EQDYNHYPER: EQUIVARIANT DYNAMIC HYPERGRAPH FOR GROUP-LEVEL MOTION PREDICTION AND GROUP DISCOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-agent motion forecasting must respect geometric symmetries while modeling both pairwise interactions and emergent collective group dynamics. Existing Euclidean-equivariant predictors rely on pairwise message passing and have no native group-level primitive, whereas hypergraph forecasters that do model groups are not Euclidean-equivariant. We present **EqDynHyper**, an equivariant hypergraph network with two disentangled outputs: trajectory forecasts, whose accuracy is insensitive to grouping quality and whose efficiency comes from *equivariant centroid aggregation* over hyperedges, and agent-group partitions, recovered without membership supervision through Euclidean-invariant soft hyperedge assignments. Building on the scalar–vector separation of GVP-GNN and PaiNN, we add an explicit affine Point type and the closure rules needed for higher-order aggregation, so that trajectories are equivariant and assignments invariant by construction. Assignments are computed once by default and can optionally be re-inferred at every layer. We find that competitive (softmax) normalization of the assignment matrix, rather than auxiliary entropy losses, is what prevents hyperedge collapse. EqDynHyper replaces $O(N^2)$ vector-valued pairwise message passing with $O(NK)$ hyperedge aggregation; a scalar pairwise topology term remains $O(N^2)$, and at $N{=}400$ inference is $9.4\times$ faster with $3.75\times$ less memory than EqMotion. On synthetic collective dynamics EqDynHyper lowers ADE by $2.9\times$ relative to EqMotion, and on ETH–UCY by 5.0% under a unified retraining protocol; on bond-dominated MD17 molecules it is less accurate than pairwise models, which we report together with other failure modes.

1 INTRODUCTION

Multi-agent motion forecasting is vital for autonomous driving, robotics and physical simulation (Salzmann et al., 2020; Ivanovic & Pavone, 2019; Chmiela et al., 2017). Real-world systems require reasoning about both pairwise interactions and emergent collective behaviors, such as groups splitting and merging or rotating as rigid bodies. Capturing such time-varying, higher-order group dependencies remains an open challenge.

Early recurrent models with social pooling encode interactions without an explicit graph, which limits interpretability (Alahi et al., 2016). Diffusion-based methods generate diverse trajectories (Gu et al., 2022) but do not enforce Euclidean symmetry, so their predictions can be physically inconsistent (Mangalam et al., 2020). Graph- and hypergraph-based approaches model interactions and agent coalitions explicitly (Battaglia et al., 2018; Xu et al., 2022), and Euclidean-equivariant networks such as EGNN and EqMotion enforce physical consistency through pairwise message passing (Satorras et al., 2021; Xu et al., 2023).

These families have complementary weaknesses. Pairwise equivariant architectures pay an $O(N^2)$ cost for vector-valued messages and have no native group representation (Satorras et al., 2021; Xu et al., 2023), whereas hypergraph and group-aware forecasters (Xu et al., 2022; 2024; Zhou et al., 2022) are not Euclidean-equivariant.

We present EqDynHyper, an equivariant hypergraph network for group-aware multi-agent forecasting that combines Euclidean-invariant group inference with typed P/V/S hypergraph message passing. Our ablations separate its two outputs. Trajectory accuracy is insensitive to grouping quality—a single hyperedge or random groups reach similar error—so the forecasting gain does not come from

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.

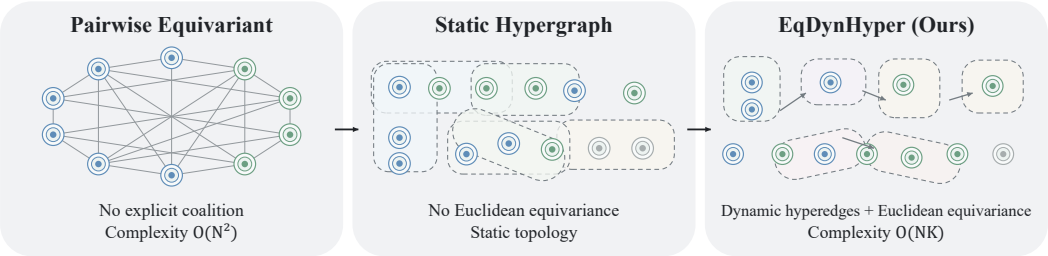

Figure 1: Comparison: (1) Pairwise equivariant baselines (EqMotion) exchange vector-valued messages over $O(N^2)$ edges and have no group representation. (2) Hypergraph forecasters (GroupNet) model groups but are not Euclidean-equivariant. (3) EqDynHyper aggregates vector-valued features over K learned, Euclidean-invariant hyperedges; its vector branch costs $O(NK)$, while a scalar pairwise topology term remains $O(N^2)$. The $O(NK)$ label in panel (3) refers to the vector branch.

recovering the correct groups; we hypothesize that it comes from equivariant centroid aggregation, a mechanism we do not fully isolate (Sec. 5.5). Group discovery is therefore an independent, interpretable output rather than a prerequisite for forecasting.

EqDynHyper replaces costly $O(N^2)$ vector-valued pairwise message passing with $O(NK)$ hyperedge aggregation, while its scalar topology computation remains $O(N^2)$; in the default static-topology mode this yields up to a $9.4\times$ inference speedup. Layer-wise re-inference of the topology is optional.

Our contributions are threefold:

1. We construct EqDynHyper on a P/V/S geometric type system that extends the scalar–vector separation of GVP-GNN (Jing et al., 2021) and PaiNN (Schütt et al., 2021) with an affine Point type, and add the two closure rules required for higher-order aggregation: affine S-weighted combinations of Points (hyperedge centroids) and Point-minus-weighted-mean Vectors. By construction, the network is $E(n)$-equivariant in its trajectory outputs and $E(n)$-invariant in its soft group assignments. EqMotion infers invariant *pairwise* interaction categories (an $N \times N$ edge typing); we infer an invariant *soft partition* of agents into K interaction units (an $N \to K$ assignment) and build message passing on the induced hyperedges.
2. Controlled ablations with a single hyperedge ($K = 1$) and random fixed groups show that forecasting accuracy is insensitive to grouping quality, which separates forecasting accuracy from group discovery. EqDynHyper lowers ADE by $2.9\times$ relative to EqMotion (up to $6.7\times$ under weak group coupling) and reduces inference latency and memory at large agent counts.
3. We show that competitive normalization of the assignment matrix suffices to prevent hyperedge collapse, and analyze when entropy-based load balancing—adapted from graph pooling and mixture-of-experts routing—helps or hurts group discovery. We evaluate on synthetic collective-dynamics, pedestrian and molecular-dynamics benchmarks, and report failure modes and trade-offs.

2 Related Work

Recurrent Social-LSTM Models. Recurrent models with social pooling aggregate the hidden states of neighboring agents (Alahi et al., 2016; Ivanovic & Pavone, 2019). Because interactions remain implicit in latent states, these models are hard to interpret and handle dynamic group-level phenomena poorly.

Graph-based Interaction Modeling for Motion Forecasting. Graph-based methods cast multi-agent prediction as relational inference over dynamic graphs (Battaglia et al., 2018; Salzmann et al., 2020), and latent-graph approaches infer the interaction structure itself (Kipf et al., 2018; Li et al., 2020). Standard graphs, however, decompose all interactions into pairwise edges, which costs $O(N^2)$ and provides no native representation of collective groups (Zhou et al., 2022).

Diffusion-based Multi-Agent Motion Modeling. Diffusion models generate multi-modal future trajectories by iterative denoising (Gu et al., 2022). Most diffusion-based forecasters do not impose Euclidean symmetry constraints, which risks physically implausible outputs in safety-critical settings (Mangalam et al., 2020).

Hypergraph and Higher-Order Interaction Modeling. Hypergraphs represent group-scale coalitions with hyperedges that connect arbitrary subsets of agents. GroupNet builds multiscale hypergraphs for crowd motion (Xu et al., 2022), and DynGroupNet (Xu et al., 2024) and Grouptron (Zhou et al., 2022) model time-varying groups, the former by inferring the topology across input time steps. Our optional re-inference operates at a different granularity: it refines a single assignment per observation window across network layers, and we do not claim it is more expressive than time-indexed topologies. We compare against its static predecessor GroupNet and leave a comparison with DynGroupNet to future work. None of these motion-oriented hypergraph models is Euclidean-equivariant.

Euclidean Equivariant Motion Prediction. Euclidean-equivariant networks behave consistently under rotations, translations and reflections (Cohen & Welling, 2016; Satorras et al., 2021; Fuchs et al., 2020). State-of-the-art models such as EGNN and EqMotion rely on pairwise message passing for physical and trajectory forecasting (Xu et al., 2023); their outputs are physically consistent, but they incur an $O(N^2)$ pairwise cost and lack explicit group-level primitives.

Equivariant Higher-Order Networks for Static Molecular Systems. MACE (Batatia et al., 2022) and related models build equivariant higher-order representations to predict properties of static molecules on fixed graphs; they do not learn soft hypergraph topologies for sequential multi-agent forecasting. EqDynHyper instead targets motion prediction and jointly infers the topology, forecasts trajectories and discovers group partitions.

Learned Soft Pooling and Assignment. Soft assignment of nodes to a small number of clusters is standard in graph pooling: DiffPool learns a softmax assignment matrix with an entropy regularizer on per-node assignments (Ying et al., 2018), and MinCutPool adds an orthogonality term to avoid degenerate clusters (Bianchi et al., 2020). Slot Attention refines soft assignments of inputs to slots through competitive normalization (Locatello et al., 2020), and load-balancing losses keep mixture-of-experts routing from collapsing onto few experts (Shazeer et al., 2017; Fedus et al., 2022). Our assignment and entropy terms follow these designs. The difference is that our assignment is constrained to be Euclidean-invariant and anchors equivariant centroid aggregation, so that the pooled geometric quantities transform correctly.

3 Background and Problem Formulation

3.1 Multi-Agent Motion Prediction

We study multi-agent trajectory forecasting under geometric constraints. The input $\mathbf{X} \in \mathbb{R}^{B\times N\times T_{\text{obs}}\times D}$ contains the observed trajectories of N agents over T_{obs} time steps in D dimensions, for a batch of B scenes. The goal is to learn a mapping $\mathcal{F} : \mathbf{X} \mapsto \widehat{\mathbf{Y}}$ that predicts the future coordinates $\widehat{\mathbf{Y}} \in \mathbb{R}^{B\times N\times T_f\times D}$ over T_f steps. Because physical and social dynamics do not depend on the choice of coordinate frame, the predictor should respect Euclidean symmetries.

3.2 Euclidean Equivariance and Invariance

Let $g = (\mathbf{R}, \mathbf{b})$ denote a Euclidean transform with $\mathbf{R} \in O(D)$ and $\mathbf{b} \in \mathbb{R}^D$, acting as $g\cdot\mathbf{X} = \mathbf{R}\mathbf{X}+\mathbf{b}$. A model $\mathcal{F}$ is *Euclidean-equivariant* if $\mathcal{F}(g \cdot \mathbf{X}) = \mathbf{R}\mathcal{F}(\mathbf{X}) + \mathbf{b}$, and a model $\mathcal{G}$ is *Euclidean-invariant* if $\mathcal{G}(g \cdot \mathbf{X}) = \mathcal{G}(\mathbf{X})$. Equivariance makes predictions consistent across coordinate frames; invariance makes the group partition independent of the frame.

3.3 P/V/S Geometric Type Primitives

Our type system extends the scalar–vector separation of GVP-GNN (Jing et al., 2021) and PaiNN (Schütt et al., 2021) with an explicit Point type that transforms affinely. Under a Euclidean transform g, the three primitive types transform as follows: (1) **P (Point)**: $p \mapsto \mathbf{R}p+\mathbf{b}$ (coordinates, hyperedge centroids); (2) **V (Vector)**: $v \mapsto \mathbf{R}v$ (velocities, displacements); (3) **S (Scalar)**: $s \mapsto s$ (distances, norms, hyperedge weights).

We add two closure rules needed for higher-order aggregation: an affine S-weighted combination of Points ($\sum_k a_k = 1$), used for hyperedge centroids, is a Point; and a Point minus a weighted mean of Points is a Vector. The full set of closure rules is given in Appendix A.2.

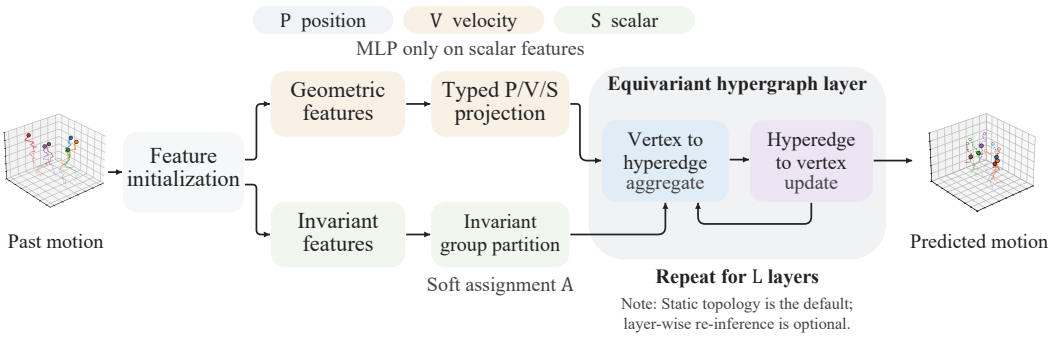

Figure 2: Overview of EqDynHyper. Past trajectories are parsed into Points $\mathbf{p}$, Vectors $\mathbf{v}$ and Scalars $\mathbf{s}$. The geometric branch computes position-aware features; the invariant branch computes rotation- and translation-invariant scalars, from which the group-partition module produces soft hyperedge assignments $\mathbf{A}$. Vertex-to-hyperedge aggregation computes centroids and pooled features, and hyperedge-to-vertex updates refine the agent states. The prediction head outputs equivariant trajectories. By default $\mathbf{A}$ is computed once (static topology); layer-wise re-inference is optional.

4 METHODOLOGY

Given trajectories $\mathbf{X}$, the initialization module $\mathcal{F}_{\text{init}}$ produces typed features $(\mathbf{r}^{(0)}, \mathbf{v}^{(0)}, \mathbf{h}^{(0)})$, which are processed by L hypergraph message-passing layers. In the default **static** mode the assignment is computed once and shared by all layers:

$$\mathbf{A} = \mathcal{F}_{\text{topo}}\big(\mathbf{r}^{(0)}, \mathbf{v}^{(0)}, \mathbf{h}^{(0)}\big), \tag{1}$$

$$\big(\mathbf{r}^{(\ell+1)}, \mathbf{v}^{(\ell+1)}, \mathbf{h}^{(\ell+1)}\big) = \mathcal{F}^{(\ell)}_{\text{hypermp}}\big(\mathbf{r}^{(\ell)}, \mathbf{v}^{(\ell)}, \mathbf{h}^{(\ell)}, \mathbf{A}\big). \tag{2}$$

In the optional **dynamic** mode the assignment is re-inferred at every layer, $\mathbf{A}^{(\ell)} = \mathcal{F}_{\text{topo}}(\mathbf{r}^{(\ell)}, \mathbf{v}^{(\ell)}, \mathbf{h}^{(\ell)})$, and $\mathbf{A}^{(\ell)}$ replaces $\mathbf{A}$ in layer ℓ. Neither mode produces time-indexed assignments. Re-inference lets deeper layers revise the grouping—for example, toward the post-switch grouping when memberships change inside the observation window (Table 3)—at a 3.3× higher inference cost for $N = 400$. The decoder outputs $\widehat{\mathbf{Y}} = \mathcal{F}_{\text{out}}\big(\mathbf{r}^{(L)}, \mathbf{v}^{(L)}, \mathbf{h}^{(L)}\big)$.

Constructive Equivariance through P/V/S Type Algebra. Equivariance is enforced by building every layer from type-closed primitives, such as the norm-gated nonlinearity $\psi(\mathbf{v}) = \sigma(\text{MLP}(\|\mathbf{v}\|)) \odot \mathbf{v}$ (V → V). Three operations are forbidden: (1) feeding raw P or V components directly into MLPs; (2) computing update weights from quantities that are not S-typed; and (3) constructing $\mathbf{A}$ from P or V inputs—only S-typed inputs are admissible.

Feature Initialization and Dual-Branch Decoupling. $\mathcal{F}_{\text{init}}$ maps $\mathbf{X}$ to $\mathbf{r}_i \in \text{P}$, $\mathbf{v}_i \in \text{V}$ and $\mathbf{h}_i \in \text{S}$. The reference point $\mathbf{r}_i = \mathbf{p}_{i,T_{\text{obs}}}$ is the last observed position. With raw velocities $\mathbf{u}_{i,t} = \mathbf{p}_{i,t} - \mathbf{p}_{i,t-1}$, the C vector channels are $\mathbf{v}_i = W_v[\mathbf{p}_{i,1:T_{\text{obs}}} - \bar{\mathbf{p}};\ \mathbf{u}_{i,1:T_{\text{obs}}}] \in \mathbb{R}^{C\times D}$, where W_v is a learned linear map along the time axis; subtracting $\bar{\mathbf{p}}$, a mean of observed positions and hence a Point, makes the inputs Vector-typed. An optional temporal DCT may precede W_v (Appendix A.4). The invariant scalars are computed from speeds and turning angles, $\mathbf{h}_i = \text{MLP}(\{\|\mathbf{u}_{i,t}\|, \angle(\mathbf{u}_{i,t}, \mathbf{u}_{i,t-1})\}_t)$.

Variable-size batches. All statistics and assignments are computed over valid agents only, since unmasked padding breaks translation equivariance (Appendix A.5). The resulting pipeline has a geometric projection branch and an invariant topology branch (Fig. 2).

Hyperedge Topology Inference via Invariant Group Partition. The soft hyperedge assignments $\mathbf{A} \in [0,1]^{B\times N\times K}$ are computed from Euclidean-invariant S-typed features only:

$$\mathbf{A}_{i,k} = \text{softmax}_k\Big(\text{MLP}_{\text{topo}}\big(\mathbf{h}_i;\ \textstyle\sum_{j\neq i} \text{MLP}_{\text{pair}}(\mathbf{h}_i, \mathbf{h}_j, \|\mathbf{r}_i - \mathbf{r}_j\|, \|\mathbf{u}_i - \mathbf{u}_j\|)\big)\Big). \tag{3}$$

Since all inputs are invariant, $\mathbf{A}(g \cdot \mathbf{X}) = \mathbf{A}(\mathbf{X})$: the group partition does not depend on the coordinate frame. Because each row of $\mathbf{A}$ sums to one, $\mathbf{A}$ is a soft partition rather than a set of freely overlapping hyperedges. This competitive normalization is our main anti-collapse mechanism: replacing the softmax with per-edge sigmoids preserves trajectory accuracy but sharply lowers grouping F1 (Sec. 5.5). Topology inference has an $O(N^2)$ scalar cost, whereas the vector-valued message passing below scales as $O(NK)$; Sec. 5.4 gives the measured cost decomposition.

Bidirectional Equivariant Hypergraph Message Passing. Each layer $\mathcal{F}_{\text{hypermp}}$ performs vertex-to-hyperedge aggregation followed by hyperedge-to-vertex updates. Define column-normalized weights $\widehat{\mathbf{A}}_{i,k} = \mathbf{A}_{i,k} / \sum_{i'} \mathbf{A}_{i',k}$, so that $\sum_i \widehat{\mathbf{A}}_{i,k} = 1$. *Vertex-to-Hyperedge Aggregation.* This step computes hyperedge centroids $\mathbf{c}_k = \sum_i \widehat{\mathbf{A}}_{i,k} \mathbf{r}_i$ (P, an affine combination), pooled scalars $m_k = \sum_i \widehat{\mathbf{A}}_{i,k} \text{MLP}(\mathbf{h}_i)$ (S), and pooled vectors $\bar{\mathbf{v}}_k = \sum_i \widehat{\mathbf{A}}_{i,k} \mathbf{v}_i$ (V). The displacement from agent i to centroid k is $\mathbf{\Delta}_{i,k} = \mathbf{c}_k - \mathbf{r}_i$ (V), and the scalar attention coefficients are

$$p_{i,k} = \text{MLP}_{\text{att}}\big([\mathbf{h}_i;\, m_k;\, \|\mathbf{r}_i - \mathbf{c}_k\|;\, \|\mathbf{v}_i - \bar{\mathbf{v}}_k\|_F]\big) \in \text{S}.$$

Hyperedge-to-Vertex Updates.

$$\Delta\mathbf{r}_i = \textstyle\sum_k \mathbf{A}_{i,k}\, p_{i,k}\, \mathbf{\Delta}_{i,k}\ (\text{V}), \qquad \mathbf{r}_i \leftarrow \mathbf{r}_i + \Delta\mathbf{r}_i\ (\text{P} + \text{V} \Rightarrow \text{P}), \tag{4}$$

$$\mathbf{h}_i \leftarrow \mathbf{h}_i + \text{MLP}\Big(\mathbf{h}_i; \textstyle\sum_k \mathbf{A}_{i,k} m_k\Big)\ (\text{S}). \tag{5}$$

Vector-feature update. The multi-channel vectors $\mathbf{v}_i$ are updated by four type-preserving operations:

1. Channel-wise scalar attention: $\mathbf{v}_i \leftarrow \big(1 + \alpha_i\big) \odot \mathbf{v}_i$, with $\alpha_i = \text{MLP}(\mathbf{h}_i) \in \text{S}$ broadcast over the spatial axis;
2. Hyperedge-relative aggregation: $\mathbf{v}_i \leftarrow \mathbf{v}_i + \sum_k \mathbf{A}_{i,k}\, \varphi_{i,k} \odot \big(\bar{\mathbf{v}}_k - \mathbf{v}_i\big)$, with $\varphi_{i,k} \in \text{S}$ from the same attention MLP as $p_{i,k}$;
3. Velocity residual: $\mathbf{v}_i \leftarrow \mathbf{v}_i + W_u \mathbf{u}_i$, a channel mixing, which stays in V;
4. Norm-gated nonlinearity: $\mathbf{v}_i \leftarrow \psi(\mathbf{v}_i)$.

Each operation is an S-weighted combination or a channel-wise linear map of Vectors, so the layer is Euclidean-equivariant.

Training Objective. The loss combines trajectory reconstruction, two entropy terms on the assignment, and optional co-group supervision:

$$\begin{aligned}\mathcal{L}_{\text{total}} = \mathbb{E}\big[\|\widehat{\mathbf{Y}} - \mathbf{Y}\|_F^2\big] + \lambda_1 \Big(\log K + \sum_{k=1}^{K} \bar{\mathbf{A}}_k \log \bar{\mathbf{A}}_k\Big) \\ + \lambda_2 \Big(-\frac{1}{N} \sum_i \sum_k \mathbf{A}_{i,k} \log \mathbf{A}_{i,k}\Big) + \lambda_3 \|\mathbf{M}_{\text{pred}} - \mathbf{M}_{\text{gt}}\|_F^2,\end{aligned} \tag{6}$$

where $\bar{\mathbf{A}}_k = \frac{1}{N} \sum_i \mathbf{A}_{i,k}$ is the **per-sample** mean occupancy of hyperedge k, and $\lambda_3 = 0$ without group annotations. The λ_1 term maximizes occupancy entropy, as in load balancing for mixture-of-experts routing (Shazeer et al., 2017; Fedus et al., 2022); the λ_2 term sharpens each agent's assignment, as in the entropy regularizer of DiffPool (Ying et al., 2018). Occupancy must be computed per sample: a batch-averaged occupancy can look balanced while every individual sample collapses onto a single hyperedge. In our ablations these terms are not what prevents collapse—competitive normalization is (Sec. 5.5)—and a large λ_1 is harmful: at $\lambda_1 \geq 0.3$ it forces all K hyperedges to be occupied and lowers grouping F1 (Table 2, Appendix Table 15). We recommend $\lambda_1 = 0.03$, $\lambda_2 = 0.002$; the values used in each reported experiment are listed in Appendix Table 12.

Equivariant Prediction Head. The decoder $\mathcal{F}_{\text{out}}$ predicts positions for $\tau \in \{1, \ldots, T_f\}$. Invariant mixture coefficients $a_i^{\tau,c} = \text{softmax}_c\big(\text{MLP}_{\text{mix}}(\mathbf{h}_i^{(L)})\big)$ combine the C vector channels into equivariant velocity increments $\boldsymbol{\delta}_i^{\tau} = \sum_c a_i^{\tau,c} \mathbf{v}_{i,c}^{(L)}$, which are integrated from the refined reference point:

$$\widehat{\mathbf{y}}_i^{\tau} = \mathbf{r}_i^{(L)} + \sum_{\tau'=1}^{\tau} \boldsymbol{\delta}_i^{\tau'} \quad (\text{P}). \tag{7}$$

This P+V composition guarantees full Euclidean equivariance $\mathcal{F}(g \cdot \mathbf{X}) = g \cdot \mathcal{F}(\mathbf{X})$. Multi-modal heads and optional DCT-based decoding are described in Appendix A.4.

Theoretical Geometric Guarantees. Full-network equivariance. In both static and dynamic modes, $\mathcal{F}(g \cdot \mathbf{X}) = g \cdot \mathcal{F}(\mathbf{X})$ and $\mathbf{A}(g \cdot \mathbf{X}) = \mathbf{A}(\mathbf{X})$ for every $g \in O(D) \ltimes \mathbb{R}^D$. *Proof sketch.* By induction over layers: initialization produces valid P/V/S primitives; topology inference consumes only S inputs and outputs an invariant $\mathbf{A}$; aggregation yields equivariant centroids and invariant scalars; every update is type-preserving (Appendix A.3). We verify the guarantee on randomly initialized, untrained networks in float64: residuals are $\leq 3.6 \times 10^{-15}$ for equivariance and $\leq$

Table 1: Synthetic Group-Hard results at $N = 12/48$, mean±std. The $N = 12$ results use five seeds (EGNN: three); all $N = 48$ results use three. "n/a" denotes no group assignment. EqDynHyper rows use dynamic topology and $\lambda_1 = 0.3$ ("Full ($\lambda_1 = .3$)" in Table 2); K equals the number of ground-truth groups. Static topology reaches the same ADE (0.0071 ± 0.0008; "w/o Dyn." in Table 2). EGNN is adapted to autoregressive rollout (Appendix B.1).

		Small-scale Evaluation ($N = 12$)			**Large-scale Evaluation** ($N = 48$)	
		Trajectory		Group	Trajectory	
Model	Superv.	ADE (↓)	FDE (↓)	F1 (↑)	ADE (↓)	FDE (↓)
EGNN (2021)	Unsup.	0.4904±0.0144	0.8081±0.0032	n/a	0.7631±0.0624	1.0147±0.0312
Non-equivariant hypergraph	Unsup.	0.0532±0.0025	0.1299±0.0064	0.789±0.146	0.0998±0.0050	0.2367±0.0097
EqMotion (2023)	Unsup.	0.0212±0.0046	0.0193±0.0026	n/a	0.0525±0.0111	0.0689±0.0155
EqDynHyper	**Unsup.**	**0.0072±0.0005**	0.0180±0.0009	0.713±0.023	**0.0124±0.0004**	**0.0331±0.0013**
EqDynHyper	**Weak**	0.0057±0.0006	**0.0122±0.0013**	**0.988±0.003**	0.0136±0.0007	0.0355±0.0018

Table 2: Architectural ablation on Group-Hard ($N = 12$). "Full" uses dynamic topology; "w/o Dyn." is the static-topology variant ($\lambda_1 = 0.3$). Mean±std over five seeds unless marked †/‡.

Variant	ADE (↓)	FDE (↓)	F1 (↑)
Full ($\lambda_1 = .3$)	0.0072±0.0005	0.0180±0.0009	0.713±0.023
Full ($\lambda_1 = .03$)	0.0062±0.0009	0.0153±0.0024	0.855±0.126
w/o Dyn.	0.0071±0.0008	0.0178±0.0022	0.667±0.019
w/o Load Reg. ($\lambda_1 = 0$)	0.0066±0.0008	0.0160±0.0020	0.859±0.047
Sigmoid†	0.0075	0.0187	0.485
$K = 1$†	0.0075	0.0195	n/a
Random‡	0.0074	0.0192	0.299±0.001
Non-equiv.	0.0532±0.0025	0.1299±0.0064	0.789±0.146

† One seed. ‡ Two seeds for ADE/FDE; F1 is the analytical random level over five samples.

Table 3: Group switch at frame 10, inside the observation window ($\lambda_1 = 0.3$; five seeds unless †). Bold: best per column.

Model	ADE (↓)	FDE (↓)	F1 (↑)
Dynamic	0.0392±.0029	0.1057±.0065	**0.698**±.014
Static	0.0400±.0028	0.1093±.0083	0.662±.015
Fixed k-means†	**0.0362**	**0.0998**	0.486
EqMotion	0.0581±.0061	0.1431±.0081	n/a

† One seed.

5.6×10^{-16} for assignment invariance across random transforms and padded batches (Appendix Table 11).

Design Rationale. Competitive (softmax) normalization is the anti-collapse mechanism. Per-layer re-inference of $\mathbf{A}$ lets deeper layers revise the grouping but costs 3.3× inference time at $N = 400$, so static topology is the default. A dedicated Point type is required: hyperedge centroids are affine combinations of positions, and expressing them with Vector and Scalar types only would require an arbitrary origin and lose translation equivariance of the aggregation anchor.

5 Experiments

Evaluation Metrics. We evaluate three aspects: trajectory accuracy (ADE and FDE), grouping quality (grouping F1; Xu et al., 2022), and computational efficiency (inference latency and peak GPU memory). On ETH–UCY we report best-of-20 ADE/FDE. Unless a caption states otherwise, results are mean±std over random seeds, with seed counts given in each caption; MD17 results and the sensitivity sweeps in the appendix are single runs.

5.1 Scenario 1: Synthetic Group Dynamics Prediction

Dataset Setup. Following Xu et al. (2022), the synthetic Group-Hard dataset simulates rigid-body rotation, translation, and group splitting and merging for small ($N = 12$) and large ($N = 48$) agent populations.

Performance Analysis. Table 1 shows that unsupervised EqDynHyper lowers ADE relative to EqMotion by 2.9× at $N = 12$ (0.0072 vs. 0.0212) and 4.2× at $N = 48$ (0.0124 vs. 0.0525). FDE is comparable at $N = 12$ (0.0180 vs. 0.0193) and 2.1× lower at $N = 48$; EqMotion's FDE below its ADE reflects per-step error that oscillates with rotation phase (Appendix Fig. 14). Weak supervision raises grouping F1 to 0.988 at $N = 12$ but slightly increases ADE at $N = 48$ (0.0136 vs. 0.0124). Throughout, K is set to the number of ground-truth groups (Appendix Table 12), so group discovery is unsupervised with respect to memberships only. The non-equivariant hypergraph reaches a higher but highly variable F1 on the test frames (0.789 ± 0.146); unlike ours, its assignments carry no guarantee under rotated or reflected scenes, which we did not evaluate separately. Ablations are discussed in Sec. 5.5; Figures 3 and 4 summarize errors and their dependence on coupling strength.

Table 4: ETH–UCY best-of-20 ADE/FDE with all models retrained under our unified protocol (up to 8 agents per sample, validation-based checkpoint selection; Sec. 5.2). Mean±std over three seeds for EqMotion and EqDynHyper; GroupNet: single run. Bold: best mean per column. FDE differences on eth, hotel and zara2 lie within one standard deviation. Published numbers under the original protocol are not comparable and are listed in Appendix Table 13.

Model	Metric	ETH	Hotel	Univ	Zara1	Zara2	Average
GroupNet (2022)	ADE	0.921	0.910	0.595	0.261	0.255	0.588
	FDE	1.765	1.665	1.096	0.476	0.466	1.093
EqMotion (2023)	ADE	0.453±.005	0.171±.006	0.257±.004	0.213±.002	0.156±.002	0.250
	FDE	0.704±.005	0.239±.006	0.469±.009	0.380±.005	0.264±.001	0.411
EqDynHyper	ADE	**0.439**±.009	**0.163**±.013	**0.241**±.004	**0.200**±.006	**0.146**±.004	**0.238**
	FDE	**0.700**±.011	**0.238**±.014	**0.451**±.010	**0.369**±.010	**0.262**±.008	**0.404**

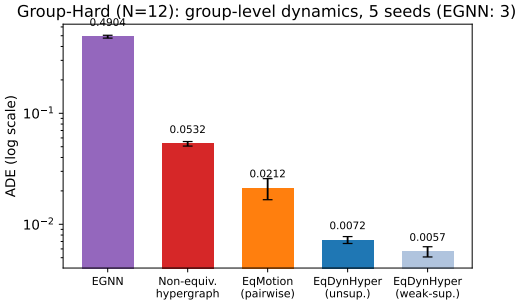

Figure 3: Group-Hard ($N = 12$) ADE on a log scale, mean±std over five seeds (EGNN: three).

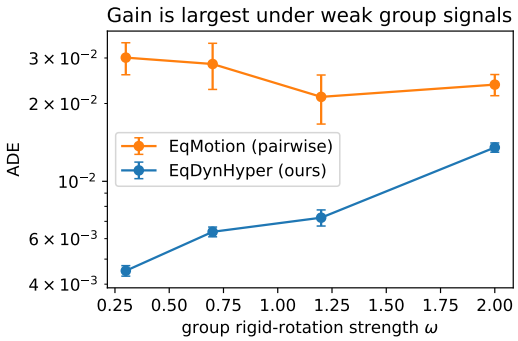

Figure 4: ADE versus group-rotation coupling ω (three seeds). The gap to EqMotion is largest under weak coupling; EqDynHyper's own error grows with ω.

5.2 Scenario 2: Pedestrian Trajectory Prediction

Dataset Setup. ETH–UCY (Alahi et al., 2016) contains five scenes; models observe 8 time steps and predict 12, with leave-one-scene-out cross-validation. **Protocol.** The commonly used pre-processing keeps only 1–3 agents per sample, which removes most interactions; we re-process the data keeping up to 8 agents per sample. EqMotion, GroupNet and EqDynHyper are all retrained under this protocol and evaluated best-of-20 on the ego agent, with checkpoints selected on a 10% validation split of the training scenes; the test scene is never used for selection. The resulting numbers are not comparable to published results, which we list separately in Appendix Table 13.

Performance Analysis. As shown in Table 4, EqDynHyper lowers mean ADE from 0.250 to 0.238 (5.0% relative) with FDE on par (0.404 vs. 0.411). The modest average gain is consistent with many samples in which pedestrians move independently. Because samples contain at most 8 agents, ETH–UCY does not test scalability; here it shows that the model transfers to real data and supports qualitative group discovery (Appendix Figures 16, 17, and 18). Retrained with its official hyper-parameters, GroupNet performs markedly worse under this protocol than in its original publication (0.588 vs. 0.25 average ADE). Unlike for EqMotion, whose published numbers we reproduce under the original protocol (0.222/0.365 average), we did not verify our GroupNet setup under that protocol, so part of this gap may stem from our configuration; we therefore treat EqMotion as the primary baseline on ETH–UCY.

5.3 Scenario 3: Molecular Dynamics Prediction

Dataset Setup. MD17 (Chmiela et al., 2017) contains molecular-dynamics trajectories governed mainly by local pairwise covalent bonds (Sanchez-Gonzalez et al., 2019). We use it as a diagnostic benchmark for settings with weak higher-order signals.

Performance Analysis. As shown in Table 5, EqDynHyper is on par with our reproduction of EqMotion on ethanol (5.22 vs. 5.30 ADE, $\times 10^{-2}$) but 24–35% worse on aspirin, benzene and malonaldehyde (32–36% relative to the published EqMotion numbers). MD17 thus marks a regime where local pairwise bonds dominate and hyperedge aggregation is a disadvantage; hybrid pairwise–

Table 5: MD17 trajectory prediction error (ADE/FDE, $\times 10^{-2}$; lower is better). Rows above the rule are as reported by Xu et al. (2023); the last two rows are trained with our pipeline (single run). Bold: best per column.

Model	Aspirin	Benzene	Ethanol	Malonaldehyde
Radial Field	17.98/26.20	7.73/12.47	8.10/10.61	16.53/25.10
TFN (2018)	15.02/21.35	7.55/12.30	8.05/10.57	15.21/24.32
SE(3)-Trans (2020)	15.70/22.39	7.62/12.50	8.05/10.86	15.44/24.47
EGNN (2021)	14.61/20.65	7.50/12.16	8.01/10.22	15.21/24.00
LSTM	17.59/24.79	6.06/9.46	7.73/9.88	15.14/22.90
S-LSTM (2016)	13.12/18.14	3.06/3.52	7.23/9.85	11.93/18.43
NRI (2018)	12.60/18.50	1.89/2.58	6.69/8.78	12.79/19.86
NMMP (2020)	10.41/14.67	2.21/3.33	6.17/7.86	9.50/14.89
GroupNet (2022)	10.62/14.00	2.02/2.95	6.00/7.88	7.99/12.49
EqMotion (2023)	**5.95/8.38**	**1.18/1.73**	**5.05/7.02**	**5.85/9.02**
EqMotion (our reproduction)	6.35/8.85	1.24/1.76	5.30/7.30	5.86/9.09
EqDynHyper	7.86/10.26	1.59/2.45	5.22/7.22	7.94/12.50

Table 6: Inference cost versus agent count N: forward-pass latency and peak GPU memory. EqDynHyper runs in static-topology mode with $K = 8$.

N	Model	Forward Latency (↓)	Peak GPU VRAM (↓)	Speedup (↑)
50	EqMotion (2023)	39.9 ms	501 MB	–
	EqDynHyper	**11.3 ms**	**143 MB**	**3.5×**
100	EqMotion (2023)	176 ms	1.9 GB	–
	EqDynHyper	**22.9 ms**	**518 MB**	**7.7×**
200	EqMotion (2023)	683 ms	7.6 GB	–
	EqDynHyper	**76.8 ms**	**2.0 GB**	**8.9×**
400	EqMotion (2023)	2697 ms	30 GB	–
	EqDynHyper	**287 ms**	**8 GB**	**9.4×**

hypergraph blocks are a natural remedy. Figure 6 shows that group recovery varies across molecules, and Appendix Figure 15 reports a low-data analysis.

5.4 Scalability and Computational Efficiency

Pairwise equivariant models pay an $O(N^2)$ cost for their vector-valued messages. EqDynHyper moves vector-valued computation onto K hyperedges ($O(NK)$), while its scalar topology term still compares all agent pairs ($O(N^2)$). Table 6 is measured in static-topology mode with $K = 8$ for all N; both models have ∼470k parameters. At $N = 400$ EqDynHyper is $9.4\times$ faster than EqMotion and uses $3.75\times$ less memory (dynamic mode is $3.3\times$ slower, which leaves it roughly $2.8\times$ faster than EqMotion). Because both models remain quadratic, the speedup saturates at a constant factor ($3.5\times$, $7.7\times$, $8.9\times$, $9.4\times$ for $N = 50$–400). At large N the scalar topology term dominates our runtime (Appendix Fig. 12); restricting it to sparse neighborhoods is a natural next step. A k-NN-sparsified EqMotion ($k = 8$) does not reduce latency in our implementation (Appendix Fig. 12), so sparsifying EqMotion's graph does not by itself reproduce the speedup; an implementation with dedicated sparse kernels may behave differently. We evaluate accuracy only up to $N = 48$ (Appendix Fig. 11): the ADE advantage over EqMotion grows from $2.9\times$ at $N = 12$ to $4.2\times$ at $N = 48$.

5.5 Ablation Studies and Structural Interpretability

Component Effectiveness. Removing equivariance raises ADE from 0.0072 to 0.0532 ($7.4\times$; Table 2) and is the only ablation that changes ADE substantially. Replacing softmax membership with per-edge sigmoids leaves ADE unchanged but lowers grouping F1 from 0.713 to 0.485 (one seed), indicating that competitive normalization is the anti-collapse mechanism. The $K = 1$ and random-assignment controls reach ADE close to the full model despite absent or random-level grouping, so forecasting accuracy does not depend on grouping quality ($K = 1$ and random: one and two seeds, respectively). The load-balancing term does not improve grouping: at $\lambda_1 = 0.3$ it lowers mean F1 from 0.859 ($\lambda_1 = 0$) to 0.713, and at $\lambda_1 = 0.03$ mean F1 is unchanged (0.855) with a larger spread across seeds (± 0.126). Under a group switch (Table 3), dynamic topology improves grouping F1 (0.698 vs. 0.662) and lowers FDE in 4 of 5 paired seeds (Appendix Fig. 9). However, the

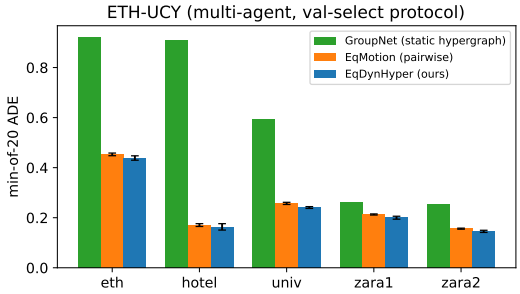

Figure 5: ETH–UCY best-of-20 ADE per scene under the unified protocol (error bars: std over three seeds).

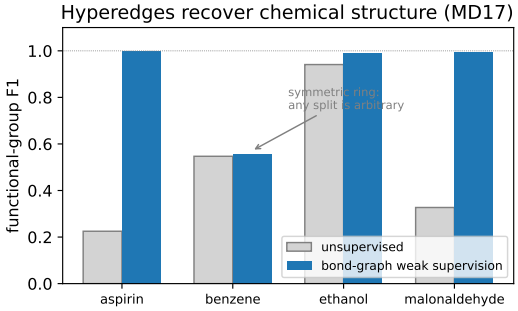

Figure 6: MD17 functional-group pairwise F1, unsupervised versus bond-graph weak supervision (single run).

Table 7: ADE across group-rotation coupling strengths ω on Group-Hard ($N = 12$), mean±std over three seeds; the $\omega = 1.2$ column is the Table 1 setting and reports its five-seed values.

Model	.3	.7	1.2	2.0
EqMotion	0.0301±.0043	0.0284±.0058	0.0212±.0046	0.0237±.0022
EqDynHyper	**0.0045**±.0002	**0.0064**±.0003	**0.0072**±.0005	**0.0135**±.0006
Gain	6.7×	4.4×	2.9×	1.8×

Table 8: ETH–UCY spatial statistics of learned groups. $r = d_{\text{in}}/d_{\text{out}}$; $r < 1$ indicates spatially coherent groups; r_{null}: random grouping. Since topology inference receives pairwise distances, $r < 1$ is partly expected from proximity alone; it shows spatial coherence, not that the groups are social.

Scene	d_{in}	d_{out}	r	r_{null}
eth	3.09	5.91	0.52	1.29
hotel	2.92	5.49	0.53	1.15
univ	3.20	4.32	0.74	1.00
zara1	3.04	5.07	0.60	0.98
zara2	2.34	4.89	0.48	0.98

mean ADE and FDE differences are within one standard deviation, and a fixed k-means topology attains the lowest ADE (single run) at much lower F1. Table 7 summarizes the gains across coupling strengths.

Latent Group Interpretability. Unsupervised EqDynHyper recovers ethanol's functional groups (pairwise F1 0.941), but recovery without bond-graph supervision is weak for aspirin (0.225) and malonaldehyde (0.327; Appendix Table 16). On ETH–UCY (Table 8), learned groups have smaller intra- than inter-group distances, consistent with spatially coherent groups. Learned assignments are invariant under the full Euclidean group, including reflections.

Limitations and Negative Results. (1) On bond-dominated MD17 molecules EqDynHyper is 24–35% less accurate than EqMotion on three of four molecules and shows no data-efficiency advantage. (2) The scalar topology term keeps the overall complexity at $O(N^2)$; our speedup is a constant factor. (3) K must be set by hand, and in our experiments equals the known group count; grouping F1 drops from 0.72 to 0.43 at $K = G + 1$ (Appendix Table 14). (4) Dynamic topology adds 3.3× inference cost for a modest grouping gain and no significant trajectory gain. (5) Rotation robustness matches EqMotion, since both are exactly equivariant; equivariance itself is not an advantage over equivariant baselines. (6) All models degrade when groups reconfigure exactly at the observation–prediction boundary. (7) We did not evaluate a variant without inter-agent aggregation, so part of the gain over EqMotion may come from the rest of the architecture (temporal channel lifting, velocity residuals) rather than from centroid aggregation. Automatic K selection and sparse topology inference are left for future work.

6 Conclusion

We presented EqDynHyper, an equivariant hypergraph network for multi-agent forecasting and group discovery. By moving vector-valued interaction computation from $O(N^2)$ edges onto $O(NK)$ hyperedges, it improves accuracy on synthetic collective dynamics and yields constant-factor speed and memory gains at large N, while a scalar $O(N^2)$ topology term remains. Group assignments are Euclidean-invariant by construction and are recovered without membership supervision given the number of groups. Limitations remain for pairwise-bond-dominated systems and for choosing K; future work will combine pairwise and hypergraph modules and sparsify topology inference.

AI USAGE STATEMENT. Generative AI tools assisted with code prototyping, unit-test scripting, bibliography formatting and manuscript editing. All technical claims, experimental results and references were checked by the authors; no AI tool was used to generate research ideas or experimental results.

ETHICS STATEMENT. All experiments use public benchmarks, and no new human-subject data were collected.

REPRODUCIBILITY STATEMENT. Source-code, scripts and checkpoints will be released upon acceptance. Hyperparameters (Appendix B.1), unit tests (Appendix Table 11) and additional experiments are in the appendix.

REFERENCES

Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 961–971. IEEE, 2016.

Inhwan Bae, Jin-Hwi Park, and Hae-Gon Jeon. Learning pedestrian group representations for multi-modal trajectory prediction. In *European Conference on Computer Vision*, pp. 270–289. Springer, 2022.

Ilyes Batatia, Péter Kovács, Gregor N. C. Simm, Christoph Ortner, and Gábor Csányi. MACE: Higher order equivariant message passing neural networks for fast and accurate force fields. In *Advances in Neural Information Processing Systems*, volume 35, pp. 11423–11436, 2022.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261, 2018.

Filippo Maria Bianchi, Daniele Grattarola, and Cesare Alippi. Spectral clustering with graph neural networks for graph pooling. In *International Conference on Machine Learning*, pp. 874–883. PMLR, 2020.

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.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pp. 2990–2999. PMLR, 2016.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. SE(3)-transformers: 3D roto-translation equivariant attention networks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1970–1981, 2020.

Tianpei Gu, Guangyi Chen, Junlong Li, Chunze Lin, Yongming Rao, Jie Zhou, and Jiwen Lu. Stochastic trajectory prediction via motion indeterminacy diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17113–17122. IEEE, 2022.

Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2255–2264. IEEE, 2018.

Yue Hu, Siheng Chen, Ya Zhang, and Xiao Gu. Collaborative motion prediction via neural motion message passing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6319–6328. IEEE, 2020.

Boris Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2375–2384, 2019.

Bowen Jing, Stephan Eismann, Pratik Suriana, Raphael J. L. Townshend, and Ron O. Dror. Geometric vector perceptrons for graph neural networks. In *International Conference on Learning Representations*, 2021.

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.

Jiachen Li, Fan Yang, Masayoshi Tomizuka, and Chiho Choi. EvolveGraph: Multi-agent trajectory prediction with dynamic relational reasoning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 19783–19794, 2020.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, volume 33, pp. 11525–11538, 2020.

Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In *European Conference on Computer Vision*, pp. 759–776. Springer, 2020.

Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *European Conference on Computer Vision*, pp. 683–700. Springer, 2020.

Alvaro Sanchez-Gonzalez, Victor Bapst, Kyle Cranmer, and Peter Battaglia. Hamiltonian graph networks with ODE integrators. arXiv preprint arXiv:1909.12790, 2019.

Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. In *international conference on machine learning*, pp. 9323–9332. PMLR, 2021.

Kristof T Schütt, Oliver T 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.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

Nathan 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, 2018.

Chenxin Xu, Maosen Li, Zhenyang Ni, Ya Zhang, and Siheng Chen. Groupnet: Multiscale hypergraph neural networks for trajectory prediction with relational reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6498–6507. IEEE, 2022.

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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1410–1420. IEEE, 2023.

Chenxin Xu, Yuxi Wei, Bohan Tang, Sheng Yin, Ya Zhang, Siheng Chen, and Yanfeng Wang. Dynamic-group-aware networks for multi-agent trajectory prediction with relational reasoning. *Neural Networks*, 170:564–577, 2024.

Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9813–9823, 2021.

Rui Zhou, Hongyu Zhou, Huidong Gao, Masayoshi Tomizuka, Jiachen Li, and Zhuo Xu. Grouptron: Dynamic multi-scale graph convolutional networks for group-aware dense crowd trajectory forecasting. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

A METHOD DETAILS

This appendix collects the notation, the P/V/S operator table, the full equivariance proof and the implementation details summarized in Sec. 4.

A.1 NOTATION

Table 9 summarizes the notation used throughout the paper.

Table 9: Notation.

Symbol	Meaning	Type
$g = (\mathbf{R}, \mathbf{b})$	Euclidean transform, $\mathbf{R} \in O(D)$, $\mathbf{b} \in \mathbb{R}^D$	–
$\mathbf{p}_{i,t}$, $\mathbf{u}_{i,t}$	observed position; raw velocity $\mathbf{p}_{i,t} - \mathbf{p}_{i,t-1}$	P; V
$\mathbf{r}_i$	reference point, refined layer by layer	P
$\mathbf{v}_i \in \mathbb{R}^{C \times D}$	C-channel vector features	V
$\mathbf{h}_i$	invariant scalar features	S
$\mathbf{A} \in [0,1]^{N \times K}$, $\widehat{\mathbf{A}}$	soft assignment (rows sum to 1); column-normalized version	S
$\mathbf{c}_k$, $\bar{\mathbf{v}}_k$, m_k	hyperedge centroid; pooled vector; pooled scalar	P; V; S
$\mathbf{\Delta}_{i,k} = \mathbf{c}_k - \mathbf{r}_i$	agent-to-centroid displacement	V
$p_{i,k}$, $\varphi_{i,k}$, α_i	attention and gating coefficients	S
K, G, K_{eff}	hyperedge count; ground-truth group count; $\exp(H(\bar{\mathbf{A}}))$	–
$\lambda_1, \lambda_2, \lambda_3$	occupancy-entropy, assignment-entropy, co-group weights	–

A.2 P/V/S OPERATOR TABLE

Every layer of EqDynHyper is composed of the type-closed operations in Table 10; composing them preserves the P/V/S signatures, which is what the proof in Appendix A.3 relies on.

Table 10: Type-closed geometric operations. Any operation not listed here is forbidden.

Operation description	Computation expression	Result type
P minus P (position difference)	$x_i - x_j$	V
V inner-product / norm / angle	$\langle v, w \rangle, \lvert v \rvert, \angle(v, w)$	S
MLP / sum / softmax / sigmoid (all-S inputs)	—	S
Scalar-weighted vector summation	$\sum a_k v_k$, $a_k \in$ S	V
Point plus Vector	$p + v$	P
Scalar-weighted Point average	$\sum a_k p_k$, $a_k \in$ S	P
Point minus scalar-weighted Point mean	$p - \bar{p}$	V
Norm-gated vector nonlinearity	$v \mapsto \sigma(\lvert v \rvert) \cdot v$	V

Forbidden operations:

1. Feeding raw P or V coordinate components directly into an MLP.
2. Computing update weights for P or V tensors from quantities that are not S-typed.
3. Constructing the assignment matrix $\mathbf{A}$ from P or V inputs.

A.3 FULL INDUCTIVE EQUIVARIANCE PROOF

Theorem 1. *Let $g = (\mathbf{R}, \mathbf{b})$ denote an arbitrary Euclidean transform in $O(D) \ltimes \mathbb{R}^D$ acting as $g \cdot \mathbf{X} = \mathbf{R}\mathbf{X} + \mathbf{b}$. In both static and dynamic modes, every layer ℓ of EqDynHyper satisfies:*

1. $\mathbf{r}^{(\ell)}(g \cdot \mathbf{X}) = \mathbf{R}\mathbf{r}^{(\ell)}(\mathbf{X}) + \mathbf{b}$ *(P-type equivariance)*
2. $\mathbf{v}^{(\ell)}(g \cdot \mathbf{X}) = \mathbf{R}\mathbf{v}^{(\ell)}(\mathbf{X})$ *(V-type equivariance, channel-wise)*
3. $\mathbf{h}^{(\ell)}(g \cdot \mathbf{X}) = \mathbf{h}^{(\ell)}(\mathbf{X})$ *(S-type invariance)*
4. $\mathbf{A}^{(\ell)}(g \cdot \mathbf{X}) = \mathbf{A}^{(\ell)}(\mathbf{X})$ *(assignment invariance; $\mathbf{A}^{(\ell)} = \mathbf{A}$ in static mode)*

Consequently $\mathcal{F}(g \cdot \mathbf{X}) = g \cdot \mathcal{F}(\mathbf{X})$.

Proof. **Base case ($\ell = 0$).** $\mathbf{r}^{(0)}$ is the last observed position (P). $\mathbf{v}^{(0)}$ is obtained by a linear map along the time axis applied to $\mathbf{p}_{i,t} - \bar{\mathbf{p}}$ (P−P⇒V) and to $\mathbf{u}_{i,t}$ (V). A map that acts only on time and

channel indices commutes with $\mathbf{R}$, so $\mathbf{v}^{(0)}$ is V; the optional DCT is such a map. $\mathbf{h}^{(0)}$ is computed from norms and angles (S).

Step 1: Topology inference. The inputs of Eq. equation 3 are $\mathbf{h}_i, \mathbf{h}_j$ and the distances $\|\mathbf{r}_i - \mathbf{r}_j\|$, $\|\mathbf{u}_i - \mathbf{u}_j\|$, all S. Hence $\mathbf{A}$ (static) or $\mathbf{A}^{(\ell)}$ (dynamic) is invariant, and so is $\widehat{\mathbf{A}}$.

Step 2: Vertex-to-hyperedge aggregation. Since $\sum_i \widehat{\mathbf{A}}_{ik} = 1$,

$$\mathbf{c}_k(g \cdot \mathbf{X}) = \sum_i \widehat{\mathbf{A}}_{ik}\big(\mathbf{R}\mathbf{r}_i + \mathbf{b}\big) = \mathbf{R}\mathbf{c}_k(\mathbf{X}) + \mathbf{b},$$

so $\mathbf{c}_k$ is P. The pooled vector $\bar{\mathbf{v}}_k$ is an S-weighted sum of V and is V; m_k is computed from S only and is S.

Step 3: Hyperedge-to-vertex update. $\mathbf{\Delta}_{ik} = \mathbf{c}_k - \mathbf{r}_i$ maps to $(\mathbf{R}\mathbf{c}_k + \mathbf{b}) - (\mathbf{R}\mathbf{r}_i + \mathbf{b}) = \mathbf{R}\mathbf{\Delta}_{ik}$, so it is V. The inputs of p_{ik} and φ_{ik} are S or norms of V, so both are S. $\Delta\mathbf{r}_i$ is an S-weighted sum of V, and $\mathbf{r}_i + \Delta\mathbf{r}_i$ is P. The update of $\mathbf{h}_i$ uses S only. The four vector operations are, respectively, an S-scaling of V, an S-weighted sum of V differences, a channel mixing of V, and the norm-gated map ψ; each returns V.

Step 4: Prediction head. $a_i^{\tau,c}$ is S, so $\boldsymbol{\delta}_i^{\tau} = \sum_c a_i^{\tau,c}\mathbf{v}_{i,c}^{(L)}$ is V (an inverse DCT along time, if used, preserves V). $\widehat{\mathbf{y}}_i^{\tau} = \mathbf{r}_i^{(L)} + \sum_{\tau' \le \tau} \boldsymbol{\delta}_i^{\tau'}$ is P+V⇒P.

By induction over $\ell = 0, \ldots, L$ the four properties hold at every layer, and the output is equivariant. □

A.4 Multi-Modal Heads and Temporal DCT

Temporal DCT. The observed sequences may be projected onto a truncated DCT basis before the time-to-channel map W_v, and the prediction head may output per-channel DCT coefficients that a fixed inverse DCT maps back to per-step increments. Both transforms are linear along the time axis and act identically on every spatial coordinate, so they preserve the V type. We found the inverse DCT to give smoother long-horizon motion on MD17.

Multi-modal heads. For best-of-20 evaluation on ETH–UCY we attach parallel prediction heads, each consisting of an additional message-passing layer and its own mixture decoder, trained with a min-over-heads (variety) loss.

A.5 Batched Padding and Equivariance Tests

With variable-size batches, padded agents must be masked before computing the normalized weights $\widehat{\mathbf{A}}$ and the hyperedge centroids; otherwise padding introduces a non-negligible equivariance error. Unit tests should therefore include padded batches and check that residuals stay near machine precision.

We document one such bug. An early implementation computed the masked centroid normalization along the time dimension, which broke translation equivariance for padded batches and raised residuals to 10^{-4}. The bug is fixed, and the unit tests now enforce correct masking for variable-size batches. Table 11 reports the resulting unit-test residuals.

Table 11: Equivariance and invariance residuals of randomly initialized, untrained networks in float64 (maximum over randomly sampled transforms, including padded variable-size batches). The same tests caught the masking bug described in Appendix A.5.

Test scenario	Numerical residual
SO(2) rotation + translation	10^{-15}
SO(3) rotation + translation	10^{-15}
O(d) reflection equivariance	10^{-16}
Hyperedge assignment $\mathbf{A}$ invariance	10^{-16}
Agent permutation equivariance	0
Large pure-translation shift ($\lvert t\rvert \approx 100$)	10^{-14}

B Experimental Setup

B.1 Hyperparameters and Training Details

Optimization. We use Adam with learning rate $5e{-4}$ (ETH–UCY: $5e{-4}$ or $1e{-3}$ per subset), batch size 100, 150–300 epochs, hidden width 64 and weight decay $1e{-12}$. Model selection uses a 10% validation split of the training data; test data are never used for checkpoint selection. Baselines use the same data splits, horizons and hardware; EqMotion and GroupNet use their official code and hyperparameters.

EGNN baseline. EGNN is not a sequence-to-sequence forecaster; we apply it autoregressively over the prediction horizon. Stable training required truncated back-propagation through time, gradient clipping at 0.5 and learning rate $5e{-5}$; full back-propagation through time diverged.

Table 12 lists the hyperedge count and loss weights used in each experiment.

Table 12: Hyperedge count and loss weights per experiment. For MD17, N counts heavy atoms. λ_3 is used only with group labels. On MD17 we use no topology regularizers for forecasting ($\lambda_1 = \lambda_2 = 0$); bond-supervised runs use only the co-membership loss with $\lambda_3 = 0.3$. The scalability benchmark measures forward-pass latency and memory with untrained models, so no loss weights apply.

Experiment	N	K	Rationale for K	λ_1	λ_2	λ_3
Group-Hard / group switch	12	3	$= G$ (ground-truth groups)	0.3	0.02	0 / 0.2
Group-Hard	48	12	$= G$	0.3	0.02	0 / 0.2
ETH–UCY	≤ 8	6	approx. social-group count	0.1	0.01	0
MD17 asp./benz./eth./malon.	13/6/3/5	4/2/2/3	approx. functional groups	0	0	0 / 0.3
Scalability (Table 6)	50–400	8	fixed capacity	–	–	–

B.2 Published ETH–UCY Results

For reference, Table 13 lists results reported in prior publications under the original ETH–UCY protocol. They are not comparable to Table 4, in which all models are retrained under our protocol (Sec. 5.2).

Table 13: ETH–UCY best-of-20 ADE/FDE as reported in prior publications under the original pre-processing (1–3 agents per sample). These numbers are *not* comparable to Table 4. Under this original protocol our reproduction of EqMotion averages $0.222/0.365$.

Method (reported)	ETH	Hotel	Univ	Zara1	Zara2	Average
SGAN (2018)	0.87/1.62	0.67/1.37	0.76/1.52	0.35/0.68	0.42/0.84	0.61/1.21
NMMP (2020)	0.61/1.08	0.33/0.63	0.52/1.11	0.32/0.66	0.43/0.85	0.41/0.82
Traj++ (2020)	0.61/1.02	0.19/0.28	0.30/0.54	0.24/0.42	0.18/0.31	0.30/0.51
PECNet (2020)	0.54/0.87	0.18/0.24	0.35/0.60	0.22/0.39	0.17/0.30	0.29/0.48
AgentFormer (2021)	0.45/0.75	0.14/0.22	0.25/0.45	0.18/0.30	0.14/0.24	0.23/0.39
GroupNet (2022)	0.46/0.73	0.15/0.25	0.26/0.49	0.21/0.39	0.17/0.33	0.25/0.44
MID (2022)	0.39/0.66	0.13/0.22	0.22/0.45	0.17/0.30	0.13/0.27	0.21/0.38
GP-Graph (2022)	0.43/0.63	0.18/0.30	0.24/0.42	0.17/0.31	0.15/0.29	0.23/0.39
EqMotion (2023)	0.40/0.61	0.12/0.18	0.23/0.43	0.18/0.32	0.13/0.23	0.21/0.35

C Additional Results

C.1 Sensitivity to K and λ_1

Table 14 and Figure 7 vary the number of hyperedges K, and Table 15 varies the load-balancing weight λ_1; all are single runs on Group-Hard ($N = 12$).

Table 14: Sensitivity to the number of hyperedges K on Group-Hard (ground-truth $G = 3$; $\lambda_1 = 0.3$; single run). $K_{\text{eff}} = \exp(H(\bar{\mathbf{A}}))$ is the effective number of occupied hyperedges. ADE is insensitive to K; grouping F1 peaks at $K = G$. Since $K_{\text{eff}} \approx K$ throughout, the F1 drop for $K > G$ reflects over-splitting of true groups under the balancing term, not collapse. We did not repeat the sweep at the recommended $\lambda_1 = 0.03$, where the balancing pressure is ten times weaker.

Number of hyperedges K	K_{eff}	Grouping F1 (↑)	ADE (↓)
2	2.00	0.521	0.0075
3 (= G)	2.99	0.721	0.0069
4	3.99	0.431	0.0066
8	7.95	0.212	0.0073
16	15.76	0.186	0.0072

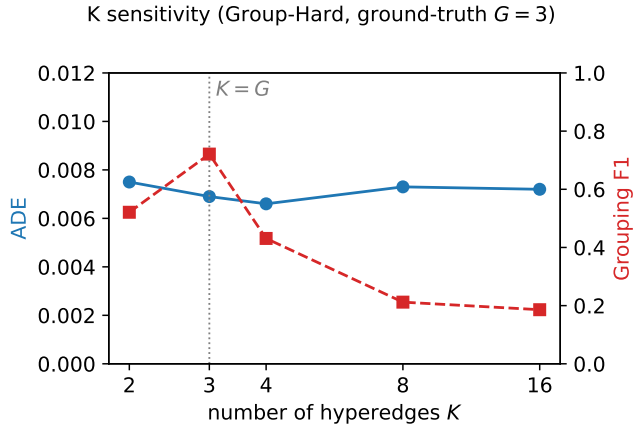

Figure 7: K sensitivity on Group-Hard ($G = 3$; $\lambda_1 = 0.3$; single run). ADE is insensitive to K, while grouping F1 peaks at $K = G$. For $K > G$ all hyperedges remain occupied ($K_{\text{eff}} \approx K$, Table 14), so the drop reflects over-splitting of true groups rather than collapse.

Table 15: Sensitivity to the load-balancing weight λ_1 on Group-Hard ($N = 12$), single run per value. Multi-seed results for $\lambda_1 \in \{0, 0.03, 0.3\}$ are in Table 2 (e.g. $\lambda_1 = 0.03$: ADE 0.0062 ± 0.0009, F1 0.855 ± 0.126), so the single-run values at $\lambda_1 = 0.03$ below overstate the expected gain. Large λ_1 degrades grouping.

Regularizer λ_1	ADE (↓)	FDE (↓)	Grouping F1 (↑)
0	0.0066	0.0165	0.888
0.03 (recommended)	0.0045	0.0112	0.979
0.1	0.0060	0.0134	0.977
0.3	0.0069	0.0184	0.721
1.0	0.0074	0.0186	0.644

C.2 Architectural Ablations

Figure 8 visualizes the ablation of Table 2.

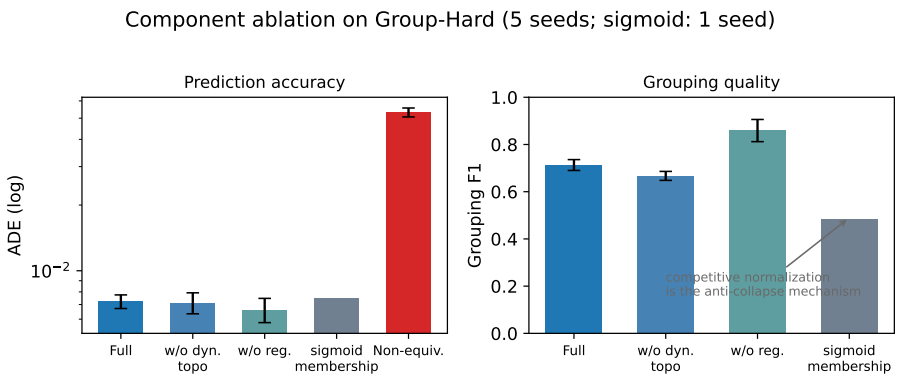

Figure 8: Ablation on Group-Hard ($N = 12$; five seeds, sigmoid one seed; $\lambda_1 = 0.3$). Left: ADE; only the non-equivariant variant changes ADE substantially. Right: grouping F1; sigmoid membership lowers F1 from 0.71 to 0.49, while removing the load-balancing term raises F1 from 0.71 to 0.86.

C.3 Dynamic Topology and Learned Memberships

Figure 9 compares dynamic and static topology seed by seed under the group switch of Table 3, and Figure 10 shows learned membership matrices.

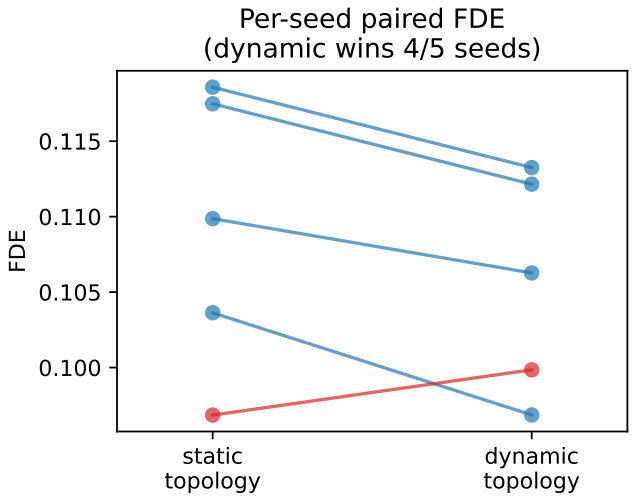

Figure 9: Per-seed paired FDE, dynamic versus static topology, under a group switch at frame 10 inside the observation window. Dynamic topology is lower for 4 of 5 seeds; mean ADE/FDE differences are within one standard deviation (Table 3).

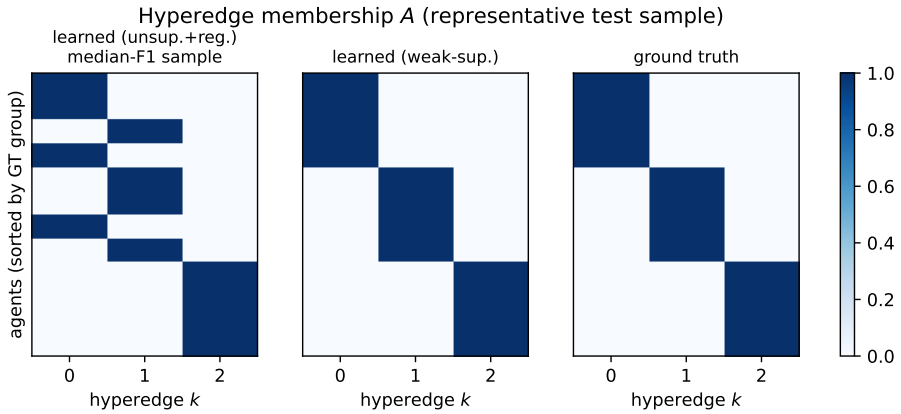

Figure 10: Membership matrices for a median-F1 test sample (chosen to avoid cherry-picking). Rows: agents sorted by ground-truth group; columns: hyperedges; brightness: soft membership. Left: unsupervised—one ground-truth group is recovered exactly, while agents of the other two are split between hyperedges 0 and 1. Middle: weakly supervised. Right: ground truth.

C.4 Scalability

EqMotion and EqDynHyper both have approximately 470k trainable parameters, so capacity differences do not explain the results. During training, enabling dynamic topology adds moderate overhead for small agent counts; full training benchmarks will be released with the code. Figure 11 shows ADE versus agent count, Figure 12 forward latency, and Figure 13 peak GPU memory.

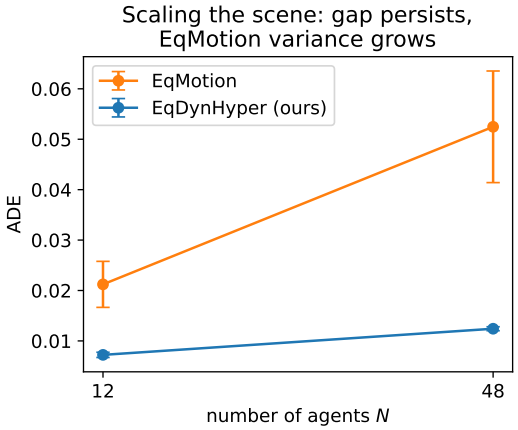

Figure 11: ADE versus agent count on Group-Hard ($N = 12$: five seeds; $N = 48$: three). Accuracy at the sizes of Table 6 ($N \geq 50$) is not evaluated.

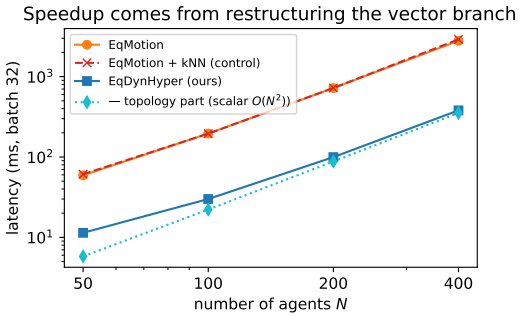

Figure 12: Forward latency versus N (log–log), including a k-NN-sparsified EqMotion control. At large N, EqDynHyper's runtime is dominated by the scalar $O(N^2)$ topology term (dotted).

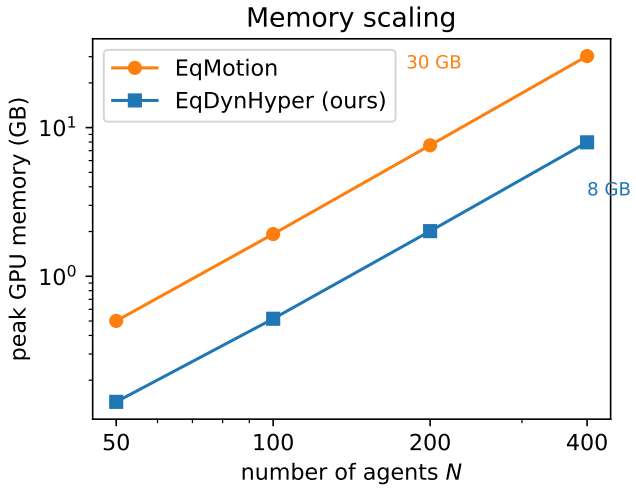

Figure 13: Peak GPU memory versus N (log–log). Both models scale quadratically (slope $\approx$ 2), because EqDynHyper's topology module still forms all agent pairs for scalar features; EqDynHyper lowers the constant by about $3.75\times$ by aggregating vector-valued features over K hyperedges instead of N^2 edges.

C.5 Prediction Horizon and Data Efficiency

Figure 14 explains why EqMotion's FDE can be lower than its ADE on Group-Hard (Table 1), and Figure 15 reports the MD17 low-data experiment.

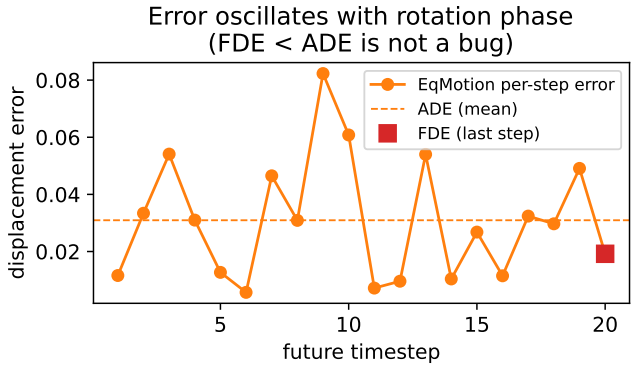

Figure 14: Per-step error of a single EqMotion run on Group-Hard, from an earlier seed whose ADE (dashed, ≈ 0.031) is higher than the five-seed mean in Table 1 (0.0212). The error oscillates with rotation phase, so the final-step error (FDE) can be smaller than the mean over the horizon (ADE).

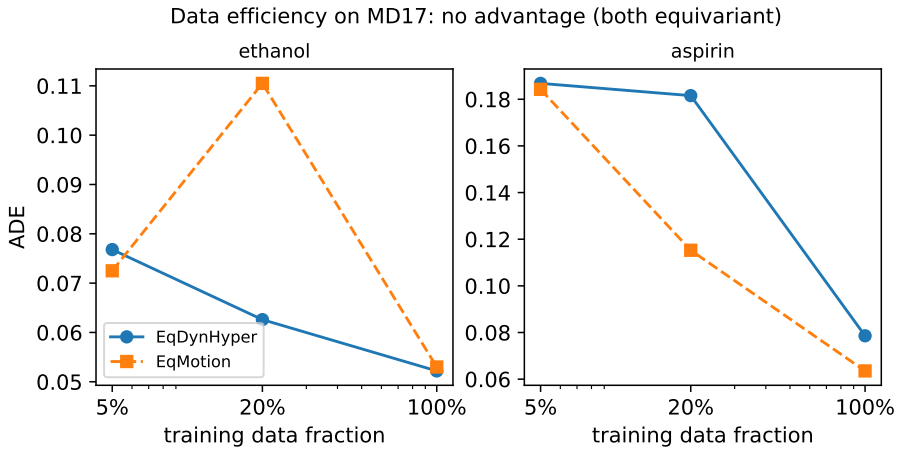

Figure 15: ADE versus training-data fraction on MD17 ethanol and aspirin (single run per point). Neither model shows a consistent data-efficiency advantage. The non-monotonic EqMotion point at 20% on ethanol comes from a single run that we did not repeat.

C.6 Group Interpretability

MD17 Functional Groups. Beyond forecasting, EqDynHyper can recover molecular functional groups. Pairwise F1 measures the agreement between inferred hyperedge memberships and chemical functional groups (Table 16). Pairwise equivariant baselines such as EqMotion provide no comparable group-discovery output. With bond-graph weak supervision, aspirin reaches pairwise F1 0.998, almost perfectly recovering its carboxyl and aromatic-ring moieties. Benzene is an informative counterexample: its fully symmetric six-carbon ring admits no preferred partition, so weak supervision barely changes F1 ($0.547 \rightarrow 0.556$).

Table 16: MD17 functional-group recovery (pairwise F1) without supervision and with bond-graph weak supervision, and the resulting change in ADE ($\times 10^{-2}$; single run). Weak supervision uses only a static per-molecule grouping obtained by clustering the bond graph; it never touches trajectory labels.

Molecule	F1 (unsup.)	F1 (bond-sup.)	ADE unsup. $\rightarrow$ bond-sup.
Aspirin	0.225	0.998	$7.86 \rightarrow 7.90$
Benzene	0.547	0.556	$1.59 \rightarrow 2.13$
Ethanol	0.941	0.990	$5.22 \rightarrow 5.22$
Malonaldehyde	0.327	0.996	$7.94 \rightarrow 6.94$

Social Groups on ETH–UCY. Figure 16 compares intra- and inter-group distances of the learned groups with a random-grouping null (numbers in Table 8). Figures 17 and 18 show the partitions that EqDynHyper discovers without membership supervision on two scenes; trajectories are colored by learned hyperedge assignment, with observed history solid and predictions dashed. Pedestrians who move together tend to share a hyperedge, in line with intuitive social groups.

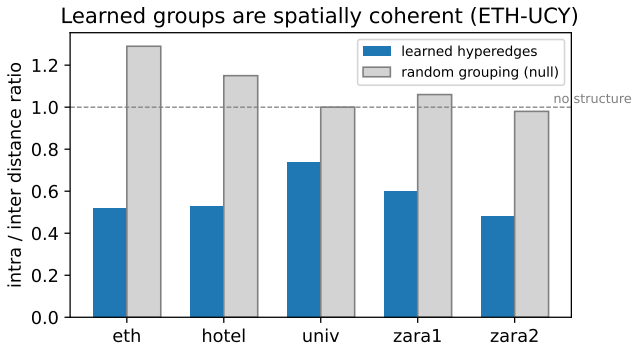

Figure 16: Intra-/inter-group distance ratio of learned hyperedges versus a random-grouping null on ETH–UCY. Learned groups are spatially coherent. Because topology inference receives pairwise distances as input, part of this effect is expected from proximity alone; the ratio shows spatial coherence, not that the groups are social.

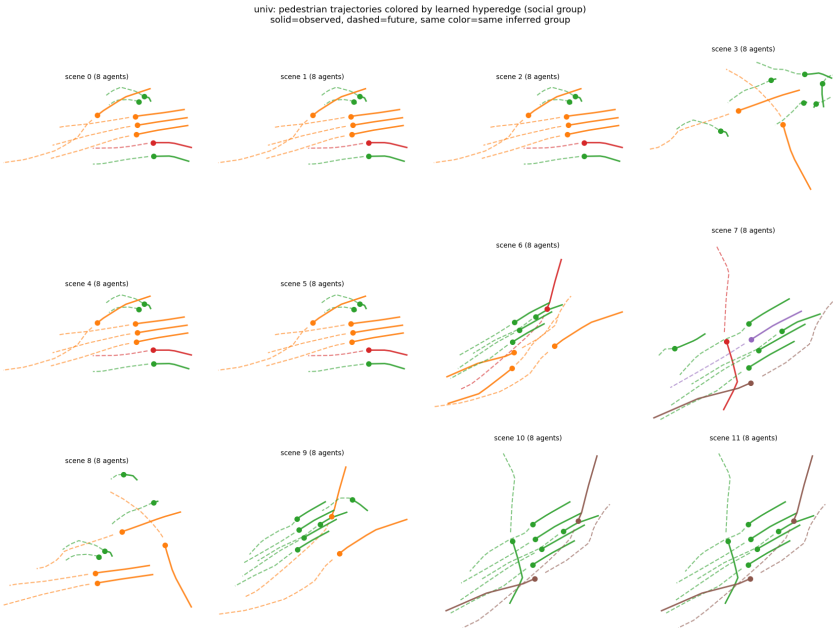

Figure 17: Univ scene: pedestrian trajectories colored by learned hyperedge assignment (solid: observed history; dashed: prediction; same color: same hyperedge). Several panels are consecutive, overlapping windows of the same scene. Pedestrians walking together tend to share a hyperedge.

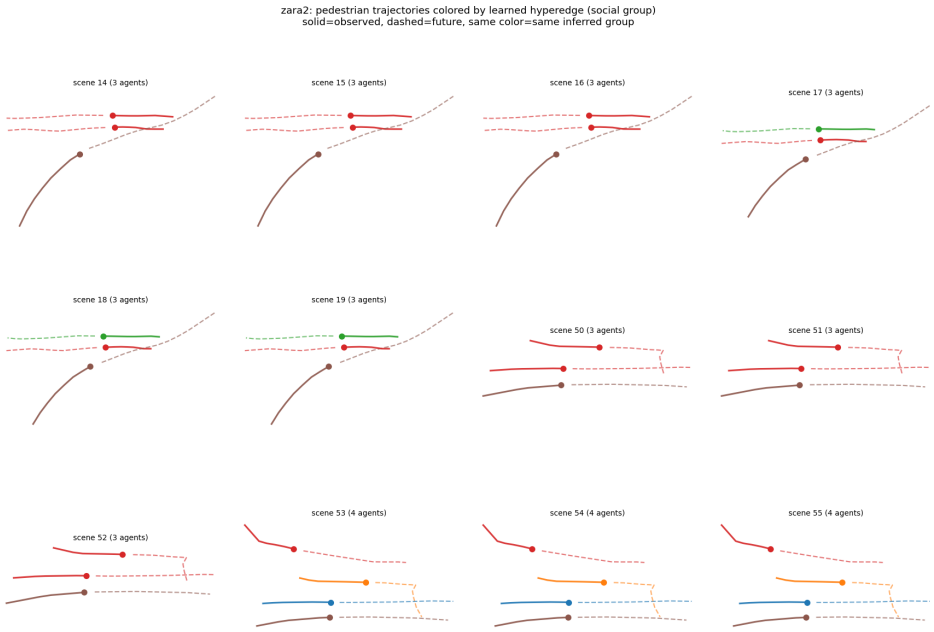

Figure 18: Zara2 scene: pedestrian trajectories colored by learned hyperedge assignment (solid: observed history; dashed: prediction). Independently moving pedestrians are assigned to different hyperedges.