

# SEGNO: GENERALIZING EQUIVARIANT GRAPH NEURAL NETWORKS WITH PHYSICAL INDUCTIVE BIASES

Yang Liu<sup>1,2\*</sup>, Jiashun Cheng<sup>1,2\*</sup>, Haihong Zhao<sup>1</sup>, Tingyang Xu<sup>3</sup>, Peilin Zhao<sup>3</sup>, Fugee Tsung<sup>1,2</sup>, Jia Li<sup>1,2†</sup>, Yu Rong<sup>3†</sup>

<sup>1</sup>The Hong Kong University of Science and Technology (Guangzhou)

<sup>2</sup>The Hong Kong University of Science and Technology <sup>3</sup>Tencent AI Lab

## ABSTRACT

Graph Neural Networks (GNNs) with equivariant properties have emerged as powerful tools for modeling complex dynamics of multi-object physical systems. However, their generalization ability is limited by the inadequate consideration of physical inductive biases: (1) Existing studies overlook the continuity of transitions among system states, opting to employ several discrete transformation layers to learn the direct mapping between two adjacent states; (2) Most models only account for first-order velocity information, despite the fact that many physical systems are governed by second-order motion laws. To incorporate these inductive biases, we propose the **Second-order Equivariant Graph Neural Ordinary Differential Equation (SEGNO)**. Specifically, we show how the second-order continuity can be incorporated into GNNs while maintaining the equivariant property. Furthermore, we offer theoretical insights into SEGNO, highlighting that it can learn a unique trajectory between adjacent states, which is crucial for model generalization. Additionally, we prove that the discrepancy between this learned trajectory of SEGNO and the true trajectory is bounded. Extensive experiments on complex dynamical systems including molecular dynamics and motion capture demonstrate that our model yields a significant improvement over the state-of-the-art baselines.

## 1 INTRODUCTION

Equivariant Graph Neural Networks (Equiv-GNNs) (Satorras et al., 2021; Han et al., 2022b; Brandstetter et al., 2021; Huang et al., 2022; Wu et al., 2024) have emerged as essential tools for simulating the multi-object physical system, i.e., N-body systems, which is relevant to numerous fundamental scientific domains, including molecular dynamics (Karplus & McCammon, 2002), protein folding (Gligorijević et al., 2021), robot motion planning/control (Siciliano et al., 2009). Specifically, given the input state, they learn to predict the output state after a specific timestep. To achieve these, Equiv-GNNs model the whole system as a geometric graph, which treats physical objects as nodes, and physical relations as edges, and encode the symmetry into a message-passing network to ensure their outputs are equivariant with respect to any translation/orientation/reflection of the inputs. Such property makes them well-suited for learning the unknown dynamics of complex physical systems that cannot be described analytically (Han et al., 2022a;b).

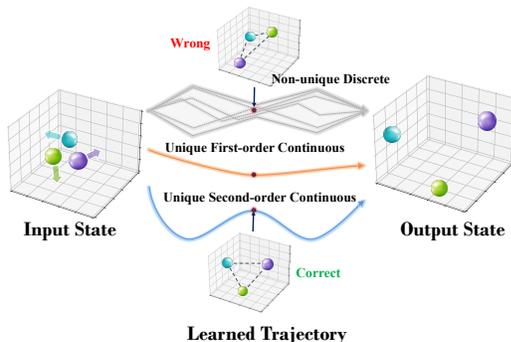


Figure 1: Learned trajectories of models with different inductive bias. All models can map input to output. However, discrete and first-order continuous models fail to learn the true intermediate states due to the lack of considering continuity and second-order laws.

\*Equal contribution. Work is done when Yang Liu and Jiashun Cheng worked as interns in Tencent AI Lab.

†Corresponding authors (jialiee@ust.hk, yu.rong@hotmail.com)

Learning to model such interacting systems is challenging. Given the vast parameter space of GNNs and finite observations of transitions of system states, there would exist multiple solutions that satisfy observed data (Curry & Morgan, 2006). Therefore, learning the real dynamic function from these solutions is crucial to model generalization ability (Gruver et al. (2022)). Though Equiv-GNNs (Han et al., 2024) have partially addressed this challenge by eliminating models that lack symmetry, existing Equiv-GNNs have yet to incorporate sufficient physical inductive bias to model the physical dynamics.

In particular, two essential inductive biases have not been well investigated in this field. First, existing models (Satorras et al., 2021; Brandstetter et al., 2021) are composed of a sequence of discrete state transformation layers, which learn a direct-mapping between adjacent states with discrete trajectories. We refer to them as *discrete models*. They are inconsistent with the continuous nature of system trajectories and fail to learn correct intermediate states. Second, most models only account for first-order information. Many physical dynamical systems, such as Newton’s equations of motion, are governed by second-order laws (Norcliffe et al., 2020). Therefore, these methods learn incomplete representations of the system’s state and fail to capture the underlying dynamics of the physical systems. In Figure 1, we illustrate the comparison of learned trajectories of models with different types of inductive bias.

In this work, we take a deep insight into the continuity and second-order inductive bias in Equiv-GNNs and propose a framework dubbed **Second-order Equivariant Graph Neural Ordinary Differential Equation (SEGNO<sup>1</sup>)**. Different from previous models that use Equiv-GNNs to fit discrete kinematic states, SEGNO introduces Neural Ordinary Differential Equations (Neural ODE) to approximate a continuous trajectory between two observed states. Furthermore, to better estimate the underlying dynamics, SEGNO is built upon second-order motion equations to update the position and velocity of the physical systems. Theoretically, we prove the uniqueness of the learned latent trajectory of SEGNO and further provide an upper bound on the discrepancy between the learned and the actual latent trajectory. Meanwhile, we prove that SEGNO can maintain equivariance properties identical to the backbone Equiv-GNNs. This property offers the flexibility to adapt various backbones in SEGNO to suit different downstream tasks in plug-and-play manner. We conduct extensive experiments on both synthetic and real-world physical systems. Our results reveal that SEGNO has a better generalization ability over the state-of-the-art baselines and second-order inductive bias is beneficial to learn complex multi-object dynamics.

## 2 BACKGROUND

**N-body System** We study N-body systems (Kipf et al., 2018; Huang et al., 2022) with a set of  $N$  particles  $\mathcal{P} = \{P_i\}_{i=1}^N$ . At time  $t$ , the state of each particle in the system is represented by:

1. geometric features including the position vector  $\mathbf{q}_i^{(t)} \in \mathbb{R}^3$  and the velocity vector  $\dot{\mathbf{q}}_i^{(t)} \in \mathbb{R}^3$ ,
2. non-geometric features such as mass or charge, denoted by  $\mathbf{h}_i \in \mathbb{R}^d$ ,
3. spatial connection with others, where an edge  $e_{ij}$  is constructed via geometric distance cutoff or physical interaction (e.g., chemical bonds) and the edge attributes (e.g., object distances, bond type) are denoted by  $a_{ij}$ .

For simplicity, we denote  $(\mathbf{q}^{(t)}, \dot{\mathbf{q}}^{(t)})$  and  $(\mathbf{h}, \mathbf{e} = \{e_{ij}\}, \mathbf{a} = \{a_{ij}\})$  as dynamic and static state information at system level correspondingly. In this work, we focus on dynamical systems that can be formulated as:

$$\ddot{\mathbf{q}}^{(t)} = \frac{d^2 \mathbf{q}^{(t)}}{dt^2} = f(\mathbf{q}^{(t)}, \mathbf{h}), \quad (1)$$

where  $\ddot{\mathbf{q}}^{(t)}$  is the acceleration at time  $t$ . Given a trajectory  $\mathbf{q}^2$  with the initial system states  $(\mathbf{q}^{(t_0)}, \dot{\mathbf{q}}^{(t_0)})$  at time  $t_0$  and static states  $(\mathbf{h}, \mathbf{e}, \mathbf{a})$ , our goal is to predict the subsequent position  $\mathbf{q}^{(t_1)}$  within a fixed time interval  $T = t_1 - t_0$ .

**$E(3)$  Equivariance** In 3-dimensional Euclidean space, the laws of physics remain invariant regardless of  $E(3)$  transformations, including translation, rotation, and reflection. Formally, a function  $\mathcal{F} : \mathbf{X} \times \mathbf{P} \rightarrow \mathbf{Y}$ , where  $\mathbf{X}, \mathbf{Y} \subset \mathbb{R}^3$ , is  $E(3)$ -equivariant, if for any transformation  $o \in E(3)$ ,

$$\mathcal{F}(o \circ \mathbf{x}, \dots) = o \circ \mathcal{F}(\mathbf{x}, \dots), \quad \mathbf{x} \in \mathbf{X}. \quad (2)$$

<sup>1</sup>SEGNO is also a musical term in Italian meaning “sign”, marking the beginning or end of a musical repeat.

<sup>2</sup>To avoid ambiguity,  $\mathbf{q}_i^{(t)}$  denotes the position of the  $i$ -th particle at time  $t$ , whereas  $\mathbf{q}$  denotes the trajectory over the entire time interval.

**Equivariant GNNs** In general, given the system state at time  $t_0$ , a modern Equiv-GNN  $\eta_\theta$  with parameters  $\theta$  directly predicts  $\mathbf{q}^{(t_1)}$  by leveraging several message passing layers on an interaction graph. To maintain brevity and avoid ambiguity, we omit the temporal superscript  $t$  here since the predictions at different times share the same model  $\eta_\theta$ . Specifically, each layer of  $\eta_\theta$  computes

$$\mathbf{m}_{ij} = \mu(\mathbf{q}_i, \mathbf{q}_j, \dot{\mathbf{q}}_i, \dot{\mathbf{q}}_j, \mathbf{h}_i, \mathbf{h}_j, a_{ij}), \quad \mathbf{q}'_i, \dot{\mathbf{q}}'_i, \mathbf{h}'_i = \nu(\mathbf{q}_i, \dot{\mathbf{q}}_i, \mathbf{h}_i, \sum_{j \in \mathcal{N}_i} \mathbf{m}_{ij}), \quad (3)$$

where  $\mu$  and  $\nu$  are the edge message function and node update function, respectively,  $\mathbf{m}_{ij}$  defines the message between node  $i$  and  $j$ .  $\mathcal{N}_i$  collects the neighbors of node  $i$ . The prediction is obtained by applying several iterations of message passing. To construct equivariant layers,  $\mu$  and  $\nu$  could be both equivariant (e.g., SEGNN (Brandstetter et al., 2021)) or alternatively,  $\mu$  is equivariant and  $\nu$  is invariant (e.g., EGNN (Satorras et al., 2021) and GMN (Huang et al., 2022)).

### 3 SEGNO FRAMEWORK

In this section, we introduce how the proposed SEGNO works and its equivariant properties. In our dynamic systems, we incorporate the ODE formulation to model the latent continuous trajectory with initial system states  $(\mathbf{q}^{(t_0)}, \dot{\mathbf{q}}^{(t_0)})$ . The position  $\mathbf{q}^{(t_0+t')}$  for any  $t' \in [0, T]$  is calculated as

$$\begin{aligned} \phi_{t',g}(\mathbf{q}^{(t_0)}) &:= \mathbf{q}^{(t_0+t')} = \mathbf{q}^{(t_0)} + \int_{t_0}^{t_0+t'} (\dot{\mathbf{q}}^{(t_0)} + \int_{t_0}^t f(\mathbf{q}^{(m)}, \mathbf{h}) dm) dt \\ &= \mathbf{q}^{(t_0)} + \int_{t_0}^{t_0+t'} g(\mathbf{q}^{(t)}, \mathbf{h}) dt, \end{aligned} \quad (4)$$

where  $g$  represents a mapping from the trajectory to its first-order derivative  $\dot{\mathbf{q}}^{(t)}$ . In this vein, we can denote  $\mathbf{q}^{(t_1)}$  as  $\phi_{T,g}(\mathbf{q}^{(t_0)})$ . Note that most physical dynamical systems follow the second-order motion law. The velocity  $\dot{\mathbf{q}}^{(t_0+t')}$  for any  $t' \in [0, T]$  can further be computed as

$$\psi_{t',g,f}(\mathbf{q}^{(t_0)}) := \dot{\mathbf{q}}^{(t_0+t')} = g(\mathbf{q}^{(t_0)}, \mathbf{h}) + \int_{t_0}^{t_0+t'} f(\mathbf{q}^{(t)}, \mathbf{h}) dt. \quad (5)$$

To incorporate the second-order inductive bias, SEGNO parameterizes the acceleration function

$$\ddot{\mathbf{q}}_\theta^{(t)} = f_\theta(\mathbf{q}^{(t)}, \mathbf{h}), \quad (6)$$

where  $f_\theta$ , an approximation of  $f$ , represents an Equiv-GNN with parameters  $\theta$  which computes:

$$\mathbf{m}_{ij}^{(t)} = \mu(\mathbf{q}_i^{(t)}, \mathbf{q}_j^{(t)}, \mathbf{h}_i, \mathbf{h}_j, a_{ij}), \quad \ddot{\mathbf{q}}_{\theta,i}^{(t)}, \mathbf{h}_i = \nu(\mathbf{q}_i^{(t)}, \mathbf{h}_i, \sum_{j \in \mathcal{N}_i} \mathbf{m}_{ij}^{(t)}), \quad (7)$$

where  $\mu$  and  $\nu$  are determined by this Equiv-GNN backbone. Let  $\mathbf{q}_\theta$  denote the approximated trajectory generated by SEGNO satisfying the initial conditions  $\mathbf{q}_\theta^{(t_0)} = \mathbf{q}^{(t_0)}, \dot{\mathbf{q}}_\theta^{(t_0)} = \dot{\mathbf{q}}^{(t_0)}$ . Then, following Eq. 4, the predicted position of SEGNO at time  $t_1$  can be represented by

$$\begin{aligned} \phi_{T,g_\theta}(\mathbf{q}^{(t_0)}) &= \mathbf{q}_\theta^{(t_1)} = \mathbf{q}^{(t_0)} + \int_{t_0}^{t_0+T} g_\theta(\mathbf{q}^{(t)}, \mathbf{h}) dt \\ &= \mathbf{q}_\theta^{(t_0)} + \int_{t_0}^{t_0+T} (\dot{\mathbf{q}}_\theta^{(t_0)} + \int_{t_0}^t f_\theta(\mathbf{q}^{(m)}, \mathbf{h}) dm) dt \\ &= \mathbf{q}_\theta^{(t_0)} + \int_{t_0}^{t_0+T} \psi_{t,g_\theta,f_\theta}(\mathbf{q}_\theta^{(t_0)}) dt, \end{aligned} \quad (8)$$

where  $g_\theta$ , a parameterized version of  $g$ , is determined by the integral of the GNN  $f_\theta$  in Eq. 4. Since it is infeasible to directly calculate the integration in  $\phi_{T,g_\theta}(\mathbf{q}^{(t_0)})$  and  $\psi_{t,g_\theta,f_\theta}(\mathbf{q}_\theta^{(t_0)})$  with parameterized  $g_\theta, f_\theta$ , in SEGNO, we utilize an ODE solver to generate a discrete trajectory that serves as an approximation of the latent continuous trajectory. Specifically, we divide the entire time interval  $T$  into  $\tau$  equal sub-intervals with timestep  $\Delta t = T/\tau$ . We then denote  $\Psi_{\Delta t,g,f}$  and  $\Phi_{\Delta t,g}$  as the numerical integrators that approach  $\psi_{\Delta t,g,f}$  and  $\phi_{\Delta t,g}$  using the following equations

$$\begin{aligned} \dot{\mathbf{q}}_\theta^{(t+\Delta t)} &= g_\theta(\mathbf{q}_\theta^{(t+\Delta t)}, \mathbf{h}) = \Psi_{\Delta t,g_\theta,f_\theta}(\mathbf{q}_\theta^{(t)}) = \dot{\mathbf{q}}_\theta^{(t)} + \mathcal{G}_1(f_\theta(\mathbf{q}_\theta^{(t)}, \mathbf{h}), \Delta t), \\ \mathbf{q}_\theta^{(t+\Delta t)} &= \Phi_{\Delta t,g_\theta}(\mathbf{q}_\theta^{(t)}) = \mathbf{q}_\theta^{(t)} + \mathcal{G}_2(\Psi_{\Delta t,g_\theta,f_\theta}(\mathbf{q}_\theta^{(t)}), \Delta t), \end{aligned} \quad (9)$$

where  $\mathcal{G}_1$  and  $\mathcal{G}_2$  are the increment functions that approximate the increment of a continuous integral given the initial value of the integrand and the integration width  $\Delta t$ . For instance, with the increment functions  $\mathcal{G}_1(x, y) = \mathcal{G}_2(x, y) = x \times y$ , the numerical integrators become the Euler integrators

$$\dot{\mathbf{q}}_{\theta}^{(t+\Delta t)} = \dot{\mathbf{q}}_{\theta}^{(t)} + f_{\theta}(\mathbf{q}_{\theta}^{(t)})\Delta t, \quad \mathbf{q}_{\theta}^{(t+\Delta t)} = \mathbf{q}_{\theta}^{(t)} + \dot{\mathbf{q}}_{\theta}^{(t+\Delta t)}\Delta t. \quad (10)$$

For approximation, SEGNO composites  $\tau$  integrator  $\Phi_{\Delta t, g_{\theta}}$  as a Neural ODE solver following

$$\mathbf{q}_{\theta}^{(t_1)} = \phi_{T, g_{\theta}}(\mathbf{q}^{(t_0)}) := \Phi_{\Delta t, g_{\theta}} \circ \dots \circ \Phi_{\Delta t, g_{\theta}}(\mathbf{q}^{(t_0)}) = (\Phi_{\Delta t, g_{\theta}})^{\tau}(\mathbf{q}^{(t_0)}). \quad (11)$$

As a result, SEGNO offers a way to reuse existing Equiv-GNNs to build second-order Neural ODEs.

However, a natural question arises: *does a Neural ODE solver compromise the equivariance of backbone GNNs?* To address this question, we show that the equivariance property of backbone GNNs could be maintained in SEGNO.

**Proposition 3.1.** *Suppose the backbone GNN  $f_{\theta}$  of SEGNO is  $O(3)$ -equivariant and translation-invariant, and the integrators' increment function  $\mathcal{G}_1, \mathcal{G}_2$  are  $O(3)$ -equivariant, then the output trajectory  $\mathbf{q}_{\theta}$  is  $E(3)$ -equivariant.*

The proof and example illustrations are provided in Appendix A.1, where we show that the general numerical integrators including symplectic Euler, Velocity Verlet, and Leapfrog satisfy Proposition 3.1. Meanwhile, since Equiv-GNNs such as EGNN (Satorras et al., 2021) and GMN (Huang et al., 2022) are built upon  $O(3)$ -equivariant and translation-invariant functions, SEGNO would preserve the same equivariant property as the backbone GNNs.

## 4 SEGNO ANALYSIS

Besides equivariance, another essential problem is how SEGNO learns from observed system states. In this section, we examine the approximation quality of SEGNO.

### 4.1 SOLUTION UNIQUENESS

It is known that continuous dynamics have a unique solution under specific continuous conditions, according to Picard's existence theorem (Coddington et al., 1956).

**Lemma 4.1.** *For the system  $\ddot{\mathbf{q}}^{(t)} = f(\mathbf{q}^{(t)}, \mathbf{h})$ , with given initial position  $\mathbf{q}^{(t_0)}$  and velocity  $\dot{\mathbf{q}}^{(t_0)}$ , if  $f$  is Lipschitz continuous, then this system has a unique solution  $\mathbf{q}^{(t)}$  over the interval  $t \in [t_0, t_1]$ .*

The proof is provided in Appendix A.2. In addition, under the SEGNO framework, we can obtain the following results:

**Proposition 4.2.** *Given the same conditions as in Lemma 4.1, if the realistic measurement on  $\mathbf{q}^{(t_1)}$  is given, there exists a  $f_{\theta^*}$  obtained by minimizing the discrepancy between  $\mathbf{q}_{\theta^*}^{(t_1)}$  and  $\mathbf{q}^{(t_1)}$ , such that  $f_{\theta^*}(\mathbf{q}_{\theta^*}^{(t)}, \mathbf{h}) = f(\mathbf{q}^{(t)}, \mathbf{h})$  holds over the interval  $t \in [t_0, t_1]$ .*

The detailed proof is given in Appendix A.3. This proposition remarks that SEGNO can be trained in line with prior studies (Satorras et al., 2021; Brandstetter et al., 2021). That is, given  $t_0$  and  $t_1$  as the input and target timesteps, SEGNO is trained to minimize the discrepancy between the exact and approximated positions:

$$\mathcal{L}_{\text{train}} = \sum_{s \in \mathcal{D}_{\text{train}}} \|\mathbf{q}_{\theta, s}^{(t_1)} - \mathbf{q}_s^{(t_1)}\|_2, \quad (12)$$

where  $\mathcal{D}_{\text{train}}$  denotes the training set. With a slight abuse of notation, here we denote  $\mathbf{q}_{\theta, s}^{(t_1)}, \mathbf{q}_s^{(t_1)}$  as the model prediction and actual position of trajectory  $s$ . If our learned model adequately approximates the system, Proposition 4.2 shows that it becomes possible for SEGNO to recover the latent trajectories of  $[t_0, t_1]$  between input and output system states via Neural ODE. In contrast, in the absence of continuous constraints, multiple discrete trajectories can exist between the input and output, making it challenging for discrete models to accurately learn the underlying dynamic functions.

## 4.2 APPROXIMATION ABILITY

In this section, we theoretically show that SEGNO is capable of learning a trajectory that remains bounded to the real solution. To achieve this, we first derive the boundness of the learned second-order derivative  $f_\theta$ . Existing theoretical findings (Zhu et al., 2022) have illustrated that training using Neural ODE solvers results in a bounded approximation of their first-order derivatives. In this work, we generalize this theorem to the second-order Neural ODE employed with Euler integrator.

### 4.2.1 BOUND OF ACCELERATION

To begin with, we first introduce relevant notations. Let  $\mathcal{B}(\mathbf{p}, r) \subset \mathbb{R}^3$  denotes a real ball of radius  $r > 0$  centered at  $\mathbf{p} \in \mathbb{R}^3$ , for a subset  $\mathcal{K} \subset \mathbb{R}^3$  and an analytic function  $\eta$ , we define

$$\mathcal{B}(\mathcal{K}, r) = \bigcup_{\mathbf{p} \in \mathcal{K}} \mathcal{B}(\mathbf{p}, r), \quad \|\eta\|_{\mathcal{K}} = \sup_{\mathbf{p} \in \mathcal{K}} \|\eta(\mathbf{p})\|_{\infty}. \quad (13)$$

We denote a set of points on the exact trajectory associated with the time increment interval  $[a, b]$  as

$$V_a^b = \{\phi_{t',g}(\mathbf{q}^{(t_0)}) | a \leq t' \leq b\}. \quad (14)$$

Given the time interval  $T = t_1 - t_0$  uniformly partitioned into  $\tau$  sub-intervals of timestep  $\Delta t$  (i.e.,  $t_0 < t_0 + \Delta t < \dots < t_1$ ), the following theorem holds:

**Theorem 4.3.** *For  $r_1, r_2 > 0$ , the ODE solver is  $\tau$  compositions of an Euler Integrator  $\Phi_{\Delta t, g_\theta}$  with  $\Delta t = T/\tau$ , and we denote the supremum norm of the approximation error for the ODE solver within  $\mathcal{B}(V_0^{2T}, r_1)$  as*

$$\mathcal{L}_0^{2T} := \frac{1}{T} \|(\Phi_{\Delta t, g_\theta})^\tau - \phi_{T,g}\|_{\mathcal{B}(V_0^{2T}, r_1)}.$$

*Suppose that the target  $f$  and the learned  $f_\theta$  are analytic and bounded by  $m_2$  on  $\mathcal{B}(V_0^{2T}, r_2)$ , and the target  $g$  and the learned  $g_\theta$  are analytic and bounded by  $m_1$  on  $\mathcal{B}(V_0^{2T}, r_1)$ . Then, there exist constants  $T_0$  such that, if  $0 < T < T_0$ ,  $\forall t \in [t_0, t_1]$ ,*

$$\|f_\theta(\mathbf{q}^{(t)}, \mathbf{h}) - f(\mathbf{q}^{(t)}, \mathbf{h})\|_{\infty} \leq O(\Delta t + \frac{\mathcal{L}_0^{2T}}{\Delta t}).$$

The proof is reported in Appendix A.4. The assumption that the true  $f, g$  and the learned  $f_\theta, g_\theta$  are analytic and bounded within specific balls is valid and grounded in real-world physical observations. Concretely, the trajectories of most dynamic physical systems exhibit smoothness, implying that there are no sudden changes in velocity or acceleration. This smooth property ensures that the system behaves in a physically realistic and predictable manner, which is a widely employed practice in control theory and dynamic systems analysis (Zhu et al., 2022).

### 4.2.2 BOUND OF TRAJECTORY

Based on Theorem 4.3, we can now analyze the error introduced by the Euler integrator. We introduce two metrics common in classical numerical analysis, namely, local and global truncation error (Poli et al., 2020). The local truncation error  $\epsilon_{t+\Delta t}$  of SEGNO is defined as:

$$\epsilon_{t+\Delta t} = \|\mathbf{q}^{(t+\Delta t)} - \mathbf{q}^{(t)} - \dot{\mathbf{q}}^{(t)} \Delta t - f_\theta(\mathbf{q}^{(t)}, \mathbf{h}) \Delta t^2\|_2, \quad (15)$$

which represents the error for a single timestep. The global truncation error  $\mathcal{E}_{t+k\Delta t}$  is defined as:

$$\mathcal{E}_{t+k\Delta t} = \|\mathbf{q}^{(t+k\Delta t)} - \mathbf{q}_\theta^{(t+k\Delta t)}\|_2, \quad (16)$$

which denotes the error accumulated in the first  $k$  steps. Then we have

**Corollary 4.4.** *Given the same conditions as in Theorem 4.3, if our learned model adequately approximates the system and  $g_\theta$  and  $f_\theta$  satisfy the Lipschitz continuity, then,  $\forall t \in [t_0, t_1]$  and  $k = 1, \dots, \tau$ , the local truncation error  $\epsilon_{t+\Delta t}$  and the global truncation error  $\mathcal{E}_{t+k\Delta t}$  are the order of  $O(\Delta t^2)$  and  $O(\Delta t)$ , respectively.*

The proof is reported in Appendix A.5. Corollary 4.4 shows that for the prediction of  $\tau$ -th iteration, its error depends on the chosen  $\Delta t$ . These statements imply that SEGNO can be trained by minimizing Eq. 12 and generalize to other timesteps.

Though there exist studies (Sanchez-Gonzalez et al., 2020; Bishnoi et al., 2022) that employ accelerations to train models, we remark that SEGNO is different from them and better. As Theorem 4.3 states, we aim to learn the **instantaneous** acceleration of each timestep by minimizing the position discrepancy, while previous works use the **average** acceleration to train the model and then adopt

semi-implicit Euler method to update the next state as well. The average acceleration is computed from the observed trajectory (i.e.,  $\bar{\mathbf{q}}^{(t_1)} = \mathbf{q}^{(t_2)} - 2\mathbf{q}^{(t_1)} + \mathbf{q}^{(t_0)}$ ). Thus, we can find that their local truncation error is  $O(T)$  even if their loss is zero, while that of SEGNO is  $O(\Delta t^2)(T = \tau\Delta t)$ . SEGNO theoretically achieves lower error without the need to obtain average acceleration by differentiating future positions.

**Empirical verification** To verify our theoretical findings, we train multiple models of EGNN (Satorras et al., 2021) and SEGNO with EGNN backbone, utilizing different random seeds on a 3-body system with two adjacent states  $\mathbf{q}^{(t_0)}, \mathbf{q}^{(t_1)}$ . We derive the intermediate state  $\mathbf{q}^{(t_{0.5})}$  from their intermediate layers/iterations, which serve as estimations of latent motion trajectories between the two states. In Figure 2, we visualize all predicted trajectories (in dotted grey line), the mean trajectory (in red line), and the mean and variance of predicted states. We can observe that EGNN predictions exhibit high error (red circle) and large variance (blue area) at the intermediate state  $\mathbf{q}^{(t_{0.5})}$ , indicating that the discrete models are incapable of learning the underlying real dynamics from observed system states. Conversely, the learned trajectory of SEGNO demonstrates a significantly lower error and small variance. The additional numerical comparisons are shown in Appendix C.1.

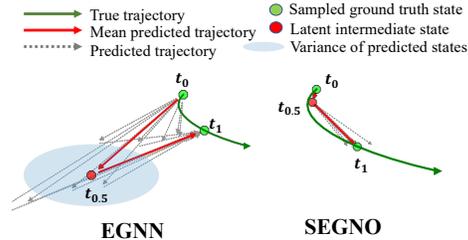


Figure 2: Illustration of learned trajectories from EGNN (left) and SEGNO (right). They are trained to predict the positions of N-body charged systems after 1000ts (See Section 5.1). The green, red, and dotted grey lines are the true, average, and predicted trajectories.  $t_0, t_1$  is the observed states.  $t_{0.5}$  is the predicted latent state. The blue area denotes the variance.

## 5 EXPERIMENTS

**Datasets** To validate the effectiveness of SEGNO, we first evaluate it on two simulated N-body systems, namely *Charged* particles and *Gravity* particles, which are driven by electromagnetic (Kipf et al., 2018) and gravitational forces (Brandstetter et al., 2021) between each pair of particles, respectively. Subsequently, we compare our model with state-of-the-art models in two challenging datasets: (1) **MD22** (Chmiela et al., 2023) which includes the trajectories of seven large and different types of molecules generated via molecular dynamics simulation; (2) **CMU motion capture** (CMU, 2003), which contains various trajectories of human motions in real-world. Note that all these dataset are symmetric, N-body systems and MD22 are  $E(3)$ -equivariant, and CMU motion capture is  $E(2)$ -equivariant<sup>3</sup>.

**Baselines** We compare SEGNO with various models: (1) Equivariant models including Radial Field (Köhler et al., 2019), TFN (Thomas et al., 2018), SE(3) Transformer (Fuchs et al., 2020), EGNN (Satorras et al., 2021), GMN (Huang et al., 2022), SEGNN (Brandstetter et al., 2021); (2) Non-equivariant models including GNN, Linear model, and Graph Neural Ordinary Differential Equation (GDE) (Poli et al., 2019).

### 5.1 SIMULATED N-BODY SYSTEM

**Implementation details** We build upon the experimental setting presented in (Satorras et al., 2021) where the task is to estimate all particle positions after a fixed timestep. Each system consists of 5 particles, each with initial positions, velocities, and attributes like positive/negative charge or mass. The graph is fully connected and the initial velocity norm is provided as additional input node features. We employ EGNN (Satorras et al., 2021) as the GNN backbone ( $f_\theta$ ) of SEGNO in these settings. For each system, besides the common setting (Satorras et al., 2021; Brandstetter et al., 2021) with 1000 timesteps (1000 ts), we introduce two extra targets—1500 ts and 2000 ts—which render the learning of latent trajectories more challenging. Other settings including the hyper-parameters are introduced in Appendix B.

<sup>3</sup>Symmetry is partially broken by gravity.

Table 1: Mean squared error ( $\times 10^{-2}$ ) of the N-body system. Bold font indicates the best result and underline is the strongest baseline. Results averaged across 5 runs. We report both mean and standard deviation in the table.

Method	Charged			Gravity		
	1000 ts	1500 ts	2000 ts	1000 ts	1500 ts	2000 ts
<b>Linear</b>	6.830 $\pm$ 0.016	20.012 $\pm$ 0.029	39.513 $\pm$ 0.061	7.928 $\pm$ 0.001	29.270 $\pm$ 0.003	58.521 $\pm$ 0.003
<b>GNN</b>	1.077 $\pm$ 0.004	5.059 $\pm$ 0.250	10.591 $\pm$ 0.352	1.400 $\pm$ 0.071	4.691 $\pm$ 0.288	10.508 $\pm$ 0.432
<b>GDE</b>	1.285 $\pm$ 0.074	4.026 $\pm$ 0.164	8.708 $\pm$ 0.145	1.412 $\pm$ 0.095	2.793 $\pm$ 0.083	6.291 $\pm$ 0.153
<b>TFN</b>	1.544 $\pm$ 0.231	11.116 $\pm$ 2.825	23.823 $\pm$ 3.048	3.536 $\pm$ 0.067	37.705 $\pm$ 0.298	73.472 $\pm$ 0.661
<b>SE(3)-Tr.</b>	2.483 $\pm$ 0.099	18.891 $\pm$ 0.287	36.730 $\pm$ 0.381	4.401 $\pm$ 0.095	52.134 $\pm$ 0.898	98.243 $\pm$ 0.647
<b>Radial Field</b>	1.060 $\pm$ 0.007	12.514 $\pm$ 0.089	26.388 $\pm$ 0.331	1.860 $\pm$ 0.075	7.021 $\pm$ 0.150	16.474 $\pm$ 0.033
<b>EGNN</b>	0.711 $\pm$ 0.029	2.998 $\pm$ 0.089	6.836 $\pm$ 0.093	0.766 $\pm$ 0.011	3.661 $\pm$ 0.055	9.039 $\pm$ 0.216
<b>GMN</b>	0.824 $\pm$ 0.032	3.436 $\pm$ 0.156	7.409 $\pm$ 0.214	0.620 $\pm$ 0.043	2.801 $\pm$ 0.194	6.756 $\pm$ 0.427
<b>SEGNN</b>	<u>0.448</u> $\pm$ 0.003	<u>2.573</u> $\pm$ 0.053	<u>5.972</u> $\pm$ 0.168	<u>0.471</u> $\pm$ 0.026	<u>2.110</u> $\pm$ 0.044	<u>5.819</u> $\pm$ 0.335
<b>SEGNO</b>	<b>0.433</b> $\pm$ 0.013	<b>1.858</b> $\pm$ 0.029	<b>4.285</b> $\pm$ 0.049	<b>0.338</b> $\pm$ 0.027	<b>1.362</b> $\pm$ 0.077	<b>4.017</b> $\pm$ 0.087

Table 2: Ablation studies ( $\times 10^{-2}$ ) on simulated N-body systems. Results averaged across 5 runs.

Order	Continuity	Charged			Gravity		
		1000 ts	1500 ts	2000 ts	1000 ts	1500 ts	2000 ts
First	Discrete	0.798 $\pm$ 0.099	2.215 $\pm$ 0.058	4.996 $\pm$ 0.148	0.466 $\pm$ 0.027	2.342 $\pm$ 0.424	5.501 $\pm$ 0.294
Second	Discrete	0.738 $\pm$ 0.026	2.125 $\pm$ 0.051	4.948 $\pm$ 0.198	0.362 $\pm$ 0.010	2.249 $\pm$ 0.591	5.343 $\pm$ 0.263
First	Continuous	<u>0.537</u> $\pm$ 0.045	<u>1.977</u> $\pm$ 0.056	<u>4.405</u> $\pm$ 0.159	<u>0.352</u> $\pm$ 0.024	<u>1.468</u> $\pm$ 0.040	<u>4.895</u> $\pm$ 0.102
Second	Continuous	<b>0.433</b> $\pm$ 0.013	<b>1.858</b> $\pm$ 0.029	<b>4.285</b> $\pm$ 0.049	<b>0.338</b> $\pm$ 0.027	<b>1.362</b> $\pm$ 0.077	<b>4.017</b> $\pm$ 0.087

**Results** Table 1 depicts the overall results of all models on two datasets. It is evident that SEGNO, equipped solely with EGNN, outperforms all baselines across all datasets and settings. Specifically, compared to the best baseline SEGNN, the average error improvement ( $\times 10^{-2}$ ) on Charged and Gravity datasets is 0.805 and 0.894 respectively, demonstrating significant improvement. Additionally, as timesteps increase, the performances of baselines largely drop while SEGNO still can model the latent trajectory. Thus, SEGNO’s performance improvement becomes more pronounced. For example, compared to the best baseline SEGNN, the relative improvement in Gravity datasets increases from 28.24% at 1000 time steps to 35.45% at 1500 time steps, demonstrating the strong generalization ability of SEGNO.

### 5.1.1 ABLATION STUDY

In this section, we conduct several ablation experiments on simulated N-body systems to scrutinize our model design, and provide empirical validation for our theoretical findings.

**Physical inductive biases in SEGNO** To validate the effect of inductive biases incorporated in SEGNO, we construct three variants of SEGNO, each featuring a unique combination of physical inductive biases. Table 2 reports the results where the term ‘First’ indicates that the model employ  $f_\theta$  to parameterize velocity rather than accelerations. ‘Discrete’ implies that SEGNO does not share the parameters across iterations, akin to discrete models. The original version of SEGNO is denoted as ‘Second’ and ‘Continuous’. From Table 2, we can observe that across all scenarios, continuous models consistently surpass discrete models and the second-order bias consistently enhances the performance of first-order models. These findings serve to corroborate the efficacy of SEGNO’s construction and further emphasize the significant advantages of integrating physical inductive biases into the learning process of dynamics.

**Effects of iteration times  $\tau$**  We further investigate the impact of the iteration number  $\tau$  in SEGNO to empirically verify Corollary 4.4. Given that the target time interval  $T$  remains constant, a larger iteration number  $\tau$  indicates a smaller interval  $\Delta t$ . The results are displayed in Figure 3. It is obvious that superior outcomes can be attained by opting for a smaller interval  $\Delta t$ . Additionally, the performance would not increase after reaching a sufficient number of iterations, which is approximately 10 for both datasets. As per Theorem 4.3, the plateau in performance improvement can be attributed to learning errors which is related to the representative ability of GNN backbone in SEGNO.

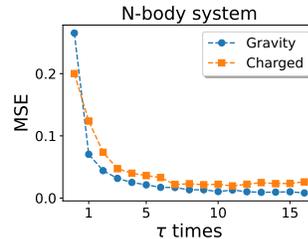


Figure 3: Effects of iteration number  $\tau$ .

Table 3: Mean squared error ( $\times 10^{-3}$ ) on MD22 dataset. Bold font indicates the best result and Underline is the strongest baseline. Results averaged across 3 runs.

Molecule	Type	# Atom	TFN	RF	EGNN	GMN	SEGNO
Ac-Ala3-NHMe	Protein	42	1.434 $\pm$ 0.716	1.340 $\pm$ 0.072	<u>0.539</u> $\pm$ 0.101	0.960 $\pm$ 0.157	<b>0.418</b> $\pm$ 0.038
DHA	Lipid	56	0.577 $\pm$ 0.075	1.487 $\pm$ 0.068	0.782 $\pm$ 0.035	<u>0.453</u> $\pm$ 0.125	<b>0.370</b> $\pm$ 0.020
AT-AT	Nucleic acid	60	1.407 $\pm$ 0.894	1.270 $\pm$ 0.067	0.583 $\pm$ 0.053	<u>0.568</u> $\pm$ 0.016	<b>0.440</b> $\pm$ 0.096
Stachyose	Carbohydrate	87	–	2.069 $\pm$ 0.074	0.629 $\pm$ 0.073	<u>0.583</u> $\pm$ 0.039	<b>0.548</b> $\pm$ 0.006
AT-AT-CG-CG	Nucleic acid	118	–	2.596 $\pm$ 1.282	0.609 $\pm$ 0.031	<u>0.581</u> $\pm$ 0.099	<b>0.394</b> $\pm$ 0.033
Buckyball Catcher	Supramolecule	148	–	0.440 $\pm$ 0.013	0.554 $\pm$ 0.093	<u>0.309</u> $\pm$ 0.092	<b>0.199</b> $\pm$ 0.020
Double-walled Nanotube	Supramolecule	370	–	0.382 $\pm$ 0.001	0.349 $\pm$ 0.261	<u>0.321</u> $\pm$ 0.065	<b>0.225</b> $\pm$ 0.008

Table 4: Mean squared error ( $\times 10^{-2}$ ) on CMU motion capture dataset. Bold font indicates the best result and Underline is the strongest baseline. Results averaged across 3 runs.

Model	TFN	SE(3)-Tr.	RF	EGNN	GMN	SEGNO	Abs. Imp.
30 ts	24.932 $\pm$ 1.023	24.655 $\pm$ 0.870	149.459 $\pm$ 0.750	24.013 $\pm$ 0.462	<u>16.005</u> $\pm$ 0.386	<b>14.462</b> $\pm$ 0.106	1.543
40 ts	49.976 $\pm$ 1.664	44.279 $\pm$ 0.355	306.311 $\pm$ 1.100	39.792 $\pm$ 2.120	<u>38.193</u> $\pm$ 0.697	<b>22.229</b> $\pm$ 1.489	15.964
50 ts	73.716 $\pm$ 4.343	68.796 $\pm$ 1.159	549.476 $\pm$ 3.461	50.930 $\pm$ 2.675	<u>47.883</u> $\pm$ 0.599	<b>29.264</b> $\pm$ 0.946	18.619

## 5.2 MOLECULAR DYNAMIC

**Implementation details** We use the atomic number and initial velocity norm as input node features. Two atoms are neighbors if their distance is less than a threshold. Our goal is to predict atom positions after 10 data frames. We run experiments on NVIDIA RTX A6000 GPU. TFN and SE(3)-Transformer run out of memory, thus we omit results of SE(3)-Transformer and part of TFN results. We use EGNN (Satorras et al., 2021) as the GNN backbone ( $f_\theta$ ) of SEGNO. Other settings including the hyper-parameters are introduced in Appendix B.

**Results** Table 3 summarizes the results of all models. It is evident that SEGNO outperforms the baselines across 7 molecules, even on the Double-walled Nanotube comprising 370 atoms, supporting the general effectiveness of encoding physical inductive biases. It is worth noting that SEGNO utilizes GMN as its backbone. In comparison to the original GMN, the errors are considerably diminished in all instances. The average relative improvement of SEGNO over GMN on 7 molecules is 15.6%. Such results demonstrate the effectiveness of enhancing physical inductive biases on equivariant GNNs in empirical applications.

## 5.3 CMU MOTION CAPTURE

**Implementation details** CMU Motion Capture (CMU, 2003) contains the trajectories of human motion under several scenarios. Following previous studies (Kipf et al., 2018), we focus on the walking motion of a single object (subject #35). The goal of this task is to predict the data frame after 30 timesteps. Similar to N-body systems, we broaden our assessment scope to include scenarios with intervals of 40 ts and 50 ts, in addition to the default settings with 30 ts. We use GMN as the backbone of SEGNO. The norm of velocity and the coordinates of the gravity axis (z-axis) are set as node features to represent the motion dynamics. Note that the human body operates through joint interactions, two joints are 1-hop neighbors if they are connected naturally and we augment the edges with 2-hop neighbors. Other settings are introduced in Appendix B.

**Results** Table 4 reports the performance of SEGNO and various compared models. We can observe that SEGNO outperforms all baseline models by a significant margin across all scenarios. Notably, the improvements are more pronounced in long-term simulations, with SEGNO achieving  $18.619 \times 10^{-2}$  lower MSE than the runner-up model GMN. To gain further insights into the superior performance of SEGNO, we illustrate the predicted motion of GMN and SEGNO in Figure 4.

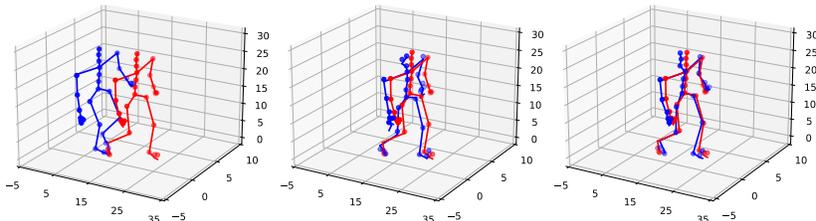


Figure 4: Visualization of Motion Capture with 50 ts. Left to Right: initial position, GMN, SEGNO (all in blue). Ground truths are in red.

Interestingly, it can be observed that the predictions of GMN appear to lag behind the ground truths, while SEGNO demonstrates a closer match. This discrepancy may be attributed to the lack of constraints imposed by modeling latent trajectories. We provide more visualizations in Appendix C.8.

## 6 RELATED WORK

Graph neural networks (Li et al., 2019; Tang et al., 2022; Gao et al., 2023; Tang et al., 2023) have shown promising performance in learning complex physical dynamics. IN (Battaglia et al., 2016), NRI (Kipf et al., 2018), and HRN (Mrowca et al., 2018) are pioneer works that model physical objects and relations as graphs and learn their interaction and evolution. Recently, researchers have taken into account the underlying physical symmetry of systems. TFN (Thomas et al., 2018) and SE(3) Transformer (Fuchs et al., 2020) employ spherical harmonics to construct models with 3D rotation equivariance in the Euclidean group for higher-order geometric representations. LieConv (Finzi et al., 2020a) and LieTransformer (Hutchinson et al., 2021) leverage the Lie convolution to extend equivariance on Lie groups. SEGNN (Brandstetter et al., 2021) proposes to use steerable vectors and their equivariant transformations to represent and process node features. In addition to these methods, a series of studies (Satorras et al., 2021; Huang et al., 2022; Schütt et al., 2021; Wang & Chodera, 2023) apply scalarization techniques to introduce equivariance into the message-passing process in GNNs. Nevertheless, these methods model dynamics in physical systems solely by learning direct mappings between discrete states. They ignore the second-order and continuous nature of observed trajectories, leading to suboptimal generalization performance.

Another research line (Thangamuthu et al., 2022) leverages ODE and Hamiltonian mechanics to capture the interactions in the systems such as Lagrangian Neural Networks (LNN) (Finzi et al., 2020b; Lutter et al., 2019; Bhattoo et al., 2023), Hamiltonian neural networks (HNN) (Greydanus et al., 2019), and Neural ODE (Chen et al., 2018; Gruver et al., 2022; Norcliffe et al., 2020). Recent studies have also enhanced these methods with GNNs. GDE (Poli et al., 2019) and HOGN (Sanchez-Gonzalez et al., 2019) combine GNNs with a differentiable ODE integrator. GNODE (Bishnoi et al., 2022) incorporates a graph-based Neural ODE with additional inductive biases, such as Newton’s third law. Compared with these works, we provide theoretical insights that show a second-order Graph Neural ODE can obtain bounded error of instantaneous acceleration and position through minimizing position discrepancy. These findings are further validated in complex applications including molecular and human motion dynamics. In particular, GNS (Sanchez-Gonzalez et al., 2020; Pfaff et al., 2021) also optimizes models via accelerations. However, they only learn the average accelerations that are calculated from the observed trajectories. In contrast, our study focuses on learning instantaneous accelerations and we show it theoretically achieves lower errors than GNS. Furthermore, it is worth mentioning that GNS does not consider equivariance, which is a critical inductive bias that captures symmetries in physical systems. We provide an experimental comparison between them in Appendix C.2.

Besides the above studies, a series of works have combined GNNs and the first-order Neural ODE to learn multi-agent systems (e.g., social networks), including CG-ODE (Huang et al., 2021), LG-ODE (Huang et al., 2020), and HOPE (Luo et al., 2023). However, due to the application difference, they do not consider equivariance and focus on historical state encoding. Thus, they are hard to extend to our task where equivariance and second-order information are vital.

## 7 CONCLUSION

In this work, we address the generalization problem of learning N-body systems and introduce SEGNO, which incorporates the equivariant property from GNN backbones and second-order physical inductive bias. We theoretically prove the uniqueness and boundedness of the trajectories inferred by SEGNO and empirically demonstrate the potential of SEGNO by applying it to a wide range of physical systems. Extensive ablation studies have further substantiated the generalization ability of SEGNO. For future works, we are interested in (1) extending our framework to solve stochastic (Salvi et al., 2022) and partial differential equations (Fortunato et al., 2022; Strönisch et al., 2023; Cao et al., 2023; Xue et al., 2023); (2) jointly considering trajectory forecasting and static tasks such as molecular property (Satorras et al., 2021) or force field predictions (Batzner et al., 2022); (3) integrating more sophisticated physical principles through advanced techniques such as pre-training (Rong et al., 2020; Cheng et al., 2023; Jiao et al., 2023), prompt tuning (Sun et al., 2023) or the utilization of large language models (Li et al., 2023).

## ACKNOWLEDGMENTS

The research of Li was supported by NSFC Grant No. 62206067, HKUST-HKUST(GZ) 20 for 20 Cross-campus Collaborative Research Scheme C019 and Guangzhou-HKUST(GZ) Joint Funding Scheme 2023A03J0673. The research of Tsung was supported by NSFC Grant No. 72371271 and Guangzhou Nansha District Key Project 2023ZD003. Particularly, Yu Rong expresses heartfelt gratitude to his wife, Yunman Huang, for the birth of their daughter Xing Rong, which brought immense joy during the pursuit of this research.

## REFERENCES

- Peter W Battaglia, Razvan Pascanu, Matthew Lai, Danilo Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. *arXiv preprint arXiv:1612.00222*, 2016.
- Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1):2453, 2022.
- Ravinder Bhattoo, Sayan Ranu, and N. M. Anoop Krishnan. Learning the dynamics of particle-based systems with lagrangian graph neural networks. *Mach. Learn. Sci. Technol.*, 4(1):15003, 2023.
- Suresh Bishnoi, Ravinder Bhattoo, Sayan Ranu, and NM Krishnan. Enhancing the inductive biases of graph neural ode for modeling dynamical systems. *arXiv preprint arXiv:2209.10740*, 2022.
- Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J Bekkers, and Max Welling. Geometric and physical quantities improve e (3) equivariant message passing. *arXiv preprint arXiv:2110.02905*, 2021.
- Yadi Cao, Menglei Chai, Minchen Li, and Chenfanfu Jiang. Efficient learning of mesh-based physical simulation with bi-stride multi-scale graph neural network. In *ICML*, volume 202, pp. 3541–3558, 2023.
- Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *NeurIPS*, pp. 6572–6583, 2018.
- Jiashun Cheng, Man Li, Jia Li, and Fugee Tsung. Wiener graph deconvolutional network improves graph self-supervised learning. In *AAAI*, pp. 7131–7139, 2023.
- Stefan Chmiela, Valentin Vassilev-Galindo, Oliver T Unke, Adil Kabylda, Huziel E Sauceda, Alexandre Tkatchenko, and Klaus-Robert Müller. Accurate global machine learning force fields for molecules with hundreds of atoms. *Science Advances*, 9(2):eadf0873, 2023.
- CMU. Carnegie-mellon motion capture database. 2003. URL <http://mocap.cs.cmu.edu>.
- Earl A Coddington, Norman Levinson, and T Teichmann. Theory of ordinary differential equations, 1956.
- Bruce Curry and Peter H Morgan. Model selection in neural networks: some difficulties. *European Journal of Operational Research*, 170(2):567–577, 2006.
- Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *ICML*, pp. 3165–3176. PMLR, 2020a.
- Marc Finzi, Ke Alexander Wang, and Andrew Gordon Wilson. Simplifying hamiltonian and lagrangian neural networks via explicit constraints. In *NeurIPS*, pp. 13880–13889, 2020b.
- Meire Fortunato, Tobias Pfaff, Peter Wirnsberger, Alexander Pritzel, and Peter W. Battaglia. Multi-scale meshgraphnets. *CoRR*, abs/2210.00612, 2022.
- Fabian B Fuchs, Daniel E Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d rotation equivariant attention networks. *arXiv preprint arXiv:2006.10503*, 2020.

- Ziqi Gao, Chenran Jiang, Jiawen Zhang, Xiaosen Jiang, Lanqing Li, Peilin Zhao, Huanming Yang, Yong Huang, and Jia Li. Hierarchical graph learning for protein–protein interaction. *Nature Communications*, 14(1):1093, 2023.
- Vladimir Gligorijević, P Douglas Renfrew, Tomasz Kosciolk, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C Taylor, Ian M Fisk, Hera Vlamakis, et al. Structure-based protein function prediction using graph convolutional networks. *Nature communications*, 12(1):3168, 2021.
- Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. In *NeurIPS*, pp. 15353–15363, 2019.
- Nate Gruver, Marc Anton Finzi, Samuel Don Stanton, and Andrew Gordon Wilson. Deconstructing the inductive biases of hamiltonian neural networks. In *ICLR*, 2022.
- Jiaqi Han, Wenbing Huang, Hengbo Ma, Jiachen Li, Josh Tenenbaum, and Chuang Gan. Learning physical dynamics with subequivariant graph neural networks. In *NeurIPS*, 2022a.
- Jiaqi Han, Wenbing Huang, Tingyang Xu, and Yu Rong. Equivariant graph hierarchy-based neural networks. In *NeurIPS*, 2022b.
- Jiaqi Han, Jiacheng Cen, Liming Wu, Zongzhao Li, Xiangzhe Kong, Rui Jiao, Ziyang Yu, Tingyang Xu, Fandi Wu, Ziheng Wang, Hongteng Xu, Zhewei Wei, Yang Liu, Yu Rong, and Wenbing Huang. A survey of geometric graph neural networks: Data structures, models and applications, 2024.
- Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.
- Wenbing Huang, Jiaqi Han, Yu Rong, Tingyang Xu, Fuchun Sun, and Junzhou Huang. Equivariant graph mechanics networks with constraints. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=SHbhHHfePhP>.
- Zijie Huang, Yizhou Sun, and Wei Wang. Learning continuous system dynamics from irregularly-sampled partial observations. In *NeurIPS*, volume 33, pp. 16177–16187, 2020.
- Zijie Huang, Yizhou Sun, and Wei Wang. Coupled graph ode for learning interacting system dynamics. In *KDD*, pp. 705–715, 2021.
- Michael J Hutchinson, Charline Le Lan, Sheheryar Zaidi, Emilien Dupont, Yee Whye Teh, and Hyunjik Kim. Lietransformer: equivariant self-attention for lie groups. In *ICML*, pp. 4533–4543. PMLR, 2021.
- Rui Jiao, Jiaqi Han, Wenbing Huang, Yu Rong, and Yang Liu. Energy-motivated equivariant pre-training for 3d molecular graphs. In *AAAI*, 2023. ISBN 978-1-57735-880-0.
- Martin Karplus and J Andrew McCammon. Molecular dynamics simulations of biomolecules. *Nature structural biology*, 9(9):646–652, 2002.
- Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. *arXiv preprint arXiv:1802.04687*, 2018.
- Jonas Köhler, Leon Klein, and Frank Noé. Equivariant flows: sampling configurations for multi-body systems with symmetric energies. *arXiv preprint arXiv:1910.00753*, 2019.
- Jia Li, Yu Rong, Hong Cheng, Helen Meng, Wenbing Huang, and Junzhou Huang. Semi-supervised graph classification: A hierarchical graph perspective. In *WWW*, pp. 972–982, 2019.
- Yuhan Li, Zhixun Li, Peisong Wang, Jia Li, Xiangguo Sun, Hong Cheng, and Jeffrey Xu Yu. A survey of graph meets large language model: Progress and future directions. *arXiv preprint arXiv:2311.12399*, 2023.
- Xiao Luo, Jingyang Yuan, Zijie Huang, Huiyu Jiang, Yifang Qin, Wei Ju, Ming Zhang, and Yizhou Sun. Hope: High-order graph ode for modeling interacting dynamics. In *ICML*, 2023.

- Michael Lutter, Christian Ritter, and Jan Peters. Deep lagrangian networks: Using physics as model prior for deep learning. In *ICLR*, 2019.
- Damian Mrowca, Chengxu Zhuang, Elias Wang, Nick Haber, Li Fei-Fei, Joshua B Tenenbaum, and Daniel LK Yamins. Flexible neural representation for physics prediction. *arXiv preprint arXiv:1806.08047*, 2018.
- Albert Musaelian, Simon Batzner, Anders Johansson, Lixin Sun, Cameron J Owen, Mordechai Kornbluth, and Boris Kozinsky. Learning local equivariant representations for large-scale atomistic dynamics. *Nature Communications*, 14(1):579, 2023.
- Alexander Norcliffe, Cristian Bodnar, Ben Day, Nikola Simidjievski, and Pietro Liò. On second order behaviour in augmented neural odes. In *NeurIPS*, volume 33, pp. 5911–5921, 2020.
- Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning mesh-based simulation with graph networks. In *ICLR*, 2021.
- Michael Poli, Stefano Massaroli, Junyoung Park, Atsushi Yamashita, Hajime Asama, and Jinkyoo Park. Graph neural ordinary differential equations. *arXiv preprint arXiv:1911.07532*, 2019.
- Michael Poli, Stefano Massaroli, Atsushi Yamashita, Hajime Asama, and Jinkyoo Park. Hyper-solvers: Toward fast continuous-depth models. In *NeurIPS*, volume 33, pp. 21105–21117, 2020.
- Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. *Advances in neural information processing systems*, 33:12559–12571, 2020.
- Cristopher Salvi, Maud Lemercier, and Andris Gerasimovics. Neural stochastic pdes: Resolution-invariant learning of continuous spatiotemporal dynamics. In *NeurIPS*, 2022.
- Alvaro Sanchez-Gonzalez, Victor Bapst, Kyle Cranmer, and Peter W. Battaglia. Hamiltonian graph networks with ODE integrators. *arXiv preprint arXiv:1909.12790*, 2019.
- Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *ICML*, pp. 8459–8468. PMLR, 2020.
- Victor Garcia Satorras, Emiel Hoogetboom, and Max Welling. E (n) equivariant graph neural networks. In *ICML*, pp. 9323–9332. PMLR, 2021.
- Kristof Schütt, Oliver Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *ICML*, pp. 9377–9388. PMLR, 2021.
- Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics*. Springer London, 2009. doi: 10.1007/978-1-84628-642-1. URL <https://doi.org/10.1007/978-1-84628-642-1>.
- Sebastian Strönisch, Maximilian Sander, Andreas Knüpfer, and Marcus Meyer. Multi-gpu approach for training of graph ML models on large CFD meshes. *CoRR*, abs/2307.13592, 2023.
- Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, and Jihong Guan. All in one: Multi-task prompting for graph neural networks. In *KDD*, pp. 2120–2131, 2023.
- Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li. Rethinking graph neural networks for anomaly detection. In *ICML*, pp. 21076–21089. PMLR, 2022.
- Jianheng Tang, Fengrui Hua, Ziqi Gao, Peilin Zhao, and Jia Li. Gadbench: Revisiting and benchmarking supervised graph anomaly detection. In *NeurIPS Datasets and Benchmarks Track*, 2023.
- Abishek Thangamuthu, Gunjan Kumar, Suresh Bishnoi, Ravinder Bhattoo, N M Anoop Krishnan, and Sayan Ranu. Unravelling the performance of physics-informed graph neural networks for dynamical systems. In *NeurIPS Datasets and Benchmarks Track*, 2022.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

Yuanqing Wang and John D Chodera. Spatial attention kinetic networks with e (n)-equivariance. *arXiv preprint arXiv:2301.08893*, 2023.

Liming Wu, Zhichao Hou, Jirui Yuan, Yu Rong, and Wenbing Huang. Equivariant spatio-temporal attentive graph networks to simulate physical dynamics. *NeurIPS*, 36, 2024.

Rui Xue, Haoyu Han, MohamadAli Torkamani, Jian Pei, and Xiaorui Liu. Lazygnn: Large-scale graph neural networks via lazy propagation. In *ICML*, volume 202, pp. 38926–38937, 2023.

Aiqing Zhu, Pengzhan Jin, Beibei Zhu, and Yifa Tang. On numerical integration in neural ordinary differential equations. In *ICML*, pp. 27527–27547, 2022.

## A PROOFS

Note that  $\mathbf{h}$  is identical for both real and learned systems. Unless otherwise specified, we omit  $\mathbf{h}$  in the following discussions.

### A.1 PROOF OF PROPOSITION 3.1

According to the constraints of the initial conditions, the initial position  $\mathbf{q}_\theta^{(t_0)}$  is  $E(3)$ -equivariant and the initial velocity  $\dot{\mathbf{q}}_\theta^{(t_0)}$  is  $O(3)$ -equivariant and translation-invariant. Given that backbone GNN  $f_\theta$  are  $O(3)$ -equivariant and translation-invariant, and  $\mathcal{G}_1$  is  $O(3)$ -equivariant, for any translation vector  $\mathbf{b} \in \mathbb{R}^3$ , orthogonal matrix  $\mathbf{A} \in \mathbb{R}^{3 \times 3}$  and  $t' \in [0, \Delta t]$ , we have

$$\begin{aligned} \Psi_{t', g_\theta, f_\theta}(\mathbf{A}\mathbf{q}_\theta^{(t_0)} + \mathbf{b}) &= g_\theta(\mathbf{A}\mathbf{q}_\theta^{(t_0)} + \mathbf{b}) + \mathcal{G}_1(f_\theta(\mathbf{A}\mathbf{q}_\theta^{(t_0)} + \mathbf{b}), t') \\ &= \mathbf{A}g_\theta(\mathbf{q}_\theta^{(t_0)}) + \mathcal{G}_1(\mathbf{A}f_\theta(\mathbf{q}_\theta^{(t_0)}), t') \\ &= \mathbf{A}\dot{\mathbf{q}}_\theta^{(t_0)} + \mathbf{A}\mathcal{G}_1(f_\theta(\mathbf{q}_\theta^{(t_0)}), t') \\ &= \mathbf{A}\dot{\mathbf{q}}_\theta^{(t_0+t')} = \mathbf{A}\Psi_{\Delta t, g_\theta, f_\theta}(\mathbf{q}_\theta^{(t_0)}). \end{aligned} \quad (17)$$

Thus,  $\Psi_{t', g_\theta, f_\theta}(\mathbf{q}_\theta^{(t_0)})$  is proved to be  $O(3)$ -equivariant and translation-invariant. Note that  $\mathcal{G}_2$  is  $O(3)$ -equivariant, then, for  $\mathbf{q}_\theta^{(t_0+t')}$ , we have

$$\begin{aligned} \Phi_{t', g_\theta}(\mathbf{A}\mathbf{q}_\theta^{(t_0)} + \mathbf{b}) &= \mathbf{A}\mathbf{q}_\theta^{(t_0)} + \mathbf{b} + \mathcal{G}_2(\Psi_{t', g_\theta, f_\theta}(\mathbf{A}\mathbf{q}_\theta^{(t_0)} + \mathbf{b}), t') \\ &= \mathbf{A}\mathbf{q}_\theta^{(t_0)} + \mathcal{G}_2(\mathbf{A}\Psi_{t', g_\theta, f_\theta}(\mathbf{q}_\theta^{(t_0)}), t') + \mathbf{b} \\ &= \mathbf{A}\mathbf{q}_\theta^{(t_0)} + \mathbf{A}\mathcal{G}_2(\Psi_{t', g_\theta, f_\theta}(\mathbf{q}_\theta^{(t_0)}), t') + \mathbf{b} \\ &= \mathbf{A}\mathbf{q}_\theta^{(t_0+t')} + \mathbf{b}. \end{aligned} \quad (18)$$

Therefore,  $\mathbf{q}_\theta^{(t_0+t')} = \Phi_{t', g_\theta}(\mathbf{q}_\theta^{(t_0)})$  is  $E(3)$ -equivariant. Note that the composition of  $E(3)$ -equivariant functions is still  $E(3)$ -equivariant. For any time  $t \in [t_0, t_1]$ ,  $\mathbf{q}_\theta^{(t)}$ , which is generated via iteratively calling integrator  $\Phi_{t', g_\theta}$  (Eq. 11) with suitable  $t' \in [0, \Delta t]$ , is  $E(3)$ -equivariant. Overall, the approximated trajectory  $\mathbf{q}_\theta$  is  $E(3)$ -equivariant.

In real-world applications, the assumption of this proposition is generally satisfied. In terms of the equivariance of backbone GNN  $f_\theta$ , for example, if  $f_\theta$  is EGNN whose message passing is defined by

$$\begin{aligned} \mathbf{m}_{ij}^{(t)} &= \phi_e \left( \|\mathbf{q}_{\theta,i}^{(t)} - \mathbf{q}_{\theta,j}^{(t)}\|^2, \mathbf{h}_i, \mathbf{h}_j, a_{ij} \right), \quad \ddot{\mathbf{q}}_{\theta,i}^{(t)} = \frac{1}{N-1} \sum_{j \in \mathcal{N}_i} (\mathbf{q}_{\theta,i}^{(t)} - \mathbf{q}_{\theta,j}^{(t)}) \phi_q(\mathbf{m}_{ij}^{(t)}). \\ \mathbf{h}_i^{(t+\Delta t)} &= \mathbf{h}_i^{(t)} + \phi_h(\mathbf{h}_i^{(t)}, \sum_{j \in \mathcal{N}_i} \mathbf{m}_{ij}^{(t)}). \end{aligned} \quad (19)$$

Here  $\phi_q$  denotes Multi-Layer Perceptron (MLP) whose output is a scalar and the output of  $\phi_e, \phi_h$  are vectors. The non-geometric features are updated via skip connections. Analogous to neural ODE methods, the model parameters are shared among all iterations. By (Satorras et al., 2021), it can be easily shown that  $f_\theta$  is  $O(3)$ -equivariant and translation-invariant.

In addition, the increment functions  $\mathcal{G}_1, \mathcal{G}_2$  of several widely used numerical integrators for motion dynamics are  $O(3)$ -equivariant, ensuring that the approximated trajectory is  $E(3)$ -equivariant. For example, a symplectic Euler integrator computes

$$\mathbf{q}^{(t+\Delta t)} = \mathbf{q}^{(t)} + \dot{\mathbf{q}}^{(t+\Delta t)} \Delta t, \quad \dot{\mathbf{q}}^{(t+\Delta t)} = \dot{\mathbf{q}}^{(t)} + \ddot{\mathbf{q}}^{(t)} \Delta t, \quad (20)$$

where  $\mathcal{G}_1(x, y) = \mathcal{G}_2(x, y) = x \times y$  is  $O(3)$ -equivariant. It is straightforward to show that

$$\mathbf{A}\dot{\mathbf{q}}^{(t)} + \mathbf{A}\ddot{\mathbf{q}}^{(t+\Delta t)} \Delta t = \mathbf{A}\dot{\mathbf{q}}^{(t+\Delta t)}, \quad \mathbf{A}\mathbf{q}^{(t)} + \mathbf{b} + \mathbf{A}\dot{\mathbf{q}}^{(t+\Delta t)} \Delta t = \mathbf{A}\mathbf{q}^{(t+\Delta t)} + \mathbf{b}. \quad (21)$$

This property also holds for Velocity Verlet

$$\mathbf{q}^{(t+\Delta t)} = \mathbf{q}^{(t)} + \dot{\mathbf{q}}^{(t)} \Delta t + \frac{1}{2} \ddot{\mathbf{q}}^{(t)} \Delta t^2, \quad \dot{\mathbf{q}}^{(t+\Delta t)} = \dot{\mathbf{q}}^{(t)} + \frac{1}{2} (\ddot{\mathbf{q}}^{(t)} + \ddot{\mathbf{q}}^{(t+\Delta t)}) \Delta t, \quad (22)$$

and Leapfrog

$$\mathbf{q}^{(t+\Delta t)} = \mathbf{q}^{(t)} + \dot{\mathbf{q}}^{(t)} \Delta t + \frac{1}{2} \ddot{\mathbf{q}}^{(t)} \Delta t^2, \quad \dot{\mathbf{q}}^{(t+\Delta t)} = \dot{\mathbf{q}}^{(t)} + \ddot{\mathbf{q}}^{(t+\Delta t)} \Delta t. \quad (23)$$

The proof is the same as symplectic Euler.

### A.2 PROOF OF LEMMA 4.1

We first prove the uniqueness of the solution in our dynamic system. The key to this proof is to convert the high-order non-linear ODE to a first-order non-linear ODE. Let's define a new variable  $\mathbf{Q}^{(t)} = (\dot{\mathbf{q}}^{(t)}, \mathbf{q}^{(t)})$  and  $\dot{\mathbf{Q}}^{(t)}$  as its first-order derivative with respect  $t$ . Then in terms of this new variable, the second-order non-linear ODE becomes

$$\dot{\mathbf{Q}}^{(t)} = (F(\mathbf{Q}^{(t)}), G(\mathbf{Q}^{(t)})), \quad (24)$$

where  $F(\mathbf{Q}^{(t)}) = f(\mathbf{q}^{(t)})$  and  $G(\mathbf{Q}^{(t)}) = \dot{\mathbf{q}}^{(t)}$ . Furthermore, let's define  $H(\mathbf{Q}^{(t)}) = (F(\mathbf{Q}^{(t)}), G(\mathbf{Q}^{(t)}))$ , and Eq. 24 is reformatted as

$$\dot{\mathbf{Q}}^{(t)} = H(\mathbf{Q}^{(t)}), \quad \mathbf{Q}^{(t_0)} = (\dot{\mathbf{q}}^{(t_0)}, \mathbf{q}^{(t_0)}), \quad (25)$$

which is exactly a first-order non-linear ODE with a known initial condition. Note that  $G$  and  $f$  are both Lipschitz continuous, thus  $H$  is Lipschitz continuous. Then, based on Picard's existence theorem (Coddington et al., 1956), the aforementioned non-linear ODE has a unique solution  $\mathbf{Q}^{(t)}$ ,  $\forall t \in [t_0, t_1]$ . Subsequently, our system has a unique solution  $\mathbf{q}^{(t)}$ ,  $\forall t \in [t_0, t_1]$ .

### A.3 PROOF OF PROPOSITION 4.2

Due to the uniqueness of the solution, if the realistic measurement  $\mathbf{q}^{(t_1)}$  is given, then the trajectory  $\mathbf{q}^{(t)}$  is fully determined over the time interval  $[t_0, t_1]$ . Under SEGNO framework, we define the discrepancy between  $\mathbf{q}_\theta^{(t_1)}$  and  $\mathbf{q}^{(t_1)}$  as

$$d(\mathbf{q}_\theta^{(t_1)}, \mathbf{q}^{(t_1)}) = \|\mathbf{q}_\theta^{(t_1)} - \mathbf{q}^{(t_1)}\|_p = \|(\Phi_{\Delta t, g_\theta})^\tau(\mathbf{q}^{(t_0)}) - \phi_{T, g}(\mathbf{q}^{(t_0)})\|_p, \quad (26)$$

where  $\|\cdot\|_p$  represents the  $p$ -norm. According to Eq. 9,  $\Phi_{\Delta t, g_\theta}$  is fully determined by  $f_\theta$ . Without loss of generality, we take SEGNO with Euler integrators as an example. Given the known initial position and velocity, along with neural ODE update scheme in SEGNO, we can convert Eq. 26 to the following form

$$\begin{aligned} d(\mathbf{q}_\theta^{(t_1)}, \mathbf{q}^{(t_1)}) &= \|\mathbf{q}_\theta^{(t_1)} - \mathbf{q}^{(t_1)}\|_p \\ &= \left\| \sum_{k=0}^{\tau-1} \left( \int_{t_0+k\Delta t}^{t_0+(k+1)\Delta t} (\dot{\mathbf{q}}^{(t_0+k\Delta t)} + \int_{t_0+k\Delta t}^t f(\mathbf{q}^{(m)}) dm) dt - (\dot{\mathbf{q}}_\theta^{(t_0+k\Delta t)} \Delta t + f_\theta(\mathbf{q}_\theta^{(t_0+k\Delta t)}) \Delta t^2) \right) \right\|_p \\ &= \left\| \sum_{k=0}^{\tau-1} \mathcal{D}(\dot{\mathbf{q}}^{(t_0+k\Delta t)}, f(\mathbf{q}^{(t_0+k\Delta t)}), \dot{\mathbf{q}}_\theta^{(t_0+k\Delta t)}, f_\theta(\mathbf{q}_\theta^{(t_0+k\Delta t)})) \right\|_p, \end{aligned} \quad (27)$$

where  $\mathcal{D}(\cdot)$  denotes the discrepancy function. Considering that there are no sudden changes in velocity or acceleration for a smooth trajectory, with sufficiently small  $\Delta t$ , we can approximate  $\mathcal{D}(\cdot)$  as follows

$$\begin{aligned} &\mathcal{D}(\dot{\mathbf{q}}^{(t_0+k\Delta t)}, f(\mathbf{q}^{(t_0+k\Delta t)}), \dot{\mathbf{q}}_\theta^{(t_0+k\Delta t)}, f_\theta(\mathbf{q}_\theta^{(t_0+k\Delta t)})) \\ &\approx (\dot{\mathbf{q}}^{(t_0+k\Delta t)} - \dot{\mathbf{q}}_\theta^{(t_0+k\Delta t)}) \Delta t + (f(\mathbf{q}^{(t_0+k\Delta t)}) - f_\theta(\mathbf{q}_\theta^{(t_0+k\Delta t)})) \Delta t^2 \\ &\approx \sum_{i=0}^k (f(\mathbf{q}^{(t_0+i\Delta t)}) - f_\theta(\mathbf{q}_\theta^{(t_0+i\Delta t)})) \Delta t^2. \end{aligned} \quad (28)$$

Note that Lemma 4.1 guarantees the uniqueness of target trajectory as well as its acceleration. Therefore, according to the aforementioned analysis, SEGNO tends to let  $f_\theta(\mathbf{q}_\theta^{(t_0+k\Delta t)})$  approximate the unique acceleration  $f(\mathbf{q}^{(t_0+k\Delta t)})$  via minimizing position loss. Then, under the assumption of widely used universal approximation theorem (Hornik et al., 1989), there exists a  $f_{\theta^*}$  such that  $f_{\theta^*}(\mathbf{q}_{\theta^*}^{(t)}, \mathbf{h}) = f(\mathbf{q}^{(t)}, \mathbf{h})$ ,  $\forall t \in [t_0, t_1]$ .

#### A.4 PROOF OF THEOREM 4.3

The proof sketch is as follows: We first revisit the core definitions pertaining to neural ODE in SEGNO and introduce its variant with Euler integrator, then derive the bound for first-order approximation error  $\|g_\theta - g\|_\infty$ , and finally extend the results to the second-order case  $\|f_\theta - f\|_\infty$  to finish the proof.

##### A.4.1 EULER INTEGRATOR IN SEGNO

As mentioned in Eq. 10, the Euler integrator  $\Phi_{\Delta t, g_\theta}$ , as mentioned in Eq. 20, approaches  $\phi_{\Delta t, g_\theta}$  via

$$\begin{aligned} g_\theta(\mathbf{q}_\theta^{(t+\Delta t)}) &= g_\theta(\mathbf{q}_\theta^{(t)}) + f_\theta(\mathbf{q}_\theta^{(t)})\Delta t, \\ \Phi_{\Delta t, g_\theta}(\mathbf{q}_\theta^{(t)}) &= \mathbf{q}_\theta^{(t)} + g_\theta(\mathbf{q}_\theta^{(t+\Delta t)})\Delta t. \end{aligned} \quad (29)$$

Considering Eq. 11 and 12, it is straightforward to reframe our learning objective in the context of a neural ODE:

$$\mathcal{L}_{\text{train}} = \sum_{s \in \mathcal{D}_{\text{train}}} \|\mathbf{q}_{\theta, s}^{(t_1)} - \mathbf{q}_s^{(t_1)}\|^2 = \sum_{s \in \mathcal{D}_{\text{train}}} \|(\Phi_{\Delta t, g_\theta})^\tau(\mathbf{q}_s^{(t_0)}) - \phi_{T, g}(\mathbf{q}_s^{(t_0)})\|^2. \quad (30)$$

##### A.4.2 APPROXIMATION ERROR OF $f_\theta$

In our dynamical system,  $g_\theta$  and  $g$  are entirely determined by  $f_\theta$  and  $f$  respectively. Thus, we first establish the boundedness of  $\|g_\theta - g\|_\infty$ , then demonstrate the approximation error of  $\|f_\theta - f\|_\infty$ .

**Lemma A.1.** *For  $\mathbf{q}^{(t_0)} \in \mathbb{R}^3$  as the initial position of a trajectory,  $r_a, r_b, T, \tau > 0$  and  $k \in \mathbb{Z}^+$ , a given ODE solver that is  $\tau$  compositions of an Euler Integrator  $\Phi_{\Delta t, g_\theta}$  with  $\Delta t = T/\tau$ , a set of points on exact trajectory associated with time increment interval  $[k\Delta t, T + k\Delta t]$  as*

$$V_{k\Delta t}^{T+k\Delta t} = \{\phi_{t', g}(\mathbf{q}^{(t_0)}) | k\Delta t \leq t' \leq T + k\Delta t\}, \quad (31)$$

, and denote

$$\mathcal{L}_{k\Delta t}^{T+k\Delta t} = \frac{1}{T} \|(\Phi_{\Delta t, g_\theta})^\tau - \phi_{T, g}\|_{\mathcal{B}(V_{k\Delta t}^{T+k\Delta t}, r_a)}, \quad (32)$$

and suppose that  $g$  and  $g_\theta$  are analytic and bounded by  $m$  within  $\mathcal{B}(V_{k\Delta t}^{T+k\Delta t}, r_a + r_b)$ , the union of complex balls centered at  $\mathbf{q} \in V_{k\Delta t}^{T+k\Delta t}$  with radius  $r_a + r_b$ . Then, there exist constants  $T_0$  and  $C$  that depends on  $r_a/m, r_b/m, \tau$ , and  $\Phi_{\Delta t, g_\theta}$ , such that, if  $0 < T < T_0, \forall t \in [t_0 + k\Delta t, t_0 + T + k\Delta t]$ ,

$$\|g_\theta(\mathbf{q}^{(t)}) - g(\mathbf{q}^{(t)})\|_\infty \leq Cm\Delta t + \frac{e}{e-1} \mathcal{L}_{k\Delta t}^{T+k\Delta t}, \quad (33)$$

where  $e$  is the base of the natural logarithm.

*Proof.* This result can be directly derived from Theorem 3.1, 3.2 and Corollary 3.3 in (Zhu et al., 2022) via replacing the integrator from Runge-Kutta integrator with Euler integrator since Euler integrator has been proven to satisfy the prerequisites of theorems in Appendix B.2 in (Zhu et al., 2022).  $\square$

Since  $\mathcal{B}(V_0^T, r_1) \subset \mathcal{B}(V_0^{2T}, r_1)$ , per our assumption, then  $g$  and  $g_\theta$  are both analytic and bounded by  $m_1$  in  $\mathcal{B}(V_0^T, r_1)$ . To utilize Lemma A.1, we set  $k = 0$  and  $T = t_1 - t_0$ . With suitable  $r_1$  and  $m_1$ , we have  $T < T_0$  and,  $\forall t \in [t_0, t_1]$ ,

$$\|g_\theta(\mathbf{q}^{(t)}) - g(\mathbf{q}^{(t)})\|_\infty \leq C_1 m_1 \Delta t + \frac{e}{e-1} \mathcal{L}_0^T, \quad (34)$$

where  $\mathcal{L}_0^T = \frac{1}{T} \|(\Phi_{\Delta t, g_\theta})^\tau - \phi_{T, g}\|_{\mathcal{B}(V_0^T, r_1)}$  and  $C_1$  is a control constant.

To establish the connection between  $g$  and  $f$ , we only focus on the first time step  $\Delta t$  instead of the entire time interval  $T$ . Recall that  $g$  is obtained by integrating  $f$ , we have

$$\psi_{\Delta t, g, f}(\mathbf{q}^{(t_0)}) = g(\mathbf{q}^{(t_0+\Delta t)}) = g(\mathbf{q}^{(t_0)}) + \int_{t_0}^{t_0+\Delta t} f(\mathbf{q}^{(t)}) dt, \quad (35)$$

as the flow map for the first-order derivative of the exact trajectory in a single step. As defined in Eq. 29, its corresponding Euler integrator has the form

$$g_\theta(\mathbf{q}_\theta^{(t_0+\Delta t)}) = \Psi_{\Delta t, g_\theta, f_\theta}(\mathbf{q}^{(t_0)}) = g_\theta(\mathbf{q}_\theta^{(t_0)}) + f_\theta(\mathbf{q}_\theta^{(t_0)})\Delta t. \quad (36)$$

Note that  $\mathcal{B}(V_0^T, r_2) \subset \mathcal{B}(V_0^{2T}, r_2)$ ,  $f$  and  $f_\theta$  are both analytic and bounded by  $m_2$  in  $\mathcal{B}(V_0^T, r_2)$ . In lemma A.1, we substitute  $\phi_{\Delta t, g_\theta}$  by  $\Psi_{\Delta t, g_\theta, f_\theta}$  and set  $T = \Delta t$  with  $\tau = 1$  and  $k = 0$ , then the corresponding loss becomes

$$\begin{aligned} \mathcal{L}_{0,1}^T &= \frac{1}{\Delta t} \|\Psi_{\Delta t, g_\theta, f_\theta} - \psi_{\Delta t, g, f}\|_{\mathcal{B}(V_0^{\Delta t}, r_2)} \\ &= \sup_{\mathbf{q} \in \mathcal{B}(V_0^{\Delta t}, r_2)} \frac{1}{\Delta t} \|\Psi_{\Delta t, g_\theta, f_\theta}(\mathbf{q}) - \psi_{\Delta t, g, f}(\mathbf{q})\|_\infty \\ &= \frac{1}{\Delta t} \|\Psi_{\Delta t, g_\theta, f_\theta}(\mathbf{q}^*) - \psi_{\Delta t, g, f}(\mathbf{q}^*)\|_\infty \\ &= \frac{1}{\Delta t} \|g_\theta(\tilde{\mathbf{q}}_\theta^{(t_0+\Delta t)}) - g(\tilde{\mathbf{q}}^{(t_0+\Delta t)})\|_\infty, \end{aligned} \quad (37)$$

where  $\mathbf{q}^*$  denote the point that maximizes the Eq. 37 and  $\tilde{\mathbf{q}}^{(t)}$  is another trajectory with  $\tilde{\mathbf{q}}^{(t_0)} = \mathbf{q}^*$  because  $\mathbf{q}^* \in V_0^{\Delta t}$  may not holds. For both  $\tilde{\mathbf{q}}^{(t)}$  and  $\mathbf{q}^t$ , the actual position and estimation from SEGNO at  $t_0 + \Delta t$  have the form

$$\begin{aligned} \tilde{\mathbf{q}}^{(t_0+\Delta t)} &= \tilde{\mathbf{q}}^{(t_0)} + \int_{t_0}^{t_0+\Delta t} g(\tilde{\mathbf{q}}^{(t)}) dt, \\ \tilde{\mathbf{q}}_\theta^{(t_0+\Delta t)} &= \tilde{\mathbf{q}}^{(t_0)} + g_\theta(\tilde{\mathbf{q}}^{(t_0)})\Delta t + f_\theta(\tilde{\mathbf{q}}^{(t_0)})\Delta t^2, \\ \mathbf{q}_\theta^{(t_0+\Delta t)} &= \mathbf{q}^{(t_0)} + g_\theta(\mathbf{q}^{(t_0)})\Delta t + f_\theta(\mathbf{q}^{(t_0)})\Delta t^2. \end{aligned} \quad (38)$$

Then, with suitable  $\Delta t$  and  $r_1$ ,  $g(\tilde{\mathbf{q}}^{(t_0+\Delta t)})$ ,  $g_\theta(\tilde{\mathbf{q}}_\theta^{(t_0+\Delta t)})$  and  $g_\theta(\mathbf{q}_\theta^{(t_0+\Delta t)})$  are bounded by  $m_1$  since  $\tilde{\mathbf{q}}^{(t_0+\Delta t)}$ ,  $\tilde{\mathbf{q}}_\theta^{(t_0+\Delta t)}$ ,  $\mathbf{q}_\theta^{(t_0+\Delta t)} \in \mathcal{B}(V_T, r_1)$ . Given such, the aforementioned loss is transformed into

$$\begin{aligned} \mathcal{L}_{0,1}^T &\leq \frac{1}{\Delta t} [\|g_\theta(\tilde{\mathbf{q}}_\theta^{(t_0+\Delta t)}) - g_\theta(\tilde{\mathbf{q}}^{(t_0+\Delta t)})\|_\infty + \|g_\theta(\tilde{\mathbf{q}}^{(t_0+\Delta t)}) - g_\theta(\mathbf{q}^{(t_0+\Delta t)})\|_\infty \\ &\quad + \|g_\theta(\mathbf{q}^{(t_0+\Delta t)}) - g(\mathbf{q}^{(t_0+\Delta t)})\|_\infty + \|g(\mathbf{q}^{(t_0+\Delta t)}) - g(\tilde{\mathbf{q}}^{(t_0+\Delta t)})\|_\infty] \\ &\leq \frac{1}{\Delta t} (6m_1 + C_1 m_1 \Delta t + \frac{e}{e-1} \mathcal{L}_0^T). \end{aligned} \quad (39)$$

Subsequently, given that  $\Delta t < T < T_0$ , we have,  $\forall t \in [t_0, t_0 + \Delta t]$ ,

$$\begin{aligned} \|f_\theta(\mathbf{q}^{(t)}) - f(\mathbf{q}^{(t)})\|_\infty &\leq C_2 m_2 \Delta t + \frac{e}{e-1} \mathcal{L}_{0,1}^T, \\ &\leq C_2 m_2 \Delta t + \frac{e}{e-1} \left[ \frac{1}{\Delta t} (6m_1 + C_1 m_1 \Delta t + \frac{e}{e-1} \mathcal{L}_0^T) \right], \\ &\leq C_2 m_2 \Delta t + \left( \frac{e}{e-1} \right)^2 \frac{\mathcal{L}_0^T}{\Delta t} + \frac{e}{e-1} \left( \frac{6m_1}{\Delta t} + C_1 m_1 \right), \\ &\leq O(\Delta t + \frac{\mathcal{L}_0^T}{\Delta t}). \end{aligned} \quad (40)$$

To extend the boundness up to  $t_1$ , we can easily utilize Lemma A.1 with different  $k = 1, \dots, \tau - 1$  (Eq. 34) to repeat the above derivation, and obtain,  $\forall t \in [t_0 + k\Delta t, t_0 + (k+1)\Delta t]$ ,

$$\|f_\theta(\mathbf{q}^{(t)}) - f(\mathbf{q}^{(t)})\|_\infty \leq O(\Delta t + \frac{\mathcal{L}_{k\Delta t}^{T+k\Delta t}}{\Delta t}), \quad (41)$$

where  $\mathcal{L}_{k\Delta t}^{T+k\Delta t} = \frac{1}{T} \|(\Phi_{\Delta t, g_\theta})^\tau - \phi_{T, g}\|_{\mathcal{B}(V_{k\Delta t}^{T+k\Delta t}, r_1)}$ . Therefore,  $\forall t \in [t_0, t_1]$ ,

$$\begin{aligned} \|f_\theta(\mathbf{q}^{(t)}) - f(\mathbf{q}^{(t)})\|_\infty &\leq \sup_{k=0, \dots, \tau-1} O(\Delta t + \frac{\mathcal{L}_{k\Delta t}^{T+k\Delta t}}{\Delta t}) \\ &\leq O(\Delta t + \frac{\mathcal{L}_0^{2T}}{\Delta t}), \end{aligned} \quad (42)$$

where  $\mathcal{L}_0^{2T} = \frac{1}{T} \|(\Phi_{\Delta t, g_\theta})^\tau - \phi_{T, g}\|_{\mathcal{B}(V_0^{2T}, r_1)}$  and it concludes the proof.

## A.5 PROOF OF COROLLARY 4.4

### A.5.1 LOCAL TRUNCATION ERROR

We use Taylor series expansion to approximate the trajectory at time  $t + \Delta t$  as

$$\mathbf{q}^{(t+\Delta t)} = \mathbf{q}^{(t)} + g(\mathbf{q}^{(t)})\Delta t + \frac{1}{2}f(\mathbf{q}^{(t)})\Delta t^2 + O(\Delta t^3). \quad (43)$$

Then the local truncation error  $\epsilon_{t+\Delta t}, \forall t \in [t_0, t_1]$ , equals to

$$\begin{aligned} \epsilon_{t+\Delta t} &= \|\mathbf{q}^{(t+\Delta t)} - \mathbf{q}^{(t)} - g(\mathbf{q}^{(t)})\Delta t - f_\theta(\mathbf{q}^{(t)})\Delta t^2\|_2 \\ &= \|\mathbf{q}^{(t)} + g(\mathbf{q}^{(t)})\Delta t + \frac{1}{2}f(\mathbf{q}^{(t)})\Delta t^2 + O(\Delta t^3) \\ &\quad - \mathbf{q}^{(t)} - g(\mathbf{q}^{(t)})\Delta t - f_\theta(\mathbf{q}^{(t)})\Delta t^2\|_2 \\ &= \|\frac{1}{2}f(\mathbf{q}^{(t)})\Delta t^2 - f_\theta(\mathbf{q}^{(t)})\Delta t^2 + O(\Delta t^3)\|_2 \\ &= \|\frac{1}{2}(f(\mathbf{q}^{(t)}) - f_\theta(\mathbf{q}^{(t)}))\Delta t^2 - \frac{1}{2}f_\theta(\mathbf{q}^{(t)})\Delta t^2 + O(\Delta t^3)\|_2 \\ &\leq \|\frac{1}{2}(f(\mathbf{q}^{(t)}) - f_\theta(\mathbf{q}^{(t)}))\Delta t^2\|_2 + \|\frac{1}{2}f_\theta(\mathbf{q}^{(t)})\Delta t^2\|_2 + O(\Delta t^3) \\ &\leq \frac{\sqrt{3}\Delta t^2}{2}\|f(\mathbf{q}^{(t)}) - f_\theta(\mathbf{q}^{(t)})\|_\infty + \|\frac{1}{2}m_2\Delta t^2\|_2 + O(\Delta t^3) \\ &\leq \frac{\sqrt{3}\Delta t^2}{2}O(\Delta t + \frac{2\mathcal{L}_0^{2T}}{\Delta t}) + O(\Delta t^2). \end{aligned} \quad (44)$$

Here the last inequality is due to Theorem 4.3. If the learned model achieves a near-perfect approximation of the true trajectory, resulting in a significantly diminished loss  $\mathcal{L}_0^{2T}$ . Then,  $\forall t \in [t_0, t_1]$ ,

$$\epsilon_{t+\Delta t} \leq O(\Delta t^2). \quad (45)$$

### A.5.2 GLOBAL TRUNCATION ERROR

Considering  $t \in [t_0, t_1]$  and  $k = 1, \dots, \tau - 1$ , the boundness of the global truncation error can be derived in a recursive way via

$$\begin{aligned} \mathcal{E}_{t+(k+1)\Delta t} &= \|\mathbf{q}^{(t+(k+1)\Delta t)} - \mathbf{q}_\theta^{(t+(k+1)\Delta t)}\|_2 \\ &= \|\mathbf{q}^{(t+(k+1)\Delta t)} - \mathbf{q}_\theta^{(t+k\Delta t)} - g_\theta(\mathbf{q}_\theta^{(t+k\Delta t)})\Delta t - f_\theta(\mathbf{q}_\theta^{(t+k\Delta t)})\Delta t^2\|_2 \\ &= \|\mathbf{q}^{(t+k\Delta t)} - \mathbf{q}_\theta^{(t+k\Delta t)} + \mathbf{q}^{(t+(k+1)\Delta t)} - \mathbf{q}^{(t+k\Delta t)} - g(\mathbf{q}^{(t+k\Delta t)})\Delta t \\ &\quad - f_\theta(\mathbf{q}^{(t+k\Delta t)})\Delta t^2 + f_\theta(\mathbf{q}^{(t+k\Delta t)})\Delta t^2 - f_\theta(\mathbf{q}_\theta^{(t+k\Delta t)})\Delta t^2 \\ &\quad + g(\mathbf{q}^{(t+k\Delta t)})\Delta t - g_\theta(\mathbf{q}^{(t+k\Delta t)})\Delta t \\ &\quad + g_\theta(\mathbf{q}^{(t+k\Delta t)})\Delta t - g_\theta(\mathbf{q}_\theta^{(t+k\Delta t)})\Delta t\|_2 \\ &\leq \mathcal{E}_{t+k\Delta t} + \epsilon_{t+(k+1)\Delta t} + \|f_\theta(\mathbf{q}^{(t+k\Delta t)}) - f_\theta(\mathbf{q}_\theta^{(t+k\Delta t)})\|_2\Delta t^2 \\ &\quad + [\|g_\theta(\mathbf{q}^{(t+k\Delta t)}) - g_\theta(\mathbf{q}_\theta^{(t+k\Delta t)})\|_2 + \|g(\mathbf{q}^{(t+k\Delta t)}) - g_\theta(\mathbf{q}^{(t+k\Delta t)})\|_2]\Delta t. \end{aligned} \quad (46)$$

Note that  $g_\theta$  and  $f_\theta$  satisfy the Lipschitz continuity, we have

$$\begin{aligned} \|g_\theta(\mathbf{q}^{(t+k\Delta t)}) - g_\theta(\mathbf{q}_\theta^{(t+k\Delta t)})\|_2 &\leq L_g\|\mathbf{q}^{(t+k\Delta t)} - \mathbf{q}_\theta^{(t+k\Delta t)}\|_2 = L_g\|\mathcal{E}_{t+k\Delta t}\|_2, \\ \|f_\theta(\mathbf{q}^{(t+k\Delta t)}) - f_\theta(\mathbf{q}_\theta^{(t+k\Delta t)})\|_2 &\leq L_f\|\mathbf{q}^{(t+k\Delta t)} - \mathbf{q}_\theta^{(t+k\Delta t)}\|_2 = L_f\|\mathcal{E}_{t+k\Delta t}\|_2, \end{aligned} \quad (47)$$

where  $L_g$  and  $L_f$  denote Lipschitz constant for  $g_\theta$  and  $f_\theta$  respectively. Given that the learned model achieves a near-perfect approximation of the true trajectory, by Lemma A.1 and Eq. 34, we obtain

$$\|g(\mathbf{q}^{(t+k\Delta t)}) - g_\theta(\mathbf{q}^{(t+k\Delta t)})\|_2 \leq \sqrt{3}[C_1 m_1 \Delta t + \frac{e}{e-1}\mathcal{L}_0^{2T}] \leq O(\Delta t). \quad (48)$$

Thus, the global truncation error in Eq. 46 is transformed into

$$\mathcal{E}_{t+(k+1)\Delta t} \leq (1 + L_g\Delta t + L_f\Delta t^2)\mathcal{E}_{t+k\Delta t} + O(\Delta t^2). \quad (49)$$



Figure 5: Example trajectories of 5-body charged system. From left to right, the number of positive and negative charges are 1, 3, 0.

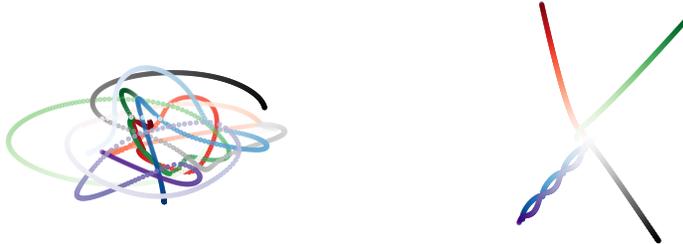


Figure 6: Example trajectories of 5-body gravity system. The system formed by gravitational force can either aggregate (i.e., left) or disperse (i.e., right).

Note that the global truncation error for the first step  $\mathcal{E}_{t+\Delta t}$ , where  $k = 1$ , is exactly the local truncation error  $\epsilon_{t+\Delta t}$  mentioned in Eq. 44. Given that  $k \leq \tau$ , we can derive,  $\forall t \in [t_0, t_1]$  and  $k = 1, \dots, \tau$ ,

$$\begin{aligned} \mathcal{E}_{t+k\Delta t} &\leq (1 + L_g\Delta t + L_f\Delta t^2)^{k-1}O(\Delta t^2) + \dots + (1 + L_g\Delta t + L_f\Delta t^2)O(\Delta t^2) \\ &\leq O(\tau\Delta t^2) = O\left(\frac{T}{\Delta t}\Delta t^2\right) = O(\Delta t). \end{aligned} \tag{50}$$

## B IMPLEMENTATION DETAILS

### B.1 MORE DETAILS ON SIMULATED N-BODY SYSTEMS

**N-body charged system** We use the same N-body charged system code<sup>4</sup> with previous work (Satorras et al., 2021; Brandstetter et al., 2021). They inherit the 2D implementation of (Kipf et al., 2018) and extend it to 3 dimensions. System trajectories are generated in 0.001 timestep and unbounded with virtual boxes. The initial location is sampled from a Gaussian distribution (mean  $\mu = 0$ , standard deviation  $\sigma = 0.5$ ), and the initial velocity is a random vector of norm 0.5. According to the charge types, three types of systems exist where the difference between the number of positive and negative charges are 1, 3, and 0. Example trajectories of these three types of systems are provided in Figure 5.

**N-body gravity system** The code<sup>5</sup> of gravitational N-body systems is provided by (Brandstetter et al., 2021). They implement it under the same framework as the above charged N-body systems. System trajectories are generated in 0.001 timestep, using gravitational interaction and no boundary conditions. Particle positions are initialized from a unit Gaussian, particle velocities are initialized with a norm equal to one, random direction, and particle mass is set to one. The system formed

<sup>4</sup><https://github.com/vgsatorras/egnn>

<sup>5</sup><https://github.com/RobDHess/Steerable-E3-GNN>

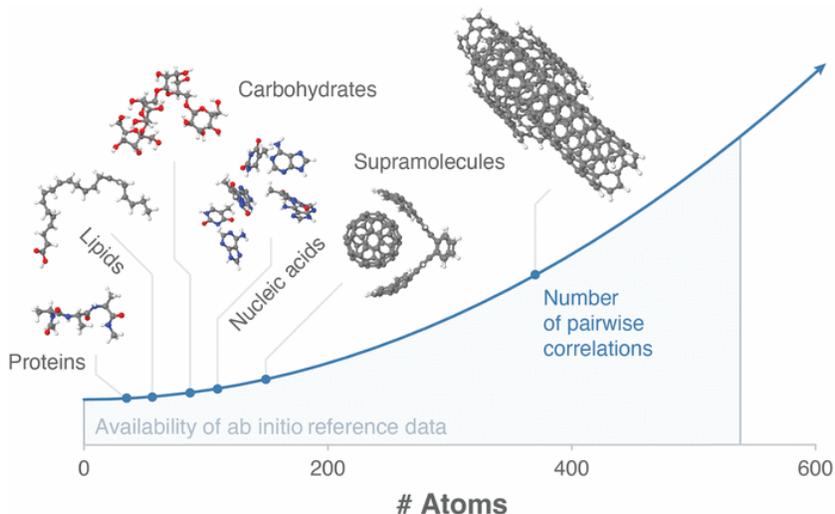


Figure 7: Molecular structures of MD22 dataset, which is borrowed from Fig.1 of original paper (Chmiela et al., 2023).

by gravitational force can either aggregate or disperse. Example trajectories of these two types of systems are provided in Figure 6.

**Hyperparameter** We empirically find that the following hyperparameters generally work well, and use them across most experimental evaluations: Adam optimizer with learning rate 0.001, the number of epochs 500, hidden dim 64, weight decay  $1 \times 10^{-12}$ , and layer number 4. We set the iteration time of SEGNO to 8. The representation degrees of SE(3)-transformer and TFN are set to 3 and 2. The number of training, validation, and testing sets are 3000, 2000, and 2000, respectively.

## B.2 MORE DETAILS ON MD22

**Molecules and Hyperparameter** The molecular structures of MD22 are displayed in Figure 7, which is borrowed from their paper (Chmiela et al., 2023). We use the following hyperparameters across all experimental evaluations: Adam optimizer with learning rate 0.001, the number of epochs 5000 with early stopping 100, hidden dim 64, weight decay  $1 \times 10^{-12}$ , and layer number 4. The iteration time of SEGNO is searched from 4 to 6. The representation degrees of SE(3)-transformer and TFN are set to 2. Due to the limited memory, the batch size of TFN is set to 10. The number of training, validation, and testing sets are 500, 2000, and 2000, respectively. The threshold for graph construction is set to 2.5 for all molecules.

## B.3 MORE DETAILS ON MOTION CAPTURE

**Hyperparameter** We use the following hyperparameters across all experimental evaluations: Adam optimizer with learning rate 0.001, the number of epochs 3000, hidden dim 64, weight decay  $1 \times 10^{-12}$ , and layer number 4. The iteration time of SEGNO is set to 4. We adopt a random split strategy introduced by Huang et al. (2022) where train/validation/test data contains 200/600/600 frame pairs.

# C ADDITIONAL EXPERIMENTS

## C.1 ACCURACY OF LEARNED LATENT TRAJECTORY

It is interesting to see how models learn the latent trajectory between the input and output states. Accordingly, we train models on 1000ts on two datasets and make the test on shorter time steps

Table 5: The generalization from long-term to short-term. All models are trained on 1000ts and test on 250/500/750/1000 ts. Mean squared error ( $\times 10^{-2}$ ) and the standard deviation are reported. Results averaged across 5 runs.

Method	Charged					Gravity				
	250 ts	500 ts	750 ts	1000 ts	Avg	250 ts	500 ts	750 ts	1000 ts	Avg
<b>GNN</b>	73.40 $\pm$ 9.60	31.79 $\pm$ 5.28	12.86 $\pm$ 2.81	0.826 $\pm$ 0.08	29.72	181.9 $\pm$ 26.1	90.33 $\pm$ 15.5	30.66 $\pm$ 12.3	0.746 $\pm$ 0.05	75.93
<b>GDE</b>	92.65 $\pm$ 25.0	43.94 $\pm$ 10.9	12.20 $\pm$ 23.2	0.652 $\pm$ 0.05	37.36	136.0 $\pm$ 135	56.80 $\pm$ 60.6	12.21 $\pm$ 14.8	0.588 $\pm$ 0.58	51.39
<b>EGNN</b>	6.756 $\pm$ 3.05	3.816 $\pm$ 4.68	3.668 $\pm$ 0.74	0.568 $\pm$ 0.09	3.702	7.146 $\pm$ 7.06	29.70 $\pm$ 20.4	9.712 $\pm$ 3.60	0.382 $\pm$ 0.11	11.89
<b>GMN</b>	10.44 $\pm$ 7.43	10.92 $\pm$ 4.67	4.518 $\pm$ 1.36	0.512 $\pm$ 0.16	6.598	7.430 $\pm$ 8.19	9.540 $\pm$ 12.1	5.730 $\pm$ 6.69	0.349 $\pm$ 0.48	5.762
<b>SEGNN</b>	21.78 $\pm$ 9.69	52.74 $\pm$ 15.6	34.13 $\pm$ 14.9	0.342 $\pm$ 0.04	27.25	10.58 $\pm$ 6.28	49.63 $\pm$ 37.7	25.82 $\pm$ 27.0	0.448 $\pm$ 0.02	21.62
<b>SEGNO</b>	0.188 $\pm$ 0.03	0.312 $\pm$ 0.06	0.360 $\pm$ 0.06	0.309 $\pm$ 0.11	0.292	0.064 $\pm$ 0.02	0.128 $\pm$ 0.03	0.176 $\pm$ 0.04	0.210 $\pm$ 0.07	0.145

by performing SEGNO on the smaller  $\tau$  step with the same ratio. For the baselines, we treat the forward timestep of each hidden layer as the same and extract their object position information as the prediction. Table 5 reports the mean and standard deviation of each setting. From Table 5 we can observe that:

- Clearly, SEGNO outperforms all other baselines across all settings by a large margin. Notably, when there is a lack of supervised signals at 250/500/750ts, the performance of all other baselines decreases significantly. By contrast, SEGNO achieves similar results as in 1000ts, demonstrating its robust generalization to short-term prediction.
- Another interesting point is that SEGNO’s error exhibits a distinct trend compared to other baselines. While the errors of other baselines significantly increase with decreasing time steps, SEGNO achieves even smaller errors with shorter time steps. This observation justifies our theoretical results that the error is bound by the chosen timestep.
- Additionally, the standard deviation of SEGNO is much smaller than that of other baselines, indicating the numerical stability of SEGNO. This result further confirms our theoretical finding that SEGNO can obtain a better latent trajectory between two discrete states.

## C.2 COMPARISON WITH GNS

We conducted additional evaluations of GNS and SEGNO-avg., which are learned by minimizing average acceleration, on two N-body systems. The results of these evaluations are presented in Table 6. We can observe that SEGNO outperforms both GNS and SEGNO-avg. In all cases, showing that training second-order neural odes on position loss outperforms training models on average acceleration.

Table 6: Comparison ( $\times 10^{-2}$ ) between SEGNO and GNS on simulated N-body systems. Results averaged across 3 runs.

Method	Charged			Gravity		
	1000 ts	1500 ts	2000 ts	1000 ts	1500 ts	2000 ts
GNS	3.245 $\pm$ 0.068	11.689 $\pm$ 0.330	31.632 $\pm$ 0.206	4.204 $\pm$ 0.081	17.095 $\pm$ 0.136	50.275 $\pm$ 0.201
SEGNO-avg.	2.146 $\pm$ 0.079	10.145 $\pm$ 0.034	24.244 $\pm$ 0.212	1.431 $\pm$ 0.047	19.488 $\pm$ 0.978	54.370 $\pm$ 1.385
SEGNO	<b>0.433</b> $\pm$ 0.013	<b>1.858</b> $\pm$ 0.029	<b>4.285</b> $\pm$ 0.049	<b>0.338</b> $\pm$ 0.027	<b>1.362</b> $\pm$ 0.077	<b>4.017</b> $\pm$ 0.087

## C.3 ROLLOUT COMPARISON ON N-BODY SYSTEMS

We evaluate the generalizability of models for rollout simulation. Specifically, we train all models on 1000ts and use rollout to make the prediction for the longer time step (over 40 rollout steps, indicating over 40000ts.). Figure 8 depicts the mean squared error of all methods on two datasets. We can observe that all baselines experience numerical explosion due to error accumulation during the rollout process, leading to a quick drop in prediction performance. In contrast, SEGNO demonstrates an order-of-magnitude error improvement over other baselines. This numerical stability can be attributed to the Neural ODE framework for modeling position and velocity.

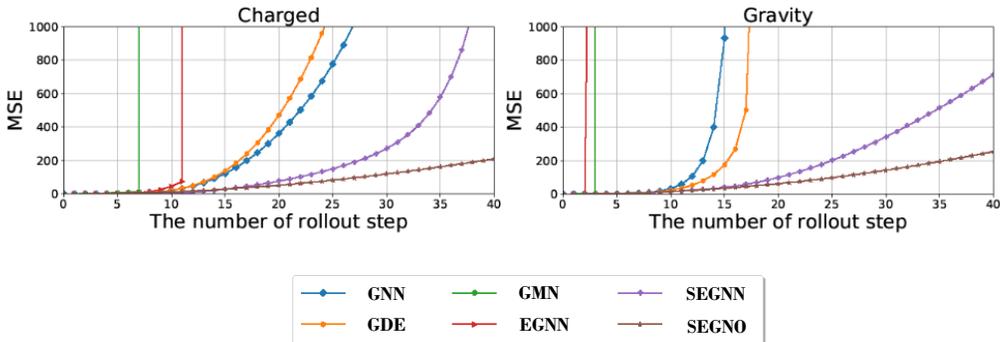


Figure 8: Mean squared error of rollout. Each rollout step is equal to 1000 ts. All models are trained on 1000ts.

#### C.4 COMPARISON WITH PHYSICS-INFORMED GNNs

Following the experimental settings in (Thangamuthu et al., 2022), we thoroughly compare the rollout position, energy, and momentum errors of SEGNO with those of GNODE (Bishnoi et al., 2022) and HGNN (Thangamuthu et al., 2022) on N-body systems. GNODE and HGNN are trained by minimizing acceleration and Hamilton respectively. The target timestep is set to 100ts. The results are demonstrated in Table 7. We can observe that SEGNO consistently outperforms GNODE and HGNN in rollout errors. Although SEGNO is only trained on the system positions, it achieves comparable performance on energy and momentum errors with GNODE, which further validates its effectiveness.

Table 7: Rollout position, energy, and momentum errors. All models are trained on 100ts. Results averaged across 3 runs.

	Method	Charged					Gravity				
		100 ts	500 ts	1000 ts	1500 ts	2000 ts	100 ts	500 ts	1000 ts	1500 ts	2000 ts
Rollout	HGNN	2.16	11.47	21.95	30.40	36.89	3.26	16.78	30.83	40.07	45.94
	GNODE	0.36	5.41	14.05	21.61	27.39	0.32	6.42	17.53	26.17	32.16
	SEGNO	<b>0.06</b>	<b>4.26</b>	<b>12.26</b>	<b>19.42</b>	<b>25.08</b>	<b>0.04</b>	<b>5.62</b>	<b>16.84</b>	<b>25.71</b>	<b>31.82</b>
Energy	HGNN	1.94	7.49	11.36	14.10	15.53	2.78	10.73	16.10	19.50	22.34
	GNODE	0.82	5.82	10.87	14.07	15.78	0.68	7.46	14.40	18.39	20.85
	SEGNO	<b>0.79</b>	<b>5.74</b>	<b>9.93</b>	<b>12.66</b>	<b>14.05</b>	<b>0.11</b>	<b>7.38</b>	<b>13.58</b>	<b>16.94</b>	<b>18.89</b>
Momentum	HGNN	10.12	33.37	45.98	51.67	54.38	6.99	25.65	39.12	46.08	49.96
	GNODE	4.17	28.65	43.97	50.84	54.06	2.35	21.43	36.93	44.92	49.19
	SEGNO	<b>2.17</b>	<b>28.39</b>	<b>43.81</b>	<b>50.46</b>	<b>53.54</b>	<b>0.75</b>	<b>21.40</b>	<b>36.88</b>	<b>44.65</b>	<b>48.42</b>

#### C.5 COMPARISON WITH ADVANCED FORCE FIELD PREDICTION MODELS

For a more comprehensive evaluation, we add empirical comparison on the MD22 dataset with NeuqIP (Batzner et al., 2022)<sup>6</sup> and Allegro (Musaelian et al., 2023)<sup>7</sup>. Both models take atom positions and numbers as input and optimize the errors between geometric outputs (generally treated as predicted forces) and true positions. We try our best to tune the models and the results are demonstrated in Table 8. It can be observed both methods do not perform well in predicting the time-dependent evolution of molecular dynamics, which is mainly attributed to the lack of dynamic information.

<sup>6</sup><https://github.com/mir-group/nequip>

<sup>7</sup><https://github.com/mir-group/allegro/tree/main>

Table 8: Comparison with equivariant GNNs in the domain of advanced force field prediction. Mean squared error ( $\times 10^{-3}$ ) on MD22 dataset averaged across 3 runs are reported.

Molecule	NequIP	Allgero	SEGNO
<b>Ac-Ala3-NHMe</b>	12.060 $\pm$ 0.220	11.785 $\pm$ 0.193	<b>0.418</b> $\pm$ 0.038
<b>DHA</b>	13.275 $\pm$ 0.164	13.126 $\pm$ 0.121	<b>0.370</b> $\pm$ 0.020
<b>Stachyose</b>	11.375 $\pm$ 0.104	11.164 $\pm$ 0.135	<b>0.440</b> $\pm$ 0.096
<b>AT-AT</b>	9.178 $\pm$ 0.123	9.032 $\pm$ 0.125	<b>0.548</b> $\pm$ 0.006
<b>AT-AT-CG-CG</b>	8.959 $\pm$ 0.066	8.866 $\pm$ 0.125	<b>0.394</b> $\pm$ 0.033
<b>Buckyball Catcher</b>	5.418 $\pm$ 0.058	5.331 $\pm$ 0.077	<b>0.199</b> $\pm$ 0.020
<b>Double-walled Nanotube</b>	3.852 $\pm$ 0.077	3.794 $\pm$ 0.022	<b>0.225</b> $\pm$ 0.008

### C.6 RUNTIME COMPARISON ON N-BODY SYSTEMS

We evaluate the running time of each model on N-body systems with Tesla T4 GPU and report the average forward time in seconds for 100 samples. The results are listed in Table 9. We can observe that SEGNO’s forward time (0.0227s) remains competitive compared to the best baseline SEGNN (0.0315s), indicating its efficiency.

Table 9: Forward time in seconds for a batch size of 100 samples on a Tesla T4 GPU.

Linear	GNN	GDE	TFN	SE(3)-Tr.	RF	EGNN	GMN	SEGNN	SEGNO
0.0002	0.0064	0.0088	0.0440	0.2661	0.0052	0.0126	0.0137	0.0315	0.0277

### C.7 GENERALIZATION CAPABILITY TO LARGE SYSTEMS

In this part, we evaluate the generalizability of models for larger system sizes. Specifically, we train all models on 5-body gravity systems and then test them on 10-body and 20-body gravity systems. Table 10 shows their results. We compare three strong baselines, i.e., EGNN, GMN, and SEGNN. The results show that the performance of all baselines significantly drops when testing on larger systems. In contrast, SEGNO still demonstrates a marked improvement over other baselines, especially on 20-body systems.

### C.8 MORE RESULTS ON CMU MOTION CAPTURE

This section illustrates more visualizations of GMN and SEGNO on modeling object motions. From Figure 9, we can observe that SEGNO is able to track the ground-truth trajectories accurately, which is consistent with the performance in Table 4.

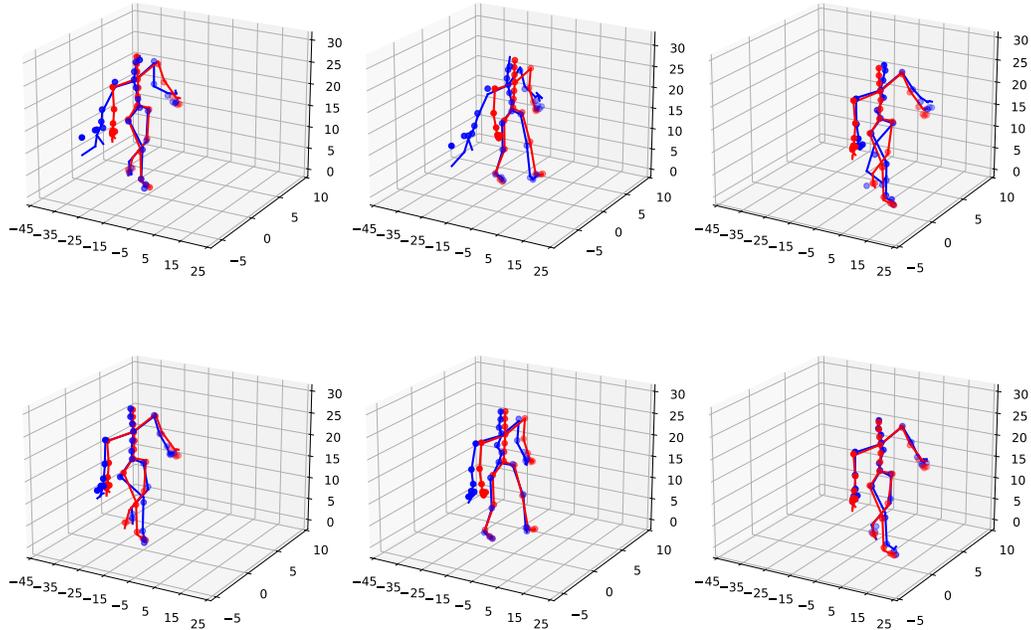


Figure 9: Example snapshots of Motion Capture with 50 time steps. *Top*: Prediction of GMN. *Bottom*: Prediction of SEGNO. Ground truth in red, and prediction in blue.

Table 10: The results on the larger systems. Mean squared error and the standard deviation are reported. Results are averaged across 3 runs.

Method	10-Body	20-Body
EGNN	$0.566 \pm 0.316$	$1.985 \pm 1.111$
GMN	$0.716 \pm 0.314$	$1.323 \pm 0.430$
SEGNN	$0.333 \pm 0.036$	$3.937 \pm 2.121$
SEGNO	$0.152 \pm 0.021$	$0.850 \pm 0.015$