A GENERAL AND EFFICIENT SE(3)-EQUIVARIANT GRAPH FRAMEWORK: ENCODING SYMMETRIES WITH COMPLETE DIFFERENTIAL INVARIANTS AND FRAMES

Anonymous authors
Paper under double-blind review

ABSTRACT

Equivariant graph neural networks (Equiv-GNNs) have demonstrated effectiveness in modeling dynamics of multi-object systems via explicit symmetry encoding. Scalarization-based Equiv-GNNs are widely adopted in practice for their superior computational efficiency, as opposed to high-steerable models. However, most existing scalarization-based approaches rely on the empirical design of invariant scalars, which fail to fully and systematically capture system dynamics, particularly when object differential components like velocities are involved, leading to limited prediction performance. To address these issues, we propose a general, efficient, and lightweight SE(3)-equivariant graph framework with **C**omplete **D**ifferential **I**nvariants and **F**rames (CDIF). Specifically, we first systematically construct a maximal functionally independent set of basic invariant scalars derived from system states, providing a theoretically complete representational basis for arbitrary SE(3)-invariant functions. We then design local frames between objects that integrate both positional and differential components to recover directional information from the constructed invariants, which enables robust recovery of directional information while enhancing the model's capability of dynamic modeling. Extensive experiments across diverse domains including molecular dynamics, formation control, human motion capture and particle simulation, validate that our CDIF framework outperforms state-of-the-art baselines on most benchmarks, and exhibits strong scalability for large-scale multi-object scenarios.

1 INTRODUCTION

Multi-object systems are widely applied in various fields such as physics (Horie & Mitsume, 2022; Gong et al., 2022; Bobkov & Astrakhantseva, 2021), robotics (Piccoli, 2023; Feng et al., 2024; Zheng & Li, 2024; Liu et al., 2024; Chen, 2024; Xu et al., 2023a), and biology (Jing et al., 2021a; Cremer et al., 2023; Roche et al., 2023; Shuvo & Bhattacharya, 2025; Yi et al., 2024). Learning how to model the dynamics of these systems is challenging because, in most practical scenarios, we only have access to observed system states, with no explicit information regarding their topologies, interaction mechanisms, or underlying dynamic models. Moreover, the strong nonlinearity inherent in these interactions and dynamics renders conventional approaches (Li et al., 2023b; 2025; 2023a; Prasse & Van Mieghem, 2022; Han et al., 2025a) inadequate for accurate system dynamics modeling and reliable future state prediction.

Graph neural networks (Skarding et al., 2021; Kipf et al., 2018; Velickovic et al., 2017; Wu et al., 2020; Vaswani et al., 2017) offer promising solutions to these challenges. However, they only capture object permutation invariance and overlook the system's geometric symmetries, such as the equivariance of object positions and interaction forces under global translations and rotations. To fill this gap, equivariant graph neural networks (Equiv-GNNs) (Han et al., 2025a) have emerged as a powerful and robust paradigm by explicitly encoding such symmetries into the models' architecture. Compared with non-equivariant graph neural networks, Equiv-GNNs narrow the solution space by incorporating symmetries, such as permutation invariance and transformation equivariance, which helps models capture the system dynamics and learn to predict system states.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Existing Equiv-GNNs can be divided into two categories (Han et al., 2025b), i.e., high-steerable models and scalarizaion models. High-steerable approaches employ group representation theory to lift geometric vectors into high-dimensional spaces, which enables symmetry representation. For example, TFN (Thomas et al., 2018) and SE(3)-Transformer (Fuchs et al., 2020) parameterize such vectors utilizing spherical harmonics, while CEGNN (Ruhe et al., 2023) employs Clifford algebra. Though these methods ensure strong expressiveness, they incur substantial computational costs, especially in large-scale scenarios. By contrast, scalarization-based methods like EGNN (Satorras et al., 2021) and GMN (Huang et al., 2022) construct invariant scalars and update states directly in the original space, which significantly reduce computational costs while achieving state-of-the-art performance across diverse tasks.

However, existing scalarization-based methods have not fully explored the relationship between the design of invariant scalars and model expressiveness. Most approaches tend to rely on empirical strategies to design invariant scalars, such as relative distances (Satorras et al., 2021) and angles (Gasteiger et al., 2021), which lack of systematic theoretical guidance. While Villar et al. (2021) has demonstrated that any invariant/equivariant function of input vectors can be represented via scalars, it does not explicitly account for structural constraints imposed by group actions, leading to potential omissions of critical features. Moreover, current methods (Huang et al., 2022) often treat differential components of object states merely as input channels to compute invariant scalars, for instance, computing relative distance alongside the magnitude of relative velocities. Actually, the combinations of these components are also critical to object dynamics, e.g., object j's position and object i's velocity jointly influence the evolution of object i. Such an empirically driven and structurally unconstrained design paradigm leads to incompleteness in the constructed invariant scalars. This incompleteness hinders the model from effectively capturing the underlying geometric features of multi-object systems, ultimately resulting in degraded predictive performance on dynamic modeling tasks. To address these issues, in this work, we propose **CDIF**, an SE(3)-equivariant graph neural network integrated with **C**omplete **D**ifferential **I**nvariants and **F**rames (CDIF). The main contributions of this work are summarized as follows:

- **(Complete Differential Invariants)** We establish the exact number of functionally independent invariant scalars sufficient to universally approximate any invariant functions dependent on multi-object system states, and develop a systematic method to construct them. These scalars integrate both positional and differential components of object states and are thus termed *complete differential invariants*. Computing these invariants imposes no excessive overhead compared with existing scalarization-based methods.
- **(Complete Differential Frames)** We then construct equivariant edge-wise local frames from object states to recover directional information from the invariants. These frames can fully represent all possible directions for object state updates and effectively capture the correlations between the positional and differential components, hence termed *complete differential frames*. Leveraging these frames to transform interaction messages into directional vector updates, we effectively compute the future object states.
- **(Complete Experimental Validation)** We conduct extensive experiments across diverse benchmarks including molecular dynamics, formation control, human motion capture, and physical particle simulation. Empirical results show that CDIF outperforms state-of-the-art baselines on most benchmarks, with particularly strong performance in long-term prediction tasks. Moreover, ablation studies confirm that both complete differential invariants and complete differential frames are critical for effectively modeling system dynamics.

2 PRELIMINARIES

Multi-object Systems The multi-object systems considered here refers to any system composed of interacting objects or nodes, such as multi-agent systems (Amirkhani & Barshooi, 2022) and N-body systems (Kipf et al., 2018). Consider a multi-object system with N objects evolving in the three-dimensional Euclidean space $\mathbb{R}^3$. At time t, the state of each object is formalized as a vector integrating positional and differential components, namely

$$\boldsymbol{s}(t) = \left[\boldsymbol{q}(t)^\top, \dot{\boldsymbol{q}}(t)^\top, \ddot{\boldsymbol{q}}(t)^\top, \cdots\right]^\top,$$

where $\boldsymbol{q}(t) \in \mathbb{R}^3$ denotes the position vector, with $\dot{\boldsymbol{q}}(t)$ and $\ddot{\boldsymbol{q}}(t)$ representing its first and second time derivatives, respectively. The interaction topology is characterized by the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$,

where $\mathcal{V} = \{1, \ldots, n\}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ define the node and edge sets. In practical applications, direct observation of second-order and higher-order differential states, e.g., accelerations and jerks, is often infeasible. Thus we focus on the scenarios where the object state includes only position and velocity, i.e., $\boldsymbol{s}(t) = \left[\boldsymbol{q}(t)^\top, \dot{\boldsymbol{q}}(t)^\top\right]^\top$. Our framework naturally generalizes to incorporate higher-order dynamics, with detailed extensions deferred to the Appendix B.

Functional Dependence Consider a family of smooth $\mathbb{R}$-valued functions $F = \{f_1, f_2, \ldots, f_n\}$. Let $p_0 \in M$ be a point on manifold M with coordinates $(x^1, x^2, \ldots, x^m)$. The *rank* of F at p_0 is defined by the rank of the Jacobian matrix $\partial f_i / \partial x^k$ for $i = 1, 2, ..., n; k = 1, 2, ..., m$. The family F is *regular* if its rank is constant across M, ensuring a uniform structure of dependencies throughout the manifold. Building on the concept of rank, we characterize *functional dependence* among functions in F. The functions $\{f_i\} \subset F$ are *functionally dependent* if, for each $p_0 \in M$ and its neighborhood U, there exists a non-trivial smooth function H, which is not identically zero on any subset of $\mathbb{R}^n$, such that

$$H(f_1(x), f_2(x), \ldots, f_n(x)) = 0, \quad \text{for all } p \in U. \tag{1}$$

Group Action Consider a Lie group G with identity element e. To describe the left action of a Lie group G on a manifold M, we introduce the permutation operator $T_g : M \to M$ that for each $g \in G$ and $p \in M$, we have $T_g(p) \in M$. The *orbit* of the point p under this action is

$$\mathcal{O}_p = \{T_g(p) : g \in G\}, \tag{2}$$

which captures all points reachable from p via left actions by elements of G. The action is termed *semi-regular* if every orbit is a smooth submanifold of M with uniform dimension. In this work, we focus on the left action of the group SO(3) on the positional and velocity vectors of objects in multi-object systems, where the corresponding group action can be typically regarded as semi-regular.

Invariance and Equivariance Let T_g, S_g denote the left group actions of G on manifolds M, N, respectively, and consider a smooth map $\phi : M \to N$. Map ϕ is called *G-equivariant* if it commutes with group actions

$$\phi(T_g(p)) = S_g(\phi(p)), \quad \text{for all } p \in M \text{ and } g \in G. \tag{3}$$

Intuitively, this property ensures that applying a group action before or after mapping ϕ yields the same result. Then, we say ϕ is *G-invariant* if group actions leave its output unchanged.

$$\phi(T_g(p)) = \phi(p), \quad \text{for all } p \in M \text{ and } g \in G. \tag{4}$$

Scalarization-Based Equivariant GNNs Let $\mathcal{I}_i$ denote the set of invariants only relevant to the state $\boldsymbol{s}_i$ of object i, while $\mathcal{I}_{ij}$ represents the invariants dependent on the states of both object i and its neighbor j. Scalarization-based Equiv-GNNs construct messages by

$$\begin{aligned} m_{ij} &= \phi_m(\{h_i, \mathcal{I}_i\}, \{h_j, \mathcal{I}_j\}, \{a_{ij}, \mathcal{I}_{ij}\}), \\ \boldsymbol{m}_{ij} &= \phi_e(m_{ij}) \boldsymbol{f}(\boldsymbol{s}_i, \boldsymbol{s}_j), \end{aligned} \tag{5}$$

where h_i, h_j and a_{ij} denote initial node and edge features, and $\boldsymbol{f}(\boldsymbol{s}_i, \boldsymbol{s}_j)$ outputs a vector. These messages are then aggregated according to the topology, with object features and states updated as follows:

$$h_i' = \phi_h(h_i, \mathcal{I}_i, \sum_{j \in \mathcal{N}_i} m_{ij}), \quad \boldsymbol{s}_i' = \boldsymbol{s}_i + c \sum_{j \in \mathcal{N}_i} \boldsymbol{m}_{ij}. \tag{6}$$

Here, c serves as a normalization coefficient, ϕ_m, ϕ_e, ϕ_h denote arbitrary mappings, and $\mathcal{N}_i$ specifies the neighbor set of object i.

3 METHODOLOGY

In this section, we introduce the derivation of complete differential invariants and complete differential frames in the three-dimensional Euclidean space under the action of the special Euclidean group SE(3). Specifically, we consider global group actions rather than gauge transformations (He et al., 2021), where the latter apply transformations independently to separated objects.

3.1 Complete Differential Invariants

In modeling dynamics of multi-object systems, messages m_{ij} characterize inter-object interactions and thus motivate the construction of edge-wise invariants rather than node-wise ones. These invariants depend on the states of two objects connected by an edge, formalized as $I(\boldsymbol{s}_i, \boldsymbol{s}_j)$. Translational effects can be readily eliminated via centralization, reducing the problem to constructing invariants under the SO(3) group. Thus, our goal is to find a set of SO(3)-invariants $\mathcal{I} = \{I_1(\boldsymbol{s}_i, \boldsymbol{s}_j), \cdots, I_m((\boldsymbol{s}_i, \boldsymbol{s}_j))\}$ such that any other SO(3)-invariant $\mathbb{R}$-valued function is functionally dependent on $\mathcal{I}$. The following proposition specifies the maximal number of non-trivial functionally independent invariants in this setting.

Theorem 3.1. *Under SO(3) group actions on object states $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$, the maximal number of non-trivial functionally independent invariants is **nine**, given by (7). Specifically, any SO(3)-invariant $\mathbb{R}$-valued function $I \in \mathcal{F}(\mathbb{R}^3)^{SO(3)}$ can be expressed by a function of these invariants, i.e., $I = H(I_1, \ldots, I_9)$.*

A detailed proof is provided in Appendix E.1. This implies that any additional invariant can be expressed through a function H of these nine invariants, which can be learned and approximated by neural networks. Explicit constructions of these invariants can be directly derived via vector inner products and cross products.

$$
\begin{aligned}
&I_{p_i} = \boldsymbol{q}_i(t)^\top \boldsymbol{q}_i(t), I_{v_i} = \dot{\boldsymbol{q}}_i(t)^\top \dot{\boldsymbol{q}}_i(t),\\
&I_{p_j} = \boldsymbol{q}_j(t)^\top \boldsymbol{q}_j(t), I_{v_j} = \dot{\boldsymbol{q}}_j(t)^\top \dot{\boldsymbol{q}}_j(t),\\
&I_{p_i v_i} = \boldsymbol{q}_i(t)^\top \dot{\boldsymbol{q}}_i(t), I_{p_j v_j} = \boldsymbol{q}_j(t)^\top \dot{\boldsymbol{q}}_j(t),\\
&I_{p_i p_j} = \boldsymbol{q}_i(t)^\top \boldsymbol{q}_j(t), I_{v_i v_j} = \dot{\boldsymbol{q}}_i(t)^\top \dot{\boldsymbol{q}}_j(t),\\
&I_{v_i p_i p_j} = \dot{\boldsymbol{q}}_i(t)^\top (\boldsymbol{q}_i(t) \times \boldsymbol{q}_j(t)),
\end{aligned}
\tag{7}
$$

Remark 3.1. This set $\mathcal{I} = \{I_{p_i}, I_{v_i}, \cdots, I_{v_i p_i p_j}\}$ is not unique. Any invariant can be substituted with an alternative as long as the functional independence of the set is preserved. In addition, Proposition 3.1 can be generalized to broader scenarios. For example, in settings where only object positions are considered such as EGNN (Satorras et al., 2021) and Radial Field (Köhler et al., 2019), the maximal number of non-trivial functionally independent invariants reduces to **three**. This implies that we only need to augment EGNN's native invariant $\|\boldsymbol{q}_i - \boldsymbol{q}_j\|^2$ with two additional functionally independent invariants like $\{\|\boldsymbol{q}_i\|^2, \|\boldsymbol{q}_j\|^2, \|\boldsymbol{q}_i - \boldsymbol{q}_j\|^2\}$ to form a maximal functionally independent set. Further details and discussions can be found in Appendix F. Furthermore, constructing an excessive number of invariants does not necessarily yield additional benefits. Empirical results in Appendix H.3 demonstrate that, under the same model architecture, a larger set of invariants may degrade model performance and introduce unnecessary overhead. Thus, Theorem 3.1 actually establishes an explicit, theoretically guaranteed connection between the invariants and model expressivity, ensuring that models achieve strong performance across all general scenarios while avoiding excessive computational costs.

The invariants constructed in (7) are readily verified to be $\mathrm{SO}(3)$-invariant functions, i.e.,

$$
\forall g \in \mathrm{SO}(3), \quad I(g \cdot \boldsymbol{s}_i, g \cdot \boldsymbol{s}_j) = I(\boldsymbol{s}_i, \boldsymbol{s}_j). \tag{8}
$$

The following corollary extends this property to arbitrary functions of the $\mathrm{SO}(3)$-invariant functions.

Corollary 3.2. *Any function f with $SO(3)$-invariant functions as inputs is $SO(3)$-invariant.*

A detailed proof is provided in Appendix E.2. Building on the Corollary 3.2, we can design SO(3)-invariant layers by taking invariants as inputs. As shown by Leshno et al. (1993), multi-layer neural networks with adequate neurons enable universal approximation of any continuous function. Therefore, by feeding $\mathcal{I} = \{I_{p_i}, I_{v_i}, \cdots, I_{v_i p_i p_j}\}$ from (7) into the layers, we can achieve universal approximation of arbitrary SO(3)-invariant functions associated with the state vectors $\boldsymbol{s}(t)$.

3.2 Complete Differential Frames

Utilizing the complete differential invariants introduced in Section 3.1, we can effectively represent SO(3)-invariant scalar messages m_{ij} to characterize the interactions between objects by neural networks. However, such scalar messages only capture the magnitude of interactions, while failing

to incorporate directional information. Thus, this section focuses on incorporating directional information into interaction messages, extending scalar messages m_{ij} to equivariant vector messages $\boldsymbol{m}_{ij}$ that can drive object state updates. Specifically, our goal is to derive a physically meaningful expression for $\boldsymbol{f}(\boldsymbol{s}_i, \boldsymbol{s}_j)$ in (5).

Du et al. (2022) proposed 3D local frames to recover directional information in positional space. Nevertheless, this single-frame design suffers from the degeneracy risk when the underlying vectors $\boldsymbol{q}_i, \boldsymbol{q}_j$ are collinear, and it neglects differential components that are vital in modeling dynamics. To address these limitations, we augment this local frames into complete differential frames, which introduce an additional 3D local frame constructed from velocities to form a dual-frame architecture. Specifically, for object i and its neighbor j, we construct two 3D orthonormal frames, the positional frame $\boldsymbol{\mathcal{F}}^p_{ij}(t)$ and the velocity frame $\boldsymbol{\mathcal{F}}^v_{ij}(t)$, defined as

$$\begin{aligned} \boldsymbol{\mathcal{F}}^p_{ij}(t) &= \{\boldsymbol{e}^{p_x}_{ij}(t), \boldsymbol{e}^{p_y}_{ij}(t), \boldsymbol{e}^{p_z}_{ij}(t)\}, \\ \boldsymbol{\mathcal{F}}^v_{ij}(t) &= \{\boldsymbol{e}^{v_x}_{ij}(t), \boldsymbol{e}^{v_y}_{ij}(t), \boldsymbol{e}^{v_z}_{ij}(t)\}. \end{aligned} \tag{9}$$

with

$$\begin{aligned} \boldsymbol{e}^{p_x}_{ij} &= \frac{\boldsymbol{q}_i - \boldsymbol{q}_j}{\|\boldsymbol{q}_i - \boldsymbol{q}_j\|}, \quad \boldsymbol{e}^{p_y}_{ij} = \frac{\boldsymbol{q}_i \times \boldsymbol{q}_j}{\|\boldsymbol{q}_i \times \boldsymbol{q}_j\|}, \\ \boldsymbol{e}^{v_x}_{ij} &= \frac{\dot{\boldsymbol{q}}_i - \dot{\boldsymbol{q}}_j}{\|\dot{\boldsymbol{q}}_i - \dot{\boldsymbol{q}}_j\|}, \quad \boldsymbol{e}^{v_y}_{ij} = \frac{\dot{\boldsymbol{q}}_i \times \dot{\boldsymbol{q}}_j}{\|\dot{\boldsymbol{q}}_i \times \dot{\boldsymbol{q}}_j\|}, \\ \boldsymbol{e}^{p_z}_{ij} &= \boldsymbol{e}^{p_x}_{ij} \times \boldsymbol{e}^{p_y}_{ij}, \quad \boldsymbol{e}^{v_z}_{ij} = \boldsymbol{e}^{v_x}_{ij} \times \boldsymbol{e}^{v_y}_{ij}. \end{aligned} \tag{10}$$

It is evident that each of the positional frame $\boldsymbol{\mathcal{F}}^p_{ij}(t)$ and velocity frame $\boldsymbol{\mathcal{F}}^v_{ij}(t)$ forms an orthonormal basis for $\mathbb{R}^3$, where orthogonality is guaranteed by the properties of vector cross products. Moreover, the cross product operation preserves SO(3)-equivariance, yielding the following proposition with the detailed proof provided in the Appendix E.3.

Proposition 3.3. *Both frames $\boldsymbol{\mathcal{F}}^p_{ij}(t)$ and $\boldsymbol{\mathcal{F}}^v_{ij}(t)$ are equivariant under actions of the SO(3) group.*

Compared with utilizing only positional local frames, our dual-frame design provides greater vector redundancy and thus substantially reduces the probability of such degeneracy. As long as any three of the six vectors from $\boldsymbol{\mathcal{F}}^p_{ij}(t)$ and $\boldsymbol{\mathcal{F}}^v_{ij}(t)$ are linearly independent in $\mathbb{R}^3$, a complete and valid 3D coordinate system can be constructed to provide reliable directional basis vectors for state updates. Another key advantage of our dual-frame design lies in its ability to jointly encode directional information for the update of both positional and velocity states. Rather than restricting the positional frame to only update positions and the velocity frame to only update velocities, this cross-frame coupling enables the model to capture the inherent dynamic correlations between objects' positions and velocities, which enhances the model's representation ability. In addition, breaking the naive one-to-one mapping also improves the robustness of directional information recovery, where noise corruption or partial degeneracy in one frame can be alleviated by the complementary information from the other, further ensuring stable state updates even in some extreme scenarios.

The complete differential invariants $\{\mathcal{I}_i, \mathcal{I}_j, \mathcal{I}_{ij}\}$ and complete differential frames $\{\boldsymbol{\mathcal{F}}^p_{ij}(t), \boldsymbol{\mathcal{F}}^v_{ij}(t)\}$ jointly form the core of our equivariant message-passing architecture. $\{\mathcal{I}_i, \mathcal{I}_j, \mathcal{I}_{ij}\}$ constitute a maximally functionally independent set under $\mathrm{SO}(3)$, ensuring no information related to rotational symmetry is lost during object state encoding. Meanwhile, the complete differential frames $\{\boldsymbol{\mathcal{F}}^p_{ij}(t), \boldsymbol{\mathcal{F}}^v_{ij}(t)\}$ provide orthonormal, equivariant edge-wise frames to recover directional information during the update process. The completeness of these two components jointly acts to guarantee the expressiveness and generalization capabilities of equivariant graph neural networks.

4 CDIF Framework

In this section, we introduce **CDIF**, a general and efficient equivariant graph framework integrated with **C**omplete **D**ifferential **I**nvariants and **F**rames. The overall architecture of CDIF is illustrated in Figure 1. To temporarily eliminate the effect of translations, we first centralize the system states with respect to the centroid, which allows us to focus exclusively on rotational symmetry, i.e., SO(3)-invariance/equivariance. Since velocities are naturally translationally invariant, centralization is applied only to system positions. We then feed the centralized states and initial features into CDIF

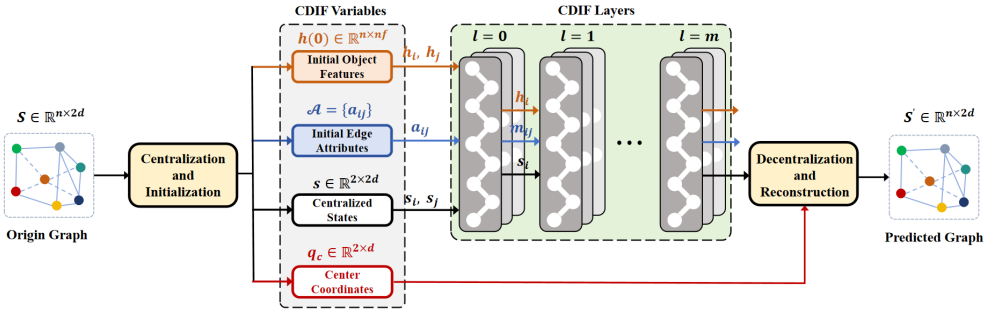


Figure 1: An overview of CDIF. The centralized object states are derived via the Centralization and Initialization module to get rid of translational effects. Then, we feed them along with the initial object features and initial edge attributes into CDIF layers to update the node features, edge messages, and object states. Finally, the predicted object states are derived by feeding local object states and center coordinates into the Decentralization and Reconstruction module.

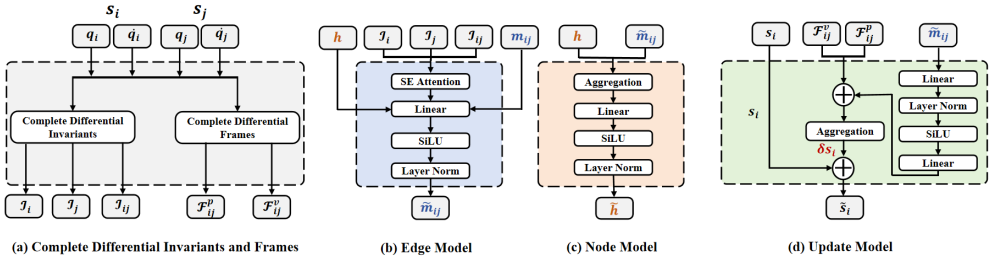


Figure 2: The detailed implementations of the CDIF Layer. The CDIF layer primarily consists of four modules. First, we construct complete differential invariants and frames, which then serve as inputs to the edge model for constructing edge messages. By aggregating these messages, the node model and update model subsequently update node features and object states,respectively. Variables marked with a tilde, such as $\tilde{h}$, denote their updated counterparts.

layers to learn for predicting system future states. The final result are derived by decentralizing the predicted centralized states. Within each CDIF layer detailed in Figure 2, we first construct complete differential invariants and frames from centralized states, as given by Eqs. (7) and (10). The CDIF layer adopts the standard message-passing mechanism (Gilmer et al., 2020) to model inter-object interactions. The edge model and node model follow the canonical design of message-passing neural networks (Kipf et al., 2018), with a key modification that the edge model takes the constructed complete differential invariants as part of its input features. To screen the importance of invariants, we introduce SE attention to gate them. Additionally, we set the activation function to SiLU and incorporate layer normalization to improve training stability. Next, with the complete differential frames and edge messages $\tilde{m}_{ij}$ updated by the edge model, each CDIF layer employs a dedicated update module to model the temporal evolution of object states. The update module predicts state variations $\delta \boldsymbol{s}$ instead of directly predicting the future object states $\tilde{s}$. Finally, to restore the effect of translations, we reintroduce the centroid $\boldsymbol{q}_c$ at the output of CDIF, which ensures translational equivariance. The implementation details of CDIF are provided in the Appendix D. Building on Corollary 3.2 and Proposition 3.3, we present the following theorem regarding the SE(3)-equivariance of the CDIF layers, with its proof detailed in the Appendix E.4.

Theorem 4.1. *The CDIF layer is equivariant under actions of the SO(3) group. Specifically, for any $g \in SO(3)$, the following holds:*

$$g \cdot \mathrm{CDIFL}(\boldsymbol{s}(l)) = \mathrm{CDIFL}(g \cdot \boldsymbol{s}(l)). \tag{11}$$

Compared with other scalarization-based methods (Satorras et al., 2021), CDIF incurs additional overhead in constructing complete differential invariants and frames. Consider a multi-object system with m edges. Note that the number of complete differential invariants and frames depends

only on the dimension of space and the orbit, regardless of the system topology. This ensures that the per-edge computational cost for their construction remains constant. Since the invariants and frames are constructed along each edge, the associated computational complexity scales as $O(m)$. In message-passing GNNs (Kipf et al., 2018), the computation of edge messages itself already exhibits a complexity of $O(m)$. Thus, the construction of complete differential invariants and frames does not introduce an increase in overall computational complexity.

5 Related Work

Graph neural networks (GNNs) (Wu et al., 2020; Skarding et al., 2021; Kipf et al., 2018) offer promising solutions for modeling system dynamics. To enhance model expressiveness and robustness, Equiv-GNNs (Han et al., 2022; 2025a) further incorporate geometric symmetries into GNN frameworks. Existing Equiv-GNNs primarily fall into two categories, i.e., scalarization-based models and high-degree steerable models (Han et al., 2025a). High-degree steerable models (Anderson et al., 2019; Fuchs et al., 2020; Batatia et al., 2022; Yu et al., 2023) treat geometric vectors as steerable tensors, leveraging high-dimensional tensor operations to ensure equivariance. Among these methods, TFN (Thomas et al., 2018) and NequIP (Batzner et al., 2022) employs spherical harmonics, while LieTransformer (Hutchinson et al., 2021) combines Lie convolution with transformers and extends the framework to Lie groups. CEGNN (Ruhe et al., 2023) utilizes Clifford algebra to handle geometric symmetries. While these methods achieve strong expressiveness, their intricate tensor manipulations lead to high computational complexity, limiting scalability to large-scale scenarios (Zitnick et al., 2022; Musaelian et al., 2023).

In contrast, scalarization-based models map vectors to invariant scalars directly in the original space and introduce equivariance during object state updating or message construction (Wang et al., 2023; Battiloro et al., 2024; Köhler et al., 2019; Schütt et al., 2021; Jing et al., 2021b; Zhang et al., 2024). For instance, EGNN (Satorras et al., 2021) and GMN (Huang et al., 2022) use relative states to derive invariant scalars and update object states along radial directions. Other methods (Du et al., 2022; Kofinas et al., 2021) construct local frames to represent directional information, with Du et al. (2023) providing guidelines for efficient frame-based model design. Some approaches additionally consider object orientations when encoding symmetries, for example, PONITA (Bekkers et al., 2024) achieves equivariance under group actions on both positions and orientations via group convolutions. Nevertheless, these methods lack theoretical guarantees in designing invariant scalars, especially when considering high-order differential components.

In modeling system dynamics, there is also much recent research devoted to integrating Equiv-GNNs with ODE-based approaches (Poli et al., 2019; Bishnoi et al., 2022; Jin et al., 2022) or trajectory prediction methods (Xu et al., 2023b; 2024) to capture temporal correlations. Since this line does not align with the core contribution of our work, more discussion about it is left for future work.

6 Experiments

To validate the effectiveness of CDIF, we benchmark it on four datasets, MD17 (Chmiela et al., 2017), CMU Motion (CMU, 2003), Formation Consensus Control, and N-body simulation (Satorras et al., 2021). Furthermore, we evaluate CDIF against a series of state-of-the-art equivariant models, including EGNN (Satorras et al., 2021), ClofNet (Du et al., 2022), TFN (Thomas et al., 2018), Radial Field (Köhler et al., 2019), GMN (Huang et al., 2022), $\mathrm{SE}(3)$-Transformer (Fuchs et al., 2020), SEGNN (Brandstetter et al., 2022) and PONITA (Bekkers et al., 2024). Ablation studies demonstrate that complete differential invariants and complete differential frames collectively enhance the model's expressiveness. Experiment details are provided in Appendix H.

6.1 Molecular Dynamic

Since complete differential invariants already encode all features related to atomic states, we only use the atomic index as inputs to obtain initial node features. In this dataset, the system topology is determined by inter-atomic distances, where atoms with a distance less than 1.6 are treated as neighbors. Our task is to predict the positions of all atoms after 2000 frames. As shown in Table 1, the CDIF framework achieves the best performance on 6 out of 10 molecular systems and ranks top 2 across all tasks. CDIF relies solely on MLPs as its architectural backbone. Yet our method is still highly competitive with SOTA approaches, and even achieves superior performance on the majority of tasks.

Table 1: Mean Square Error ($\times 10^{-2}$) for position prediction on MD17. The best results are **bolded** and the second best are underlined.

Molecule	EGNN	ClofNet	RF.	GMN	TFN	SEGNN	PONITA	CDIF(ours)
Aspirin	6.71	5.29	12.18	6.29	7.19	5.89	**2.92**	$4.25_{\pm 0.04}$
Azobenzene	5.77	4.74	15.36	4.65	5.88	5.19	3.97	$\mathbf{3.77}_{\pm 0.06}$
Benzene	0.17	0.16	0.33	0.17	0.17	0.21	0.61	$\mathbf{0.15}_{\pm 0.01}$
Ethanol	3.95	3.15	4.66	3.96	3.95	3.87	2.95	$\mathbf{2.70}_{\pm 0.18}$
Malonaldehyde	12.23	12.29	21.29	12.13	12.30	12.31	**9.18**	$11.87_{\pm 0.75}$
Naphthalene	0.27	0.25	0.90	0.25	0.27	0.26	0.24	$\mathbf{0.23}_{\pm 0.00}$
Paracetamol	16.66	8.18	23.63	16.03	16.88	10.01	6.60	$\mathbf{4.61}_{\pm 0.47}$
Salicylic	0.64	0.59	1.62	0.98	0.68	0.63	0.55	$\mathbf{0.53}_{\pm 0.01}$
Toluene	5.15	3.78	6.20	4.36	4.70	5.31	**1.87**	$1.98_{\pm 0.10}$
Uracil	0.44	0.36	1.06	0.43	0.50	0.39	**0.27**	$0.34_{\pm 0.01}$

6.2 Formation Consensus Control

Formation consensus control, a fundamental problem in control theory, aims to coordinate a group of agents to achieve unified states through inter-agent interactions. In this study, we revisit the conventional formation consensus (FC) framework by adding a velocity cross product term into the agent control inputs, and further partition the datasets into three levels, "Easy", "Medium", and "Difficult", based on the complexity of the agent interactions. The details for dataset generation are provided in Appendix G, and each level includes tasks with 5 agents (FC(5)) and 10 agents (FC(10)). As shown in Table 2, CDIF achieves the best performance on both FC(5) and FC(10) tasks across three levels. Regarding computational efficiency, CDIF introduces a marginal increase in forward time (**Time** in Table 2) compared with other scalarization-based baselines like EGNN (Satorras et al., 2021) and ClofNet (Du et al., 2022), which is primarily due to the construction of complete differential invariants and frames. However, it significantly outpaces high-degree steerable models such as TFN (Thomas et al., 2018) and $\mathrm{SE}(3)$-Transformer (Fuchs et al., 2020), particularly in FC(10) tasks. Moreover, a critical observation is that the forward time of CDIF remains remarkably close between FC(5) and FC(10) tasks, demonstrating its potential in large-scale scenarios.

Table 2: Mean Square Error (MSE) and forward time ($\times 10^{-2}$) for position prediction on FC(5) and FC(10) tasks across different levels. The best results are **bolded** and the second best are underlined.

Task	FC(5)				FC(10)			
Level	Easy	Medium	Difficult		Easy	Medium	Difficult	
	MSE	**MSE**	**MSE**	**Time**	**MSE**	**MSE**	**MSE**	**Time**
EGNN	0.189	0.430	0.583	0.10	0.070	0.136	0.962	0.11
ClofNet	0.179	0.465	0.629	0.17	0.039	0.142	0.906	0.16
TFN	0.114	0.252	0.372	0.64	0.034	0.080	0.601	1.39
SEGNN	0.134	0.395	0.582	0.48	0.098	0.176	1.742	1.03
Radial Field	0.194	0.460	0.696	0.07	0.187	0.321	1.145	0.07
$\mathrm{SE}(3)$-Tr.	0.126	0.286	0.400	1.38	0.033	0.083	0.599	3.71
GMN	0.145	0.369	0.551	0.14	0.122	0.210	0.844	0.14
PONITA	0.105	0.326	0.475	0.33	0.054	0.138	0.594	0.48
CDIF(ours)	**0.073**	**0.173**	**0.326**	0.32	**0.017**	**0.037**	**0.510**	0.33

6.3 CMU Motion Capture

Unlike the other three datasets that require manual construction of system topology, the CMU Motion Capture dataset (CMU, 2003) provides a predefined topology that forms a human skeleton-like structure. Our objective is to predict the positions of all nodes after 10, 20, and 30 timesteps. As shown in Table 3, CDIF achieves the best performance on 2 out of 3 tasks, particularly in long-range prediction tasks. During experiments, we found that PONITA struggled to converge stably despite considerable efforts to stabilize the training process. Across these three tasks, the resulting prediction errors were 71.25, 80.05, and 89.13. This is likely due to misalignment between the

orientations constructed by PONITA and the inherent orientations of human motion. Therefore, we omitted PONITA from the comparisons.

Table 3: Mean Square Error ($\times 10^{-2}$) for position prediction across different timesteps on CMU Motion. The best results are **bolded** and the second best are underlined.

Motion	EGNN	ClofNet	RF.	SEGNN	GMN	TFN	CDIF
10	1.116	1.086	8.296	11.017	1.218	**0.884**	0.953
20	3.096	3.467	63.267	19.158	3.065	3.016	**2.958**
30	5.827	7.322	199.475	26.547	7.011	5.983	**5.430**

6.4 N-BODY SIMULATION

The n-body simulation (Satorras et al., 2021) is widely adopted for evaluating the performance of models in position prediction tasks. In this experiment, we test model performance at 10, 20, 40 timesteps on 5-agent (Charged(5)) and 10-agent (Charged(10)) tasks. To demonstrate that complete differential invariants can be readily transferred to other models, we conduct additional experiments where these invariants are fed only as initial inputs to EGNN (Satorras et al., 2021) and GMN (Huang et al., 2022) (denoted as EGNN+CDIs and GMN+CDIs), without modifying their other original settings. The results presented in Table 4 show that CDIF achieves top performance on 4 out of 6 tasks and ranks within the top 2 across all evaluations. Notably, CDIF demonstrates superior performance in longer-term predictions compared with short-range forecasts, owing to the embedded differential components. Additionally, integrating complete differential invariants into baseline models improves their prediction accuracy by an average of 6.35%, verifying the critical role of complete invariants.

Table 4: Mean Square Error (MSE) for position prediction over Charged(5) and Charged(10) tasks at different prediction time steps. The best results are **bolded** and the second best are underlined.

Task	Charged(5)			Charged(10)		
Timestep	10	20	40	10	20	40
Model	**MSE**	**MSE**	**MSE**	**MSE**	**MSE**	**MSE**
EGNN	0.0849	0.5739	3.0840	0.1129	0.7893	4.6088
ClofNet	0.0807	0.5102	2.9128	0.1051	0.6880	4.3226
TFN	0.2259	1.2246	5.7761	0.2901	1.5932	7.8833
Radial Field	0.1573	0.9562	5.3055	0.2401	1.4727	7.6604
SE(3)-Tr.	0.2739	1.3820	6.6055	0.3394	1.7705	8.5021
GMN	0.1021	0.5809	3.1795	0.1178	0.7968	4.6437
PONITA	**0.0459**	0.4757	2.9013	**0.0702**	0.6635	4.2281
EGNN+CDIs	0.0845	0.5363	2.8754	0.1058	0.7457	4.3436
GMN+CDIs	0.0902	0.5424	3.0426	0.1056	0.7712	4.4403
CDIF(ours)	0.0689	**0.4214**	**2.3834**	0.0803	**0.5829**	**3.5131**

7 CONCLUSION

In this paper, we propose CDIF, a general SE(3)-equivariant graph neural network integrated with complete differential invariants and frames. We develop this framework to tackle the critical limitations of existing scalarization-based Equiv-GNNs which typically rely on empirically designed invariant scalars and neglect the differential components of object states in dynamic systems. We systematically construct a set of functionally independent complete differential invariants which guarantee the universal approximation of arbitrary SE(3)-invariant functions. This provides meaningful theoretical guidance for the design of invariant scalars. We further introduce complete differential frames with a dual-frame design that enhances vector redundancy, reduces degeneracy risks and enables cross-frame coupling to capture system dynamics. Extensive experiments across various domains validate the effectiveness and scalability of the CDIF. Ablation studies further demonstrate that both the complete differential invariants modules and the complete differential frames modules enhance the model's performance. For future work, we plan to extend the construction of complete differential invariants and frames beyond the SE(3) group to more general Lie groups.

IMPACT STATEMENT

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

AI USE STATEMENT

In this work, we used generative AI tools for the implementation of baseline methods, assisting with translation, support of qualitative and thematic data analysis and interpretation of results. We have not used generative AI tools for providing critical ingredients for proving mathematical claims and assisting in the writing of proofs, propose or refine hypotheses, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for code generation, text generation and brainstorming, (specifically for grammar checking, language polishing, drafting boilerplate code and discussing potential experimental setups). We have reviewed all AI-assisted work. The AI-generated code was executed and tested by the authors to ensure correctness. analysis and suggestions from AI tools were critically evaluated and integrated only after manual verification. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our work, we provide detailed implementation details, including model architecture, hyperparameters, and dataset processing steps, in Appendix D and Appendix H. Specifically, the CDIF framework's architecture and training procedures are described in Section 4 and Appendix D, while experimental setups for molecular dynamics (MD17), formation consensus control, CMU motion capture, and N-body simulation are detailed in Section 6 and Appendix H. All hyperparameters, including the number of layers, feature dimensions, learning rates, and batch sizes, are listed in Table 5. The source code[1] for our model and experiments will be made publicly available upon publication. We encourage readers to refer to these sections for a comprehensive understanding of our methodology and experimental design.

REFERENCES

Abdollah Amirkhani and Amir Hossein Barshooi. Consensus in multi-agent systems: a review. *Artificial Intelligence Review*, 55(5):3897–3935, 2022.

Emma Andersdotter, Daniel Persson, and Fredrik Ohlsson. Equivariant manifold neural odes and differential invariants. *arXiv preprint arXiv:2401.14131*, 2024.

Brandon Anderson, Truong Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. *Advances in neural information processing systems*, 32, 2019.

Ilyes Batatia, David P Kovacs, Gregor Simm, Christoph Ortner, and Gábor Csányi. Mace: Higher order equivariant message passing neural networks for fast and accurate force fields. *Advances in neural information processing systems*, 35:11423–11436, 2022.

Claudio Battiloro, Mauricio Tec, George Dasoulas, Michelle Audirac, Francesca Dominici, et al. E (n) equivariant topological neural networks. *arXiv preprint arXiv:2405.15429*, 2024.

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

Erik J Bekkers, Sharvaree P Vadgama, Rob Hesselink, Putri A van der Linden, and David W Romero. Fast, expressive se (n) equivariant networks through weight-sharing in position-orientation space. In *International Conference on Learning Representations*, 2024.

[1] https://github.com/WXY-0324/cfins

Suresh Bishnoi, Ravinder Bhattoo, Sayan Ranu, and NM Krishnan. Enhancing the inductive biases of graph neural ode for modeling dynamical systems. *arXiv preprint arXiv:2209.10740*, 2022.

Sergey Petrovich Bobkov and Irina Aleksandrovna Astrakhantseva. The use of multi-agent systems for modeling technological processes. In *Journal of Physics: Conference Series*, 2021.

Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J Bekkers, and Max Welling. Geometric and physical quantities improve e (3) equivariant message passing. In *International Conference on Learning Representations*, 2022.

Hanxiao Chen. Graph neural networks with model-based reinforcement learning for multi-agent systems. *arXiv preprint arXiv:2407.09249*, 2024.

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

CMU. Carnegie-mellon motion capture database, 2003. URL `http://mocap.cs.cmu.edu`.

Julian Cremer, Leonardo Medrano Sandonas, Alexandre Tkatchenko, Djork-Arné Clevert, and Gianni De Fabritiis. Equivariant graph neural networks for toxicity prediction. *Chemical Research in Toxicology*, 2023.

Weitao Du, He Zhang, Yuanqi Du, Qi Meng, Wei Chen, Nanning Zheng, Bin Shao, and Tie-Yan Liu. Se (3) equivariant graph neural networks with complete local frames. In *International Conference on Machine Learning*, pp. 5583–5608. PMLR, 2022.

Yuanqi Du, Limei Wang, Dieqiao Feng, Guifeng Wang, Shuiwang Ji, Carla P Gomes, Zhi-Ming Ma, et al. A new perspective on building efficient and expressive 3d equivariant graph neural networks. *Advances in neural information processing systems*, 36:66647–66674, 2023.

Pu Feng, Rongye Shi, Size Wang, Junkang Liang, Xin Yu, Simin Li, and Wenjun Wu. Safe and efficient multi-agent collision avoidance with physics-informed reinforcement learning. *IEEE Robotics and Automation Letters*, 2024.

Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in neural information processing systems*, 33:1970–1981, 2020.

Johannes Gasteiger, Florian Becker, and Stephan Günnemann. Gemnet: Universal directional graph neural networks for molecules. *Advances in Neural Information Processing Systems*, 34:6790–6802, 2021.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Message passing neural networks. In *Machine learning meets quantum physics*, pp. 199–214. Springer, 2020.

Shiqi Gong, Qi Meng, Jue Zhang, Huilin Qu, Congqiao Li, Sitian Qian, Weitao Du, Zhi-Ming Ma, and Tie-Yan Liu. An efficient lorentz equivariant graph neural network for jet tagging. *Journal of High Energy Physics*, 2022.

Jiaqi Han, Yu Rong, Tingyang Xu, and Wenbing Huang. Geometrically equivariant graph neural networks: A survey. *arXiv preprint arXiv:2202.07230*, 2022.

Jiaqi Han, Jiacheng Cen, Liming Wu, Zongzhao Li, Xiangzhe Kong, Rui Jiao, Ziyang Yu, Tingyang Xu, Fandi Wu, Zihe Wang, et al. A survey of geometric graph neural networks: Data structures, models and applications. *Frontiers of Computer Science*, 19(11):1911375, 2025a.

Jiaqi Han, Jiacheng Cen, Liming Wu, Zongzhao Li, Xiangzhe Kong, Rui Jiao, Ziyang Yu, Tingyang Xu, Fandi Wu, Zihe Wang, et al. A survey of geometric graph neural networks: Data structures, models and applications. *Frontiers of Computer Science*, 19(11):1911375, 2025b.

Lingshen He, Yiming Dong, Yisen Wang, Dacheng Tao, and Zhouchen Lin. Gauge equivariant transformer. *Advances in Neural Information Processing Systems*, 34:27331–27343, 2021.

Masanobu Horie and Naoto Mitsume. Physics-embedded neural networks: Graph neural PDE solvers with mixed boundary conditions. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2022.

Wenbing Huang, Jiaqi Han, Yu Rong, Tingyang Xu, Fuchun Sun, and Junzhou Huang. Equivariant graph mechanics networks with constraints. In *International Conference on Learning Representations*, 2022.

Michael J Hutchinson, Charline Le Lan, Sheheryar Zaidi, Emilien Dupont, Yee Whye Teh, and Hyunjik Kim. Lietransformer: Equivariant self-attention for lie groups. In *International Conference on Machine Learning*, pp. 4533–4543. PMLR, 2021.

Ming Jin, Yu Zheng, Yuan-Fang Li, Siheng Chen, Bin Yang, and Shirui Pan. Multivariate time series forecasting with dynamic graph neural odes. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):9168–9180, 2022.

Bowen Jing, Stephan Eismann, Pratham N Soni, and Ron O Dror. Equivariant graph neural networks for 3d macromolecular structure. *arXiv preprint arXiv:2106.03843*, 2021a.

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2021b.

Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In *International Conference on Machine Learning*, pp. 2688–2697. PMLR, 2018.

Miltiadis Kofinas, Naveen Nagaraja, and Efstratios Gavves. Roto-translated local coordinate frames for interacting dynamical systems. *Advances in Neural Information Processing Systems*, 34:6417–6429, 2021.

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

Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993.

Yushan Li, Jianping He, Lin Cai, and Xinping Guan. Local topology inference of mobile robotic networks under formation control. *IEEE Transactions on Automatic Control*, 2023a.

Yushan Li, Jianping He, Cailian Chen, and Xinping Guan. Topology inference for network systems: Causality perspective and nonasymptotic performance. *IEEE Transactions on Automatic Control*, 69(6):3483–3498, 2023b.

Yushan Li, Jianping He, and Dimos V Dimarogonas. Inverse inference on cooperative control of networked dynamical systems. *arXiv preprint arXiv:2504.13701*, 2025.

Yefeng Liu, Jingjing Liu, Zengpeng He, Zhenhong Li, Qichun Zhang, and Zhengtao Ding. A survey of multi-agent systems on distributed formation control. *Unmanned Systems*, 2024.

Albert Musaelian, Simon Batzner, Anders Johansson, Lixin Sun, Cameron J Owen, Mordechai Kornbluth, and Boris Kozinsky. Learning local equivariant representations for large-scale atomistic dynamics. *Nature Communications*, 14(1):579, 2023.

Peter J Olver. *Equivalence, invariants and symmetry*. Cambridge University Press, 1995.

Benedetto Piccoli. Control of multi-agent systems: Results, open problems, and applications. *Open Mathematics*, 2023.

Michael Poli, Stefano Massaroli, Junyoung Park, Atsushi Yamashita, Hajime Asama, and Jinkyoo Park. Graph neural ordinary differential equations. *arXiv preprint arXiv:1911.07532*, 2019.

Bastian Prasse and Piet Van Mieghem. Predicting network dynamics without requiring the knowledge of the interaction graph. *Proceedings of the National Academy of Sciences*, 119(44): e2205517119, 2022.

Rahmatullah Roche, Bernard Moussad, Md Hossain Shuvo, and Debswapna Bhattacharya. E (3) equivariant graph neural networks for robust and accurate protein-protein interaction site prediction. *PLoS Computational Biology*, 2023.

David Ruhe, Johannes Brandstetter, and Patrick Forré. Clifford group equivariant neural networks. *Advances in Neural Information Processing Systems*, 36:62922–62990, 2023.

Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International Conference on Cachine Learning*, pp. 9323–9332. PMLR, 2021.

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

Md Hossain Shuvo and Debswapna Bhattacharya. Equirank: Improved protein-protein interface quality estimation using protein language-model-informed equivariant graph neural networks. *Computational and Structural Biotechnology Journal*, 2025.

Joakim Skarding, Bogdan Gabrys, and Katarzyna Musial. Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey. *iEEE Access*, 9:79143–79168, 2021.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 2017.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. Graph attention networks. *stat*, 2017.

Soledad Villar, David W Hogg, Kate Storey-Fisher, Weichi Yao, and Ben Blum-Smith. Scalars are universal: Equivariant machine learning, structured like classical physics. *Advances in neural information processing systems*, 34:28848–28863, 2021.

Zun Wang, Guoqing Liu, Yichi Zhou, Tong Wang, and Bin Shao. Efficiently incorporating quintuple interactions into geometric deep learning force fields. *Advances in Neural Information Processing Systems*, 36:77043–77055, 2023.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

Chenxin Xu, Robby T Tan, Yuhong Tan, Siheng Chen, Yu Guang Wang, Xinchao Wang, and Yanfeng Wang. Eqmotion: Equivariant multi-agent motion prediction with invariant interaction reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1410–1420, 2023a.

Chenxin Xu, Robby T Tan, Yuhong Tan, Siheng Chen, Yu Guang Wang, Xinchao Wang, and Yanfeng Wang. Eqmotion: Equivariant multi-agent motion prediction with invariant interaction reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023b.

Zhao Xu, Haiyang Yu, Montgomery Bohde, and Shuiwang Ji. Equivariant graph network approximations of high-degree polynomials for force field prediction. *Transactions on machine learning research*, 2024.

Yiqiang Yi, Xu Wan, Kangfei Zhao, Le Ou-Yang, and Peilin Zhao. Equivariant line graph neural network for protein-ligand binding affinity prediction. *IEEE Journal of Biomedical and Health Informatics*, 2024.

Haiyang Yu, Zhao Xu, Xiaofeng Qian, Xiaoning Qian, and Shuiwang Ji. Efficient and equivariant graph networks for predicting quantum hamiltonian. In *International Conference on Machine Learning*, pp. 40412–40424. PMLR, 2023.

Yuelin Zhang, Jiacheng Cen, Jiaqi Han, Zhiqiang Zhang, Jun Zhou, and Wenbing Huang. Improving equivariant graph neural networks on large geometric graphs via virtual nodes learning. In *International Conference on Machine Learning*, pp. 58662–58679. PMLR, 2024.

Jiafeng Zheng and Jianzhen Li. Affine formation control of discrete-time second-order multi-agent systems with a fixed communication period. In *Journal of Physics: Conference Series*. IOP Publishing, 2024.

Larry Zitnick, Abhishek Das, Adeesh Kolluru, Janice Lan, Muhammed Shuaibi, Anuroop Sriram, Zachary Ulissi, and Brandon Wood. Spherical channels for modeling atomic interactions. *Advances in Neural Information Processing Systems*, 35:8054–8067, 2022.

A Appendix

B Mathematical Supplements

In Section 2, we discuss the functional dependence. Now we further present a lemma that bridges the concepts of rank and functional dependence.

Lemma B.1 (Theorem 1.10 (Olver, 1995)). *If F is regular of rank k, then in a neighborhood of any $p_0 \in M$, there exist k functionally independent functions $f_1, \dots, f_k \in F$ such that any other function $f \in F$ can be expressed as $f = H(f_1, \dots, f_k)$ for some smooth function H.*

Lemma B.1 shows that regular families of rank k admit a local basis of k independent functions generating the entire family. On an m-dimensional manifold M, locally, at most m functions can be found functionally independent. This implies that for the action of the group G on the manifold M, and a point $p_0 \in M$, there must exist a finite number (not exceeding m) of functionally independent invariants. To formalize the group actions on system dynamics with differential components, we introduce the jet bundle framework. Let $u(t) : \mathbb{R} \to M$ be a smooth curve on the manifold M, and $u^{(k)}(t)$ be the k-th order derivative of $u(t)$ with respect to t. Define $E = \mathbb{R} \times M$ as the *total space*, where $\mathbb{R}$ is the *fiber* and M is the *base space*. The *graph* of $u(t)$ is $\gamma_u = \{t, u(t)\} \subset E$. To include the derivatives $u^{(1)}(t), \dots, u^{(k)}(t)$ of $u(t)$, we introduce the *jet bundle* $J^{(k)}E = \mathbb{R} \times M^{(k)}$. Here, $M^{(k)}$ contains all points in M along with their derivatives up to order k. Similar to the graph γ_u in the total space E, the *prolonged graph* of the jet bundle $J^{(k)}E$ is

$$\gamma_u^{(k)} = \{t, u(t), \dots, u^{(k)}(t)\} \subset J^{(k)}E. \tag{12}$$

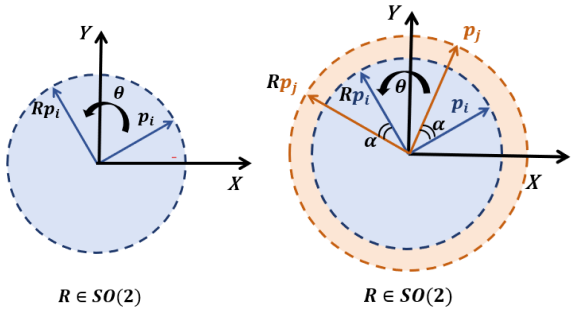
Group actions typically leave the time variable t unchanged. Thus, the left-action of the group G on the total space E is restricted to the base space M. For $\gamma_u = \{t, u(t)\}$, the group action becomes $\{t, T_g u(t)\}$. Extending this, the *prolonged group action* $T_g^{(k)}$ on $\gamma_u^{(k)}$ is

$$T_g^{(k)}\gamma_u^{(k)} = \{t, T_g u(t), (T_g u(t))^{(1)}, \dots, (T_g u(t))^{(k)}\}. \tag{13}$$

For a Lie group G with Lie algebra $\mathfrak{g}$, induced vector fields X_i form a basis of $\mathfrak{g}$, with $T_g p = (\exp X_g)p$. The *prolonged vector field* $X^{(k)}$ describes actions on derivatives.

For a node i and its neighbors $\mathcal{N}_i$, the prolonged graph of the extended jet bundle $J^{(k)}E_\Lambda$ is

$$\gamma_\Lambda^{(k)} = \{t, u_i(t), \dots, u_i^{(k)}(t), u_{i_1}(t), \dots, u_{i_1}^{(k)}(t), \dots, u_{i_m}(t), \dots, u_{i_m}^{(k)}(t)\} \subset J^{(k)}E_\Lambda. \tag{14}$$



(a) Single vector (b) Multiple vectors

Figure 4: The dashed lines represent the orbits. Both p_i and p_j are rotated counterclockwise by the same angle θ, maintaining their relative angle α.

where $i_1, \ldots, i_m \in \mathcal{N}_i$ and Λ denotes the set of $\{i, \mathcal{N}_i\}$. The prolonged group action $T_g^{(k)}$ on $\gamma_\Lambda^{(k)}$ acts simultaneously on node and neighbor states,

$$\begin{aligned} T_g^{(k)}\gamma_\Lambda^{(k)} = \{&t, T_g u_i(t), (T_g u_i(t))^{(1)}, \ldots, (T_g u_i(t))^{(k)}, \\ &T_g u_{i_1}(t), (T_g u_{i_1}(t))^{(1)}, \ldots, (T_g u_{i_1}(t))^{(k)}, \\ &T_g u_{i_2}(t), (T_g u_{i_2}(t))^{(1)}, \ldots, (T_g u_{i_2}(t))^{(k)}, \\ &\ldots, \\ &T_g u_{i_m}(t), (T_g u_{i_m}(t))^{(1)}, \ldots, (T_g u_{i_m}(t))^{(k)}\}. \end{aligned} \tag{15}$$

C Rotations in Euclidean space

In this section, we discuss the rotations in Euclidean spaces, specifically in the 2D and 3D Euclidean spaces $\mathbb{R}^2$ and $\mathbb{R}^3$. The special orthogonal group $\mathrm{SO}(n)$ is defined as:

$$\mathrm{SO}(n) \triangleq \{R \in \mathbb{R}^{n\times n} \mid R^\top R = I,\ \det(R) = 1\}, \tag{16}$$

where I denotes the n-dimensional identity matrix. Geometrically, $\mathrm{SO}(n)$ represents the group of orientation-preserving isometries (i.e., rotations) on the Euclidean space $\mathbb{R}^n$. In 2D Euclidean space $\mathbb{R}^2$, rotations are fully characterized by the special orthogonal group $\mathrm{SO}(2)$. Every rotation matrix $R \in \mathrm{SO}(2)$ can be parameterized by a unique angle θ, and is explicitly given by

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \tag{17}$$

As illustrated in Figure 4, for any vector $\boldsymbol{p}_i \in \mathbb{R}^2$, the action of $\mathrm{SO}(2)$ induces an orbit with dimension $s_k = 1$, a circle in the 2D plane. Geometrically, this corresponds to rotating $\boldsymbol{p}_i$ counterclockwise around the origin by an angle θ. When considering multiple vectors, e.g., $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$, the orbit dimension remains $s_k = 1$, as all vectors are rotated by the same angle θ. Certain functions dependent on $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$, such as $\|p_i\|$, $\|p_i - p_j\|$, and relative angle α, remain invariant under rotations. Hence, these functions are so-called $\mathrm{SO}(2)$-invariant. In 3D Euclidean space $\mathbb{R}^3$, we first consider a single vector $\boldsymbol{p} \in \mathbb{R}^3$. As shown in Figure 3, the orbit induced by the action of the $\mathrm{SO}(3)$ group on the single vector $\boldsymbol{p}$ is a sphere, whose dimension $s_k = 2$. Geometrically, the actions of the $\mathrm{SO}(3)$ group preserve a vector's norm while altering its orientation to another direction on the sphere, which can be parameterized by two parameters θ and ϕ. However, unlike the 2D case, the orbit dimension $s_k = 3$ for multiple vectors, which aligns with the dimension of the $\mathrm{SO}(3)$ group itself. This stems from the fundamental principle in Lie group actions; namely the orbit dimension is jointly determined by the

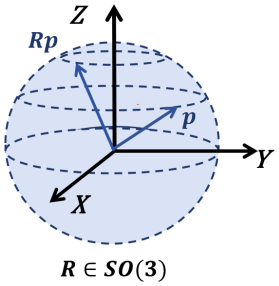


Figure 3: $\mathrm{SO}(3)$ acting on a single vector in 3D spaces

geometric constraints of the acted-upon object and the intrinsic degrees of freedom of the group. For a single vector $\boldsymbol{p} \in \mathbb{R}^3$, its orbit under $\mathrm{SO}(3)$ is a 2-dimensional submanifold, which is constrained by the norm and allows only directional changes parameterized by two angles (θ, ϕ). In contrast, when $\mathrm{SO}(3)$ acts on multiple linearly independent vectors, the group governs their global orientation, which requires 3 independent parameters, corresponding to rotations about the three orthogonal axes (e.g., Euler angles α, β, γ). Since $\mathrm{SO}(3)$ is a 3-dimensional Lie group, the orbit dimension here is fully determined by the group's dimension, i.e., $s_k = 3$. This aligns with the intuition that a rigid body's rotation in 3D space demands three independent angles to specify its orientation uniquely.

D Implementation Details of CDIF

Centralization. Let t_0 be the initial time. For the positional configuration of the system

$$\boldsymbol{q}(t_0) = \left[\boldsymbol{q}_1^\top(t_0), \ldots, \boldsymbol{q}_n^\top(t_0)\right], \tag{18}$$

the centroid $\boldsymbol{q}_c$ is calculated as

$$\boldsymbol{q}_c = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{q}_i(t_0). \tag{19}$$

By redefining the coordinate system origin to this centroid, each object's position is centralized via

$$\overline{\boldsymbol{q}}_i(t_0) = \boldsymbol{q}_i(t_0) - \boldsymbol{q}_c. \tag{20}$$

For any time t, the centralized position $\overline{\boldsymbol{q}}_i(t)$ remains invariant under translations applied at t_0. Specifically, for an arbitrary translation vector $\boldsymbol{T}$ acting on the centralized position $\boldsymbol{T}(\overline{\boldsymbol{q}}_i(t))$, we have

$$\boldsymbol{T}(\overline{\boldsymbol{q}}_i(t)) = (\boldsymbol{q}_i(t) + \boldsymbol{T}) - (\boldsymbol{q}_c + \boldsymbol{T}) = \boldsymbol{q}_i(t) - \boldsymbol{q}_c = \overline{\boldsymbol{q}}_i(t), \tag{21}$$

which demonstrates that centralized object positions are invariant under global translations.

Initialization. To encode object-specific attributes unrelated to geometric information, such as object charges, we employ a linear layer to map these attributes into initial object feature embeddings $h(0) \in \mathbb{R}^{n \times n_f}$, where n_f denotes the dimension of hidden features. Meanwhile, the initial edge attributes are denoted as $\mathcal{A} = (a_{ij})$, which incorporate topological attributes such as connectivity and edge weights.

CDIF Layer. Building upon the centralized object states and initial features, we introduce the implementation details of the CDIF layer:

$$\boldsymbol{\mathcal{F}}_{ij}^{p}(l), \boldsymbol{\mathcal{F}}_{ij}^{v}(l) = \textbf{CompFrame}(\boldsymbol{s}_i(l), \boldsymbol{s}_j(l)) \tag{22}$$

$$\mathcal{I}_i(l), \mathcal{I}_j(l), \mathcal{I}_{ij}(l) = \textbf{CompInvar}(\boldsymbol{s}_i(l), \boldsymbol{s}_j(l)) \tag{23}$$

$$m_{ij}(l) = \phi_m^l(h_i(l), h_j(l), a_{ij}(l), \mathcal{I}_i(l), \mathcal{I}_j(l), \mathcal{I}_{ij}(l)) \tag{24}$$

$$\boldsymbol{s}_i(l+1) = \boldsymbol{s}_i(l) + C\sum_{j \in \mathcal{N}_i} \textbf{Update}(m_{ij}(l), \boldsymbol{\mathcal{F}}_{ij}^{p}(l), \boldsymbol{\mathcal{F}}_{ij}^{v}(l)) \tag{25}$$

$$h_i(l+1) = \phi_h^l(h_i(l), \mathcal{I}_i(l), \sum_{j \in \mathcal{N}_i} m_{ij}(l)) \tag{26}$$

From the input-output perspective, at layer l, the CDIF layer takes as input node feature embeddings $h(l) = \left[h_1(l)^\top; \cdots; h_n(l)^\top\right] \in \mathbb{R}^{n \times n_f}$, object states $\boldsymbol{s}(l) \in \mathbb{R}^{n \times 6}$, and initial edge attributes $\mathcal{A}$, outputting updated features and states such that

$$(h(l+1), \boldsymbol{s}(l+1)) = \mathrm{CDIFL}(h(l), \boldsymbol{s}(l), \mathcal{A}). \tag{27}$$

Specifically, for each object i and one of its neighbors j, we construct complete differential frames and invariants via the functions **CompFrame**$(\cdot)$ and **CompInvar**$(\cdot)$ in (22) and (23), whose explicit forms are provide in (7) and (10). Next, inter-object messages $m_{ij}(l)$ are generated by an MLP ϕ_m^l that integrates object features $h_i(l), h_j(l)$, edge attributes $a_{ij}(l)$, and the extracted invariants in (24). Then, the state update model in (25) leverages the complete differential frames to fully recover directional information from invariants. The function **Update**$(\cdot)$ employs MLPs $\phi_p^l(m_{ij}), \phi_v^l(m_{ij})$:

$R^{nf} \to R^6$ to map messages $m_{ij}(l)$ to scalars $\{\lambda^l_{p1}, \ldots, \lambda^l_{p6}\}$ and $\{\lambda^l_{v1}, \ldots, \lambda^l_{v6}\}$. These scalars serve as weights for the frame basis vectors $\{\boldsymbol{e}^{p_x}_{ij}, \boldsymbol{e}^{p_y}_{ij}, \boldsymbol{e}^{p_z}_{ij}, \boldsymbol{e}^{v_x}_{ij}, \boldsymbol{e}^{v_y}_{ij}, \boldsymbol{e}^{v_z}_{ij}\}$. By linearly combining this basis with the learnable scalars, the interaction between object i and its neighbor j can be formulated as

$$\begin{aligned} \delta \boldsymbol{q}_{ij} &= \lambda_{p_1} \boldsymbol{e}^{p_x}_{ij} + \lambda_{p_2} \boldsymbol{e}^{p_y}_{ij} + \lambda_{p_3} \boldsymbol{e}^{p_z}_{ij} + \lambda_{p_4} \boldsymbol{e}^{v_x}_{ij} + \lambda_{p_5} \boldsymbol{e}^{v_y}_{ij} + \lambda_{p_6} \boldsymbol{e}^{v_z}_{ij}, \\ \delta \dot{\boldsymbol{q}}_{ij} &= \lambda_{v_1} \boldsymbol{e}^{v_x}_{ij} + \lambda_{v_2} \boldsymbol{e}^{v_y}_{ij} + \lambda_{v_3} \boldsymbol{e}^{v_z}_{ij} + \lambda_{v_4} \boldsymbol{e}^{v_x}_{ij} + \lambda_{v_5} \boldsymbol{e}^{v_y}_{ij} + \lambda_{v_6} \boldsymbol{e}^{v_z}_{ij}. \end{aligned} \tag{28}$$

The variations $\delta \boldsymbol{q}_{ij}$ and $\delta \dot{\boldsymbol{q}}_{ij}$ are aggregated across all neighbors of object i, scaled by constants C_p and C_v to regulate magnitude, yielding the updated states

$$\overline{\boldsymbol{q}}_i(l+1) = \overline{\boldsymbol{q}}_i(l) + C_p \sum_{j \in \mathcal{N}_i} \delta \boldsymbol{q}_{ij}, \quad \overline{\dot{\boldsymbol{q}}}_i(l+1) = \overline{\dot{\boldsymbol{q}}}_i(l) + C_v \sum_{j \in \mathcal{N}_i} \delta \dot{\boldsymbol{q}}_{ij}. \tag{29}$$

At last, node feature embeddings are updated in (26) by combining the previous embeddings $h_i(l)$, object-specific invariants $\mathcal{I}_i(l)$, and aggregated messages from the neighbors.

E Proofs of Things

E.1 Proof of Theorem 3.1

Proof. We first introduce an important lemma before proceeding with the proof.

Lemma E.1 (Theorem 2.4 (Andersdotter et al., 2024)). *If the prolonged group $G^{(k)}$ of a group G acts semi-regularly on $J^{(k)}E$ with orbit dimension s_k, then there are* $\dim J^{(k)}E - s_k$ *functionally independent local differential invariants of order k.*

The notations in this lemma is detailed in Section 2 and Appendix B. Lemma E.1 indicates that the maximal number of functionally independent invariants is determined by the dimension of the jet bundle and the prolonged group action orbit. Now let us back to the the proof of the proposition.

When constructing invariants, we only consider the positions and velocities of objects i and j, implying that the order k equals one. The corresponding prolonged graph $\gamma^{(1)}_{ij}$ can be formulated as

$$\gamma^{(1)}_{ij} = \{t, x_i, y_i, z_i, x_j, y_j, z_j, \dot{x}_i, \dot{y}_i, \dot{z}_i, \dot{x}_j, \dot{y}_j, \dot{z}_j\}, \tag{30}$$

which means the dimension of $J^{(1)}E_{ij}$ is 13. Moreover, the orbit dimension of $\mathrm{SO}(3)$ acting on $J^{(1)}E_{ij}$ is 3. Thus, by Lemma E.1, the number of functionally independent invariants is $13 - 3 = 10$. Note that t is a trivial invariant, reducing the number of non-trivial functionally independent differential invariants to $10 - 1 = 9$. Then, by Lemma B.1, any other function $I \in \mathcal{F}(\mathbb{R}^3)^{\mathrm{SO}(3)}$ can be expressed by these invariants. □

E.2 Proof of Corollary 3.2

Proof. Let $f = f(I_1, I_2, \ldots, I_n)$ where each I_k is an $\mathrm{SO}(3)$-invariant function. For any $g \in \mathrm{SO}(3)$, we have:

$$\begin{aligned} f(g \cdot \boldsymbol{s}_1, \ldots, g \cdot \boldsymbol{s}_m) &= f\big(I_1(g \cdot \boldsymbol{s}_1, \ldots), \ldots, I_n(g \cdot \boldsymbol{s}_1, \ldots)\big) \\ &= f\big(I_1(\boldsymbol{s}_1, \ldots), \ldots, I_n(\boldsymbol{s}_1, \ldots)\big) \\ &= f(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_m) \end{aligned} \tag{31}$$

Thus, f is $\mathrm{SO}(3)$-invariant. □

E.3 Proof of Proposition 3.3

Proof. Let $R \in \mathrm{SO}(3)$ be a rotation matrix, and the transformed positions are denoted as $\widetilde{\boldsymbol{q}}_i = R\boldsymbol{q}_i$ and $\widetilde{\boldsymbol{q}}_j = R\boldsymbol{q}_j$. Denote the transformed positional frame as $\widetilde{\boldsymbol{\mathcal{F}}}^p_{ij}(t) = \{\widetilde{\boldsymbol{e}}^{p_x}_{ij}, \widetilde{\boldsymbol{e}}^{p_y}_{ij}, \widetilde{\boldsymbol{e}}^{p_z}_{ij}\}$,we have

$$\begin{aligned}
\widetilde{\boldsymbol{e}}^{p_x}_{ij} &= \frac{\widetilde{\boldsymbol{q}}_i - \widetilde{\boldsymbol{q}}_j}{\|\widetilde{\boldsymbol{q}}_i - \widetilde{\boldsymbol{q}}_j\|} = \frac{R(\boldsymbol{q}_i - \boldsymbol{q}_j)}{\|\boldsymbol{q}_i - \boldsymbol{q}_j\|} = R\boldsymbol{e}^{p_x}_{ij}, \\
\widetilde{\boldsymbol{e}}^{p_y}_{ij} &= \frac{\widetilde{\boldsymbol{q}}_i \times \widetilde{\boldsymbol{q}}_j}{\|\widetilde{\boldsymbol{q}}_i \times \widetilde{\boldsymbol{q}}_j\|} = \frac{R(\boldsymbol{q}_i \times \boldsymbol{q}_j)}{\|\boldsymbol{q}_i \times \boldsymbol{q}_j\|} = R\boldsymbol{e}^{p_y}_{ij}, \\
\widetilde{\boldsymbol{e}}^{p_z}_{ij} &= \widetilde{\boldsymbol{e}}^{p_x}_{ij} \times \widetilde{\boldsymbol{e}}^{p_y}_{ij} = (R\boldsymbol{e}^{p_x}_{ij}) \times (R\boldsymbol{e}^{p_y}_{ij}) = R(\boldsymbol{e}^{p_x}_{ij} \times \boldsymbol{e}^{p_y}_{ij}) = R\boldsymbol{e}^{p_z}_{ij},
\end{aligned} \tag{32}$$

Thus, $\boldsymbol{\mathcal{F}}^p_{ij}(t)$ is SO(3)-equivariant, i.e.,

$$\boldsymbol{\mathcal{F}}^p_{ij}(R\boldsymbol{q}_i, R\boldsymbol{q}_j) = \{R\boldsymbol{e}^{p_x}_{ij}, R\boldsymbol{e}^{p_y}_{ij}, R\boldsymbol{e}^{p_z}_{ij}\} = R\boldsymbol{\mathcal{F}}^p_{ij}(\boldsymbol{q}_i, \boldsymbol{q}_j) \tag{33}$$

The proof of the SO(3)-equivariance of $\boldsymbol{\mathcal{F}}^v_{ij}(t)$ is similar to that of $\boldsymbol{\mathcal{F}}^p_{ij}(t)$, and it only requires replacing the superscript p with v. Therefore both $\boldsymbol{\mathcal{F}}^p_{ij}(t)$ and $\boldsymbol{\mathcal{F}}^v_{ij}(t)$ are equivariant under actions of the $\mathrm{SO}(3)$ group. □

E.4 Proof of Theorem 4.1

Proof. Since $h_i(l), h_j(l), a_{ij}(l)$ are scalars independent of object states and the invariants $\mathcal{I}$ are $\mathrm{SO}(3)$-invariant functions, the construction of messages (24) remains invariant under $\mathrm{SO}(3)$ group actions. Therefore, we have

$$\begin{aligned}
\forall g \in \mathrm{SO}(3), \quad m_{ij}(g \cdot \boldsymbol{s}(l)) &= \phi^l_m(\mathcal{I}_i(g \cdot \boldsymbol{s}(l)), \mathcal{I}_j(g \cdot \boldsymbol{s}(l)), \mathcal{I}_{ij}(g \cdot \boldsymbol{s}(l))) \\
&= \phi^l_m(\mathcal{I}_i(\boldsymbol{s}(l)), \mathcal{I}_j(\boldsymbol{s}(l)), \mathcal{I}_{ij}(\boldsymbol{s}(l))) = m_{ij}(\boldsymbol{s}(l)).
\end{aligned} \tag{34}$$

Moreover, by Proposition 3.3 the frames $\{\boldsymbol{\mathcal{F}}^p_{ij}(t), \boldsymbol{\mathcal{F}}^v_{ij}(t)\}$ are $\mathrm{SO}(3)$-equivariant functions. Thus, the state update process (25) is equivariant with respect to the $\mathrm{SO}(3)$ group.

$$\begin{aligned}
\forall g \in \mathrm{SO}(3), \quad g \cdot \boldsymbol{s}(l+1) &= g \cdot \boldsymbol{s}(l) + C \sum_{j \in \mathcal{N}_i} \textbf{Update}(m_{ij}(\boldsymbol{s}(l)), g \cdot \boldsymbol{\mathcal{F}}^p_{ij}(\boldsymbol{s}(l)), g \cdot \boldsymbol{\mathcal{F}}^v_{ij}(\boldsymbol{s}(l))) \\
&= g \cdot \boldsymbol{s}(l) + C \sum_{j \in \mathcal{N}_i} \textbf{Update}(m_{ij}(g \cdot \boldsymbol{s}(l)), \boldsymbol{\mathcal{F}}^p_{ij}(g \cdot \boldsymbol{s}(l)), \boldsymbol{\mathcal{F}}^v_{ij}(g \cdot \boldsymbol{s}(l))) \\
&= g \cdot \boldsymbol{s}(l) + C \sum_{j \in \mathcal{N}_i} \textbf{Update}(g \cdot \boldsymbol{s}(l))
\end{aligned} \tag{35}$$

Since the node model ϕ^l_h takes as inputs only $\mathrm{SO}(3)$-invariant functions, by Corollary 3.2, Object feature embeddings (26) is $\mathrm{SO}(3)$-invariant. Thus, the CDIF layer is $\mathrm{SO}(3)$-equivariant, i.e.,

$$\forall g \in \mathrm{SO}(3), \quad g \cdot \mathrm{CDIFL}(\boldsymbol{s}(l)) = \mathrm{CDIFL}(g \cdot \boldsymbol{s}(l)). \tag{36}$$

□

F Analysis on more cases

In this section, we demonstrate the construction of complete differential invariants and frames for additional cases. Moreover, we show that these invariants can be systematically derived leveraging characteristic equations.

Two-dimensional Euclidean space with object positions only First, we construct the complete invariants. The vector field induced by $\mathrm{SO}(2)$ on $\mathbb{R}^2$ is as following:

$$X = -y\frac{\partial}{\partial x} + x\frac{\partial}{\partial y}, \tag{37}$$

and the graph of jet bundle JE_{ij} can be expressed as

$$\gamma_{ij} = \{t, x_i(t), y_i(t), x_j(t), y_j(t)\}. \tag{38}$$

By Lemma E.1, the number of functionally independent invariants is $\dim JE_{ij} - s_k = 4$, where $\dim JE_{ij} = 5$ and $s_k = 1$. A trivial invariant is time t. Thus, our goal is to find other three invariants. The vector field X can be written as

$$X = -y_i \frac{\partial}{\partial x_i} + x_i \frac{\partial}{\partial y_i} - y_j \frac{\partial}{\partial x_j} + x_j \frac{\partial}{\partial y_j}, \tag{39}$$

The invariants should satisfy $X(\mathcal{I}) = 0$. Then, the corresponding characteristic equation can be expressed as

$$\frac{dx_i}{-y_i} = \frac{dy_i}{x_i} = \frac{dx_j}{-y_j} = \frac{dy_j}{x_j} \tag{40}$$

From the characteristic equation (40), the object-specific invariants are

$$\begin{aligned} \frac{dx_i}{-y_i} = \frac{dy_i}{x_i} &\implies I_{p_i} = r_i^2 = x_i^2 + y_i^2 \\ \frac{dx_j}{-y_j} = \frac{dy_j}{x_j} &\implies I_{p_j} = r_j^2 = x_j^2 + y_j^2 \end{aligned} \tag{41}$$

and the edge differential invariants are as following:

$$\frac{dx_i}{-y_i} = \frac{dy_j}{x_j}, \frac{dy_i}{x_i} = \frac{dx_j}{-y_j} \implies I_{p_i p_j} = x_i x_j + y_i y_j \tag{42}$$

Therefore, we have found all 3 non-trivial differential invariants as

$$\mathcal{I} = \{I_{p_i}, I_{p_j}, I_{p_i p_j}\}. \tag{43}$$

It is clear to find that the rank of $\mathbf{J}(\mathcal{I})$ is 3, indicating that we have found all functionally independent non-trivial invariants. Next, we construct complete frames. To construct them, we first extend 2D object positions to 3D via zero-padding $p_i(t) = [x_i(t), y_i(t), 0]^\top$. Analogous to the construction in (10), the complete frames are formulated as:

$$\boldsymbol{e}_{ij}^{p_x}(t) = \frac{\boldsymbol{q}_i(t) - \boldsymbol{q}_j(t)}{\|\boldsymbol{q}_i(t) - \boldsymbol{q}_j(t)\|}, \quad \boldsymbol{e}_{ij}^{p_y}(t) = \frac{\boldsymbol{q}_i(t) \times \boldsymbol{q}_j(t)}{\|\boldsymbol{q}_i(t) \times \boldsymbol{q}_j(t)\|} \times \boldsymbol{e}_{ij}^{p_x}(t). \tag{44}$$

After constructing these frames, we project them back to the 2D space by discarding the zero-padding component.

Two-dimensional Euclidean space with object positions and velocities First, we construct complete differential invariants. The vector field induced by $\mathrm{SO}(2)$ on $\mathbb{R}^2$ is in (37), and the prolonged graph of jet bundle $J^{(1)}E_{ij}$ can be expressed as

$$\gamma_{ij}^{(1)} = \{t, x_i(t), y_i(t), \dot{x}_i(t), \dot{y}_i(t), x_j(t), y_j(t), \dot{x}_j(t), \dot{y}_j(t)\}. \tag{45}$$

According to Lemma E.1, the number of functionally independent local differential invariants is $\dim J^{(1)}E - s_k = 8$, where $\dim J^{(1)}E = 9$ and $s_k = 1$. Obviously, a trivial differential invariant is time t. Thus, our goal is to find other 7 invariants.The prolonged vector field $X^{(1)}$ can be written as

$$X^{(1)} = -y_i \frac{\partial}{\partial x_i} + x_i \frac{\partial}{\partial y_i} - y_j \frac{\partial}{\partial x_j} + x_j \frac{\partial}{\partial y_j} - \dot{y}_i \frac{\partial}{\partial \dot{x}_i} + \dot{x}_i \frac{\partial}{\partial \dot{y}_i} - \dot{y}_j \frac{\partial}{\partial \dot{x}_j} + \dot{x}_j \frac{\partial}{\partial \dot{y}_j}, \tag{46}$$

Then, the corresponding characteristic equation can be expressed as

$$\frac{dx_i}{-y_i} = \frac{dy_i}{x_i} = \frac{d\dot{x}_i}{-\dot{y}_i} = \frac{d\dot{y}_i}{\dot{x}_i} = \frac{dx_j}{-y_j} = \frac{dy_j}{x_j} = \frac{d\dot{x}_j}{-\dot{y}_j} = \frac{d\dot{y}_j}{\dot{x}_j} \tag{47}$$

From the characteristic equation (47), the object-specific differential invariants are found to be

$$\begin{aligned} \frac{dx_i}{-y_i} = \frac{dy_i}{x_i} &\implies I_{p_i} = r_i^2 = x_i^2 + y_i^2 \\ \frac{d\dot{x}_i}{-\dot{y}_i} = \frac{d\dot{y}_i}{\dot{x}_i} &\implies I_{v_i} = v_i^2 = \dot{x}_i^2 + \dot{y}_i^2 \\ \frac{dx_j}{-y_j} = \frac{dy_j}{x_j} &\implies I_{p_j} = r_j^2 = x_j^2 + y_j^2 \\ \frac{d\dot{x}_j}{-\dot{y}_j} = \frac{d\dot{y}_j}{\dot{x}_j} &\implies I_{v_j} = v_j^2 = \dot{x}_j^2 + \dot{y}_j^2 \\ \frac{dy_i}{x_i} = \frac{d\dot{x}_i}{-\dot{y}_i}, \frac{dx_i}{-y_i} = \frac{d\dot{y}_i}{\dot{x}_i} &\implies I_{p_i v_i} = x_i \dot{x}_i + y_i \dot{y}_i, \end{aligned} \tag{48}$$

and the edge differential invariants are as following:

$$\begin{aligned}\frac{dx_i}{-y_i}=\frac{dy_j}{x_j},\frac{dy_i}{x_i}=\frac{dx_j}{-y_j} &\implies I_{p_ip_j}=x_ix_j+y_iy_j\\ \frac{d\dot{x}_i}{-\dot{y}_i}=\frac{d\dot{y}_j}{\dot{x}_j},\frac{d\dot{y}_i}{\dot{x}_i}=\frac{d\dot{x}_j}{-\dot{y}_j} &\implies I_{v_iv_j}=\dot{x}_i\dot{x}_j+\dot{y}_i\dot{y}_j.\end{aligned} \tag{49}$$

Up to this point, we have found all 7 differential invariants as

$$\mathcal{I}=\{I_{p_i},I_{v_i},I_{p_j},I_{v_j},I_{p_iv_i},I_{p_ip_j},I_{v_iv_j}\}. \tag{50}$$

It is easy to check the Jacobian matrix $\mathbf{J}(\mathcal{I})$ is full rank. Next, we construct complete differential frames. Similarly, we first extend 2D states to 3D via zero-padding

$$p_i(t)=[x_i(t),y_i(t),0]^\top, v_i(t)=[\dot{x}_i(t),\dot{y}_i(t),0]^\top. \tag{51}$$

Analogous to the 3D frame construction in (10), the 2D differential frames are defined as:

$$\begin{aligned}\boldsymbol{e}_{ij}^{p_x}(t)&=\frac{\boldsymbol{q}_i(t)-\boldsymbol{q}_j(t)}{\|\boldsymbol{q}_i(t)-\boldsymbol{q}_j(t)\|},\quad \boldsymbol{e}_{ij}^{p_y}(t)=\frac{\boldsymbol{q}_i(t)\times\boldsymbol{q}_j(t)}{\|\boldsymbol{q}_i(t)\times\boldsymbol{q}_j(t)\|}\times\boldsymbol{e}_{ij}^{p_x}(t),\\ \boldsymbol{e}_{ij}^{v_x}(t)&=\frac{\dot{\boldsymbol{q}}_i(t)-\dot{\boldsymbol{q}}_j(t)}{\|\dot{\boldsymbol{q}}_i(t)-\dot{\boldsymbol{q}}_j(t)\|},\quad \boldsymbol{e}_{ij}^{v_z}(t)=\frac{\dot{\boldsymbol{q}}_i(t)\times\dot{\boldsymbol{q}}_j(t)}{\|\dot{\boldsymbol{q}}_i(t)\times\dot{\boldsymbol{q}}_j(t)\|}\times\boldsymbol{e}_{ij}^{v_x}(t).\end{aligned} \tag{52}$$

After constructing these frames in 3D, we project them back to the 2D space by discarding the zero-padding component. It is straightforward to verify that the frames defined in (52) are equivariant under $\mathrm{SO}(2)$ group actions.

Three-dimensional Euclidean space with object positions only Compared with the 2D case, constructing complete invariants in 3D space is more challenging. The vector field induced by $\mathrm{SO}(2)$ is single, while the vector fields induced by $\mathrm{SO}(3)$ are multiple, which can be expressed as

$$X_1=-y\frac{\partial}{\partial z}+z\frac{\partial}{\partial y},\quad X_2=-z\frac{\partial}{\partial x}+x\frac{\partial}{\partial z},\quad X_3=-x\frac{\partial}{\partial y}+y\frac{\partial}{\partial x}. \tag{53}$$

The graph of jet bundle JE_{ij} is

$$\gamma_{ij}=\{t,x_i(t),y_i(t),z_i(t),x_j(t),y_j(t),z_j(t)\}. \tag{54}$$

By Lemma E.1, the number of functionally independent invariants is $\dim J^{(k)}E-s_k=4$ with $\dim J^{(k)}E=7$ and $s_k=3$. Similarly, t serves as a trivial invariant, leaving us to identify the remaining 3 functionally independent invariants. For the jet bundle JE_{ij}, the vector fields are defined as

$$\begin{aligned}X_1&=-y_i\frac{\partial}{\partial z_i}+z_i\frac{\partial}{\partial y_i}-y_j\frac{\partial}{\partial z_j}+z_j\frac{\partial}{\partial y_j},\\ X_2&=-z_i\frac{\partial}{\partial x_i}+x_i\frac{\partial}{\partial z_i}-z_j\frac{\partial}{\partial x_j}+x_j\frac{\partial}{\partial z_j},\\ X_3&=-x_i\frac{\partial}{\partial y_i}+y_i\frac{\partial}{\partial x_i}-x_j\frac{\partial}{\partial y_j}+y_j\frac{\partial}{\partial x_j}.\end{aligned} \tag{55}$$

The invariants $\mathcal{I}$ must satisfy

$$X_i(\mathcal{I})=0,\quad i=1,2,3. \tag{56}$$

Leveraging the properties of Lie groups, we can reduce this problem to satisfying just two of these three equations. For group $\mathrm{SO}(3)$, the Lie bracket relations hold:

$$[X_1,X_2]=X_3,\quad [X_2,X_3]=X_1,\quad [X_3,X_1]=X_2. \tag{57}$$

With the property (57), we can reduce the problem (56) to satisfying just two of these three equations. Specifically, if the invariants satisfy $X_1(\mathcal{I})=0$ and $X_2(\mathcal{I})=0$, then the third condition $X_3(\mathcal{I})=0$ follows automatically

$$X_3(\mathcal{I})=[X_1,X_2](\mathcal{I})=X_1(X_2(\mathcal{I}))-X_2(X_1(\mathcal{I}))=0. \tag{58}$$

Thus, we first compute $X_1(\mathcal{I}) = 0$. The corresponding characteristic equation is

$$\frac{dz_i}{-y_i} = \frac{dy_i}{z_i} = \frac{dz_j}{-y_j} = \frac{dy_j}{z_j} \tag{59}$$

The same as SO(2) acting on 2D plane, there are 3 differential invariants at this stage, as shown in (60).

$$\begin{aligned} \frac{dz_i}{-y_i} = \frac{dy_i}{z_i} &\implies I_1 = y_i^2 + z_i^2 \\ \frac{dz_j}{-y_j} = \frac{dy_j}{z_j} &\implies I_2 = y_j^2 + z_j^2 \\ \frac{dz_i}{-y_i} = \frac{dy_j}{z_j}, \frac{dy_i}{z_i} = \frac{dz_j}{-y_j} &\implies I_3 = y_i y_j + z_i z_j. \end{aligned} \tag{60}$$

Then, the invariants $\mathcal{I}$ can be treated as functions of $(x_i, x_j, I_1, I_2, I_3)$, i.e., $\mathcal{I} = F(x_i, x_j, I_1, I_2, I_3)$. Substitute the invariants I to $X_2\mathcal{I} = 0$:

$$\begin{aligned} X_2\mathcal{I} = -z_i\frac{\partial F}{\partial x_i} - z_j\frac{\partial F}{\partial x_j} \\ + 2x_i z_i\frac{\partial F}{\partial I_1} + 2x_j z_j\frac{\partial F}{\partial I_2} + (x_i z_j + x_j z_i)\frac{\partial F}{\partial I_3} = 0, \end{aligned} \tag{61}$$

whose corresponding characteristic equation can be expressed as

$$\frac{dx_i}{-z_i} = \frac{dx_j}{-z_j} = \frac{dI_1}{2x_i z_i} = \frac{dI_2}{2x_j z_j} = \frac{dI_3}{x_i z_j + x_j z_i} \tag{62}$$

From this, 3 functionally independent invariants are derived as:

$$\begin{aligned} \frac{dx_i}{-z_i} = \frac{dI_1}{2x_i z_i} &\implies I_{p_i} = x_i^2 + y_i^2 + z_i^2 \\ \frac{dx_j}{-z_j} = \frac{dI_2}{2x_j z_j} &\implies I_{p_j} = x_j^2 + y_j^2 + z_j^2 \\ \frac{dx_i}{-z_i} = \frac{dx_j}{-z_j} = \frac{dI_3}{x_i z_j + x_j z_i} &\implies I_{p_i p_j} = x_i x_j + y_i y_j + z_i z_j. \end{aligned} \tag{63}$$

Up to this point, we have found all 3 functionally nontrivial differential invariants as

$$\mathcal{I} = \{I_{p_i}, I_{p_j}, I_{p_i p_j}\}. \tag{64}$$

It is easy to check the Jacobian matrix $\mathbf{J}(\mathcal{I})$ is full rank. Next, we construct complete frames as follows

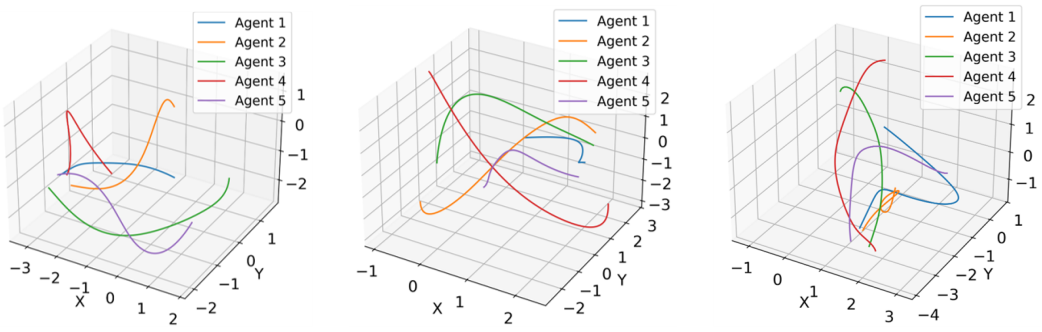
$$\boldsymbol{e}_{ij}^{p_x}(t) = \frac{\boldsymbol{q}_i(t) - \boldsymbol{q}_j(t)}{\|\boldsymbol{q}_i(t) - \boldsymbol{q}_j(t)\|}, \quad \boldsymbol{e}_{ij}^{p_y}(t) = \frac{\boldsymbol{q}_i(t) \times \boldsymbol{q}_j(t)}{\|\boldsymbol{q}_i(t) \times \boldsymbol{q}_j(t)\|}, \quad \boldsymbol{e}_{ij}^{p_z}(t) = \boldsymbol{e}_{ij}^{p_x}(t) \times \boldsymbol{e}_{ij}^{p_y}(t). \tag{65}$$

It is straightforward to verify that the frames defined in (65) are equivariant under SO(3) group actions.

Complete differential invariants on more complex cases Complete differential invariants can be applied to more complex scenarios. For instance, when considering two-hop neighbors, the jet bundle can be denoted as $J^{(1)}E_{ijk}$, where $j \in \mathcal{N}_i$ and $k \in \mathcal{N}_j$. The corresponding prolonged graph $\gamma_{ijk}^{(1)}$ is expressed as

$$\begin{aligned} \gamma_{ijk}^{(1)} = \{t, &x_i(t), y_i(t), z_i(t), \dot{x}_i(t), \dot{y}_i(t), \dot{z}_i(t), \\ &x_j(t), y_j(t), z_j(t), \dot{x}_j(t), \dot{y}_j(t), \dot{z}_j(t), \\ &x_k(t), y_k(t), z_k(t), \dot{x}_k(t), \dot{y}_k(t), \dot{z}_k(t).\} \end{aligned} \tag{66}$$

By Lemma E.1, the number of non-trivial functionally independent invariants is $\dim J^{(1)}E_{ijk} - s_k = 19 - 3 - 1 = 15$. Another scenario arises when higher-order derivatives are incorporated, such as



(a) Easy: 3D Trajectory (b) Medium: 3D Trajectory (c) Difficult: 3D Trajectory

Figure 5: Formation dynamics across different difficulty levels (Easy/Medium/Difficult, left to right). Each column presents the 3D trajectories. Trajectories are stable, with velocities reflecting increasing inter-agent interaction complexity encoded in ω_{ij}.

observable object states including second-order derivatives (i.e., acceleration $\boldsymbol{a}(t)$). In this case, the prolonged graph $\gamma_{ij}^{(2)}$ of the jet bundle $J^{(2)}E_{ij}$ can be formulated as (67) and the number of non-trivial functionally independent invariants is 15.

$$\begin{aligned}\gamma_{ij}^{(2)} = \{t, &x_i(t), y_i(t), z_i(t), x_j(t), y_j(t), z_j(t),\\ &\dot{x}_i(t), \dot{y}_i(t), \dot{z}_i(t), \dot{x}_j(t), \dot{y}_j(t), \dot{z}_j(t),\\ &\ddot{x}_i(t), \ddot{y}_i(t), \ddot{z}_i(t), \ddot{x}_j(t), \ddot{y}_j(t), \ddot{z}_j(t), \}\end{aligned} \tag{67}$$

It is evident that the number of non-trivial functionally independent invariants can be readily determined by Lemma E.1 in such cases. While this count increases as scenarios become more complex, constructing explicit forms of complete differential invariants remains not difficult but rather tedious, as it entails a systematic, step-by-step derivation of each invariant to ensure functional independence.

G A Novel Benchmark: Formation Consensus Control

In this section, we introduce the implementation of a novel benchmark, formation consensus control, designed to evaluate equivariant graph neural networks across hierarchical interaction complexities. The core task of formation consensus control is coordinating agents to achieve unified states, such as synchronized positions, velocities, through inter-agent interactions. In this system, the agents follow double-integrator dynamics, where their accelerations depend on neighbors' positions and velocities

$$\dot{\boldsymbol{v}}_i = -\sum_{j\in\mathcal{N}_i} \omega_{ij}\left(k_p(\boldsymbol{q}_i - \boldsymbol{q}_j) + k_v(\dot{\boldsymbol{q}}_i - \dot{\boldsymbol{q}}_j) + k_d(\dot{\boldsymbol{q}}_j \times \dot{\boldsymbol{q}}_i)\right), \tag{68}$$

with stability ensured by $k_p > 0$ and $k_v > 0$. The dataset is divided into three levels based on the complexity of interaction coefficients ω_{ij}. At the easy level, ω_{ij} involves only edge-wise position and velocity dot products, forming low-complexity interactions with linearity:

$$\omega_{ij} = 0.2 \times \|\boldsymbol{q}_i \cdot \boldsymbol{q}_j + \dot{\boldsymbol{q}}_i \cdot \dot{\boldsymbol{q}}_j\|. \tag{69}$$

The medium level introduces a velocity-position cross-term ($\dot{\boldsymbol{q}}_i \cdot (\boldsymbol{q}_i \times \boldsymbol{q}_j)$), introducing bilinear interactions and geometric dependencies:

$$\omega_{ij} = 0.2 \times \|\boldsymbol{q}_i \cdot \boldsymbol{q}_j + \dot{\boldsymbol{q}}_i \cdot \dot{\boldsymbol{q}}_j + \dot{\boldsymbol{q}}_i \cdot (\boldsymbol{q}_i \times \boldsymbol{q}_j)\| \tag{70}$$

The difficult level further incorporates higher-order cross interactions between positions and velocities, such as triple products and cross product norms, creating a highly non-linear, multi-modal interaction structure:

$$\begin{aligned}\omega_{ij} = 0.15 \times \|&\boldsymbol{q}_i \cdot \boldsymbol{q}_j + \dot{\boldsymbol{q}}_i \cdot \dot{\boldsymbol{q}}_j + \dot{\boldsymbol{q}}_i \cdot (\boldsymbol{q}_i \times \boldsymbol{q}_j)\\ &+ \dot{\boldsymbol{q}}_j \cdot (\boldsymbol{q}_i \times \boldsymbol{q}_j) + \boldsymbol{q}_i \cdot (\dot{\boldsymbol{q}}_i \times \dot{\boldsymbol{q}}_j)\\ &+ \boldsymbol{q}_j \cdot (\dot{\boldsymbol{q}}_i \times \dot{\boldsymbol{q}}_j) + (\boldsymbol{q}_i \times \boldsymbol{q}_j) \cdot (\dot{\boldsymbol{q}}_i \times \dot{\boldsymbol{q}}_j)\|.\end{aligned} \tag{71}$$

This hierarchy progressively challenges equivariant graph neural networks to model increasingly intricate spatial and velocity interactions. Moreover, the agent dynamics in (68) transcend radial interactions ($\boldsymbol{q}_i - \boldsymbol{q}_j$) by introducing a velocity cross-term $\dot{\boldsymbol{q}}_j \times \dot{\boldsymbol{q}}_i$, which drives state updates along tangential directions in the space. This term requires equivariant graph neural networks to handle state update process in entire space rather than merely relying on a single direction.

Figure 5 visualizes trajectories and velocity magnitudes across levels, illustrating system behaviors under varying interaction complexities. To ensure data validity, we truncate sequences at the first 1000 timesteps, excluding post-convergence data where agents reach unified states. Importantly, all trajectories exhibit stable and convergent behavior, confirming that the benchmark provides a rigorous testbed for evaluating network expressiveness.

H Implementation Details, Ablation Studies, and Additional Experiments

H.1 Implementation Details.

For the molecular dynamics experiment, the dataset is partitioned into 3000 training, 2000 validation, and 2000 test samples. In the CMU motion capture experiment, the corresponding splits are 2000 training, 600 validation, and 600 test samples. For the formation consensus control experiment, the dataset is divided into 6000 training samples, 2000 validation samples, and 1000 test samples, with the objective of predicting object positions after 600 timesteps. The n-body simulation experiments evaluate models on two scenarios, charged(5) and charged(10). Each scenario employs 3000 training trajectories, 2000 validation trajectories, and 2000 test trajectories, with each trajectory containing 10,000 timesteps. Across all experiments, the training procedure maintains a batch size of 100 and a maximum of 500 training epochs. Moreover, Models are optimized using the AdamW optimizer with mean squared error (MSE) loss. Additionally, models are optimized using the AdamW optimizer with mean squared error (MSE) loss. To ensure fair comparisons, hyperparameters are individually tuned for each model and task. All experiments are conducted on a single NVIDIA GeForce RTX 4090 24GB GPU. Detailed hyperparameters specific to CDIF across different experimental setups are listed in Table 5.

Table 5: Model and training hyperparameters for our method on different tasks.

Hyperparameter	Scenarios			
	FC	MD17	Charged	CMU Motion
Number of layers	4	4	4	4
Feature embeddings	64	64	32	64
Epochs	500	500	500	500
Batch size	100	100	100	100
Learning rate	1e-3	1e-3	1e-3	1e-3
Learning rate scheduler	steplr	steplr	steplr	steplr
Learning rate decay factor	0.9	0.98	0.9	0.98
Learning rate decay epochs	100	100	100	100

H.2 Standard Deviations

We provide the standard deviations in Table 6 and 7.

H.3 Analysis of Invariant Complexity

This section investigates the impact of varying numbers of invariants on model performance for the FC(5) task. Frames are constructed using the complete differential frame in (10) to preserve directional information. For fair comparison, the initialization stage encodes no agent geometric features,

Table 6: Prediction results of CDIF on FC(5) and FC(10) tasks. Results (mean ± standard deviation) are computed over 5 runs

Task	FC(5)			FC(10)		
Level	Easy	Medium	Difficult	Easy	Medium	Difficult
	MSE	**MSE**	**MSE**	**MSE**	**MSE**	**MSE**
CDIF	0.073±0.008	0.173±0.020	0.326±0.016	0.017±0.010	0.037±0.015	0.510±0.042

Table 7: Mean Square Error (MSE) for position prediction over Charged(5) and Charged(10) tasks at different prediction time steps. The best results are **bolded** and the second best are underlined.

Task	Charged(5)			Charged(10)		
Timestep	10	20	40	10	20	40
Model	**MSE**	**MSE**	**MSE**	**MSE**	**MSE**	**MSE**
EGNN+CDIs	0.0845 ± 0.0010	0.5363 ± 0.0013	2.8754 ± 0.0032	0.1058 ± 0.0028	0.7457 ± 0.0016	4.3436 ± 0.0105
GMN+CDIs	0.0902 ± 0.0013	0.5424 ± 0.0007	3.0426 ± 0.0027	0.1056 ± 0.0024	0.7712 ± 0.0030	4.4403 ± 0.0102
CDIF(ours)	0.0689 ± 0.0013	0.4214 ± 0.0015	2.3834 ± 0.0025	0.0803 ± 0.0021	0.5829 ± 0.0047	3.5131 ± 0.0070

including only intrinsic properties like mass and charge. We design six models with incremental invariant complexity.

Model 1: the inputs include only pairwise dot products $\boldsymbol{q}_i \cdot \boldsymbol{q}_j$.

Model 2: we replace dot products with squared Euclidean distances $\|\boldsymbol{q}_i - \boldsymbol{q}_j\|^2$ to compare positional encoding strategies.

Model 3: we introduce complete positional invariants to capture all geometric symmetries dependent on agent positions.

$$\{\|\boldsymbol{q}_i\|^2, \|\boldsymbol{q}_j\|^2, \|\boldsymbol{q}_i - \boldsymbol{q}_j\|^2\}, \tag{72}$$

Model 4: we extend the case 3 with velocity-related invariants

$$\{\|\boldsymbol{q}_i\|^2, \|\boldsymbol{q}_j\|^2, \|\boldsymbol{q}_i - \boldsymbol{q}_j\|^2\}, \quad \{\|\dot{\boldsymbol{q}}_i\|^2, \|\dot{\boldsymbol{q}}_j\|^2, \|\dot{\boldsymbol{q}}_i - \dot{\boldsymbol{q}}_j\|^2\}, \tag{73}$$

Model 5: we employ complete differential invariants in (7) as inputs of the layers.

Model 6: we further augment the complete differential invariants to investigate the impact of incorporating additional invariants on model performance.

$$\begin{aligned}
&\{\|\boldsymbol{q}_i\|^2, \|\dot{\boldsymbol{q}}_i\|^2, \boldsymbol{q}_i \cdot \dot{\boldsymbol{q}}_i, \|\boldsymbol{q}_i \times \dot{\boldsymbol{q}}_i\|^2\}, \quad \{\|\boldsymbol{q}_j\|^2, \|\dot{\boldsymbol{q}}_j\|^2, \boldsymbol{q}_j \cdot \dot{\boldsymbol{q}}_j, \|\boldsymbol{q}_j \times \dot{\boldsymbol{q}}_j\|^2\}, \\
&\{\boldsymbol{q}_i \cdot \boldsymbol{q}_j, \|\boldsymbol{q}_i - \boldsymbol{q}_j\|, \dot{\boldsymbol{q}}_i \cdot \dot{\boldsymbol{q}}_j, \|\dot{\boldsymbol{q}}_i - \dot{\boldsymbol{q}}_j\|\}, \quad \{(\boldsymbol{q}_i \times \boldsymbol{q}_j) \cdot (\dot{\boldsymbol{q}}_i \times \dot{\boldsymbol{q}}_j)\} \\
&\{\dot{\boldsymbol{q}}_i \cdot (\boldsymbol{q}_i \times \boldsymbol{q}_j), \dot{\boldsymbol{q}}_j \cdot (\boldsymbol{q}_i \times \boldsymbol{q}_j), \boldsymbol{q}_i \cdot (\dot{\boldsymbol{q}}_i \times \dot{\boldsymbol{q}}_j), \boldsymbol{q}_j \cdot (\dot{\boldsymbol{q}}_i \times \dot{\boldsymbol{q}}_j)\}, \\
&\{\boldsymbol{q}_j \cdot (\boldsymbol{q}_i \times \dot{\boldsymbol{q}}_i), \dot{\boldsymbol{q}}_j \cdot (\boldsymbol{q}_i \times \dot{\boldsymbol{q}}_i), \boldsymbol{q}_i \cdot (\boldsymbol{q}_j \times \dot{\boldsymbol{q}}_j), \dot{\boldsymbol{q}}_i \cdot (\boldsymbol{q}_j \times \dot{\boldsymbol{q}}_j)\},
\end{aligned} \tag{74}$$

Then, we evaluate the proposed models across the easy, medium, and difficult levels of the FC(5) task. The results are shown in Table 8.

Table 8: Mean Square Error for position prediction on FC(5) tasks across three different levels. The best results are **bolded** and the second best are underlined.

FC(5)	**Model 1**	**Model 2**	**Model 3**	**Model 4**	**Model 5**	**Model 6**
Easy	0.1109	0.1382	0.0934	0.0884	**0.0681**	0.0814
Medium	0.3173	0.3204	0.2785	0.2194	**0.1623**	0.1717
Difficult	0.4357	0.4380	0.3973	0.3683	0.3242	**0.2781**

Models 1 and 2 exhibit nearly identical performance across all levels, indicating that $\boldsymbol{q}_i \cdot \boldsymbol{q}_j$ and $\|\boldsymbol{q}_i - \boldsymbol{q}_j\|^2$ serve similar roles in encoding the relative angular relationships between agents i and j. A significant performance jump occurs between model 2 and 3, with average improvements of 18.3%. This highlights the critical role of complete positional invariants, which enable smooth function approximation of all position-dependent invariants. Incorporating velocity-related invariants in model 4 yields average improvements of 11.3% over model 3, which demonstrates that invariants based on velocities enhance model's expressiveness. Utilizing complete differential invariants in model 5 achieves an average improvement of 20.4% over model 4. This corroborates the importance of differential completeness, which significantly boosts modeling of complex dynamics. Regarding model 6, its performance varies across difficulty levels. It performs worse than model 5 on Easy and Medium tasks, suggesting that excessive invariants may introduce redundant features and degrade performance in simpler scenarios due to increased input dimensionality. However, on the Difficult level, model 6 achieves the best overall result, outperforming model 5 significantly. This indicates that while redundant features may hinder simple tasks, the comprehensive invariants are crucial for modeling highly complex dynamics.

H.4 Ablation Study.

We introduce two core modules to ensure completeness, namely complete differential invariants and complete differential frames. To demonstrate the importance of each module, we conduct experiments on FC(5) and FC(10) tasks, with results summarized in Table 9. The results show that introducing complete differential invariants alone improves the performance of the model lacking both complete invariants and frames by an average of 33.5%. Furthermore, incorporating complete positional frames yields moderate improvements on FC(5) tasks (averaging 12.3%), but offers limited benefits for FC(10) tasks. Finally, adding complete differential frames yields an additional average improvement of 33.2%. The results highlight the critical role of two modules in enhancing model performance.

Table 9: Ablation study on FC(5) and FC(10) tasks. Best results are **bolded**. CDIF(w/o CDIs and CDFs). CDIF(CDIs only) incorporates complete differential invariants as inputs at each layer while retaining radial direction via relative displacements. CDIF(CDIs + positional CFs) constructs positional complete frames (65) for the update model.

Model	FC(5) (MSE)			FC(10) (MSE)		
	Easy	Medium	Difficult	Easy	Medium	Difficult
CDIF(w/o CDIs and CDFs)	0.1896	0.4304	0.5827	0.0701	0.1363	0.9616
CDIF(CDIs only)	0.1109	0.3173	0.4357	0.0343	0.1076	0.6063
CDIF(CDIs + positional CFs)	0.0934	0.2785	0.3973	0.0318	0.1137	0.5624
CDIF	**0.0732**	**0.1726**	**0.3268**	**0.0174**	**0.0368**	**0.5103**