainable copy

Proposed GeoAda. The proposed architecture in the main paper.

Without Zero Convolution. Replacing the zero convolutions with standard convolution layers initialized with Gaussian weights.

Lightweight Layers. This architecture does not use a trainable copy, and directly initializes single convolution layers.

Results We present the results of this ablative study in Table 22, Table 23, Table 24 and Table 25.

Table 22: Comparisons for Molecular Dynamics prediction on MD17 dataset (all results reported by $\times 10^{-1}$). The best results are highlighted in bold. Results averaged over 5 runs

Scenarios	Aspirin				Benzene			
Task	Downstream		Pretrain		Downstream		Pretrain	
Metric	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE
w/o trainable copy	2.232 ±0.008	3.501 ±0.022	2.197 ±0.007	3.449±0.010	0.607±0.002	0.952±0.010	0.584±0.002	0.948±0.006
w/o zero conv	0.929 ±0.002	1.619±0.009	1.203±0.009	1.975±0.018	0.214±0.001	0.359±0.004	0.291±0.001	0.469±0.007
GeoAda	**0.891** ±0.003	**1.533** ±0.008	**1.060**±0.003	**1.852** ±0.012	**0.191** ±0.000	**0.319**±0.002	**0.240**±0.002	**0.394**±0.005

Table 23: Ablation study of Short-term prediction on **running** from the CMU Mocap dataset.

scenarios	**running**				**pretrain**			
millisecond (ms)	80	160	320	400	80	160	320	400
w/o trainable copy	44.52±0.48	76.98±1.20	134.91±2.04	159.75±2.45	198.16±2.97	139.24±0.23	270.04±0.56	314.89±0.68
w/o zero conv	19.07±0.37	34.25±0.82	51.75±1.39	55.74±1.53	nan	nan	nan	nan
GeoAda	**18.70** ±0.37	**33.56**±0.25	**50.26**±0.42	**55.54**±0.36	**7.972**±0.02	**16.91**±0.05	**40.06**±0.42	**52.61**±0.07

Table 24: Ablation study of Short-term prediction on **walking** from the CMU Mocap dataset.

scenarios	**walking**				**pretrain**			
millisecond (ms)	80	160	320	400	80	160	320	400
w/o trainable copy	23.77±0.15	43.43±0.31	84.79±0.77	105.08±1.28	nan	nan	nan	nan
w/o zero conv	12.62±0.07	20.67±0.20	36.75±0.52	44.69±0.45	36.18±0.12	58.18±0.08	102.06±0.05	123.82±0.04
GeoAda	**8.92**±1.02	**13.82**±1.26	**22.99**±1.30	**26.68**±1.31	**7.932**±0.03	**16.90**±0.04	**38.96**±0.47	**52.54**±0.10

Table 25: Ablation study of long-term prediction on **running, walking** from the CMU Mocap dataset.

scenarios	**running**		**pretrain**		**walking**		**pretrain**	
millisecond (ms)	560	1000	560	1000	560	1000	560	1000
w/o trainable copy	198.16±2.97	228.41±2.96	365.41±0.63	374.74±0.48	143.35±2.26	214.78±2.82	nan	nan
w/o zero conv	64.69±0.97	74.64±0.66	nan	nan	60.28±1.07	93.99±1.41	165.86±0.15	249.90±0.38
GeoAda	**60.88** ±0.82	**70.22**±2.02	77.22 ±0.45	**130.17** ±0.26	**34.52** ±2.26	**50.49** ±0.33	78.12 ±0.49	**129.97** ±0.30

https://doi.org/10.52202/085713-3172

10.4 Standard Deviations

We have already provided the standard deviations in App. 10.2.

11 Discussion

Limitation While GeoAda demonstrates strong empirical performance and theoretical grounding in preserving SE(3)-equivariance during fine-tuning, several limitations remain: The effectiveness of GeoAda hinges on the design of coupling and decoupling operators for control injection. While theoretically sound, these handcrafted designs may not generalize well to control signals with high-dimensional or structured semantics. Moreover, although GeoAda is validated across multiple domains (e.g., particles, molecules, human motion), the evaluations are limited to medium-scale datasets and relatively small models. Assessing its scalability to larger systems—such as full proteins or macromolecular assemblies—remains an important direction for future work.