Towards Geometric Normalization Techniques in SE(3) Equivariant Graph Neural Networks for Physical Dynamics Simulations

Ziqiao Meng 1*, Liang Zeng 3, Zixing Song 1, Tingyang Xu 2, Peilin Zhao 2 and Irwin King 1

1 The Chinese University of Hong Kong  
2 Tencent AI Lab  
3 Tsinghua University  
king@cse.cuhk.edu.hk, masonzhao@tencent.com,

Abstract

SE(3) equivariance is a fundamental property that is highly desirable to maintain in physical dynamics modeling. This property ensures neural outputs to remain robust when the inputs are translated or rotated. Recently, there have been several proposals for SE(3) equivariant graph neural networks (GNNs) that have shown promising results in simulating particle dynamics. However, existing works have neglected an important issue that current SE(3) equivariant GNNs cannot scale to large particle systems. Although some simple normalization techniques are already in use to stabilize the training dynamics of equivariant graph networks, they actually break the SE(3) equivariance of the architectures. In this work, we first show the numerical instability of training equivariant GNNs on large particle systems and then analyze some existing normalization strategies adopted in modern works. We propose a new normalization layer called GEONORM, which can satisfy the SE(3) equivariance and simultaneously stabilize the training process. We conduct comprehensive experiments on N-body system simulation tasks with larger particle system sizes. The experimental results demonstrate that GEONORM successfully preserves the SE(3) equivariance compared to baseline techniques and stabilizes the training dynamics of SE(3) equivariant GNNs on large systems.

1 Introduction

Particle dynamics simulations have recently been successfully formulated as geometric graph learning problems. The message passing mechanism of graph neural networks (GNNs) [Kipf and Welling, 2017; Hamilton et al., 2017; Veličković et al., 2018] can effectively capture the interactions between each pair of particles. However, applying GNNs to particle dynamics modeling poses a special challenge in preserving physical symmetry biases, particularly when scaling to larger particle systems. Specifically, the position update step of EGNN shown in Eq. 2 can be highly unstable, and this instability becomes more severe with deeper models and larger systems. We illustrate this issue in Figure 2 through n-body system simulation experiments involving charged particles following simple physics rules. To address this issue, some recent works have employed simple normalization techniques. One popular strategy is scaling geometric vectors by their norm to make them unit-length vectors. However, this approach vio-

*This work is done when Ziqiao Meng worked as an intern in Tencent AI Lab.

Figure 1: The illustration of particle dynamics simulations. At time step \( T - 1 \), input particle locations \( X(T - 1) \in \mathbb{R}^{3 \times N} \), where each column \( x_i \) denotes the 3D coordinate of particle \( i \). Then we predict next location \( X(T) \in \mathbb{R}^{3 \times N} \) at time step \( T \). Note that \( \Delta T \) is a pre-defined delta time frame, which determines the time slice to discretize an inherent continuous trajectory.
lates SE(3) equivariance, although it can stabilize the training dynamics. In the following sections, we will systematically analyze the weaknesses of existing approaches. In general, designing novel SE(3) equivariant normalization layers that can be easily integrated into current equivariant GNN architectures is challenging yet necessary. This will pave the way for training SE(3) equivariant GNNs on large particle system dynamics.

In this study, we begin by examining various normalization strategies. We analyze their properties and shortcomings through theoretical analysis. Following this, we introduce a new normalization layer called GEONorm. This ensures stable training dynamics and upholds the SE(3) equivariance constraints with a theoretical guarantee. We perform extensive experiments on n-body system simulation tasks with large system sizes, specifically $N = 50, 100$. It is observed that GEONorm effectively stabilizes training, irrespective of different weight initialization and learning rate choices. We further evaluate the SE(3) equivariance preservation of GEONorm by applying random rotations and translations to test inputs. Empirical results show that GEONorm achieves the lowest rotational and translational errors (RTE) when compared to baseline methods. In conclusion, our contributions can be summarized as follows:

- We systematically analyze the equivariance properties of several simple and existing normalization strategies. To the best of our knowledge, this is the first attempt to investigate geometric normalization techniques with equivariance analysis;
- We propose a SE(3) equivariant normalization layer, called GEONorm, which can stabilize the training dynamics of EGNN on much larger systems;
- We conduct extensive experiments to verify that our proposed GEO Norm method can effectively address the numerical explosion issue, while maintaining the SE(3) group equivariance of the original EGNN architecture.

2 Related Works

Normalization. Normalization layers are essential components of modern large models. In language and vision modeling, various classic normalization techniques have been introduced to improve the stability of large model training. These normalization layers aim to normalize the output of neurons by removing the mean and dividing by the standard deviation along specific dimensions, ensuring that the values fall within a reasonable range. For example, BatchNorm [Ioffe and Szegedy, 2015] standardizes the output across the entire batch of samples to address internal covariate shift. LayerNorm [Ba et al., 2016], on the other hand, normalizes the output across the feature dimensions of each individual sample, resulting in improved training stability for transformer-based models. Other variations follow a similar normalization approach but with different dimensions [Ulyanov et al., 2016; Wu and He, 2018; Brock et al., 2021], or use simplified versions by scaling with different norms [Singh and Krishnan, 2020; Daneshmand et al., 2020; Salimans and Kingma, 2016; Miyato et al., 2018]. Recent research has explored incorporating symmetry priors into normalization layers. In the context of graph structures, normalization techniques have been developed to stabilize deep GNN training by leveraging graph-specific information [Cai et al., 2021; Zhao and Akoglu, 2020; Yang et al., 2020; Li et al., 2022; Zhou et al., 2020; Dwivedi et al., 2023]. Similarly, there are SO(3) equivariant normalization layers designed for point cloud modeling [Deng et al., 2021; Shen et al., 2020]. Additionally, different normalization techniques have been proposed for chemical modeling [Meng et al., 2023] and protein modeling, taking into account corresponding geometric considerations [Schütt et al., 2017; Batatia et al., 2022; Jing et al., 2021].

Equivariant Graph Neural Networks. There are three main types of equivariant GNNs. Irreducible representation methods [Fuchs et al., 2020; Anderson et al., 2019; Thomas et al., 2018; Batzner et al., 2022; Züodie et al., 2022; Brandstetter et al., 2022; Frank et al., 2022] leverage specific equivariant basis functions to process relative position signals. However, these models suffer from excessive computational overhead and produce higher-order outputs. Regular representations [Finzi et al., 2020; Hutchinson et al., 2021] map each geometric vector to a group element by designing equivariant group convolutions. However, these models are usually SE(3)-invariant. Scalarization methods [Satorras et al., 2021b; Schütt et al., 2017; Gasteiger et al., 2020; Liu et al., 2022; Köhler et al., 2020; Jing et al., 2021; Huang et al., 2022; Schütt et al., 2021; Thölke and Fabritius, 2022; Klicpera et al., 2021] transform relative position information into invariant scalars and use...
this scalar information to update coordinates. This class of

![Diagram](image)

Figure 4: Two important geometric notions in a particle system.

models dominates current research and benefits many downstream applications like molecules [Hoogeboom et al., 2022], proteins [Ganea et al., 2022] and materials [Jiao et al., 2023; Song et al., 2024].

### 3 Preliminary

We first define the problem formulation of particle dynamics modeling. Given a set of $N$ particles $P = \{p_i\}_{i=1}^N$ with an input 3D position vector matrix $X(T) \in \mathbb{R}^{3 \times N}$ at time $T$, each particle $p_i$ is associated with an input geometric vector $x_i(T) \in \mathbb{R}^3$ ($i$-th column of $X(0)$) and an input scalar feature vector $h_i(T) \in \mathbb{R}^H$ with hidden dimension $H$. A geometric graph $G = (P, E)$ can be constructed by building local edges between each pair of particles. Usually, each particle connects with its neighbors within a pre-defined cutoff distance threshold $\gamma$ such that $E = \{e_{ij} ||x_i(T) - x_j(T)||_2 < \gamma\}$. In this work, we only consider fully connected geometric graphs. Each edge $e_{ij}$ can be associated with an edge feature $e_{ij}$, such as bond types in chemical molecules. The target of dynamics modeling is predicting the system state $X(T+\Delta T)$ after $\Delta T$ steps given the initial state $X(T)$ as input.

**SE(3) Equivariance.** Let $T_g: \mathcal{X} \to \mathcal{X}$ be a set of transformations on $\mathcal{X}$ for the abstract group $g \in G$. We say a function $f: \mathcal{X} \to \mathcal{Y}$ is equivariant to $g$ if there exists an equivalent transformation on its output space $S_g: \mathcal{Y} \to \mathcal{Y}$ such that:

$$f(T_g(X)) = S_g(f(X)).$$

If $S_g$ is an identity mapping function, then the function $f(\cdot)$ is invariant to group action $g$. Function $f(\cdot)$ is considered to be SE(3) equivariant if it is equivariant to both 3D rotations $R \in \mathbb{R}^{3\times3}$ (SO(3) group elements) and 3D translations $t \in \mathbb{R}^3$.

Specifically, function $f(\cdot)$ is SE(3)-equivariant if $f(RX + t) = R(f(X) + t)$.

**Equivariant Graph Neural Networks (EGNN).** In this work, we mainly focus on the EGNN architecture since its output is not higher-order outputs like irreducible representation-based methods and it is the fundamental prototype of various scalarization-based methods. Currently, EGNN is still regarded as the most efficient equivariant GNN models. The following is the general message passing scheme proposed by EGNN [Satorras et al., 2021b]:

$$m_{ij}^l = \phi_m(h_i^l, h_j^l, ||x_i^l - x_j^l||_2^2, e_{ij}),$$

$$x_i^{l+1} = x_i^l + \sum_{j \neq i} (x_j^l - x_i^l)\phi_h(m_{ij}^l),$$

$$(\text{2})$$

where $x_i^l$ denotes the coordinate of particle $i$ at layer $l$, $h_i^l$ denotes the invariant embedding of particle $i$ at layer $l$, $\phi_m$ and $\phi_h$ denote two separate multi-layer perceptrons (MLPs).

For dynamics modeling, the EGNN is trained in an Markov way with input $x_i(T)$ and output $x_i(T + \Delta T)$. The layer input $x_i^0$ is the input system state $x_i(T)$ at time $T$ and the final layer output $x_i^L$ of the above learning mechanism (stacking $L$ layers) is the output predicted $x_i(T + \Delta T)$.

### 4 Proposed Methods

In this section, we first demonstrate the issue of numerical instability in SE(3) equivariant GNNs and identify the specific challenges in designing geometric normalization. Next,
we summarize the temporary normalization techniques used in previous works and analyze their limitations. To address this challenge, we introduce a new normalization layer called GEONORM, which provides SE(3) equivariance with theoretical guarantee. Finally, we discuss some trade-off issues recently discovered in GEONORM.

4.1 Numerical Instability & Particular Challenges

From Eq. 2, we can see that there are two separate updating steps which are invariant embedding updates and geometric vector updates. Indicated by equivariant normalizing flow [Satorras et al., 2021a], the coordinate update step easily explodes when utilized in an ODE. Based on our experimental observations shown in Figure 2, the coordinate update step also explodes when training EGNN on large particle systems. In fact, both invariant embedding updates and geometric vector updates can lead to numerical explosions. The numerical instability of the invariant update part can be alleviated through a vanilla layer normalization LAYERORM. Indicated by equivariant normalizing flow [Ba et al., 2016]. If we apply LAYERORM on geometric vectors, the normalization layer would be as follows:

$$\mu(x_i^l) = \frac{\sum_{d=1}^{D} x_{id}^l}{\sigma(x_i^l)} = \sqrt{\sum_{d=1}^{D} (x_{id}^l - \mu(x_i^l))^2},$$

$$x_i^{l+1} = F(x_i^l) = g_i \cdot \frac{x_i^l - \mu(x_i^l)}{\sigma(x_i^l)} + b_i, \quad g_i, b_i \in \mathbb{R},$$

(3)

where $x_{id}^l \in \mathbb{R}$ denotes the $d$th dimension of the position coordinate of particle $i$ in layer $l$, $\mu(x_i^l) \in \mathbb{R}$ and $\sigma(x_i^l)$ denote the mean and standard deviation across coordinate dimensions respectively, $x_i^{l+1}$ denotes the output position vector of particle $i$ after the $l$th layer block, $g_i \in \mathbb{R}$ denotes a learnable re-scaling scalar parameter, $b_i \in \mathbb{R}^3$ denotes a learnable bias parameter. Note that unlike the vanilla LAYERORM on the invariant embedding update, the standard deviation $\sigma(x_i^l)$ in Eq. 3 is a scalar value since each dimension of the particle should be scaled by the same magnitude. However, LAYERORM has two important defects. First, the $\mu(x_i^l)$ and $\sigma(x_i^l)$ computation are measuring the mean and variance across coordinate dimension values, which does not summarize useful statistics that can reflect geometric information. Second, the above transformation is not SE(3) equivariant and thereby we present the following lemma:

**Lemma 1.** The vanilla LAYERORM on geometric vectors are neither rotation equivariant nor translation equivariant.

The proof of lemma 1 is straightforward since $\mu(x_i^l)$ is clearly not SE(3) equivariant and thus we show the detailed procedures in Appendix. To sum up, the geometric vector update has instability issue and it cannot be easily solved by simple modifications of the vanilla LAYERORM. Hence, designing SE(3) equivariant normalization layers for geometric vectors is a non-trivial challenge.

4.2 Existing Solutions

There are some straightforward solutions tackling the instability issue. One of the popular strategies adopted in many recent works [Liao and Smidt, 2023; Fuchs et al., 2020; Deng et al., 2021; Jing et al., 2021; Batatia et al., 2022] is scaling the geometric vector by its norm. Specifically, most of the variants can be concluded in the following form:

$$x_i^{l+1} = F(x_i^l) = g_i \cdot \frac{x_i^l}{\text{norm}(x_i^l)}, \quad g_i \in \mathbb{R},$$

(4)

where $g_i$ is still a learnable re-scaling scalar parameter, norm$(\cdot)$ denotes a vector norm computation on $x_i$. Some adopt the L2-norm and some adopt the RSM-norm (Root-Mean-Square norm). To summarize, the temporary solution presented in Eq. 4 involves using various norms to scale geometric vectors into unit-length vectors, depending on the selected norm. Experimental evidence from related works suggests that this method of feature scaling can significantly facilitate the training stability of EGNN. However, it also has issues with SE(3) equivariance constraints and we present the following lemma:

**Lemma 2.** The feature scaling norm SCALEORM is rotation equivariant but not translation equivariant.

We use L2-norm as an example to prove the Lemma 2:

**Proof.** $F(R_{x_i}) = g_i \cdot \frac{Rx_i}{\|Rx_i\|_2} = g_i \cdot \frac{Rx_i}{\sqrt{(Rx_i)^2}} = R(g_i \cdot \frac{x_i}{\sqrt{(x_i)^2}}) = RF(x_i)$

$$F(x_i + t) = g_1 \cdot \frac{x_i + t}{\sqrt{(x_i + t)^2}} \neq g_i \cdot \frac{x_i}{\sqrt{(x_i)^2}} = F(x_i) + t \quad \Box$$

Therefore, the SCALEORM is rotation equivariant but not translation equivariant. Also, this feature scaling normalization does not collect the mean statistics and consequently it does not contain learnable bias parameters, which impairs the expressiveness of the normalization layer.

Another temporary solution proposed in E-NF [Satorras et al., 2021a] is turning the relative coordinate difference $(x_i^l - x_j^l)$ in the second line of Eq. 2 into the scaled relative coordinate difference $(x_i^l - x_j^l) / \|x_i^l - x_j^l\|_2$. This solution is not a real normalization layer since it does not contain the re-scaling operation and $(x_i^l - x_j^l)$ is not a direct neural network output in EGNN. Consequently, no recovery of the coordinate difference norm would hurt the expressiveness of the networks as it results in some geometry and force information loss. In addition, this method still cannot stabilize the training dynamics of EGNN on real large particle systems.

4.3 New Geometric Normalization Layer GEONORM

The very first challenge we need to solve is finding summary statistics $\mu(x)$ and $\sigma(x)$ in particle systems that make geometric sense. Inspired by point cloud geometry observations, two important geometric notions are emerged as strong candidates for $\mu(x)$ and $\sigma(x)$. First, we adopt the center of mass of the particle system as $\mu(x)$. Since our problem setting assumes that each particle has exactly the same mass to simplify the problem setting, the center of mass is equal to the geometric center of the particle system such that $\mu(x) = \frac{1}{N} \sum x_i$.
For the standard deviation $\sigma(x)$, we adopt the euclidean distance between each particle and the geometric center such that

$$\sigma(x_i) = \|x_i - \mu(x)\|_2 = \sqrt{(x_i - \mu(x))^T (x_i - \mu(x))}.$$  

Here we abuse the notation $\sigma$ a bit for alignment with the terminologies used in the vanilla LN setting although the selected $\sigma(x_i)$ is not a typical standard deviation.  

We illustrate these two geometric notions in Figure 4. Importantly, these two statistics include fundamental geometric information. The geometric center $\mu(x)$ reflects the global state of the whole particle system and the $\sigma(x_i)$ describes how distant from each particle to the geometric center. Furthermore, these two geometric notions enjoy great SE(3) equivariance properties, which brings extra convenience for designing SE(3) equivariant normalization layers. Therefore, we present the following two lemmas:

**Lemma 3.** The geometric center $\mu(x)$ is both rotation equivariant and translation equivariant.

**Lemma 4.** The standard deviation $\sigma(x_i)$ is both rotation invariant and translation invariant.

The proof procedures of the above two lemmas are shown in the Appendix.

To consider the SE(3) equivariance guarantee of GeoNORM design, our strategy is splitting the normalization layer into two independent operations. The first operation is the normalization that projects geometric vectors to the standard normal distribution. The second operation is the re-scaling operation that learns to recover the normalized vectors back to the original. The general idea is turning the first operation into a rotation equivariant and translation invariant operation and then adding the translation back with a rotation-equivariant-re-scaling operation. From the Lemma 2, we know that the feature scaling norm is rotation equivariant but not translation equivariant. Inspired by [Satorras et al., 2021a] and the two previously selected geometric summary statistics, we modify the feature scaling operation in Eq. 4 to be a rotation equivariant and translation invariant operation

$$\frac{x_i - \mu(x)}{\|x_i - \mu(x)\|_2} = \frac{x_i - \mu(x)}{\sigma(x_i)}.$$  

We present the following lemma:

**Lemma 5.** The feature scaling of coordinate difference between each particle coordinate and the geometric center is rotation equivariant and translation invariant.

The proof of this lemma is shown in the Appendix. At this time, the feature scaling norm becomes $g_i \cdot \frac{x_i - \mu(x)}{\sigma(x_i)}$.

The remaining problem is how to add the missing translation back in the re-scaling operation while still maintaining the rotation equivariance. Note that since the geometric center has been removed from position coordinates, we need a learnable bias to recover the geometric center such that $g_i \cdot \frac{x_i - \mu(x)}{\sigma(x_i)} + b_i$. However, adding the additional bias vector $b_i$ would break the rotation equivariance. A straightforward solution to solve this tricky issue is multiplying a rotation equivariant and translation invariant vector on $b$ and replace the bias vector $b_i$ with a scalar parameter $b_i$. Therefore, we multiply a direction vector $\frac{\mu(x) - \mu(x_0)}{\|\mu(x) - \mu(x_0)\|_2}$ to $b_i$ such that the norm becomes $g_i \cdot \frac{x_i - \mu(x)}{\sigma(x_i)} + \frac{\mu(x) - \mu(x_0)}{\|\mu(x) - \mu(x_0)\|_2} \cdot b_i$ where $\mu(x_0)$ denotes the computed geometric center of the input state of particle system (not the intermediate layer $l$ state).

Note that we apply $\frac{\mu(x) - \mu(x_0)}{\|\mu(x) - \mu(x_0)\|_2}$ as the bias direction instead of $\frac{x_i - \mu(x)}{\|x_i - \mu(x)\|_2}$ and $\frac{\mu(x) - \mu(x_0)}{\|\mu(x) - \mu(x_0)\|_2}$. If we use $\frac{x_i - \mu(x)}{\|x_i - \mu(x)\|_2}$, then the bias $b_i$ learning would degenerate into the $g_i$ learning since the mean removal direction and given bias direction are exactly the same, which makes the bias learning very trivial. If we use $\frac{\mu(x) - \mu(x_0)}{\|\mu(x) - \mu(x_0)\|_2}$, then GeoNORM would be unstable particularly when a single particle position vector $x_i$ has extreme values. Therefore, using $\frac{\mu(x) - \mu(x_0)}{\|\mu(x) - \mu(x_0)\|_2}$ would avoid both problems. However, this leads to a direction inconsistency issue between the mean removal direction $\frac{x_i - \mu(x)}{\|x_i - \mu(x)\|_2}$ and the bias learning direction $\frac{\mu(x) - \mu(x_0)}{\|\mu(x) - \mu(x_0)\|_2}$. The scalar bias $b_i$ is not able to calibrate the bias direction to align with the mean removal direction. Fortunately, this problem can be perfectly solved by adding an additional $\mu(x_0)$ since $\frac{\mu(x) - \mu(x_0)}{\|\mu(x) - \mu(x_0)\|_2} + b_i + \mu(x_0)$ can perfectly recover the removed $\mu(x)$ with $b_i$ learns different information from $g_i$ (No degeneracy). We illustrate this mechanism in Figure 3 (d).

To conclude, the complete solution of our GeoNORM normalization layer is as follows:

$$\mu(x) = \frac{1}{N} \sum_{i=1}^{N} x_i,$$

$$\sigma(x_i) = \|x_i - \mu(x)\|_2,$$

$$F(x_i) = g_i \cdot \frac{x_i - \mu(x)}{\sigma(x_i)} + \frac{\mu(x) - \mu(x_0)}{\|\mu(x) - \mu(x_0)\|_2} \cdot b_i + \mu(x_0),$$

$$x_i^{t+1} = F(x_i), \quad \mu_0(x) = \mu(x_0), \quad g_i \in \mathbb{R}, b_i \in \mathbb{R}$$  

(5)

Then we prove the following theorem:

**Theorem 1.** GeoNORM is an SE(3) equivariant normalization layer, which is both rotation-equivariant and translation-equivariant.

We show the proof of the above theorem as follows:

Proof. $F(Rx_i + t) = g_i \cdot \frac{(Rx_i + t - R\mu(x_0) - t)}{\sigma(x_i)}$

$$+ \frac{R\mu(x) + t - R\mu_0 - t}{\|R(x_i) - \mu_0\|_2} \cdot b_i + R\mu_0 + t$$

$$= g_i \cdot R \left( \frac{x_i - \mu(x)}{\sigma(x_i)} \right) + \frac{\mu(x) - \mu(x_0)}{\|\mu(x) - \mu(x_0)\|_2} \cdot b_i + R\mu_0 + t$$

$$= R(g_i \cdot \frac{x_i - \mu(x)}{\sigma(x_i)} + \frac{\mu(x) - \mu(x_0)}{\|\mu(x) - \mu(x_0)\|_2} \cdot b_i + \mu_0) + t$$

$$= RF(x_i) + t \quad \Box$$

Therefore, GeoNORM is an SE(3) equivariant normalization layer which would not break the whole SE(3) equivariance of the particle system.

### 4.4 Further Discussions

Although we obtain an SE(3) equivariant normalization layer, a new challenge arises in the bias learning design. Note that in Eq. 5, the bias $b_i$ is a scalar parameter, which is only able to learn scalar information such as distance particularly the
direction information is determined by both $\frac{\mu(x) - \mu_0}{\|\mu(x) - \mu_0\|_2}$ and $\mu_0$. This design apparently downgrades the expressiveness of GeoNorm although we avoid the bias $b_i$ to be degenerated into the re-scaling parameter $g_i$. In contrast, if we want to transform the bias $b_i$ into a vector parameter $b_i \in \mathbb{R}^3$, the rotation matrix $R$ cannot be extracted to the front to ensure the rotation equivariance of GeoNorm. To sum up, there exists an temporary trade-off issue between expressiveness and SE(3) equivariance guarantee for the current GeoNorm design. To enable the bias parameter to learn both direction and distance information, we need to come up with a new solution that set biases to be learnable vectors instead of scalars meanwhile not breaking the rotation equivariance of GeoNorm. Since this is a tough challenge that requires tremendous additional research efforts, it is hard to cover the corresponding solutions in this work. Therefore, we leave this problem as an important future work to explore.

5 Experiments

Dataset & Experimental Setting. We also adopt the 3D extension [Fuchs et al., 2020] of the $N$-body system simulation dataset provided by Kipf [Kipf et al., 2018]. It provides simulation trajectories of charged particle systems following simple physical force rules. Specifically, we train EGNNs on thousands of provided training trajectories that can generalize to the validation and test sets containing 2000 trajectories. We utilize the MSE (Mean-Squared-Error) loss as the evaluation metric to measure the state prediction accuracy between predicted particle positions and ground-truth particle positions after 1000 steps. Unlike the previous works adopting small particle systems ($N = 3, 5$), we set large system simulations with $N = 20, 50, 100$ particles to test the model training stability. The two commonly used datasets QM9 and MD-17 are not used in this work since the involved system particle size is around 5-10 (small chemical molecular conformers), which is not large enough to test the stability of large EGNN training. In addition to stability testing, we also evaluate whether GeoNorm satisfy the SE(3) equivariance constraints. During testing, we apply random 3D rotations and translations to mutate the inputs. Then we report MSE scores of each selected baseline after SE(3) transformations mutations. We mainly report mean, median and incremental rotational and translational errors (RTE).

Implementation Details, Hyperparameters and Configurations. In preliminary sections, we explore the training instability issue by setting the number of particles $n = 20, 50, 100$, the number of layers $l = 3, 5, 8, 10$. For the equivariant GNN model selection, we adopt the alternative velocity version of EGNN as our backbone model (Note that the velocity vector update is omitted from Eq. 2, please refer details in [Satorras et al., 2021b]). In main experiments for normalization comparisons, there are mainly two important configurations which are random seed and learning rate. We select random seed $= 30, 40, 50, 60$ to test different weight initialization. For learning rate, we select $lr = 5e - 3, 5e - 4$ to test different learning rates. The optimization is conducted using Adam [Kingma and Ba, 2015] and the weight decay of $1 \times 10^{-12}$. The feature dimensions in EGNN is 64 as its original paper reported. All algorithms have been trained under the same conditions with batch size of 50. All experiments on N-body particle system use the default trajectory sampling method in EGNN [Satorras et al., 2021b]. We train all models with 1000 total number of epochs with batch size 50. The target we optimize is the averaged Mean Squared Error (MSE) between predicted positions and corresponding ground-truth positions. A single Tesla V100 GPU is needed to reproduce each hyperparameter’s experimental results. Running each configuration may take up to 6 hours.

Experimental Results. We mainly compare the training stability of vanilla EGNN, the feature scaling norm SCALENORM and our GeoNorm. From the training dynamics shown in Figure 5, we can see that both GeoNorm and SCALENORM can stabilize the training dynamics over all system configurations while the vanilla EGNN easily encounters numerical explosions. Furthermore, compared to SCALENORM, our GeoNorm perform better convergence properties in optimizations over different settings. We clearly see that in most cases GeoNorm (highlighted by red curves) can reach lower training errors compared to SCALENORM. Even in some cases, SCALENORM may converge faster than
GeoNorm, GeoNorm can still catch up with the performance of ScaleNorm in fewer epochs. This demonstrates the effectiveness of GeoNorm compared to baseline methods. However, we can also observe that GeoNorm is still not good enough since the training errors cannot be further minimized at some training stage. Although the vanilla EGNN is very unstable, its training errors can be minimized to very decent accuracy at some certain epochs. This may suggest that GeoNorm is still not expressive enough and some operations may hurt the expressiveness of the original EGNN.

From the table 5, we can see that using the LayerNorm, ScaleNorm and GeoNorm concatenated with EGNN training. We mainly report 8 metrics: Mean $R_{err}(\circ)$, Median $R_{err}(\circ)$, Max $R_{err}(\circ)$, $\Delta R_{err}$, Median $T_{err}$, Max $T_{err}$, $\Delta T_{err}$, which are reporting the mean, median, max and incremental RTE errors over all testing samples after rotation and translation mutations. Note that the most important metrics are $\Delta R_{err}$ and $\Delta T_{err}$ which reflect additional MSE errors caused by SE(3) transformations to the original testing input. System configurations include the number of particles $N = 50, 100$, the EGNN layers $l = 3, 5$ and the random seed $30, 40$.

### 7 Conclusion

In this work, we investigate geometric normalization techniques for deep SE(3) equivariant graph neural networks on larger particle system simulation tasks. We demonstrate that both naive and existing solutions cannot strictly satisfy the SE(3) equivariance geometric constraints. To address this issue and maintain both stable training dynamics and SE(3) equivariance, we propose a new technique called GeoNorm as a replacement for previous normalization strategies. The effectiveness of our approach is supported by theoretical analysis and experimental results.

### Acknowledgments

The work described in this paper was partially supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (RGC GRF 2151185). Ziqiao Meng and Liang Zeng contributed equally to this work.
References


[Li et al., 2022] Guohao Li, Chenxin Xiong, Guocheng Qian, Ali Thabet, and Bernard Ghanem. DeeperGCN: All you need to train deeper GCNs, 2022.


