Relaxing Continuous Constraints of Equivariant Graph Neural Networks for Broad Physical Dynamics Learning

Zinan Zheng*
Hong Kong University of Science and
Technology (Guangzhou)
Guangzhou, China
zzheng078@connect.hkust-gz.edu.cn

Yang Liu*
Hong Kong University of Science and Technology
Hong Kong University of Science and
Technology (Guangzhou)
Hong Kong SAR, China
yliukj@connect.ust.hk

Jia Li†
Hong Kong University of Science and
Technology (Guangzhou)
Guangzhou, China
jialee@ust.hk

Jianhua Yao
Yu Rong†
Tencent AI Lab
Shenzhen, China
jianhua.yao@gmail.com
yu.rong@hotmail.com

ABSTRACT

Incorporating Euclidean symmetries (e.g. rotation equivariance) as inductive biases into graph neural networks has improved their generalization ability and data efficiency in unbounded physical dynamics modeling. However, in various scientific and engineering applications, the symmetries of dynamics are frequently discrete due to the boundary conditions. Thus, existing GNNs either overlook necessary symmetry, resulting in suboptimal representation ability, or impose excessive equivariance, which fails to generalize to unobserved symmetric dynamics. In this work, we propose a general **D**iscrete **E**quivariant **G**raph **N**eural **N**etwork (DEGNN) that guarantees equivariance to a given discrete point group. Specifically, we show that such discrete equivariant message passing could be constructed by transforming geometric features into permutation-invariant embeddings. Through relaxing continuous equivariant constraints, DEGNN can employ more geometric feature combinations to approximate unobserved physical object interaction functions. Two implementation approaches of DEGNN are proposed based on ranking or pooling permutation-invariant functions. We apply DEGNN to various physical dynamics, ranging from particle, molecular, crowd to vehicle dynamics. In twenty scenarios, DEGNN significantly outperforms existing state-of-the-art approaches. Moreover, we show that DEGNN is data efficient, learning with less data, and can generalize across scenarios such as unobserved orientation.

---

*Equal contribution. Listing order is random. Yang Liu designed and implemented the first DEGNN. Zinan designed, implemented and evaluated countless model variants.
†Corresponding author.

---

1 INTRODUCTION

Accurate physical dynamic modeling is a fundamental task in numerous applications. For example, understanding crowd (i.s., a group of pedestrians) dynamics is essential for public safety, urban planning, and architecture design [26, 30, 53]. In scientific domains, simulating molecular dynamics is pivotal in material science [27, 33], drug discovery [10], and protein folding [20]. These systems have complex latent behavior patterns (e.g., agents' intentions or physical laws), which are generally difficult to observe. Consequently, the object interactions are either intractable (e.g., social interactions among pedestrians) [1, 36] or with high computation complexity (e.g., interactions between atoms or proteins) [11, 16, 19, 24, 47, 52]. In particular, the core aspect of their behavior forecasting is representing and reasoning the interactions among system objects. To achieve this, multiple Graph Neural Networks (GNNs) [19, 23, 32, 35, 37, 42–44, 52, 56–58] have been proposed for learning interactions of various physical systems such as crowds, particles, and molecules. They represent system objects as nodes, physical relations as edges, and their interactions as the message passing thereon.

Basically, physical dynamics demonstrates particular symmetries. Figure 1 displays two examples. Vehicles exhibit analogous behavior patterns on two opposing directions highways, and the

(a) Highway Vehicle Dynamics
f: **Equivariant to 180° rotation**

(b) Boundary Molecular Dynamics
f: **Equivariant to diagonal plane reflection**
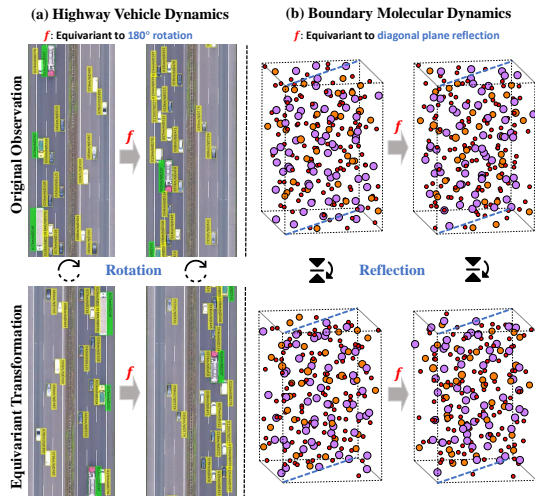
Figure 1: Illustration of discrete equivariance. (a) Rotation equivariance of vehicle trajectories. They have similar patterns when they drive in lanes of opposite directions; (b) Reflection equivariance of molecular dynamics. The effects of boundaries and molecules are symmetric when we reflect the entire system.

trajectories of molecules remain equivariant under reflection. Regardless of such inductive bias, current methods such as GNS [42] and Transformer [17] fail to generalize to unobserved directions. Considering highway vehicle dynamics, if training data are in a downward direction, GNS can predict well during testing when the vehicles are in the same direction but perform poorly if the vehicles are from the bottom up (See Section 4.3). This observation implies that conventional GNNs are insufficient for capturing the true dynamics and tend to overfit the observed trajectories.

Therefore, geometrically equivariant graph neural networks [21] have been proposed to leverage symmetry as an inductive bias to efficiently model physical dynamics. They force their outputs to be strictly equivariant under a given group, e.g., SE(3) (rotation and translation) [15, 46] and E(3) [3, 5] (rotation, translation, and reflection). However, strict continuous symmetry is rarely observed in physical dynamics. The widely existing boundary, such as sidewalks, highways, and boundary conditions in molecular simulation, generally breaks such symmetry into discrete elements. Learning such dynamics is highly challenging due to (1) **Discrete symmetry**. GNNs equivariant to continuous groups, like EGNN [43] and Eqmotion [51], are so strong and overly restrictive, limiting their flexibility and hindering their potential applications. Relaxing the continuous assumption and building GNNs that are equivariant to a *discrete group* is necessary; (2) **Intricate interaction**, which is divergent and dense. Specifically, the underlying interactions of different physical systems are diverse. In addition, we consider high-density scenarios where interactions occur with great frequency. Therefore, the model needs to be flexible enough to approximate these interactions.

In this paper, we propose a novel **D**iscrete **E**quivariant **G**raph **N**eural **N**etwork framework, called DEGNN, to forecast various types of physical dynamics. It first augments the input geometric features with transformations of the given group and then employs

a permutation-invariant function, followed by radial direction multiplication, to obtain the message embedding. We show that such a general framework is equivariant to a given point group. In addition, since it relaxes the constraints on geometric features, it could utilize more expressive feature combinations to approximate various types of object interactions. Under this framework, we propose two realizations, ranking and pooling methods. Extensive experiments on four physical systems, including macro-level vehicles and crowds, and micro-level particle and molecular dynamics, demonstrate that DEGNN has a better generalization ability over state-of-the-art models and is beneficial for learning the discrete symmetry of dynamics. Our main contributions are summarized as follows:

- We highlight the importance of learning discrete equivariance in physical dynamic modeling, which is challenging and vital in scientific and engineering applications. To the best of our knowledge, it is the first time discrete symmetry learning is considered in such domains.
- We propose a novel and general DEGNN framework that is proven to be equivariant to discrete groups.
- We propose two realizations of the DEGNN framework, via permutation-invariant ranking and pooling functions.
- We conduct extensive experiments to evaluate the performance of DEGNN on **twenty** scenarios of four types of physical systems. Experimental results show that the proposed model achieves significantly better performance than the state-of-the-art models. Additional results including ablation studies, generalization experiments, and sensitivity analysis further demonstrate the generalization ability of DEGNN. The code is available at the link: https://github.com/compasszzn/DEGNN.

2 PRELIMINARY

2.1 Problem Definition

In this work, we study various physical dynamics that consist of complex interacting objects (e.g., atoms, pedestrians, and vehicles) in static and bounded environments, and their trajectories are governed by complex physical rules or social interactions (e.g., vehicles tend to avoid collision). Therefore, our goal is to learn interactions of N objects and forecast their positions after a fixed time interval. At time t, each object i is represented by:

- Geometric features including the position vector $q_i^{(t)} \in \mathbb{R}^2$ and the velocity vector $\dot{q}_i^{(t)} \in \mathbb{R}^2$.
- Non-geometric features such as the atom number or vehicle type, denoted by u_i.
- Spatial connection where an edge e_{ij} is constructed via distance cutoff or physical relations. Generally, edges are associated with attributes such as object distance, denoted by a_{ij}.

For simplicity, we denote $(q^{(t)}, \dot{q}^{(t)})$ and $(u, e = \{e_{ij}\}, a = \{a_{ij}\})$ as dynamic and static state information of the entire graph correspondingly. Formally, the dynamic forecasting problem is defined as follows:

Definition 2.1. (Dynamic Forecasting) Given the initial system states $(q^{(t)}, \dot{q}^{(t)})$ at time t and static states (u, e, a), the objective

is to predict the subsequent position $q^{(t+\Delta t)}$, where Δt is the target time interval.

2.2 Equivariance and Invariance

*2.2.1 **Group**.* Equivariance and invariance play a pivotal role in physical dynamic modeling. Formally, it is defined on a specific group [45]:

Definition 2.2. (Group) A group is a set of operations that satisfy: closure, associativity, the existence of an identity element, and the existence of inverse elements for each element in the set.

For instance, a widely studied group in geometric graph learning is Euclidean group $E(n)$, which includes transformations of translation, rotation, and reflection.

*2.2.2 **Equivariance and invariance**.* Given a group G, the definition of G-Equivariance and G-Invariance is:

Definition 2.3. (G-Equivariance and G-Invariance) A function f is equivariant to group $\mathcal{G}$, if for any transformation $g \in \mathcal{G}$, $f(g \circ x) = g \circ f(x)$, $x \in X$. Similarly, f is invariant to group $\mathcal{G}$, if for any transformation $g \in \mathcal{G}$, $f(g \circ x) = f(x)$, $x \in X$.

The group transformation $\circ$ is instantiated as $g \circ x := Ox + t$ where $O \in O(n) := \{O \in \mathbb{R}^{n \times n} | O^T O = I\}$ is orthogonal transformation (rotation and reflection), and $t \in \mathbb{R}^n$ is a translation vector.

*2.2.3 **Permutation-invariance**.* Another important inductive bias is permutation-invariance. That is, the function output remains unchanged regardless of the order in which its inputs are presented. Formally, given the input is a set $\{x_1.x_2, \cdots, x_d\}$ where d is the size, its definition is:

Definition 2.4. (Permutation-invariance) A function acting on sets is permutation-invariant, if for any permutation π,

$$f(x_1, x_2, \cdots, x_d) = f(x_{\pi(1)}, x_{\pi(2)}, \cdots, x_{\pi(d)}). \tag{1}$$

2.3 Equivariant Message Passing

Since the composition of equivariant functions is again equivariant, equivariant GNNs [23, 43] are built via stacking multiple equivariant message-passing layers. Thus, their output will change in the same way as the input changes. In general, their l-th layer computes:

$$m_{ij}^{(l)} = \mu(q_i^{(l)}, q_j^{(l)}, \dot{q}_i^{(l)}, \dot{q}_j^{(l)}, h_i^{(l)}, h_j^{(l)}, a_{ij}),$$
$$q_i^{(l+1)}, \dot{q}_i^{(l+1)}, h_i^{(l+1)} = \nu(q_i^{(l)}, \dot{q}_i^{(l)}, h_i^{(l)}, \sum_{j \in \mathcal{N}_i} m_{ij}^{(l)}), \tag{2}$$

where $h_i^{(l)}$ is the l-th layer embedding of node i and $m_{ij}^{(l)}$ denotes the l-th layer message embedding between node i and node j. $\mathcal{N}_i$ collects the neighbors of node i. μ and ν are the equivariant message embedding function and node state updating function, respectively.

*2.3.1 **EGNN framework**.* It is non-trivial to design the above equivariant message embedding functions. A feasible solution is based on the EGNN framework [43], which relies on the inner product to transform geometric vectors into invariant features, followed by an MLP and radial directions multiplication. They have the following form:

$$\mu(x, h) = x\sigma(x^T x, h), \tag{3}$$

where x, h denotes abbreviated geometric and non-geometric terms and σ is an MLP. For example, EGNN employs the relative squared distance as the invariant feature:

$$\mu_{\text{egnn}}(q_i, q_j, h) = (q_i - q_j)\sigma(||q_i - q_j||^2, h), \tag{4}$$

due to that for any orthogonal transformation O ($O^T O = I$) and translation vector t in $E(n)$:

$$||(Oq_i + t) - (Oq_j + t)||^2 = ||O(q_i - q_j)||^2 = ||q_i - q_j||^2. \tag{5}$$

Then by multiply the MLP output with $q_i - q_j$, μ_{egnn} is rotation and reflection equivariant. Since involving $E(n)$-invariant features will not change its equivariant property, other feasible features are the inner product of velocity [27] or expanding the relative distance with radial basis functions [49].

Given the equivariant message embedding, the node states are updated through a physics-inspired integration:

$$\dot{q}_i^{(l+1)} = \psi(h_i^{(l)})\dot{q}_i^{(l)} + \sum_{j \in \mathcal{N}_i} m_{ij}^{(l)},$$
$$q_i^{(l+1)} = q_i^{(l)} + \dot{q}_i^{(l+1)}, \quad h_i^{(l+1)} = \sigma_h(q_i^{(l)}, \sum_{j \in \mathcal{N}_i} m_{ij}^{(l)}) \tag{6}$$

where $\psi(h_i^{(l)}) \in \mathbb{R}$ is a scalar that controls the magnitude of velocity. The non-geometric features could be updated via $h_i^{(l+1)} = \sigma_h(q_i^{(l)}, \sum_{j \in \mathcal{N}_i} m_{ij}^{(l)})$. However, such frameworks follow the constraint of continuous groups and are not straightforward to be relaxed to a discrete group.

*2.3.2 **Model training**.* The final prediction is obtained by applying several iterations of equivariant message passing. Then the model parameters can be optimized by minimizing the discrepancy between exact and approximated positions:

$$\mathcal{L}_{\text{train}} = \sum_{s \in \mathcal{D}_{\text{train}}} ||\hat{q}_s^{(t+\Delta t)} - q_s^{(t+\Delta t)}||_2, \tag{7}$$

where $\mathcal{D}_{\text{train}}$ denotes the training set. $\hat{q}_s^{(t+\Delta t)}, q_s^{(t+\Delta t)}$ are the model prediction and actual position of sample s.

3 METHOD

3.1 Framework

The core of the EGNN framework lies in the invariant inner product. Thus, our goal is to replace it via an invariant function to point group and consequently relax the $E(n)$ equivariant message passing. In this section, we first introduce the point group and then elaborate on our equivariant message-passing layer.

*3.1.1 **Point group**.* The symmetry of dynamics is highly affected by their boundary conditions, which only have finite discrete elements in many circumstances. Mathematically, it can be described by a point group P [4], the set of isometries that maps the boundary structure to itself. As isometries, point groups are subgroups of the orthogonal group $O(n)$, which will be rotations and reflections. Different symmetric boundaries correspond to different point groups. For example, the symmetry of a square is described by the $D4$ group, which consists of identity, 90°/180°/270° rotation, horizontal/vertical reflection, and reflection of two diagonal lines, and the point group of a non-rectangular parallelogram is D_1, which only contains the identity and 180° rotation.
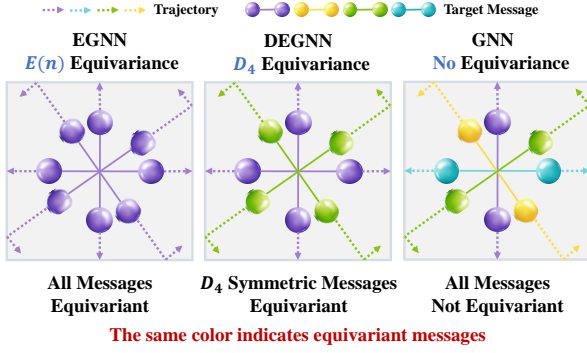
Figure 2: Examples of N-body system trajectories. The square is the boundary where its symmetry is described by the D_4 group, a specific point group. EGNN and GNN fail to distinguish the interactions of the objects that result in different dynamics, while DEGNN successfully maps different messages to different equivariant embeddings.

3.1.2 P-equivariant message embeddings. It is challenging to derive a straightforward P-equivariant function. Nevertheless, note that the inner product in Eq. 3 is essentially a function $\phi : \{Ox + t\}_{O,t \in E(n)} \rightarrow \mathbb{R}^n$ that maps all transformed geometric vectors $Ox + t$ in $E(n)$ to the same value $x^T x$. Inspired by this insight, we relax equivariance to a point group P via only reducing the function input to transformed geometric vectors in $\{Ox\}_{O \in P}$. Specifically, we first enhance the input using the point group to generate a set of transformed geometric vectors. Then, the function ϕ converts this set of vectors into the same embedding. Consequently, the P-equivariant function μ_P is defined as follows:

$$\mu_P(x, h) = (q_i - q_j)\phi(\{Ox\}_{O \in P}, h). \tag{8}$$

Here x is **symmetry-breaking** features. For example, they can be $(q_i, q_j, \dot{q}_i, \dot{q}_j)$ or $(q_i, q_j, \dot{q}_i, \dot{q}_j, q_i - q_j, ||q_i - q_j||^2)$. Such construction leads to P-equivariance, otherwise employing a set of $E(n)$-invariant features (e.g., $||x_i - x_j||^2$ or $||v_i||^2$) will enhance the symmetry to $E(n)$-equivariance.

REMARK. *The feature $\{Ox\}_{O \in P}$ is invariant to the point group P.*

Since the set is permutation-invariant and the group is closure (the result of two elements within a group is still an element of the group), for any transformation $O \in P$, the input x and Ox result in the same $\{Ox\}_{O \in P}$. That is, let $O_1, O_2 \in P$ be transformations in point group P, we have $O_1 O_2 x \in \{Ox\}_{O \in P}$. Thus, we immediately have the following theory:

THEOREM 3.1. *For arbitrary rotation or reflection matrix $O \in P$, if ϕ is a permutation-invariant function, the function μ_P satisfies the P-equivariance:*

$$\mu_P(Ox, h) = O\mu_P(x, h). \tag{9}$$

The proof is provided in Appendix A.1 where we show that ϕ is invariant to P and by multiplying it with radial directions $q_i - q_j$, we achieve P-equivariant message embedding.

Finally, following the same equivariant node updating functions (i.e., Eq. 6) and training strategy, we obtain a message-passing layer equivariant to a given point group P.
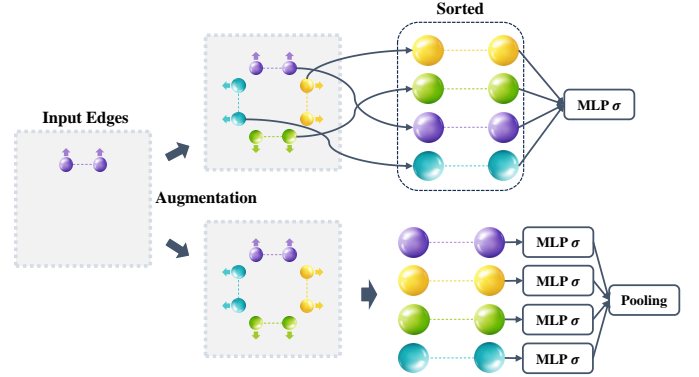

Figure 3: Two realizations of permutation-invariant embedding functions. We omit velocity features in MLP inputs for simplicity. Ranking-based methods rearrange the unordered set to ordered features while pooling-based approaches utilize a permutation-invariant function to aggregate features.

3.1.3 Comparison with existing GNNs. Figure 2 compares the equivariant message functions of non-equivariant GNN, EGNN, and DEGNN. Since EGNN is equivariant to rotations, it will map all isometric interactions to equivariant embeddings. And without equivariant constraints, GNN tends to map all distinct interactions to different embeddings. Thus, they all fail to distinguish interactions that lead to different trajectories in Figure 2. In contrast, DESIGN only maintains the necessary symmetry and successfully learns the message embeddings.

Equivariant inductive biases are crucial to the representation ability of GNNs since they have to maintain symmetry when performing non-linear transformations. For example, EGNN-like models (e.g., GMN [23]) rely on the inner product of translation-invariant geometric features (e.g., relative distance, relative velocity, and absolute velocity), which is not able to capture all types of interactions (e.g., $q_i q_j$ or $q_i \dot{q}_j$). As illustrated in Eq. 8, by relaxing the equivariant group from $E(n)$ to point groups, DEGNN can utilize any symmetry-breaking feature set, which is beneficial to approximate the underlying object interaction functions [7, 18].

3.2 Implementation

In this section, we present two realizations of the permutation-invariant function ϕ - ranking-based and pooling-based. Our objective is to generate a representation of the geometric feature set. Figure 3 illustrates their designs.

3.2.1 Ranking. Since MLP is permutation-sensitive, directly concatenating a set of geometric features as input will produce different outputs when the permutation is changed. To ensure unique embedding for any permutations, we could first sort the features before feeding them into the MLP. Thus, we have:

$$\phi(\{(Ox, h)\}_{O \in P}) = \sigma(\text{Sort}(\{(Ox)\}_{O \in P}), h). \tag{10}$$

Here we sort these vectors by the magnitude of each dimension in a descending order. However, the choice of ordering may affect the model performance and manually finding the optimal ordering is challenging and time-consuming.

3.2.2 ***Permutation-invariant Pooling***. Since the augmented geometric features could be viewed as independent spatial points, an efficient way to obtain permutation-invariant embedding is through embedding pooling. We first generate embeddings of each point and then employ a permutation-invariant function to aggregate their embeddings. Specifically, the pooling function is defined as follows:

$$\phi(\{(Ox, h)\}_{O \in P}) = \text{Aggregation}(\{\sigma((Ox), h)\}_{O \in P}), \quad (11)$$

where σ is an MLP and the common choices of the aggregation function are mean or sum.

To better describe the symmetry environment in practice, we further propose to employ self-attention to aggregate these geometric embeddings. Let e_i denotes the edge embedding of transformation $O_i \in P$. Then the computation of message embedding m is:

$$
\begin{aligned}
m &= \sum_{j=1}^{d} \alpha_{ij} \sigma_v(e_j), \\
\alpha_{ij} &= \frac{\exp(< \sigma_q(e_i), \sigma_k(e_j) >)}{\sum_{j=1}^{d} \exp(< \sigma_q(e_i), \sigma_k(e_j) >)},
\end{aligned}
\quad (12)
$$

where $< \cdot, \cdot >$ denotes the inner product of two vectors. $\sigma_q, \sigma_k, \sigma_v$ are MLPs and α_{ij} represents the attention weight.

PROPOSITION 3.2. *The self-attention pooling function is permutation-invariant.*

The proof is reported in the Appendix A.2 where we show that changing the input permutation will not lead to different message embedding.

4 EXPERIMENTS

In this section, we extensively evaluate our methods on micro-level and macro-level dynamic systems to answer the following research questions:

- **RQ1**: How does the proposed model perform on different types of physical systems compared to the state-of-the-art baselines?
- **RQ2**: Can our model learn the discrete symmetry of the complex real-world dynamics?
- **RQ3**: What are the effects of discrete equivariant message-passing components and graph pooling strategies?
- **RQ4**: How powerful is our method when we vary training size and target interval?

4.1 Settings

4.1.1 ***Datasets***. We evaluate model performance on diverse micro-level and macro-level physical dynamics. The micro-level systems are particles and molecules:

- **Particle**: Two Simulated 5-body systems driven by Coulomb or gravitational forces. To evaluate the model performance on particle dynamics of different symmetries, we generate particle trajectories within the cube, square prism, and rectangle prism boundary, which are described by different point groups. We randomly generate 200/2000/2000 trajectories for training/validation/testing.

- **Molecule**: We further evaluate our model on two complex and large molecular simulation datasets [3] - LiPS and $Li_4P_2O_7$ contain 83 and 208 atoms within a rectangular prism and parallelepiped boundaries, respectively. 2000/2000/2000 frames are randomly selected for training/validation/testing.

Besides, we conduct experiments on macro-level agent dynamics as well. For these datasets, 70%/10%/20% frames are randomly selected for training/validation/testing. Their details are as follows:

- **Crowd**: We use public bidirectional crowd dynamic data that are built up by the Institute for Advanced Simulation 7:Civil Safety Research of ForschungszentrumJülich [6]. Pedestrians in these scenarios exhibit a high frequency of interactions. Based on whether they form social groups and the pedestrian number of starting positions, we classify the dataset into low-imbalanced individuals, low-imbalanced groups, high-imbalanced individuals, and high-imbalanced groups.
- **Vehicle**: The vehicle dynamics are obtained from HighD [31] dataset, which is recorded on multi-lane highway. The dataset includes 110,500 vehicles covering a total driving distance of 44,500 kilometers and a total of 147 driving hours. Besides, it comprises a total of six distinct highway scenarios and we conducted experiments for each scenario to validate the effectiveness of our model.

The dataset statistics are shown in Appendix Table 6 and Table 5. More details on N-body system generation, molecule data, and the figure illustration of macro-level systems are shown in Appendix B.

4.1.2 ***Baselines***. We compare DEGNN against various baselines: (1) non-equivariant GNN; (2) equivariant GNNs: Radial Field [29], EGNN [43], GMN [23], EqMotion [51]; (3) in addition, for particle and molecular systems, we further compare DEGNN with TFN [46], SE(3) Transformer [15], and SAKE [49]; (4) for crowd and vehicle, we further compare DEGNN with trajectory forecasting models S-LSTM [1] and TransF [17], and graph simulation models GNS [42], and CrowdSim [44].

4.1.3 ***Implementation details***. we empirically find that the following hyper-parameters generally work well, and use them across all experiments: Batch size 100, the hidden dimension 64, weight decay 1×10^{-12}. All models are set to four layers. Due to insufficient GPU memory, the batch size for the EqMotion model is set to 50 for the LiPS dataset and 5 for the $Li_4P_2O_7$ dataset. The learning rates of the micro-level and macro-level datasets are 0.0003 and 0.0005. All models are trained for 5000 epochs with an early stopping strategy of 100. We use Mean Square Error (MSE) as our metric to measure the loss between prediction and the ground truth. The point group used in DEGNN can be found in Appendix 4. All models are implemented based on Pytorch and PyG library [12], trained on GeForce RTX 4090 GPU.

4.2 Overall Performance Comparison (RQ1)

In this section, we validate the effectiveness of our model by comparing it to the baselines.

4.2.1 ***Micro-level***. Table 1 presents the comparison of different dynamic modeling methods on particle and molecule systems. Based on the results, we have the following observations:

Table 1: MSE of all models on micro-level systems. Bold font indicates the best result and Underline is the strongest baseline. We report both mean and standard deviation that are computed over 5 runs. "OOM" denotes out of memory. The official implementation of Eqmotion report "Nan" on $Li_4P_2O_7$ dataset. In particle systems, (a,b,c) denotes the side length of the boundary, where (5,5,5) and (4,4,4) are cube, (5,4,4) and (5,4,3) are square prism and rectangle prism.

Model	Charged Particle ($\times 10^{-1}$)				Gravitational Particle ($\times 10^{-1}$)				Molecular($\times 10^{-1}$)	
	(5,5,5)	(5,4,4)	(5,4,3)	(4,4,4)	(5,5,5)	(5,4,4)	(5,4,3)	(4,4,4)	LiPS	$Li_4P_2O_7$
GNN	3.12± 0.49	4.46± 0.15	3.24± 0.13	3.14±0.19	3.81± 0.54	<u>3.32± 0.37</u>	<u>3.21± 0.27</u>	3.20± 0.61	<u>4.39± 0.71</u>	**8.59± 0.51**
TFN	5.03± 0.33	5.53± 0.65	6.15± 0.30	5.53±0.65	6.70± 0.17	7.49± 0.24	8.07± 0.55	7.74± 0.32	OOM	OOM
Radial Field	1.94± 0.03	2.58± 0.07	3.04± 0.01	2.77± 0.10	4.76± 0.03	5.51± 0.06	5.85± 0.12	5.29± 0.20	10.0± 0.06	13.3± 0.02
SE(3)-Tr.	4.65± 0.08	5.15± 0.01	5.70± 0.10	5.29± 0.12	9.03± 0.20	10.1± 0.27	10.54± 0.23	10.3± 0.30	OOM	OOM
EGNN	4.92± 0.12	5.62± 0.03	6.33± 0.05	5.67± 0.02	10.3± 0.04	11.5± 0.03	12.8± 0.06	11.5± 0.04	7.97± 0.49	13.8± 0.04
GMN	<u>1.82± 0.13</u>	<u>2.23± 0.03</u>	<u>2.64± 0.07</u>	<u>2.41± 0.05</u>	<u>3.78± 0.15</u>	4.48± 0.13	5.07± 0.10	4.44± 0.20	6.26± 0.65	13.0± 0.04
SAKE	4.22± 2.42	3.88± 0.55	3.87± 0.05	3.80± 0.10	5.88± 0.33	7.21± 0.34	7.60± 0.37	6.89± 0.40	11.5± 9.76	12.8± 0.27
EqMotion	2.14± 0.24	2.70± 0.34	3.27± 0.25	2.96± 0.34	4.23± 0.24	4.95± 0.18	5.62± 0.18	4.90± 0.19	15.2± 9.8	Nan
DEGNN	**1.23± 0.06**	**1.43± 0.07**	**1.24± 0.05**	**1.54± 0.07**	**1.46± 0.17**	**1.73± 0.21**	**1.56± 0.09**	**2.02± 0.18**	**4.36± 0.31**	<u>9.86± 0.97</u>

Table 2: MSE of all models on macro-level systems. Bold font indicates the best result and Underline is the strongest baseline. We report both mean and standard deviation that are computed over 5 runs.

Model	Crowd ($\times 10^{-3}$)				Vehicle ($\times 10^{-2}$)					
	Ind.-Low	Ind.-High	Group-Low	Group-High	Highway 1	Highway 2	Highway 3	Highway 4	Highway 5	Highway 6
GNN	1.43± 0.25	<u>0.86±0.24</u>	<u>0.8± 0.28</u>	1.20± 0.10	<u>0.61± 0.28</u>	0.20± 0.07	<u>0.67± 0.29</u>	<u>1.02± 0.77</u>	<u>0.39± 0.19</u>	2.27± 1.27
S-LSTM	3.43± 0.01	1.89±0.09	3.83± 1.25	2.57±0.49	1.66± 0.05	0.49±0.03	0.78± 0.03	1.72±0.01	1.22± 0.05	1.23±0.07
Radial Field	3.21± 1.16	2.43±0.17	3.76± 0.45	3.39± 0.87	512± 395	303± 270	180± 127	44.0± 34.9	477± 301	240± 211
GNS	3.7± 0.49	4.14±1.02	5.25± 1.61	3.66±0.71	1.58± 0.78	3.00± 2.46	1.54± 0.55	5.13± 2.07	4.53± 5.03	2.43± 0.40
TransF	1.59±0.07	1.76±0.04	2.72±0.04	1.89±0.04	1.61±0.87	0.59±0.08	0.70±0.25	1.36±0.70	0.68±0.30	<u>0.94±0.34</u>
EGNN	5.94± 1.05	2.17±0.84	2.02± 0.32	1.48± 0.30	24.9± 7.79	52.4± 27.4	42.6± 8.25	19.3± 6.38	46.1± 4.84	35.8± 23.8
GMN	<u>0.44± 0.07</u>	1.17±0.05	0.91± 0.36	<u>0.71± 0.48</u>	1.41± 0.76	0.74± 0.27	1.30± 0.45	2.48± 0.90	2.94± 2.00	5.20± 2.99
CrowdSim	3.60± 1.25	3.14±0.95	5.77± 0.90	2.93± 1.11	1.85± 0.67	2.39± 0.36	1.25± 0.45	11.3± 12.5	13.2± 24.3	3.52± 1.26
EqMotion	8.00± 7.34	5.98±1.22	6.27± 4.82	2.91± 2.54	29.8± 23.6	1.18± 2.01	39.4± 44.0	3.77± 7.3	541± 432	144± 247
DEGNN	**0.32± 0.04**	**0.41±0.04**	**0.71±0.13**	**0.36± 0.04**	**0.07± 0.01**	**0.04± 0.07**	**0.35± 0.35**	**0.56± 0.08**	**0.11± 0.02**	**0.27± 0.04**

- DEGNN outperforms all baselines on all particle datasets in a large gap. Compared with the best baseline, the average enhancements on the Charged and Gravity datasets are 0.9 and 1.7 respectively. For molecular systems, our model surpasses the performance of other models on the LiPS dataset while achieving comparable performance to the best baseline on the $Li_4P_2O_7$ dataset.
- The basic GNN performs better than several equivariant GNNs (e.g., TFN and EGNN), especially in Molecular systems. These results show that $E(n)$-equivariant constraint is too strong to model these dynamics, leading to a decrease in performance.
- In particle systems, as the boundary changes from the cube (5,5,5), square prim (5,4,4), to rectangle prism (5,4,3), the degree of discrete symmetry gradually decreases, indicating the size of the corresponding point group is reduced. In these systems, we can observe that the error of most models rises when the symmetry weakens. For instance, in Charged particle systems, the error of Emotion in (5,5,5), (5,4,4), and (5,4,3) boundaries are 2.14, 2.70, 3.27, respectively. In contrast, DEGNN maintains low error in different systems.
- Note that dynamics with small boundaries will have more boundary effects. By comparing the results on (5,5,5) and (4,4,4) systems, we can observe that almost all models achieve larger errors in

(4,4,4) systems, indicating these dynamics are more complex. DEGNN still achieves the best performance in such cases.

4.2.2 ***Macro-level***. Table 2 depicts the overall results on real-world crowd and vehicle datasets. The observations are as follows:

- GNN outperforms all baselines in 7 of 10 scenarios, indicating incorporating continuous equivariance does not enhance model performance in most cases. On the contrary, DEGNN outperforms all baseline models, with the lowest errors on crowd and vehicle datasets. Such improvement verifies the effectiveness of encoding the discrete symmetry in the dynamic modeling of agents.
- Another discovery is that our model DEGNN exhibits a lower standard deviation compared to other equivariant NN models. In the Crowd dataset, DEGNN demonstrates better stability (0.06 average standard deviation) compared to EGNN (0.6), GMN (0.24), and EqMotion (4).

4.3 Generalization Experiments (RQ2)

In this section, we conduct two additional experiments across different directions and scenarios on vehicle datasets to validate the generalization ability of DEGNN.

4.3.1 ***Different directions***. In each scenario, models are trained on trajectories in one direction (from left to right) and evaluated
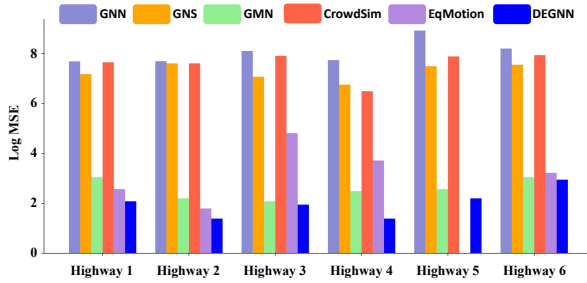
Figure 4: MSE of models on vehicle dataset. They are trained on trajectories from the left to right and tested on those from the right to left. (Official EqMotion reports "Nan" in Highway 5 datasets; $Log\ MSE = log(100 \times MSE + 1)$)

on trajectories in the opposite direction (from right to left). We compare DEGNN with Eqmotion, GNN, GMN, GNS, and CrowdSim. Figure 4 displays the result. We can find that without observing the opposite direction data, GNN, GNS, and CrowdSim achieve high errors, indicating their weak generalization ability in dynamic directions. $E(n)$-equivariant models exhibit much stronger generalization than the above models and DEGNN outperforms GMN and EqMotion by a significant margin, demonstrating the discrete equivariance achieves the best generalization performance.

4.3.2 Cross scenarios. Models are trained on trajectories from one scenario and evaluated on trajectories from other scenarios. We present the results of GNN, DEGNN, GMN, and GNS in Figure 5 where a lighter color indicates a lower error. From the figure, we can observe that all compared methods achieve high generalization errors in multiple cases, indicating they overfit the training data instead of learning the real interaction patterns. Compared to these methods, DEGNN demonstrates significantly better cross-scenario generalization capabilities.

4.4 Ablation Study (RQ3)

To illustrate the effectiveness of each model deisgn to the overall performance of our model, we compare the default settings of DEGNN with three model feature variants by replacing the feature in Eq. 8 with : (1) **E(n)**: $x = (||q_i - q_j||^2)$, which satisfied rotation, translation, and reflection equivariance; (2) **O(n)**: $x = (||q_i||^2, ||q_j||^2)$, which satisfied rotation and reflection equivariance; (3) **T(n)**: $x = (q_i - q_j)$, which satisfied translation equivariance, and three permutation invariant function variants: (4) **Ranking**: sort the feature vector based on each dimension of geometric vectors; (5) **Sum Pooling**: sum the feature embedding; (6) **Mean Pooling**: averaging the feature embedding; Table 3 shows the results on particle systems. The results of the other systems are shown in Table 7.

4.4.1 Effect of discrete equivariant message passing. We can find that variant **T(n)** exhibits the highest error among the three variants. For example, the average loss for **E(n)**, **O(n)** and **T(n)** in the Charged dataset is 2.07, 2.10, and 2.88 respectively. This discrepancy is mainly attributed to the fact that, in the presence of boundaries, the dynamic is not equivariant to translation. DEGNN
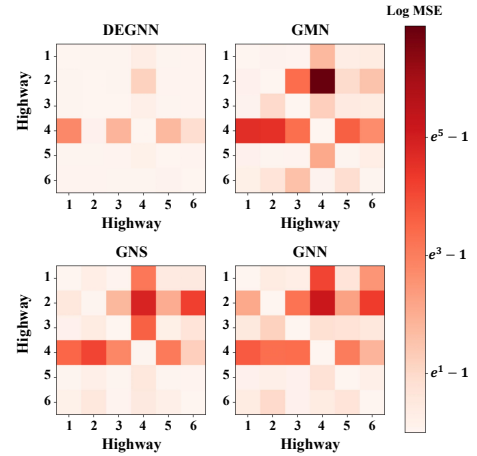
Figure 5: Generalization experiments across different scenarios on vehicle dynamic datasets. Row/Column denotes the training/testing scenarios. ($Log\ MSE = log(MSE + 1)$)

outperforms variants in all cases, demonstrating the effectiveness of discrete equivariance.

4.4.2 Effect of graph pooling strategies. Among the three implementations of DEGNN, sum pooling demonstrates the lowest model performance. In scenarios with high symmetry, such as Charged (5,5,5) and Charged (4,4,4), the losses reach 25 and 13, respectively. The reason is that the size of O_h (i.e., the point group of Cube) is large which makes the sum pooling unstable. On the contrary, the self-attention pooling is stable and consistently outperforms other variants.

4.5 Sensitivity Analysis (RQ4)

In this section, we conducted experiments on vehicle datasets by varying the training set size and target time interval.

4.5.1 Performance w.r.t. training size. The first row of Figure 6 illustrates the results of leveraging 10%, 30%, 50%, and 70% of the dataset for training. As the training dataset size decreases to 10%, the loss of DEGNN consistently remains below 10, while other models may increase to 40 or even 50. Such results show that DEGNN is data efficient.

4.5.2 Performance w.r.t. target interval. Figure 6 depicts how the model loss changes with varying prediction frame intervals, ranging from 50 frames (2s) to 200 (8s) frames. With the increment in interval frames, DEGNN consistently sustains a loss below 1, whereas other models might escalate to 2 or 3. Such results demonstrate that DEGNN can generalize to long-term predictions.

5 RELATED WORK

In this section, we review the most related works from the perspective of physical systems.

Particle and Molecule. Graph neural networks [34] have shown promising performance in learning complex dynamics of N-body systems (e.g., particles and molecules). IN [2], NRI [28], and HRN [38] are pioneer works that model physical objects and relations as

Table 3: Ablation Studies on model designs. Bold font indicates the best result. We report both mean and standard deviation that are computed over 5 runs. In molecular systems, (a,b,c) denotes the side length of the boundary, where (5,5,5) and (4,4,4) are cube, (5,4,4) and (5,4,3) are square prism and rectangle prism.

Model	Charged Particle ($\times 10^{-1}$)				Gravitational Particle ($\times 10^{-1}$)			
	(5,5,5)	(5,4,4)	(5,4,3)	(4,4,4)	(5,5,5)	(5,4,4)	(5,4,3)	(4,4,4)
E(n)	1.78±0.15	2.12±0.14	2.01± 0.06	2.36±0.13	2.32±0.05	2.83± 0.12	3.35±0.19	3.29±0.17
O(n)	2.46±0.82	2.13±0.15	1.79± 0.07	2.03±0.26	1.73±0.13	3.49± 0.92	2.27±0.08	2.52±0.48
T(n)	2.79±0.04	2.80± 0.13	2.18± 0.07	3.76±0.06	5.98±0.68	3.57± 0.24	3.35±0.09	6.63±0.08
Ranking	1.95± 0.22	1.63±0.10	1.51±0.07	2.11±0.18	2.06±0.09	1.79± 0.10	1.75± 0.08	2.36±0.11
Sum Pooling	25.3± 12.2	2.47±0.64	1.82± 0.09	13.0± 5.15	13.55±4.77	4.16± 0.67	3.21± 0.53	52.9±61.4
Mean Pooling	2.09± 0.23	2.39± 0.12	1.52± 0.14	2.10± 0.11	2.13±0.42	2.10± 0.24	1.78± 0.19	2.60±0.24
Default	1.23± 0.06	1.43± 0.07	1.24± 0.05	1.54± 0.07	1.46± 0.17	1.73± 0.21	1.56± 0.09	2.02± 0.18
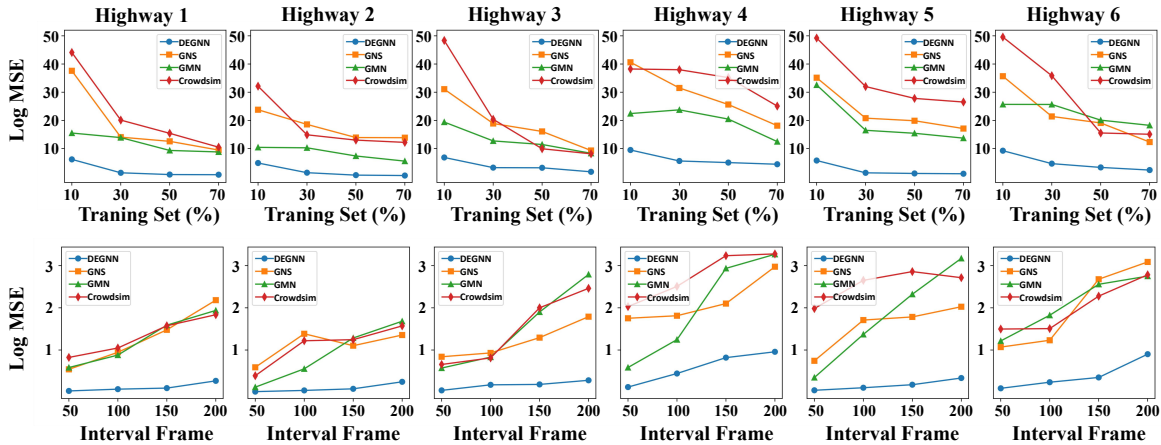
Figure 6: The rollout-MSE curves on 6 vehicle scenarios. DEGNN achieves the best performance.($Log\ MSE = log\ (100 \times MSE + 1)$)

graphs and learn their interaction and evolution. Recent studies have considered the underlying physical symmetry of systems. TFN [46], SE(3) Transformer [15], and SEGNN [5] utilize spherical harmonics to construct rotation equivariant models with higher-order geometric representations. Several works [13, 25] leverage the Lie convolution to extend equivariance on Lie groups. In addition to these methods, a series of studies [23, 43, 49–51] apply scalarization techniques to introduce equivariance into the message-passing process in GNNs. Nevertheless, these methods incorporate continuous equivariant constraints, which are too strong for various physical systems.

Crowd. Traditional crowd dynamic modeling models include rule-based [39], force-based [22, 40], and velocity-based models [14]. Nevertheless, the designed formulas are insufficient to model complex and uncertain real-world dynamics. With the development of deep learning, recent studies [54, 54] seek neural solutions to improve model performance. For example, CrowdSim [44] learns crowd trajectories under the framework of GNS [42]. However, these methods ignore the symmetry of crowd dynamics, leading to suboptimal generalization ability.

Vehicle. Current methods for vehicle trajectory prediction are mainly based on sequence models such as recurrent models [8, 9, 41] and transformers [17, 55]. ECCO [48] embeds the rotation symmetries in continuous convolution to improve data efficiency

and make physically consistent predictions. However, it still follows the continuous equivariance, which constrains its representation ability.

6 CONCLUSION AND FUTURE WORK

In this work, we propose a general DEGNN framework for learning micro-level and macro-level physical dynamics in boundary environments. It is proved to be equivariant to the given point group via data augmentation and permutation-invariant embedding functions. We propose ranking-based and pooling-based implementations of DEGNN. Extensive experiments conducted on various physical dynamics show that DEGNN outperforms all competing methods. Generalization experiments and sensitivity analysis demonstrate the superior generalization ability of our model. Ablation studies further validate the effectiveness of our model designs. In the future, we are interested in (1) extending our work to solve multiple physical dynamics jointly; (2) evaluating DEGNN on multi-step trajectories; (3) learning a ranking function for ranking-based DEGNN.

REFERENCES

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. 2016. Social LSTM: Human Trajectory Prediction in Crowded Spaces. In *CVPR*. 961–971.

[2] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. 2016. Interaction networks for learning about objects, relations and physics. *Advances in neural information processing systems* 29 (2016).

[3] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. 2022. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications* 13, 1 (2022), 2453.

[4] Christopher Bradley and Arthur Cracknell. 2010. *The mathematical theory of symmetry in solids: representation theory for point groups and space groups.* Oxford University Press.

[5] Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J. Bekkers, and Max Welling. 2022. Geometric and Physical Quantities improve E(3) Equivariant Message Passing. In *ICLR*.

[6] Shuchao Cao, Armin Seyfried, Jun Zhang, Stefan Holl, and Weiguo Song. 2017. Fundamental diagrams for multidirectional pedestrian flows. *Journal of Statistical Mechanics: Theory and Experiment* 2017, 3 (2017), 033404.

[7] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *DLRS@RecSys*. 7–10.

[8] Nachiket Deo and Mohan M. Trivedi. 2018. Convolutional Social Pooling for Vehicle Trajectory Prediction. In *CVPR*. 1468–1476.

[9] Nachiket Deo and Mohan M. Trivedi. 2018. Multi-Modal Trajectory Prediction of Surrounding Vehicles with Maneuver based LSTMs. In *IV*. 1179–1184.

[10] Jacob D Durrant and J Andrew McCammon. 2011. Molecular dynamics simulations and drug discovery. *BMC biology* 9, 1 (2011), 1–9.

[11] Tao Feng, Ziqi Gao, Jiaxuan You, Chenyi Zi, Yan Zhou, Chen Zhang, and Jia Li. 2024. Deep Reinforcement Learning for Modelling Protein Complexes. *arXiv preprint arXiv:2405.02299* (2024).

[12] Matthias Fey and Jan Eric Lenssen. 2019. Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428* (2019).

[13] Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. 2020. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *ICML*. PMLR, 3165–3176.

[14] Paolo Fiorini and Zvi Shiller. 1998. Motion planning in dynamic environments using velocity obstacles. *IJRR* 17, 7 (1998), 760–772.

[15] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. 2020. Se (3)-transformers: 3d roto-translation equivariant attention networks. *NeurIPS* (2020).

[16] Ziqi Gao, Chenran Jiang, Jiawen Zhang, Xiaosen Jiang, Lanqing Li, Peilin Zhao, Huanming Yang, Yong Huang, and Jia Li. 2023. Hierarchical graph learning for protein–protein interaction. *Nature Communications* 14, 1 (2023), 1093.

[17] Francesco Giuliari, Irtiza Hasan, Marco Cristani, and Fabio Galasso. 2021. Transformer networks for trajectory forecasting. In *ICPR*. IEEE, 10335–10342.

[18] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *IJCAI*. 1725–1731.

[19] Jiaqi Han, Jiacheng Cen, Liming Wu, Zongzhao Li, Xiangzhe Kong, Rui Jiao, Ziyang Yu, Tingyang Xu, Fandi Wu, Zihe Wang, et al. 2024. A Survey of Geometric Graph Neural Networks: Data Structures, Models and Applications. *arXiv preprint arXiv:2403.00485* (2024).

[20] Jiaqi Han, Wenbing Huang, Tingyang Xu, and Yu Rong. 2022. Equivariant Graph Hierarchy-Based Neural Networks. In *NeurIPS*.

[21] Jiaqi Han, Yu Rong, Tingyang Xu, and Wenbing Huang. 2022. Geometrically Equivariant Graph Neural Networks: A Survey. *CoRR* abs/2202.07230 (2022).

[22] Dirk Helbing and Peter Molnar. 1995. Social force model for pedestrian dynamics. *Physical review E* 51, 5 (1995), 4282.

[23] Wenbing Huang, Jiaqi Han, Yu Rong, Tingyang Xu, Fuchun Sun, and Junzhou Huang. 2022. Equivariant Graph Mechanics Networks with Constraints. In *ICLR*.

[24] Zijie Huang, Yizhou Sun, and Wei Wang. 2023. Generalizing graph ode for learning complex system dynamics across environments. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 798–809.

[25] Michael J Hutchinson, Charline Le Lan, Sheheryar Zaidi, Emilien Dupont, Yee Whye Teh, and Hyunjik Kim. 2021. Lietransformer: equivariant self-attention for lie groups. In *ICML*. PMLR, 4533–4543.

[26] Anders Johansson, Dirk Helbing, Habib Z Al-Abideen, and Salim Al-Bosta. 2008. From crowd dynamics to crowd safety: a video-based analysis. *Advances in Complex Systems* 11, 04 (2008), 497–527.

[27] Sékou-Oumar Kaba and Siamak Ravanbakhsh. 2022. Equivariant Networks for Crystal Structures. In *NeurIPS*.

[28] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. 2018. Neural relational inference for interacting systems. In *International conference on machine learning*. PMLR, 2688–2697.

[29] Jonas Köhler, Leon Klein, and Frank Noé. 2019. Equivariant flows: sampling configurations for multi-body systems with symmetric energies. *arXiv preprint arXiv:1910.00753* (2019).

[30] Parth Kothari, Sven Kreiss, and Alexandre Alahi. 2022. Human Trajectory Forecasting in Crowds: A Deep Learning Perspective. *IEEE Trans. Intell. Transp. Syst.* 23, 7 (2022), 7386–7400.

[31] Robert Krajewski, Julian Bock, Laurent Kloeker, and Lutz Eckstein. 2018. The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems. In *ITSC*.

[32] Yuhan Li, Zhixun Li, Peisong Wang, Jia Li, Xiangguo Sun, Hong Cheng, and Jeffrey Xu Yu. 2023. A survey of graph meets large language model: Progress and future directions. *arXiv preprint arXiv:2311.12399* (2023).

[33] Zijie Li and Amir Barati Farimani. 2022. Graph neural network-accelerated Lagrangian fluid simulation. *Computers & Graphics* 103 (2022), 201–211.

[34] Yang Liu, Liang Chen, Xiangnan He, Jiaying Peng, Zibin Zheng, and Jie Tang. 2020. Modelling high-order social relations for item recommendation. *IEEE Transactions on Knowledge and Data Engineering* 34, 9 (2020), 4385–4397.

[35] Yang Liu, Jiashun Cheng, Haihong Zhao, Tingyang Xu, Peilin Zhao, Fugee Tsung, Jia Li, and Yu Rong. 2023. SEGNO: Generalizing Equivariant Graph Neural Networks with Physical Inductive Biases. In *ICLR*.

[36] Yang Liu, Yu Rong, Zhuoning Guo, Nuo Chen, Tingyang Xu, Fugee Tsung, and Jia Li. 2023. Human mobility modeling during the COVID-19 pandemic via deep graph diffusion infomax. In *AAAI*, Vol. 37. 14347–14355.

[37] Hehuan Ma, Yatao Bian, Yu Rong, Wenbing Huang, Tingyang Xu, Weiyang Xie, Geyan Ye, and Junzhou Huang. 2022. Cross-dependent graph neural networks for molecular property prediction. *Bioinformatics* 38, 7 (2022), 2003–2009.

[38] Damian Mrowca, Chengxu Zhuang, Elias Wang, Nick Haber, Li F Fei-Fei, Josh Tenenbaum, and Daniel L Yamins. 2018. Flexible neural representation for physics prediction. *Advances in neural information processing systems* 31 (2018).

[39] Craig W Reynolds. 1987. Flocks, herds and schools: A distributed behavioral model. In *SIGGRAPH*. 25–34.

[40] Priscila Saboia and Siome Goldenstein. 2012. Crowd simulation: applying mobile grids to the social force model. *The Visual Computer* 28 (2012), 1039–1048.

[41] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. 2020. Trajectron++: Dynamically-Feasible Trajectory Forecasting with Heterogeneous Data. In *ECCV*, Vol. 12363. 683–700.

[42] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W. Battaglia. 2020. Learning to Simulate Complex Physics with Graph Networks. In *ICML*, Vol. 119. 8459–8468.

[43] Víctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. 2021. E (n) equivariant graph neural networks. In *ICML*. 9323–9332.

[44] Hongzhi Shi, Quanming Yao, and Yong Li. 2023. Learning to Simulate Crowd Trajectories with Graph Networks. In *WWW*. 4200–4209.

[45] Michio Suzuki. 1986. *Group theory II*. Springer.

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

[47] Oliver T Unke, Stefan Chmiela, Huziel E Sauceda, Michael Gastegger, Igor Poltavskyi, Kristof T Schütt, Alexandre Tkatchenko, and Klaus-Robert Müller. 2021. Machine Learning Force Fields. *Chemical Reviews* (2021).

[48] Robin Walters, Jinxi Li, and Rose Yu. 2021. Trajectory Prediction using Equivariant Continuous Convolution. In *ICLR*. OpenReview.net.

[49] Yuanqing Wang and John D. Chodera. 2023. Spatial Attention Kinetic Networks with E(n)-Equivariance. In *ICLR*. OpenReview.net.

[50] Liming Wu, Zhichao Hou, Jirui Yuan, Yu Rong, and Wenbing Huang. 2024. Equivariant Spatio-Temporal Attentive Graph Networks to Simulate Physical Dynamics. *Advances in Neural Information Processing Systems* 36 (2024).

[51] Chenxin Xu, Robby T Tan, Yuhong Tan, Siheng Chen, Yu Guang Wang, Xinchao Wang, and Yanfeng Wang. 2023. EqMotion: Equivariant Multi-agent Motion Prediction with Invariant Interaction Reasoning. In *CVPR*. 1410–1420.

[52] Jinyu Yang, Peilin Zhao, Yu Rong, Chaochao Yan, Chunyuan Li, Hehuan Ma, and Junzhou Huang. 2021. Hierarchical graph capsule network. In *AAAI*.

[53] Shanwen Yang, Tianrui Li, Xun Gong, Bo Peng, and Jie Hu. 2020. A review on crowd simulation and modeling. *Graph. Model.* 111 (2020), 101081.

[54] Zihan Yu, Guozhen Zhang, Yong Li, and Depeng Jin. 2023. Understanding and Modeling Collision Avoidance Behavior for Realistic Crowd Simulation. In *CIKM*.

[55] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris Kitani. 2021. AgentFormer: Agent-Aware Transformers for Socio-Temporal Multi-Agent Forecasting. In *ICCV*.

[56] Haihong Zhao, Aochuan Chen, Xiangguo Sun, Hong Cheng, and Jia Li. 2024. All in One and One for All: A Simple yet Effective Method towards Cross-domain Graph Pretraining. *arXiv preprint arXiv:2402.09834* (2024).

[57] Haihong Zhao, Bo Yang, Jiaxu Cui, Qianli Xing, Jiaxing Shen, Fujin Zhu, and Jiannong Cao. 2023. Effective fault scenario identification for communication networks via knowledge-enhanced graph neural networks. *IEEE Transactions on Mobile Computing* (2023).

[58] Haihong Zhao, Chenyi Zi, Yang Liu, Chen Zhang, Yan Zhou, and Jia Li. 2024. Weakly Supervised Anomaly Detection via Knowledge-Data Alignment. In *Proceedings of the ACM on Web Conference 2024*. 4083–4094.
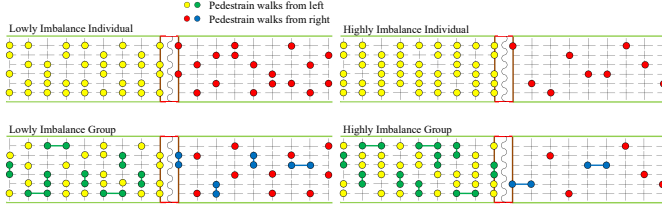
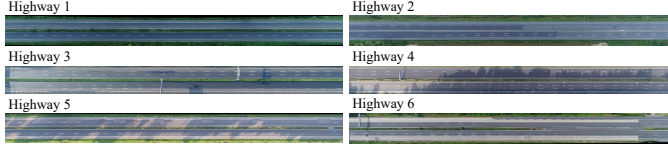**Figure 7: Illustrations of starting positions of crowd scenes.**

Figure 8: Representative highway scene.

A PROOF

A.1 Proof of Theorem 3.1

Since the set is unordered and the point group satisfied closure, let $O_1, O_2 \in P$ be transformations in point group P, we have $O_1O_2 x \in \{Ox\}_{O \in P}$. Given that ϕ is a permutation-invariant function, for any transformation $O_1 \in P$, we have

$$\begin{aligned} \mu_P(O_1 x, h) &= (O_1 q_i - O_1 q_j)\phi(\{O_2 O_1 x\}_{O_1, O_2 \in P}, h) \\ &= O_1(q_i - q_j)\phi(\{Ox\}_{O \in P}, h) = O_1\mu_P(x, h) \end{aligned} \tag{13}$$

Therefore, μ_P is equivariant to the point group P if ϕ is permutation-invariant.

A.2 Proof of Proposition 3.2

For simplicity and with no loss of generality, the message embedding in Eq. 12 can be rewritten as

$$m = \sum_{j=1}^{d} \frac{\exp(< \sigma_q(e_i), \sigma_k(e_j) >)\sigma_v(e_j)}{\sum_{j=1}^{d} \exp(< \sigma_q(e_i), \sigma_k(e_j) >)}. \tag{14}$$

Then the message embedding m' for any permutation $\pi \in \Pi_d$ is:

$$\begin{aligned} m' &= \sum_{j=1}^{d} \frac{\exp(< \sigma_q(e_{\pi(i)}), \sigma_k(e_{\pi(j)}) >)\sigma_v(e_{\pi(j)})}{\sum_{j=1}^{d} \exp(< \sigma_q(e_{\pi(i)}), \sigma_k(e_{\pi(j)}) >)} \\ &= \sum_{j=1}^{d} \frac{\exp(< \sigma_q(e_{\pi(i)}), \sigma_k(e_{\pi(j)}) >)\sigma_v(e_{\pi(j)})}{\sum_{j=1}^{d} \exp(< \sigma_q(e_{\pi(i)}), \sigma_k(e_j) >)} \\ &= \sum_{j=1}^{d} \frac{\exp(< \sigma_q(e_{\pi(i)}), \sigma_k(e_j) >)\sigma_v(e_j)}{\sum_{j=1}^{d} \exp(< \sigma_q(e_{\pi(i)}), \sigma_k(e_j) >)}. \end{aligned} \tag{15}$$

The second and third equation hold because sum is permutation-invariant. Finally, since $e_{\pi(i)} = e_i$, $m' = m$.

B MORE DETAILS ON DATASETS

B.0.1 **Charged N-body system**. We adopt the 3D N-body simulation code introduced by [43], which builds upon the 2D implementation from [28] and extends it to three dimensions. System trajectories are generated at a 0.02 time step and are constrained with four different sizes of virtual boxes: $5 \times 5 \times 5$, $5 \times 4 \times 4$, $5 \times 4 \times$

3, and $4 \times 4 \times 4$, respectively. Initial locations are sampled from a Gaussian distribution (mean $\mu = 0$, standard deviation $\sigma = 1$), and the initial velocity is a random vector with a norm of 0.5.

B.0.2 **Gravity N-body system**. The gravitational N-body systems code is sourced from [5], implemented within the same framework as the aforementioned charged N-body systems. System trajectories are generated at a time step of 0.02, constrained within four different sizes of virtual boxes $5 \times 5 \times 5$, $5 \times 4 \times 4$, $5 \times 4 \times$ 3, and $4 \times 4 \times 4$, respectively. Initial locations are sampled from a Gaussian distribution (mean $\mu = 0$, standard deviation $\sigma = 1$), and the initial velocity is a random vector with a norm of 0.5. Particle mass is consistently set to one.

Table 4: The Point group for each system.

System	Point group	#Elements	#Dimension
Particle-Cube	O_h	48	3
Particle-Square prim	D_{4h}	16	3
Particle-Rectangle prism	D_{2h}	8	3
LiPS	C_i	2	3
$Li_4P_2O_7$	D_{2h}	8	3
Crowd	D_2	4	2
Vehicle	D_2	4	2

Table 5: Dataset statistics for Crowd and Vehicle dataset.

Dataset	#Sample	Mean #Ped.	Max #Ped.	Target frame
Ind.-Low	1938	23	42	10
Ind.-High	1961	23	37	10
Group-Low	2121	25	44	10
Group-High	2142	24	42	10
Highway 1	2000	10	19	100
Highway 2	2000	9	13	100
Highway 3	2000	13	22	100
Highway 4	2000	31	39	100
Highway 5	2000	13	26	100
Highway 6	2000	20	30	100

Table 6: Dataset statistics for particle and molecular dataset.

Dataset	#Sample	#Atom	Target frame
Particle	4200	5	10
LiPS	6000	83	10
$Li_4P_2O_7$	6000	208	10

B.0.3 **LiPS**. Lithium phosphorus sulfide (LiPS) is a crystalline superionic lithium conductor related to battery development. We adopt this dataset from [3]. The LiPS data consists of 83 atoms. The datasets comprise 25,001 frames. We randomly select 2000/2000/2000 samples for the training, validation, and test sets respectively in the first 6000 frame.

B.0.4 **$Li_4P_2O_7$**. The $Li_4P_2O_7$ were generated using an ab-initio meltquench MD simulation. We adopt this dataset from [3]. A stoichiometric crystal of 208 atom interact with each other in a periodic box of $10.4 \times 14.0 \times 16$. We adopt this dataset from [3]. The datasets comprise 25,000 frames. We randomly select 2000/2000/2000 samples for the training, validation, and test sets respectively in the first 6000 frame.

B.0.5 **Crowd**. we classify the dataset into low-imbalanced individuals, low-imbalanced groups, high-imbalanced individuals, and

Table 7: Ablation Studies on model designs. Bold font indicates the best result. We report both mean and standard deviation that are computed over 5 runs.

Model	Crowd ($\times 10^{-3}$)				Vehicle ($\times 10^{-2}$)						Molecular($\times 10^{-1}$)	
	Ind.-Low	Ind.-High	Group-Low	Group-High	Highway 1	Highway 2	Highway 3	Highway 4	Highway 5	Highway 6	LiPS	Li$_4$P$_2$O$_7$
E(n)	0.65± 0.09	0.96±0.08	1.09± 0.24	0.98± 0.09	0.05±0.02	0.11±0.05	0.14±0.05	0.72±0.11	0.14±0.03	0.54±0.53	5.44± 0.33	12.1± 0.46
O(n)	0.61± 0.05	0.92±0.11	1.59± 0.21	1.00± 0.08	**0.04±0.04**	0.05±0.02	**0.13±0.08**	0.80±0.11	0.16±0.07	0.51±0.53	6.46± 0.33	10.7± 0.89
T(n)	0.83± 0.10	1.26±0.08	1.79± 0.14	1.11± 0.06	0.13±0.09	0.09±0.04	0.25±0.14	0.99±0.14	0.18±0.05	0.43±0.20	5.07± 0.08	10.8± 0.51
Ranking	0.34± 0.02	0.50±0.01	0.73± 0.09	0.37± 0.03	0.10±0.04	0.06±0.01	0.20±0.02	1.11±0.40	0.16±0.01	0.61±0.29	4.63± 1.19	**9.18± 0.74**
Sum Pooling	0.33± 0.02	0.42±0.02	**0.61± 0.09**	0.35± 0.03	1.95± 0.06	2.37± 0.97	3.26± 3.10	20.8± 15.1	5.46± 5.30	9.84± 2.56	4.37± 0.53	9.79± 0.97
Mean Pooling	0.33± 0.05	0.52±0.03	0.74± 0.13	**0.34± 0.04**	0.19±0.13	0.05±0.05	0.53±0.5	1.45± 0.59	0.14±0.06	0.78± 0.53	**4.11± 0.14**	9.80± 0.54
DEGNN	**0.32± 0.04**	**0.41±0.04**	0.71±0.13	0.36± 0.04	0.07± 0.04	**0.04± 0.01**	0.19± 0.05	**0.56± 0.08**	**0.11± 0.02**	**0.27± 0.04**	4.36± 0.31	9.86± 0.97

Table 8: Forward time in seconds for a batch size of 100 samples in LiPS dataset on GeForce RTX 4090 GPU.

Model	GNN	S-LSTM	TFN	Radial Field	GNS	TransF	SE(3)-Trans.	EGNN	GMN	CrowdSim	SAKE	EqMotion	DEGNN
Time (s)	0.0019	0.00047	OOM	0.0023	0.0025	0.0029	OOM	0.0033	0.0039	0.0026	0.0063	0.1201	0.0088

Table 9: MAE metrics of two baselines and our proposed DEGNN on the charged dataset and vehicle dataset.

Model	Charged Particle				Vehicle ($\times 10^{-2}$)					
	(5,5,5)	(5,4,4)	(5,4,3)	(4,4,4)	Highway 1	Highway 2	Highway 3	Highway 4	Highway 5	Highway 6
GNN	0.340	0.372	0.374	0.397	0.053	0.052	0.067	0.083	0.058	0.061
GMN	0.243	0.290	0.331	0.310	0.059	0.061	0.131	0.129	0.107	0.113
DEGNN	0.239	0.282	0.325	0.307	0.019	0.015	0.031	0.047	0.025	0.034

high-imbalanced groups. Detailed information is as follows. The scenes are visualized in Figure 7.

- **Low imbalance individual flow**: bidirectional pedestrian movement with the presence of individuals. A total of 36 people walk from left to right, while 18 people walk from right to left.
- **Highly imbalance Individual flow**: bidirectional pedestrian movement with the presence of individuals. A total of 45 people walk from left to right, while 9 people walk from right to left.
- **Lowly imbalance Group flow**: bidirectional pedestrian movement with the presence of group. A total of 36 people walk from left to right, while 18 people walk from right to left.
- **Highly imbalance Group flow**: bidirectional pedestrian movement with the presence of group. A total of 45 people walk from left to right, while 9 people walk from right to left.

*B.0.6 Highway.* The scenes of Highways can be seen in Figure 8

C ADDITIONAL RESULTS

C.1 Running Time Comparison

To compare the efficiency, we report the running time of all methods in the LiPS dataset in Table 8. The results show that DEGNN is faster than EqMotion and yields comparable results to SAKE.

C.2 Additional Ablation Study

Table 7 shows the results of ablation studies on the other datasets. We can observe that the default settings of DEGNN outperform the other equivariant variants in almost all cases, verifying the idea of discrete equivariance. Additionally, compared with Ranking, Mean, and Sum Pooling variants, self-attention outperforms them in the vehicle and half of the crowd datasets, demonstrating its effectiveness on macro-level physical systems.
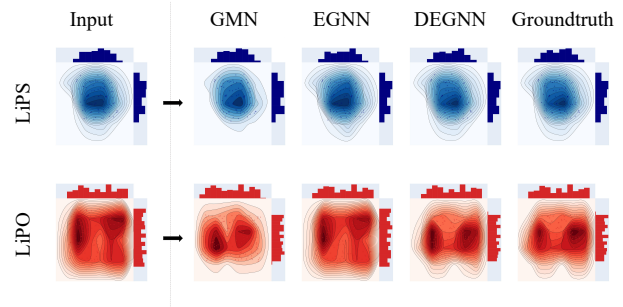
Figure 9: Visualization of DEGNN and other baselines prediction of molecular. Our method produces a molecular distribution that is closer to the groundtruth.

C.3 Other evaluation metrics on model accuracy

The MAE of two baselines and our proposed DEGNN across two datasets are shown in the Table 9.

C.4 Visualization

To investigate deeper insight into DEGNN, we visualize predicted samples of molecular datasets by projecting the position onto the xy-plane. Figure 9 shows their density. Based on the visualization results, we can find that DEGNN accurately recovers the distribution across all directions, whereas other models only capture patterns in specific directions. For example, in the LiPS dataset, the y-axis of EGNN output is similar to the ground truth, but it has an obvious difference on the x-axis. Such results again demonstrate the effectiveness of our model.