

Multi-Agent Dynamic Relational Reasoning for Social Robot Navigation

Jiachen Li^{1*}, Chuanbo Hua^{2*}, Jianpeng Yao¹, Hengbo Ma³, Jinkyoo Park², Victoria Dax⁴, and Mykel J. Kochenderfer⁴

arXiv:2401.12275v2 [cs.RO] 11 Nov 2024

Abstract—Social robot navigation can be helpful in various contexts of daily life but requires safe human-robot interactions and efficient trajectory planning. While modeling pairwise relations has been widely studied in multi-agent interacting systems, the ability to capture larger-scale group-wise activities is limited. In this paper, we propose a systematic relational reasoning approach with explicit inference of the underlying dynamically evolving relational structures, and we demonstrate its effectiveness for multi-agent trajectory prediction and social robot navigation. In addition to the edges between pairs of nodes (i.e., agents), we propose to infer hyperedges that adaptively connect multiple nodes to enable group-wise reasoning in an unsupervised manner. Our approach infers dynamically evolving relation graphs and hypergraphs to capture the evolution of relations, which the trajectory predictor employs to generate future states. Meanwhile, we propose to regularize the sharpness and sparsity of the learned relations and the smoothness of the relation evolution, which proves to enhance training stability and model performance. The proposed approach is validated on synthetic crowd simulations and real-world benchmark datasets. Experiments demonstrate that the approach infers reasonable relations and achieves state-of-the-art prediction performance. In addition, we present a deep reinforcement learning (DRL) framework for social robot navigation, which incorporates relational reasoning and trajectory prediction systematically. In a group-based crowd simulation, our method outperforms the strongest baseline by a significant margin in terms of safety, efficiency, and social compliance in dense, interactive scenarios. We also demonstrate the practical applicability of our method with real-world robot experiments. The code and experiment videos can be found on the project website: <https://relational-reasoning-nav.github.io/>.

Index Terms—Relational reasoning, social interactions, reinforcement learning, trajectory prediction, social robot navigation, graph neural network, crowd simulation

I. INTRODUCTION

Social robot navigation has emerged as a crucial area of study in the rapidly evolving landscape of robotics, particularly in environments densely populated with humans (e.g., restaurants, hospitals, hotels). The integration of intelligent mobile robots into these shared spaces requires understanding human interactive behavior [1], [2]. In the early literature, mobile robot navigation systems have been designed with a focus on static obstacle avoidance and efficiency, often neglecting

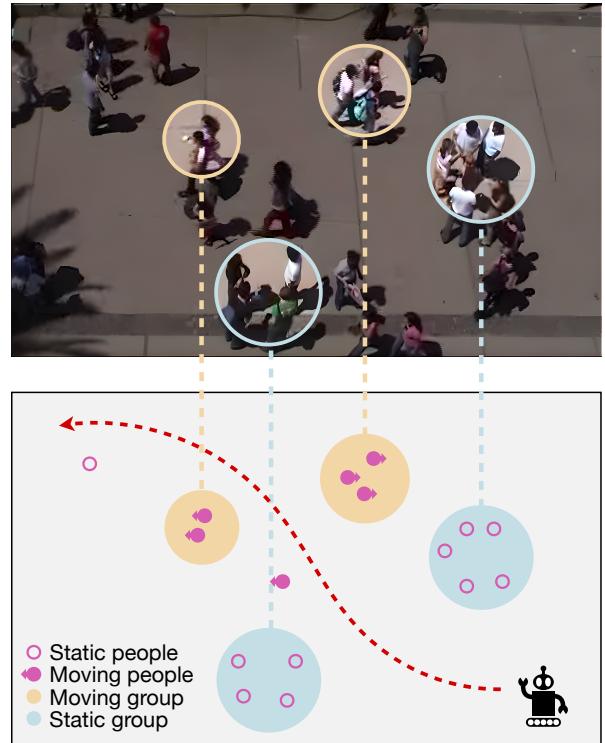


Fig. 1. An illustrative example of social robot navigation in a dynamic, crowded scenario where the robot plans to reach its destination at the top left of the scene from its current location. Some pedestrians move individually while others behave similarly as a group. The relations between agents may evolve over time. The ellipses denote different groups of interacting agents that exhibit distinct group-wise relations, and the arrows indicate agents' motions. The robot needs to reach its destination safely and efficiently without colliding with pedestrians or intruding into group spaces.

the subtleties of human-robot interactions and social norms [3]. This oversight has limited their applicability in highly interactive human-centric environments. As human-robot interactions and collaborations become increasingly common in recent years, it is inevitable to enable safe, efficient, and socially compliant robot navigation.

A major challenge in social robot navigation lies in accurately modeling and predicting the future behavior of multi-agent interacting systems, which requires effective relational reasoning. Fig. 1 provides a scenario that requires the robot to accurately understand both pairwise and group-wise relations between humans and infer how those relations influence their future behaviors. Many research efforts in multi-agent trajectory prediction have been devoted to investigating pairwise

* Equal contribution

¹ J. Li and J. Yao are with the University of California, Riverside, CA, USA. {jiachen.li, jyao073}@ucr.edu.

² C. Hua and J. Park are with the Korea Advanced Institute of Science and Technology, Korea. {cbhua, jinkyoo.park}@kaist.ac.kr.

³ H. Ma is with the University of California, Berkeley, CA, USA. hengbo_ma@berkeley.edu.

⁴ V. Dax and M. J. Kochenderfer are with Stanford University, CA, USA. {vmdax, mykel}@stanford.edu.

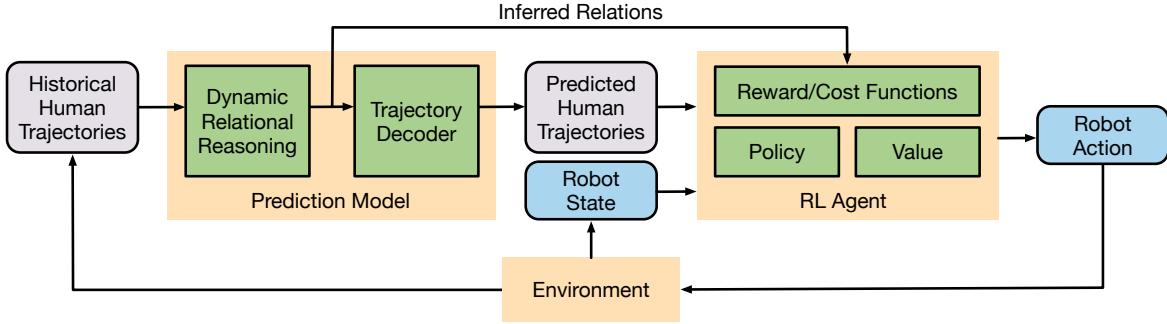


Fig. 2. An overall diagram of our proposed pipeline, which consists of a trajectory prediction model and an RL agent for robot navigation. The prediction model takes in the historical observations of human motion and generates future trajectory hypotheses while inferring the underlying relations between humans. The RL agent decides the optimal robot actions based on the predicted trajectories and the current robot state. The inferred relations are used in designing the reward function. We decouple the training of the prediction model and the RL agent into two sequential phases. More details about the architectures and training strategy of the two components can be found in Sections IV and V.

relational reasoning [4]–[7] based on the graph representations where nodes and edges represent agents and their pairwise relations, respectively. However, group-wise relational reasoning and its implications in social navigation remain underexplored.

Although the recognition of group activities in images and videos has been extensively researched in the field of computer vision in recent years [8], these approaches primarily rely on the visual appearance information and human skeletal movements as informative cues for identifying group-level behaviors. In contrast, only the historical state information is available in the context of trajectory prediction, which makes group recognition more difficult. Some prior work proposed group identification methods that use heuristics based on the proximity of interactive agents [9], [10], which work well in a specific scenario. However, it is difficult for these methods to directly generalize to diverse scenarios where the heuristics may need additional tuning. For example, Group-LSTM [9] uses the coherent filtering technique [11] to cluster the agents based on their collective movements. This group clustering strategy may work in scenarios where the agents in the same group behave similarly, but it may not generalize well to scenarios where agents in the same group can have highly distinct or even adversarial behaviors. Moreover, Group-LSTM models interactions implicitly by learning a feature embedding for each agent through a social pooling layer, which is fixed in the whole prediction horizon and difficult to explain. Grouptron [10] employs a spatio-temporal clustering technique based on the proximity of interactive agents and extracts interaction patterns at different scopes. However, group identification highly depends on the proximity threshold and cannot differentiate various types of relations. GroupNet [12] is a more flexible approach for group inference and modeling based on graph neural networks. However, it cannot handle the dynamic evolution of relations over time and does not provide a way to leverage the inferred relations in social robot navigation.

To address these issues, we propose a systematic framework for trajectory prediction with dynamic pairwise and group-wise relational reasoning for multi-agent interacting systems, which extends our prior work [5] that only handles pairwise relations. A natural representation for group-wise relations is

a hypergraph where each hyperedge can connect more than two nodes (i.e., agents). Hypergraphs have been employed in areas such as chemistry, point cloud processing, and sensor network analysis. Most existing works propose message passing mechanisms over hypergraphs with a pre-defined topology. In contrast, we propose an effective mechanism to infer the hypergraph topology for group-aware relational reasoning in a fully data-driven manner. Meanwhile, we propose a hypergraph evolution mechanism to enable dynamic reasoning about both pairwise and group-wise relations over time. It is also necessary to encourage smooth evolutions of the relational structures and adjust their sharpness and sparsity in a flexible way for different applications.

Moreover, we present a deep reinforcement learning (DRL) framework for social robot navigation adapted from CrowdNav++ [13] and a systematic way to employ our relational reasoning approach in the decision making process. More specifically, we employ multi-head attention mechanisms to enable the robot to focus on important robot-human interactions and human-human interactions that may potentially influence the robot's planning, especially in highly crowded scenarios with a large number of agents. Meanwhile, we employ the inferred relations between agents in the reward function design and incorporate the predicted future trajectories into the input of the policy network, which demonstrates the effectiveness of relational reasoning in social navigation. The overall diagram of our proposed pipeline is illustrated in Fig. 2, which integrates the prediction model and reinforcement learning framework.

The main contributions of this paper are as follows:

- We propose a systematic dynamic relational reasoning framework by inferring dynamically evolving latent interaction graphs and hypergraphs based on historical observations, and demonstrate its effectiveness for multi-agent trajectory prediction and social robot navigation.
- We design a flexible mechanism to construct hyperedges and introduce a novel hypergraph attention mechanism to extract group features. We also propose new mechanisms to regularize the sharpness of learned relations, smoothness of relation evolution, and the sparsity of the learned graphs and hypergraphs, which not only enhance training stability but also reduce prediction error.

- We introduce a modified deep reinforcement learning framework for social robot navigation, which takes advantage of the inferred relations and predicted trajectories in a principled manner to improve safety and efficiency while complying with human social norms.
- We validate the trajectory prediction method with relational reasoning on both crowd simulations and real-world benchmark datasets. Our method infers reasonable evolving hypergraphs for group-wise relational reasoning and achieves state-of-the-art performance. We also evaluate the DRL-based robot navigation method on our group-based crowd simulations and real-world scenarios, demonstrating enhanced navigation performance, social awareness, and appropriateness in robot motion planning.

This paper builds upon our previous work [5] in several important ways. First, the relational reasoning in our earlier work only considers the fundamental pairwise relations while we additionally reason about group-wise relations in this work to enable systematic relational reasoning at different scales. Second, we propose a series of effective regularization mechanisms to improve the learning of the dynamic evolution of relations in terms of smoothness, sharpness, and sparsity. Third, our prior work only deals with the trajectory prediction problem while in this work we further investigate a systematic way to employ relational reasoning and trajectory prediction components in the downstream decision making task for social robot navigation. We also develop a group-based crowd simulator to evaluate our approach.

The remainder of the paper is organized as follows. Section II reviews related work concisely. Section III provides the necessary background related to the proposed approaches. Section IV introduces the relational reasoning framework for trajectory prediction, and Section V introduces the deep reinforcement learning framework for social robot navigation. Section VI presents experimental settings, implementation details, and empirical results. Finally, Section VII concludes the paper by summarizing the key findings, discussing potential limitations, and suggesting directions for future research.

II. RELATED WORK

A. Trajectory Prediction

The objective of trajectory prediction is to forecast a sequence of future states based on historical observations, which has been widely studied in dense scenarios with highly interactive entities (e.g., pedestrians [1], [2], [14]–[16], vehicles [17], sports players [4]–[6], [12]). Traditional approaches such as physics-based models [18], [19], heuristics-based models [20], [21], or probabilistic graphical models [22], [23] achieved satisfactory performance in simple scenarios where agents mostly move independently. Recent advances in deep learning models have enhanced the capability of capturing not only the motion patterns of individuals but also the complex interactions between agents, which improves prediction accuracy. For instance, the temporal information of agent trajectories is captured by using sequential models such as long short-term memory (LSTM) [9], [24]. Multi-agent interaction behavior is modeled by graph neural networks (GNNs) [10], [12], [25].

Also, generative models are incorporated in the prediction framework to capture the multi-modal distribution of future trajectories [25]–[28].

Due to the promising performance of transformers in natural language processing [29], [30], transformer-based approaches have also been prevailing in trajectory prediction [31]–[35]. For example, a unified transformer-based approach is proposed to consider different modalities such as road elements, agent interactions, and time steps [33]. Shi et al. propose a transformer-based prediction framework that incorporates motion intention priors to improve prediction performance [34]. An autoregressive transformer-based method called MotionLM generates joint predictions in a single autoregressive pass [35]. These approaches usually model multi-agent interactions implicitly while not providing explicit, explainable relational structures that can be used in downstream tasks.

B. Social Robot Navigation

Social robot navigation not only considers human behavior modeling but also robot motion planning [36]. The robot navigation system should consider the safety and efficiency of the robot and humans in the environment. Besides developing effective human behavior modeling methods for trajectory prediction to capture the interactions, another challenge is how to ensure the safety of humans during navigation. On one hand, some work focuses on game-theoretic approaches. For example, the human-robot interaction problem can be formulated as a non-cooperative non-zero-sum game [37]. Multi-agent interaction problems can be formulated as a repeated linear-quadratic game [38]. On the other hand, some work proposes reinforcement learning based methods to implicitly capture the human-robot interactions and synthesize the robot planning policy directly based on the observations [13], [39]–[42]. Liu et al. [13] propose an RL-based social navigation method with trajectory prediction of surrounding pedestrians. However, their prediction and planning models do not consider group-wise relations or activities. Katyal et al. [42] propose to learn a group-aware policy for robot navigation. However, their method only uses the historical state information for robot decision making while not considering the future behaviors of pedestrians. Also, their method requires ground truth group membership information during training and does not consider the evolution of group relations over time. In contrast, our approach explicitly infers dynamic pairwise and group-wise relations in an unsupervised manner, which incorporates interaction modeling and human behavior modeling into the reinforcement learning framework systematically.

C. Relational Learning and Reasoning

The relational learning and reasoning module is crucial in multi-agent trajectory prediction [1], [17]. It also shows effectiveness in social robot navigation [36], physical dynamics modeling [43], [44], human-object interaction detection [45], group activity recognition [46], and community detection in various types of networks [47]. The performance of these tasks highly depends on the learned relations or inferred interactions between different entities. Traditional statistical relational

learning approaches were proposed to infer the dependency between random variables based on statistics and logic, which are difficult to leverage high-dimensional data (e.g., images / videos) to infer complex relations [48]. Recently, deep learning has been investigated to model the relations or interactions such as pooling layers [24], [49], attention mechanisms [50], [51], transformers [31], [32], [52], and graph neural networks [4]–[7], [53], which can extract flexible relational features used in downstream tasks by aggregating the information of individual entities. In contrast with most existing graph-based methods, which only use edges to model pairwise relations between agents, we propose to adopt effective hyperedges to additionally capture dynamic group-wise relations by inferring evolving hypergraphs.

Relational reasoning is an effective component for enabling safe and efficient robot navigation in dynamic environments. Model-free RL policies that use one-step predictions as observations can only capture instantaneous velocities, which are not sufficient for complex situations [54], [55]. While some studies predict the long-term goals of pedestrians to enhance planning, their reliance on a finite set of potential goals restricts generalizability [56], [57]. Liu et al. [13] extend this by predicting pedestrian movement in a long-term, continuous two-dimensional space. However, their approach does not account for group-wise relations. Our method considers both pairwise and group-wise relations to enhance the safety and social compliance of the learned robot policy.

III. PRELIMINARIES

A. Graph Neural Networks

Graph Neural Networks (GNNs) are a class of deep learning models designed to capture the dependencies in graph-structured data. Consider a graph $G = (V, E)$, where V is the set of vertices (or nodes) and $E \subseteq V \times V$ is the set of edges. Each node $v \in V$ is associated with a feature vector \mathbf{x}_v . The goal of a GNN is to learn a state embedding \mathbf{h}_v for each node that captures both its features and the neighbors' information given the structure of the graph.

The core idea behind GNNs is to update the representation of a node by aggregating the features of its neighbors:

$$\mathbf{h}_v^{(k)} = g^{(k)} \left(\mathbf{x}_v, \text{AGG}^{(k)} \left(\{\mathbf{h}_u^{(k-1)} : u \in \mathcal{N}(v)\} \right) \right), \quad (1)$$

where $\mathbf{h}_v^{(k)}$ is the feature vector of node v at the k -th iteration, $\mathcal{N}(v)$ denotes the set of neighbors of v , and $\text{AGG}^{(k)}$ and $g^{(k)}$ are differentiable functions that aggregate and update node features, respectively. In many GNN architectures, the aggregation function is a simple operation, such as sum, mean, or max. For instance, a common choice for the $\text{AGG}^{(k)}$ function is the mean aggregator:

$$\text{AGG}^{(k)}(\cdot) = \frac{1}{|\mathcal{N}(v)|} \sum_{u \in \mathcal{N}(v)} \mathbf{h}_u^{(k-1)}, \quad (2)$$

where $|\mathcal{N}(v)|$ is the number of nodes in the set of neighbors. After K iterations of aggregation, the final node representations $\mathbf{h}_v^{(K)}$ can be used for various graph-related tasks, such as node classification, link prediction, and graph classification.

The effectiveness of GNNs lies in their ability to capture both the relational structure and node features in learning tasks.

B. Markov Decision Process and Reinforcement Learning

A Markov decision process (MDP) provides a mathematical framework used to model sequential decision making in discrete-time stochastic control [58]. An MDP is defined by a tuple (S, A, P, R, γ) where:

- S denotes a set of the agent's states.
- A denotes a set of actions available to the agent.
- $P : S \times A \times S \rightarrow [0, 1]$ denotes the transition function $P(s_{t+1} | s_t, a_t)$, which defines the probability of transitioning from state s_t to s_{t+1} after taking action a_t .
- $R : S \times A \rightarrow \mathbb{R}$ denotes the reward function $R(s_t, a_t)$.
- γ denotes the discount factor satisfying $0 \leq \gamma \leq 1$.

The environment is defined by a set of states and actions, with transitions between states being probabilistic and dependent on the actions taken. The agent receives rewards or penalties after each action, guiding its learning process. A reinforcement learning agent learns to make decisions by interacting with an environment modeled as an MDP to maximize the expected discounted sum of rewards over time.

C. Multi-Head Attention

The attention mechanism [59] is an important feature of many state-of-the-art neural network architectures, particularly in handling sequential data such as language processing or time series analysis. It allows the model to focus on different parts of the input sequence simultaneously.

The mechanism operates on queries (Q), keys (K), and values (V), which are typically derived from the same input in self-attention models. For a single attention head, we have

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V, \quad (3)$$

where d_k is the dimension of keys, and the softmax function is applied row-wise. This formulation computes a set of attention weights and applies them to the values, effectively determining which parts of the input sequence are emphasized. In the multi-head attention mechanism, this process is repeated across multiple heads, each with its own set of learned linear transformations for Q , K , and V .

IV. DYNAMIC RELATIONAL REASONING FOR TRAJECTORY PREDICTION

Trajectory prediction involves forecasting the future trajectory of dynamic objects based on historical observations. We assume that, without loss of generality, there are N agents in the scene. The number of agents may vary in different cases. We denote a set of agents' state sequences as

$$\mathbf{X}_{1:T} = \{\mathbf{x}_t^i \mid T = T_h + T_f, i = 1, \dots, N\}, \quad (4)$$

where N represents the number of involved agents, T_h and T_f represent the history and future prediction horizons, respectively. More specifically, we have $\mathbf{x}_t^i = (x_t^i, y_t^i)$ as the state of agent i , where (x, y) represents the 2D coordinate.

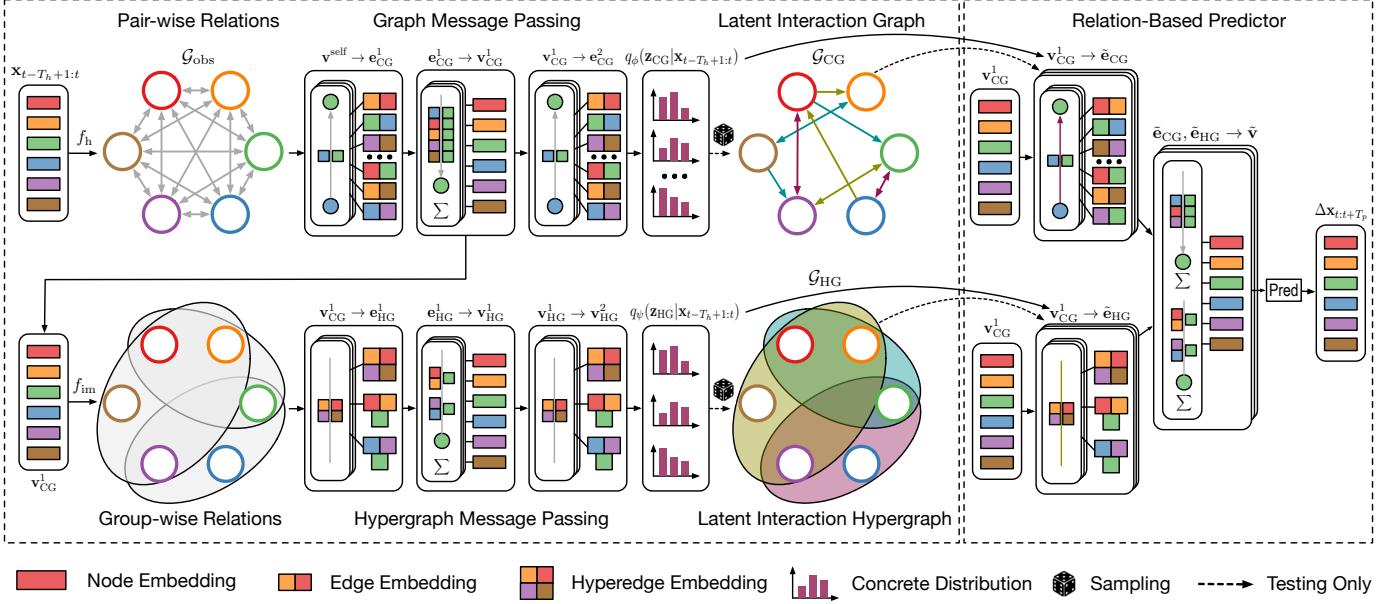


Fig. 3. A diagram of our prediction method within a single prediction period, which consists of an encoder for relational reasoning and a decoder to generate prediction hypotheses. Different colors of edges in the latent interaction graph and hyperedges (i.e., ellipses) in the latent interaction hypergraph indicate different relation types. Best viewed in color.

The objective of probabilistic trajectory prediction is to estimate the conditional distribution for future trajectories given the historical observation $p(\mathbf{X}_{T_h+1:T_h+T_f} | \mathbf{X}_{1:T_h})$. As an intermediate step, we also aim to infer the underlying relational structures in the form of dynamic graphs (i.e., pairwise relations) and hypergraphs (i.e., group-wise relations), which may evolve over time and largely influence the predicted trajectories. More specifically, we aim to estimate the conditional distribution of the edge types and hyperedge types given their extracted features. We treat “no edge / hyperedge” as a special type, which indicates no relation between the involved agents. Long-term prediction can be obtained by iterative short-term predictions with a horizon of T_p ($\leq T_f$).

A. Overview

Fig. 3 shows the major components of our trajectory prediction approach, which follows an encoder-decoder architecture with an iterative sliding window to enable dynamic relational reasoning for long-term prediction. The encoder infers relation graphs and hypergraphs in the latent space based on historical state observations without direct supervision of relations. We introduce effective regularizations on the learned relations. The decoder employs the inferred latent relational structures to generate the distribution of future trajectory hypotheses. We present a dynamic reasoning mechanism to capture the probabilistic evolution of pairwise and group-wise relations.

We employ a fully connected observation graph $\mathcal{G}_{obs} = \{\mathcal{V}_{obs}, \mathcal{E}_{obs}\}$ to represent the involved agents, where $\mathcal{V}_{obs} = \{\mathbf{v}_i | i \in \{1, \dots, N\}\}$ and $\mathcal{E}_{obs} = \{\mathbf{e}_{ij} | i, j \in \{1, \dots, N\}\}$. \mathbf{v}_i and \mathbf{e}_{ij} denote the attribute of node i and the attribute of the directed edge from node j to node i , respectively. Each agent node has two types of attributes: a *self-attribute*

encoding its own information and a *social-attribute* encoding the aggregation of others’ information.

B. Encoder: Relational Reasoning

The encoder consists of two parallel branches: *pairwise* relational reasoning using a conventional graph (\mathcal{G}_{CG}) and *group-wise* relational reasoning using a hypergraph (\mathcal{G}_{HG}), which are correlated via the node attributes obtained after a round of message passing across the observation graph \mathcal{G}_{obs} .

Computing pairwise edge features. The self-attribute $\mathbf{v}_i^{\text{self}}$ and social attribute $\mathbf{v}_i^{\text{social}}$ of agent i are obtained by

$$\begin{aligned} \mathbf{v}_i^{\text{self}} &= f_h(\mathbf{x}_{t-T_h+1:t}), \\ \alpha_{ij} &= \frac{\exp(\text{LeakyReLU}(\mathbf{a}^\top [\mathbf{Wv}_i^{\text{self}}, \mathbf{Wv}_j^{\text{self}}]))}{\sum_{k \in \mathcal{N}_i} \exp(\text{LeakyReLU}(\mathbf{a}^\top [\mathbf{Wv}_i^{\text{self}}, \mathbf{Wv}_k^{\text{self}}]))}, \\ \mathbf{e}_{CG,ij}^1 &= f_{CG,e}^1 \left(\left[\frac{\alpha_{ij}}{\alpha_{ij} + \alpha_{ji}} \mathbf{v}_i^{\text{self}}, \frac{\alpha_{ji}}{\alpha_{ij} + \alpha_{ji}} \mathbf{v}_j^{\text{self}} \right] \right), \\ \mathbf{v}_i^{\text{social}} &= f_{CG,v}^1 \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{e}_{CG,ij}^1 \right), \quad \sum_{j \in \mathcal{N}_i} \alpha_{ij} = 1, \end{aligned} \quad (5)$$

where $f_h(\cdot)$ denotes the history embedding function, α_{ij} denotes the learnable attention weights, $\mathbf{e}_{CG,ij}^1$, $f_{CG,e}^1(\cdot)$ and $f_{CG,v}^1(\cdot)$ denote the edge attribute, edge and node update functions during the first round of message passing across the conventional graph, respectively. Then we can obtain the complete node attribute of agent i and conduct a second round of edge update by

$$\mathbf{v}_{CG,i}^1 = [\mathbf{v}_i^{\text{self}}, \mathbf{v}_i^{\text{social}}], \quad \mathbf{e}_{CG,ij}^2 = f_{CG,e}^2([\mathbf{v}_{CG,i}^1, \mathbf{v}_{CG,j}^1]). \quad (6)$$

Computing hyperedge features. We infer the topology of the hypergraph \mathcal{G}_{HG} in the form of incidence matrix I_{HG} by

$$I_{HG} = f_{im}(\mathbf{V}_{CG}^1). \quad (7)$$

Specifically, we first obtain a probabilistic incidence matrix by

$$I_{\text{PIM}} = f_{\text{PIM}}(\mathbf{V}_{\text{CG}}^1), \quad (8)$$

where $I_{\text{PIM},im}$ is the probability of node i being connected by hyperedge \mathcal{H}_m , the i -th row of the node attribute matrix \mathbf{V}_{CG}^1 is $\mathbf{v}_{\text{CG},i}^1$, and $f_{\text{PIM}}(\cdot)$ is a multi-layer perceptron (MLP). Then, we obtain the incidence matrix I_{HG} by sampling the value of $I_{\text{HG},im}$ (1 or 0) from a Gumbel-Softmax distribution [60] defined by $I_{\text{PIM},im}$. In summary, we have

$$f_{\text{im}}(\mathbf{V}_{\text{CG}}^1) := \text{Gumbel-Softmax}(f_{\text{PIM}}(\mathbf{V}_{\text{CG}}^1)). \quad (9)$$

Note that we need to set the maximum number of hyperedges denoted as M according to specific applications. Based on the inferred I_{HG} , the message passing across the hypergraph is given by

$$\begin{aligned} \alpha_{im} &= \frac{1}{N_{\mathcal{H}_m} - 1} \sum_{i \neq j, i,j \in \mathcal{H}_m} \alpha_{ij}, \\ \mathbf{e}_{\text{HG},m}^1 &= f_{\text{HG},e}^1 \left(\sum_{i \in \mathcal{H}_m} \frac{\alpha_{im}}{\sum_{n:i \in \mathcal{H}_n} \alpha_{in}} \mathbf{v}_{\text{CG},i}^1 \right), \\ \alpha_{mi} &= \frac{\exp(\text{LeakyReLU}(\mathbf{a}^\top [\mathbf{W}_1 \mathbf{e}_{\text{HG},m}^1, \mathbf{W}_2 \mathbf{v}_{\text{CG},i}^1])))}{\sum_{j \in \mathcal{H}_m} \exp(\text{LeakyReLU}(\mathbf{a}^\top [\mathbf{W}_1 \mathbf{e}_{\text{HG},m}^1, \mathbf{W}_2 \mathbf{v}_{\text{CG},j}^1]))}, \\ \mathbf{v}_{\text{HG},i}^1 &= f_{\text{HG},v}^1 \left(\sum_{m:i \in \mathcal{H}_m} \alpha_{mi} \mathbf{e}_{\text{HG},m}^1 \right), \\ \mathbf{e}_{\text{HG},m}^2 &= f_{\text{HG},e}^2 \left(\sum_{i \in \mathcal{H}_m} \mathbf{v}_{\text{HG},i}^1 \right), \end{aligned} \quad (10)$$

where α_{im} denotes learnable attention weights from nodes to hyperedges, $N_{\mathcal{H}_m}$ denotes the number of nodes in the hyperedge \mathcal{H}_m , $\mathbf{e}_{\text{HG},m}^1$ denotes the attribute of hyperedge \mathcal{H}_m ($m \in \{1, \dots, M\}$), α_{mi} denotes learnable attention weights from hyperedges to nodes, $f_{\text{HG},e}^1(\cdot)$ / $f_{\text{HG},e}^2(\cdot)$ and $f_{\text{HG},v}^1(\cdot)$ denote the hyperedge and node update functions during the message passing across the hypergraph.

Infering relation patterns. The interaction graph \mathcal{G}_{CG} and hypergraph \mathcal{G}_{HG} represent relation patterns with a distribution of edge / hyperedge types, which are inferred based on the edge / hyperedge attributes. We denote the number of possible edge / hyperedge types as L_{CG} and L_{HG} which can be adjusted for different applications. Since not all pairs of agents have direct relations, we set the first type as “no edge” which prevents message passing through the corresponding edges and can be used to regularize the sparsity of relations. Formally, we formulate a classification problem which is to infer the edge / hyperedge (i.e., relation) types based on $\mathbf{e}_{\text{CG},ij}^2$ and $\mathbf{e}_{\text{HG},m}^2$ by

$$\begin{aligned} \mathbf{z}_{\text{CG},ij} &= \text{Softmax}((\mathbf{e}_{\text{CG},ij}^2 + \mathbf{g}) / \tau), \quad i, j \in \{1, \dots, N\}, \\ \mathbf{z}_{\text{HG},m} &= \text{Softmax}((\mathbf{e}_{\text{HG},m}^2 + \mathbf{g}) / \tau), \quad m \in \{1, \dots, M\}, \end{aligned} \quad (11)$$

where \mathbf{g} denotes a vector of i.i.d. samples from the $\text{Gumbel}(0, 1)$ distribution and τ denotes the temperature coefficient which controls the smoothness of samples [60]. We can use the reparametrization trick to get gradients from this approximation, which is adapted from NRI [4] and EvolveGraph [5]. The latent variables $\mathbf{z}_{\text{CG},ij}$ and $\mathbf{z}_{\text{HG},m}$ are multinomial

random vectors, which indicate the relation types for a certain edge / hyperedge. More formally, we have

$$\begin{aligned} \mathbf{z}_{\text{CG},ij} &= [z_{\text{CG},ij,1}, \dots, z_{\text{CG},ij,L_{\text{CG}}}], \\ \mathbf{z}_{\text{HG},m} &= [z_{\text{HG},m,1}, \dots, z_{\text{HG},m,L_{\text{HG}}}], \end{aligned} \quad (12)$$

where L_{CG} and L_{HG} denote the number of edge types and hyperedge types, respectively. Here, $z_{\text{CG},ij,r}$ ($1 \leq r \leq L_{\text{CG}}$) and $z_{\text{HG},m,s}$ ($1 \leq s \leq L_{\text{HG}}$) denote the probability of a certain edge / hyperedge belongs to a certain type, respectively. The operations in the encoder can be summarized as

$$\begin{aligned} q_{\text{CG}}(\mathbf{z}_{\text{CG},\beta} | \mathbf{X}_{t-T_h+1:t}), \quad q_{\text{HG}}(\mathbf{z}_{\text{HG},\beta} | \mathbf{X}_{t-T_h+1:t}) = \\ \text{Encoder}(\mathbf{X}_{t-T_h+1:t}). \end{aligned} \quad (13)$$

C. Decoder: Relation-Based Predictor

Relational feature aggregation. To enable dynamic relational reasoning, the whole prediction process consists of multiple recurrent prediction periods with a time horizon T_p during which the relation graphs / hypergraphs are re-inferred. In a single prediction period, the decoder is designed to predict the distribution of future trajectories conditioned on the historical observations or the prediction hypotheses in the last prediction period as well as the relation graphs / hypergraphs inferred by the encoder, which is written as $p(\mathbf{X}_{t+1:t+T_p} | \mathbf{X}_{t-T_h+1:t}, \mathcal{G}_{\text{CG}}, \mathcal{G}_{\text{HG}})$. We have

$$\begin{aligned} \tilde{\mathbf{e}}_{\text{CG},ij} &= \sum_{l=1}^{L_{\text{CG}}} z_{\text{CG},ij,l} \tilde{f}_{\text{CG},e}^l([\mathbf{v}_{\text{CG},i}^1, \mathbf{v}_{\text{CG},j}^1]), \\ \tilde{\mathbf{e}}_{\text{HG},m} &= \sum_{l'=1}^{L_{\text{HG}}} z_{\text{HG},m,l'} \tilde{f}_{\text{HG},e}^{l'} \left(\sum_{i \in \mathcal{H}_m} \mathbf{v}_{\text{CG},i}^1 \right), \\ \tilde{\mathbf{v}}_i &= \tilde{f}_v \left(\left[\sum_{i \neq j} \tilde{\mathbf{e}}_{\text{CG},ij}, \sum_{i \in \mathcal{H}_m} \tilde{\mathbf{e}}_{\text{HG},m} \right] \right), \end{aligned} \quad (14)$$

where $\tilde{\mathbf{e}}_{\text{CG},ij}$ and $\tilde{\mathbf{e}}_{\text{HG},m}$ are, respectively, the aggregated edge / hyperedge attributes computed from the node attributes $\mathbf{v}_{\text{CG},i}^1$ obtained in the encoding process. The edge update functions corresponding to a certain edge / hyperedge type in \mathcal{G}_{CG} and \mathcal{G}_{HG} are denoted as $\tilde{f}_{\text{CG},e}^l(\cdot)$ and $\tilde{f}_{\text{HG},e}^{l'}(\cdot)$. The node update function in the decoding process is denoted as $\tilde{f}_v(\cdot)$.

Predicting trajectory distributions. To capture the uncertainty of future trajectories, the predicted distribution of displacement $\Delta \mathbf{x}_t^i$ at each time step is designed to be a Gaussian distribution with an equal constant covariance. The output function $\tilde{f}_{\text{output}}(\cdot)$ takes in the updated node attribute $\tilde{\mathbf{v}}_i$ and outputs a sequence of the mean of the Gaussian kernel of displacement $\Delta \mathbf{x}_{t:t+T_p-1}^i$ for agent i . We iteratively calculate the mean of the coordinates of agent i in the future by

$$\hat{\mu}_{t+1}^i = \hat{\mathbf{x}}_t^i + \Delta \hat{\mathbf{x}}_t^i = \hat{\mathbf{x}}_t^i + \tilde{f}_{\text{output},t}(\tilde{\mathbf{v}}_i). \quad (15)$$

Then, the predicted conditional distribution is obtained by

$$p(\hat{\mathbf{x}}_{t+1}^i | \mathbf{z}_{\text{CG}}, \mathbf{z}_{\text{HG}}, \mathbf{x}_{t-T_h+1:t}^i) = \mathcal{N}(\hat{\mathbf{x}}_{t+1}^i | \hat{\mu}_{t+1}^i, \sigma^2 \mathbf{I}), \quad (16)$$

where σ^2 denotes the constant variance of Gaussian kernels, and \mathbf{I} denotes the identity matrix.

D. Dynamic Evolution of Relations

We generalize the graph evolving mechanism proposed in our prior work [5] to the dynamic inference of hypergraphs to capture evolving relations and the uncertainty of future trajectories. The operations include:

$$\begin{aligned} q_{\text{CG}}(\mathbf{z}_{\text{CG},\beta} | \mathbf{X}_{1+\beta\tau:T_h+\beta\tau}), q_{\text{HG}}(\mathbf{z}_{\text{HG},\beta} | \mathbf{X}_{1+\beta\tau:T_h+\beta\tau}) \\ = \text{Encoder}(\mathbf{X}_{1+\beta\tau:T_h+\beta\tau}), \end{aligned} \quad (17)$$

$$q_{\text{CG}}(\mathbf{z}'_{\text{CG},\beta} | \mathbf{X}_{1+\beta\tau:T_h+\beta\tau}) = \text{GRU-CG}(q(\mathbf{z}_{\text{CG},\beta} | \mathbf{X}_{1+\beta\tau:T_h+\beta\tau})), \quad (18)$$

$$q_{\text{HG}}(\mathbf{z}'_{\text{HG},\beta} | \mathbf{X}_{1+\beta\tau:T_h+\beta\tau}) = \text{GRU-HG}(q(\mathbf{z}_{\text{HG},\beta} | \mathbf{X}_{1+\beta\tau:T_h+\beta\tau})), \quad (19)$$

where $\mathbf{z}'_{\text{CG},\beta}$ and $\mathbf{z}'_{\text{HG},\beta}$ denote the updated pairwise and groupwise relational structures, τ denotes the time gap between two consecutive relational inference steps, and β denotes the index of relation graphs / hypergraphs starting from 0. Finally, the future trajectory distribution is

$$\begin{aligned} p(\mathbf{X}_{T_h+\beta\tau+1:T_h+(\beta+1)\tau} | \mathcal{G}'_{\text{CG},\beta}, \mathcal{G}'_{\text{HG},\beta}, \mathbf{X}_{1:T_h}, \hat{\mathbf{X}}_{T_h+1:T_h+\beta\tau}) \\ = \text{Decoder}(\mathcal{G}'_{\text{CG},\beta}, \mathcal{G}'_{\text{HG},\beta}, \mathbf{X}_{1:T_h}, \hat{\mathbf{X}}_{T_h+1:T_h+\beta\tau}). \end{aligned} \quad (20)$$

E. Regularizations on the Learned Relations

We propose effective regularization techniques on learned underlying relational structures in three aspects: smoothness, sharpness, and sparsity. They influence the learned relations by shaping the edge and hyperedge type distributions.

First, both pairwise and group-wise relations tend to evolve gradually and smoothly over time. Thus, we propose a smoothness regularization loss that indicates the difference between two consecutive graphs or hypergraphs, which is written as

$$\begin{aligned} L_{\text{SM}} = \alpha_{\text{SM,CG}} \text{KL}(q_{\text{CG}}(\mathbf{z}'_{\text{CG},\beta}) || q_{\text{CG}}(\mathbf{z}'_{\text{CG},\beta+1})) \\ + \alpha_{\text{SM,HG}} \text{KL}(q_{\text{HG}}(\mathbf{z}'_{\text{HG},\beta}) || q_{\text{HG}}(\mathbf{z}'_{\text{HG},\beta+1})), \end{aligned} \quad (21)$$

where $\alpha_{\text{SM,CG}}$ and $\alpha_{\text{SM,HG}}$ denote the corresponding coefficients and $\text{KL}(\cdot)$ denotes the Kullback–Leibler divergence. The intuition is that the consecutive graphs / hypergraphs should have similar topologies in the form of edge / hyperedge type distributions.

Second, to learn different relation types that represent distinct relation patterns, the learned relation distributions should have an appropriate sharpness. To avoid the distributions being too close to uniform, we propose a sharpness regularization loss based on the entropy of edge / hyperedge type distributions, which is written as

$$\begin{aligned} L_{\text{SH}} = -\alpha_{\text{SH,CG}} q_{\text{CG}}(\mathbf{z}'_{\text{CG},\beta}) \log q_{\text{CG}}(\mathbf{z}'_{\text{CG},\beta}) \\ -\alpha_{\text{SH,HG}} q_{\text{HG}}(\mathbf{z}'_{\text{HG},\beta}) \log q_{\text{HG}}(\mathbf{z}'_{\text{HG},\beta}), \end{aligned} \quad (22)$$

where $\alpha_{\text{SH,CG}}$ and $\alpha_{\text{SH,HG}}$ denote the coefficients.

Third, human interactions tend to be sparse. They usually only interact with a subset of surrounding agents that may influence their behaviors, which implies that not all pairs of agents have a direct relation. Thus, we propose a sparsity regularization on the learned relations, which is written as

$$\begin{aligned} L_{\text{SP}} = \alpha_{\text{SP,CG}} \text{KL}(q_{\text{CG}}(\mathbf{z}'_{\text{CG},\beta}) || q_{\text{CG},0}) \\ + \alpha_{\text{SP,HG}} \text{KL}(q_{\text{HG}}(\mathbf{z}'_{\text{HG},\beta}) || q_{\text{HG},0}) \end{aligned} \quad (23)$$

where $\alpha_{\text{SP,CG}}$ and $\alpha_{\text{SP,HG}}$ denote the coefficients, $q_{\text{CG},0}$ and $q_{\text{HG},0}$ are categorical distributions where the probability of the “no-relation” type is 1. This encourages sparser connections between nodes in the interaction graphs / hypergraphs.

F. Training Strategy and Loss Function

Since the inference of hyperedges highly depends on the intermediate node attributes \mathbf{V}_{CG}^1 obtained after the message passing across the observation graph \mathcal{G}_{obs} , we adopt a warm-up training stage during which only the modules related to the pairwise relational reasoning is trained. This training strategy provides a good initialization for the inference of hyperedges, which proves to accelerate convergence and improve the final performance. The complete loss at the formal training stage consists of a reconstruction loss, KL divergence losses, and regularization losses, written as

$$\begin{aligned} L &= L_{\text{Rec}} + L_{\text{KL}} + L_{\text{SM}} + L_{\text{SH}} + L_{\text{SP}}, \\ L_{\text{Rec}} &= \sum_{i=1}^N \sum_{t=T_h+1}^{T_h+T_f} \|\mathbf{x}_t^i - \boldsymbol{\mu}_t^i\|^2, \\ L_{\text{KL}} &= \alpha_{\text{KL,CG}} \text{KL}(q_{\text{CG}}(\mathbf{z}'_{\text{CG},\beta} | \mathbf{X}_{1+\beta\tau:T_h+\beta\tau}) || p_{\text{CG}}(\mathbf{z}'_{\text{CG},\beta})) \\ &\quad + \alpha_{\text{KL,HG}} \text{KL}(q_{\text{HG}}(\mathbf{z}'_{\text{HG},\beta} | \mathbf{X}_{1+\beta\tau:T_h+\beta\tau}) || p_{\text{HG}}(\mathbf{z}'_{\text{HG},\beta})), \end{aligned} \quad (24)$$

where $p_{\text{CG}}(\mathbf{z}'_{\text{CG},\beta})$ and $p_{\text{HG}}(\mathbf{z}'_{\text{HG},\beta})$ denotes the prior uniform categorical distributions, and $\alpha_{\text{KL,CG}}$ and $\alpha_{\text{KL,HG}}$ denote the corresponding coefficients. The whole model can be trained in an end-to-end manner.

V. SOCIAL ROBOT NAVIGATION

Social robot navigation entails determining a sequence of actions for a robot to reach its destination safely and efficiently based on the current robot’s state, the behavior of surrounding pedestrians, and other contextual information. We formulate this problem as a Markov decision process [13]. The state space S can include the robot’s position, orientation, and the states of surrounding humans. The action space A may consist of different movement options. The transition probabilities P reflect the dynamics of the robot and its interactions with the environment, including the behavior of people around it. The reward function R is designed to encourage robot behaviors that are safe, efficient, and socially compliant.

In this section, we adopt the same notations for trajectories from Section IV. Given the historical positions of agent i (i.e., $\mathbf{x}_{1:T_h}^i$), we employ our method in Section IV to predict the future positions of the agent as $\hat{\mathbf{x}}_{T_h+1:T_h+T_f}^i$. We denote the robot’s states as \mathbf{w}_t , which includes current robot position $\mathbf{x}_{R,t} = (x_r^r, y_r^r)$, the goal position $\mathbf{g} = (g_x, g_y)$, the robot’s velocity $\mathbf{v}_t = (v_{x,t}, v_{y,t})$, the maximum robot speed v_{\max} , and the robot radius ρ . Furthermore, to enhance the realism and complexity of the environment, we also account for static obstacles as a global state, denoted as $\mathbf{O} = \{\mathbf{o}_k\}$. Each \mathbf{o}_k represents the state of static obstacle k , which includes both its position and radius. In this work, we only consider circular static obstacles while more complex shapes are left for future work. We define the state $s_t \in \mathcal{S}$ of the MDP as

$$s_t = [\mathbf{w}_t, \mathbf{x}_{T_h}, \hat{\mathbf{x}}_{T_h+1:T_h+T_f}, \mathbf{O}]. \quad (25)$$

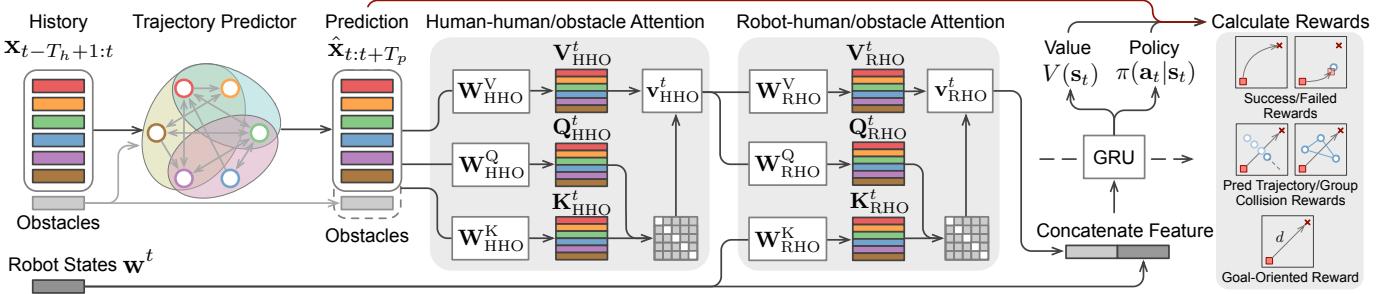


Fig. 4. A diagram of the social robot navigation framework, which consists of a trajectory predictor, a human-human/obstacle attention layer, a robot-human/obstacle attention layer, a value network, and a policy network. Best viewed in color.

The objective of social robot navigation is to learn an optimal robot policy $\pi(a_t | s_t)$, which navigates the robot to its destination efficiently within a certain time horizon without colliding with surrounding pedestrians or intruding into human personal spaces.

A. Overview

Fig. 4 illustrates the overall pipeline of our robot navigation method, which integrates deep reinforcement learning with multi-agent relational reasoning and trajectory prediction. The pipeline consists of our trajectory prediction model, two attention modules (i.e., human-human/obstacle attention and robot-human/obstacle attention) for feature aggregation adapted from CrowdNav++ [13], and a recurrent layer to output value and policy. Specifically, the trajectory predictor processes the historical trajectories of observable agents to generate their future state hypotheses. These predictions are first transformed by a human-human/obstacle attention module into a human-centric embedding. Then, these embeddings, along with the current robot state, are processed by the robot-human attention module, resulting in a refined feature representation of human agents considering human-robot interactions. This enhanced representation, concatenated with the current robot state, is then fed into a recurrent layer, which outputs the policy $\pi(a_t | s_t)$ and the value $V(s_t)$.

B. Attention Modules

Attention layers in our model play an important role by assigning weights to each edge connecting to an agent, thereby emphasizing important interactions. The human-human/obstacle and robot-human/obstacle attention layers employ the scaled dot-product attention technique.

In the human-human/obstacle attention layer, the current and predicted future states of humans are concatenated and passed through linear layers to derive the query, key, and value:

$$(\mathbf{Q}_{\text{HHO}}, \mathbf{K}_{\text{HHO}}, \mathbf{V}_{\text{HHO}}) = (W_{\text{HHO}}^Q \tilde{\mathbf{x}}, W_{\text{HHO}}^K \tilde{\mathbf{x}}, W_{\text{HHO}}^V \tilde{\mathbf{x}}), \quad (26)$$

where $\tilde{\mathbf{x}} = [\mathbf{x}_{T_h}, \hat{\mathbf{x}}_{T_h+1:T_h+T_f}]$ represents the concatenated state for all agents. The attention weights are obtained from $\mathbf{Q}_{\text{HHO}} \mathbf{K}_{\text{HHO}}^\top$, and the human embeddings at the current time step $\mathbf{v}_{\text{HHO}}^t$ are derived using a multi-head dot-product attention mechanism with eight attention heads. Note that the attention weights between static obstacles are set to 0 since they do not influence each other.

In the robot-human/obstacle attention layer, the human embeddings obtained from the human-human/obstacle attention layer serve as inputs for deriving the query and value:

$$(\mathbf{Q}_{\text{RHO}}, \mathbf{V}_{\text{RHO}}) = (W_{\text{RHO}}^Q \mathbf{v}_{\text{HHO}}^t, W_{\text{RHO}}^V \tilde{\mathbf{x}} \mathbf{v}_{\text{HHO}}^t), \quad (27)$$

while the key is computed using a linear embedding of the robot state:

$$\mathbf{K}_{\text{RHO}} = W_{\text{RHO}}^K \mathbf{w}^t. \quad (28)$$

The attention weights are calculated using $\mathbf{Q}_{\text{RHO}} \mathbf{K}_{\text{RHO}}^\top$, and the human embeddings at the current time after the second-layer attention $\mathbf{v}_{\text{RHO}}^t$ are obtained by applying a single-head dot-product attention mechanism.

C. Policy and Value Estimation

After obtaining the human embeddings, we concatenate them with the robot embedding $h_R^t = \text{Linear}(\mathbf{w}^t)$, which are fed into a GRU layer [61]:

$$h^t = \text{GRU}(h^{t-1}, ([\mathbf{v}_{\text{RHO}}^t, h_R^t])),$$

where h^t is the hidden state of the GRU at time t . The hidden state h^t of the GRU is fed into a fully connected layer to obtain the value of the state $V(s^t)$ and the policy $\pi(a_t | s_t)$.

D. Reward Function

The design of the reward function is crucial for social navigation based on reinforcement learning. The primary objective of the robot is to reach a specified goal while adhering to social norms and avoiding collisions with individual pedestrians and groups. The components of the reward function are as follows:

Goal-oriented rewards. To guide the robot toward its goal, we define a potential-based reward r_{goal} , which is a function of the L_2 distance between the robot's position $\mathbf{x}_{R,t}$ and the goal position \mathbf{g} at time t . Mathematically, it is written as

$$r_{\text{goal}}(t) = -\alpha \cdot \|\mathbf{x}_{R,t} - \mathbf{g}\|_2, \quad (29)$$

where α is a scaling factor. In addition, we introduce r_{success} , a reward given when the robot successfully reaches the goal.

Human collision and prediction rewards. To encourage safe navigation, we penalize collisions and potential intrusions into the space of humans. The prediction reward $r_{\text{pred_h}}$ penalizes the robot based on its proximity to the predicted future positions of humans, which is written as

$$r_{\text{pred_h}} = \beta_1 \cdot \min_{\mathbf{p}_{\text{agent}} \in \mathbf{P}_{\text{pred}}} (\|\mathbf{x}_{R,t} - \mathbf{p}_{\text{agent}}\|_2), \quad (30)$$

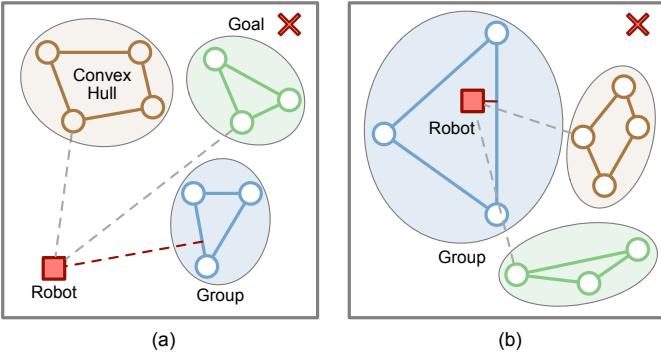


Fig. 5. An illustrative visualization of two typical scenarios where the robot navigates in human crowds. (a) The robot is located outside the human groups' convex hulls, which complies with social norms. (b) The robot is located inside a human group's convex hull, which intrudes into human personal spaces and violates social norms. Best viewed in color.

where β_1 is a scaling factor, and \mathbf{P}_{pred} represents the set of predicted positions of agents. The collision reward $r_{\text{collision_h}}$ is given when the robot collides with any agents, which heavily penalizes such events.

Obstacle collision reward. Similar to the human collision and prediction rewards, we have a distance reward $r_{\text{dist_o}}$ that penalizes the robot based on its proximity to the nearest obstacles, as expressed by:

$$r_{\text{dist_o}} = \beta_2 \cdot \min_{\mathbf{o}_k \in \mathbf{O}} (\|\mathbf{x}_{R,t} - \mathbf{o}_k\|_2), \quad (31)$$

where β_2 is a scaling factor. The collision reward $r_{\text{coll_o}}$ is applied when the robot collides with any obstacle, imposing a heavy penalty for such events. We set lower penalty weights for $r_{\text{dist_o}}$ and $r_{\text{coll_o}}$ compared to those associated with humans to emphasize safe navigation around humans.

Group interaction reward. To minimize discomfort to individuals and discourage intrusion into human groups, we consider the geometry of group formations. Fig. 5 illustrates the scenarios where the robot complies with or violates the social norms. For each group, we calculate a convex hull encompassing all agents within the group. For example, with a group of pedestrian $G\{\mathbf{x}^i\}, i = 1, \dots, m$, where m is the number of pedestrians in this group, we define the convex hull H of this group as

$$H(G) = \bigcap \{C \text{ is a convex polygon, } G \subseteq C\},$$

i.e., the convex hull is the intersection of all convex sets that contain G . We employ the Qhull algorithm [62] to construct these convex hulls by initially selecting the extreme points based on x -coordinates and dividing the points into two subsets. Using a recursive divide-and-conquer approach, it identifies the farthest point from each line segment, refining the subsets and expanding the convex boundary until no external points remain. This method can efficiently delineate the smallest convex polygon enclosing all points.

The reward r_{group} is defined as a function of the robot's distance to the closest point on the group's convex hull, normalized by the area of the convex hull, which is written as

$$r_{\text{group}} = \gamma \cdot \frac{c \|\mathbf{x}_{R,t} - H(\tilde{G})\|_2}{\text{Area}(H(\tilde{G}))}, \quad \tilde{G} = \operatorname{argmin}_G \|\mathbf{x}_{R,t} - H(\tilde{G})\|_2, \quad (32)$$

here the $\|\mathbf{x}_{R,t} - H(\tilde{G})\|_2$ is the shortest distance from the robot to the convex hull, γ is a scaling factor. c is a condition variable, which is set to 1 when the robot is located outside the convex hull. Conversely, it is set to -1 to incentivize the robot to move out from the group with the shortest path.

Overall reward. The overall reward function $r(s_t, a_t)$ at any time step t is defined as follows:

$$r(s_t, a_t) = \begin{cases} r_{\text{success}}, & \text{if } s_t \in S_{\text{goal}}, \\ r_{\text{coll_h}}, & \text{if } s_t \in S_{\text{fail_h}}, \\ r_{\text{coll_o}}, & \text{if } s_t \in S_{\text{fail_o}}, \\ r_{\text{group}} + r_{\text{pred_h}} + r_{\text{dist_o}} + r_{\text{goal}}, & \text{otherwise,} \end{cases} \quad (33)$$

where S_{goal} are the states of the robot reaching the goal, $S_{\text{fail_h}}$ are the states of robot collision with a human, and $S_{\text{fail_o}}$ are the states of robot collision with a static obstacle.

This reward design ensures that the robot receives a high reward when reaching the goal while also incorporating other critical aspects of social navigation, including collision avoidance and adherence to social norms during navigation.

E. Training Strategy and Loss Function

We train the trajectory prediction model and the reinforcement learning policy sequentially for the following reasons:

- 1) *Distinct Objectives:* The trajectory prediction model and the RL policy serve different purposes. The former is focused on accurately predicting human trajectories, which mainly considers pairwise and group-wise human-human interactions. On the other hand, the RL policy aims to optimize safe and efficient robot navigation behavior, which mainly considers robot-human interactions. An effective prediction model serves as a prerequisite for high-quality decision making for the robot.
- 2) *Stability and Efficiency:* Joint training of both components can lead to instability and inefficiency due to the differing nature of the learning tasks, especially during the initial stage when the prediction model generates unreasonable future trajectories. Separating the training processes allows each component to learn more effectively from its specific objective function.

To learn the RL policy, we employ Proximal Policy Optimization (PPO) [63], a robust and efficient model-free policy gradient algorithm. To enhance training stability and speed, we deploy 16 parallel environments, allowing the collection of diverse experiences. This parallelization facilitates efficient exploration of the state space. During policy updates, we use experiences collected over 30 steps from six episodes, providing a rich dataset for each update iteration. The detailed training procedures are summarized in Algorithm 1.

VI. EXPERIMENTS

In this section, we validate the proposed approaches on benchmark datasets for trajectory prediction and a simulation for social robot navigation. The empirical results are analyzed and compared with state-of-the-art baselines.

Algorithm 1 Social Robot Navigation (Training Phase)

Require: initial parameters of the reinforcement learning policy network θ_0 , clipping threshold $\epsilon = 0.2$

- 1: **for** $k = 1, 2, \dots, K$ **do**
- 2: // Collect a trajectory of the robot by iteratively processing the action
- 3: **for** $t = 1, 2, \dots, T$ **do**
- 4: Predict future trajectories of pedestrians $\hat{x}_{t+1:t+T_f}$ based on historical observations $\mathbf{x}_{t-T_h+1:t}$.
- 5: Estimate the value of the state $V(s_t)$ and the policy $\pi(a_t | s_t)$ with the current robot state, predicted pedestrians' trajectories, and static obstacles' states.
- 6: Select an action using policy π_k and obtain the next state and the reward from the environment.
- 7: **end for**
- 8: // Update the learnable networks.
- 9: Compute the rewards $r(s_t, a_t)$.
- 10: Estimate advantages \hat{A}_t .
- 11: Iteratively update the policy by optimizing a surrogate objective function with PPO. Update the policy network by maximizing the PPO objective function via stochastic gradient ascent:
$$\theta_{k+1} = \operatorname{argmax}_{\theta} \mathbb{E} \left[\min \left(\frac{\pi_{\theta}(a_t | s_t)}{\pi_{\theta_k}(a_t | s_t)} \hat{A}_t, \right. \right. \\ \left. \left. \text{clip} \left(\frac{\pi_{\theta}(a_t | s_t)}{\pi_{\theta_k}(a_t | s_t)}, 1 - \epsilon, 1 + \epsilon \right) \hat{A}_t \right) \right],$$
- where θ represents the policy parameters, \hat{A}_t is an estimator of the advantage function at time t , and $\epsilon = 0.2$ is a hyperparameter denoting the clipping range.
- 12: **end for**

A. Benchmark Datasets and Crowd Simulation

1) *Human Trajectory Datasets*: We evaluated human trajectory prediction and relational reasoning on three widely used real-world datasets and a synthetic human trajectory dataset.

ETH-UCY [64], [65]: These datasets were collected in real-world traffic scenarios where pedestrians move on the street or the university campus. Some scenarios are crowded and full of pairwise and group-wise interactions between pedestrians who naturally form groups and move together. We used the standard split of training, validation, and testing datasets. Moreover, we also selected the test cases with more than eight pedestrians to construct a difficult test subset to compare the model performance in dense scenarios with strong interactions.

SDD [66]: This is a real-world pedestrian dataset captured on the university campus with top-down view videos with annotations on the positions of pedestrians. Most scenarios are crowded with interacting pedestrians and group activity. We used the standard split for training, validation, and testing.

NBA¹: Although we focus on pedestrian trajectory prediction in this work, we also use a basketball player dataset to demonstrate the effectiveness of dynamic relational reasoning since the players have very complex interactions during the

games. This dataset provides recordings of real NBA games that offer precise trajectory information of basketball players and the ball, which show complex team sports dynamics and feature high-intensity interactions among basketball players. The dataset is particularly suited for studying relational reasoning and group behavior dynamics in sports, highlighting the contrasting behaviors of cooperative team members and adversarial opponents.

These datasets enable a comprehensive evaluation of the model performance across various contexts. The synthetic dataset is introduced below.

2) *Group-Aware Crowd Simulation*: We evaluate social robot navigation with a crowd simulation as an interactive environment for reinforcement learning, which is also used to collect a synthetic dataset. We develop a group-based crowd simulator that incorporates the ORCA model [67] to enable the simulation of crowd dynamics with human groups. More specifically, we design a specific scenario to observe changes in group behaviors. The simulation environment is a 2D square space ($12 \text{ m} \times 12 \text{ m}$), where we randomly initialize five groups of pedestrians. For both training and evaluation, each group contains 2-5 individuals randomly, i.e., in total 10-25 pedestrians. For each group, all members will be positioned within a circular area. Different groups do not have overlapping initial spaces. Every group is assigned a random destination along the neighboring or the opposite boundary and moves at a random maximum speed (v_{\max}) ranging from 0.5 m/s to 1.5 m/s. Each pedestrian is modeled with a radius (ρ) between 0.3 m and 0.5 m. In the square, a random number of static obstacles, ranging from one to the number of groups, will be randomly positioned within a circular area. The center of this area coincides with the center of the square, and it has a radius of 6 m, which is an inscribed circle. Each static obstacle is modeled with a radius between 0.6 m and 1.0 m.

For the robot simulation, during each episode, its starting and goal positions are randomly determined on the plane, ensuring a minimum distance of 10 m to avoid simple cases. The robot is equipped with a limited circular sensor range of 5 m, reflecting a realistic hardware constraint. We assume the robot can move continuously at a maximum speed of 1 m/s and can instantly achieve its desired velocity. To train and evaluate the trajectory prediction model, we generated 5k cases for training, 2k cases for validation, and 2k cases for testing.

B. Evaluation Metrics

1) *Trajectory Prediction*: To evaluate the trajectory prediction performance on both synthetic and benchmark datasets, we adopted two standard evaluation metrics [5]:

- **minADE₂₀**: the minimum distance among 20 trajectory samples between the predicted trajectories and the ground truth over all the entities in the prediction horizon.
- **minFDE₂₀**: the minimum deviated distance among 20 trajectory samples at the last predicted time step.

2) *Social Robot Navigation*: To assess the performance of social robot navigation, we adopt a comprehensive set of evaluation metrics [36], which are designed to evaluate various

¹<https://github.com/linouk23/NBA-Player-Movements>

aspects of social navigation, including safety, efficiency, and social appropriateness:

- **Success Rate (SR)**: SR indicates the robot’s capability to complete its navigation task within a certain period without collisions.
- **Collision Rate (CR)**: CR indicates navigation safety by quantifying how frequently it collides with humans.
- **Timeout Rate (TR)**: TR tracks the proportion of navigation tasks where the robot fails to reach its destination within a certain period without collisions, which indicates navigation efficiency.
- **Navigation Time (NT)**: NT shows the average duration, in seconds, for the robot to complete a navigation task successfully, which indicates navigation efficiency.
- **Path Length (PL)**: PL shows the length of the robot’s path from its initial position to the goal, which indicates the navigation efficiency.
- **Intrusion Time Ratio (ITR)**: ITR is defined as c/C where c is the number of timesteps of the robot intrudes into any human’s future position from $t + 1$ to $t + 5$, and C is the length of that episode. The ITR assesses the robot’s respect for socially sensitive spaces.
- **Social Distance (SD)**: SD is defined as the average distance between the robot and its closest human when an instruction to human spaces occurs, which gauges the robot’s ability to maintain appropriate distances from individuals or groups, reflecting its social awareness.
- **Group Intersections (GI)**: GI shows the ratio of scenarios where the robot intrudes into human groups, indicating the social appropriateness of robot navigation.

C. Baselines

1) *Trajectory Prediction*: We compare our approach with a series of state-of-the-art baselines:

- **MID** [14]: This is a stochastic trajectory prediction framework with motion indeterminacy diffusion, which contains a transformer-based architecture to capture the temporal dependencies in trajectories.
- **GroupNet** [12]: This trajectory prediction model learns to capture multi-scale interaction patterns based on graph representations and infers interaction strengths.
- **Graph-TERN** [15]: This is a graph-based pedestrian trajectory estimation and refinement network, which consists of a control point prediction module, a trajectory refinement module, and a multi-relational graph convolution.
- **EigenTrajectory** [16]: This is a state-of-the-art trajectory prediction approach that uses a trajectory descriptor to form a compact space in place of Euclidean space for representing pedestrian movements.

2) *Social Robot Navigation*: To demonstrate the effectiveness of dynamic relational reasoning in social robot navigation, we compare our method with both traditional and state-of-the-art baselines:

- **ORCA** [67]: This is a rule-based approach to reciprocal collision avoidance, where multiple independent agents avoid collisions with each other without communication among agents while moving in a common workspace.

- **DSOA** [68]: This is a closed-form solution for dynamic spatial navigation, which creates smooth motion fields to ensure the robot navigates safely within confined spaces and around both static and dynamic obstacles.
- **RL (no prediction)**: This is a standard reinforcement learning method without predicting the future behavior of surrounding humans. The observations only include past and current information.
- **RL (constant velocity)**: This is a reinforcement learning method with constant velocity prediction of human trajectories. While considering the future information, this method does not model the complex interactions.
- **Group-Aware** [42]: This model employs deep reinforcement learning to develop navigation policies that recognize and respect dynamic human group formations, which enhances robot navigation by reducing collisions and adhering to social norms.
- **CrowdNav++** [13]: This is a state-of-the-art social robot navigation method based on reinforcement learning. It infers the intentions of moving pedestrians, which prevents the robot from intruding into the humans’ intended paths.

D. Implementation Details

1) *Trajectory Prediction*: We use a batch size of 32 and an initial learning rate of 0.001. The model is trained for 10k epochs with an Adam optimizer and a step scheduler with a decay rate of 0.85. We set the weight of SM, SH, and SP to 0.001. The details of the model components are below:

- *Agent node embedding function*: a three-layer MLP with hidden size = 128.
- *Inferring hypergraph topology function*: a three-layer MLP with hidden size = 128.
- *Edge feature updating function*: a distinct three-layer MLP with hidden size = 128 for each type of edge or hyperedge.
- *Encoding*: a three-layer MLP with hidden size = 128.
- *Decoding*: a three-layer MLP with hidden size = 128.
- *Recurrent graph / hypergraph evolution module*: a two-layer GRU with hidden size = 128.

Specific experimental details of different datasets:

- *Synthetic*: 3 edge / hyperedge types, maximum hyperedge = 5, prediction period = 1, encoding horizon = 4.
- *ETH-UCY*: 3 edge / hyperedge types, maximum hyperedge = 5, prediction period = 4, encoding horizon = 8.
- *SDD*: 5 edge / hyperedge types, maximum hyperedge = 8, prediction period = 4, encoding horizon = 8.
- *NBA*: 3 edge / hyperedge types, maximum hyperedge = 5, prediction period = 5, encoding horizon = 5.

2) *Social Robot Navigation*: To simulate the behaviors of the robot and pedestrians, we employ holonomic kinematics. The action space for each agent, including the robot, at time t is represented by their desired velocities along the x and y axes, denoted as $a_t = [v_{x,t}, v_{y,t}]$. This action space is continuous, with the robot’s maximum speed at 1 m/s. Humans are controlled by ORCA [67] and interact with surrounding humans and obstacles. We operate under the assumption that all agents can instantaneously achieve their desired velocities

TABLE I
MINADE₂₀ / MINFDE₂₀ (METERS) ON THE ETH / UCY DATASETS (OVERALL).

Dataset	Baseline Methods				Ours									
	MID [14]	GroupNet [12]	Graph-TERN [15]	EigenTraj [16]	SCG	SHG	SCG + SHG	DCG + DHG	DCG + DHG + SM	DCG + DHG + SM + SH	DCG + DHG + SM + SH + SP (Equal Attention)	DCG + DHG + SM + SH + SP	DCG + DHG + SM + SH + SP	
ETH	0.39 / 0.66	0.46 / 0.73	0.42 / 0.58	0.36 / 0.57	0.41 / 0.68	0.45 / 0.76	0.40 / 0.66	0.37 / 0.63	0.35 / 0.62	0.33 / 0.59	0.34 / 0.60	0.33 / 0.54		
HOTEL	0.13 / 0.22	0.15 / 0.25	0.14 / 0.23	0.13 / 0.21	0.16 / 0.27	0.19 / 0.28	0.16 / 0.24	0.15 / 0.23	0.13 / 0.21	0.13 / 0.18	0.14 / 0.21	0.12 / 0.17		
UNIV	0.22 / 0.45	0.26 / 0.49	0.26 / 0.45	0.24 / 0.43	0.24 / 0.49	0.28 / 0.51	0.22 / 0.47	0.21 / 0.45	0.20 / 0.43	0.18 / 0.41	0.19 / 0.40	0.18 / 0.36		
ZARA1	0.17 / 0.30	0.21 / 0.39	0.21 / 0.37	0.20 / 0.35	0.23 / 0.36	0.25 / 0.40	0.22 / 0.35	0.21 / 0.32	0.20 / 0.31	0.18 / 0.30	0.19 / 0.29	0.17 / 0.29		
ZARA2	0.13 / 0.27	0.17 / 0.33	0.17 / 0.29	0.14 / 0.25	0.21 / 0.30	0.23 / 0.31	0.20 / 0.27	0.18 / 0.24	0.17 / 0.20	0.15 / 0.21	0.16 / 0.23	0.13 / 0.20		
AVG	0.21 / 0.38	0.25 / 0.44	0.24 / 0.38	0.21 / 0.36	0.24 / 0.45	0.26 / 0.47	0.23 / 0.42	0.21 / 0.39	0.20 / 0.35	0.19 / 0.32	0.21 / 0.36	0.18 / 0.30		

TABLE II
MINADE₂₀ / MINFDE₂₀ (METERS) ON THE ETH / UCY DATASETS (DIFFICULT SUBSET).

Dataset	Baseline Methods				Ours									
	MID [14]	GroupNet [12]	Graph-TERN [15]	EigenTraj [16]	SCG	SHG	SCG + SHG	DCG + DHG	DCG + DHG + SM	DCG + DHG + SM + SH	DCG + DHG + SM + SH + SP (Equal Attention)	DCG + DHG + SM + SH + SP	DCG + DHG + SM + SH + SP	
ETH	0.42 / 0.77	0.49 / 0.89	0.52 / 0.84	0.40 / 0.73	0.45 / 0.79	0.54 / 0.96	0.43 / 0.72	0.41 / 0.71	0.38 / 0.65	0.36 / 0.63	0.36 / 0.65	0.35 / 0.61		
HOTEL	0.15 / 0.25	0.18 / 0.30	0.17 / 0.28	0.14 / 0.24	0.20 / 0.31	0.24 / 0.35	0.17 / 0.29	0.17 / 0.26	0.15 / 0.24	0.14 / 0.23	0.15 / 0.25	0.14 / 0.20		
UNIV	0.23 / 0.55	0.27 / 0.58	0.29 / 0.61	0.26 / 0.54	0.28 / 0.55	0.33 / 0.69	0.26 / 0.54	0.24 / 0.51	0.23 / 0.48	0.22 / 0.46	0.22 / 0.49	0.21 / 0.45		
ZARA1	0.21 / 0.38	0.26 / 0.45	0.23 / 0.42	0.22 / 0.37	0.25 / 0.41	0.29 / 0.52	0.23 / 0.38	0.23 / 0.36	0.21 / 0.35	0.20 / 0.33	0.21 / 0.33	0.19 / 0.31		
ZARA2	0.20 / 0.34	0.21 / 0.36	0.21 / 0.32	0.18 / 0.28	0.26 / 0.32	0.28 / 0.41	0.23 / 0.29	0.21 / 0.28	0.19 / 0.25	0.18 / 0.24	0.19 / 0.26	0.16 / 0.23		
AVG	0.31 / 0.50	0.29 / 0.49	0.27 / 0.47	0.25 / 0.44	0.30 / 0.56	0.34 / 0.65	0.28 / 0.50	0.25 / 0.45	0.23 / 0.41	0.22 / 0.39	0.24 / 0.43	0.20 / 0.34		

TABLE III
MINADE₂₀ / MINFDE₂₀ ON THE SDD (PIXELS) AND NBA (METERS) DATASETS.

Dataset	Baseline Methods				Ours									
	MID [14]	GroupNet [12]	Graph-TERN [15]	EigenTraj [16]	SCG	SHG	SCG + SHG	DCG + DHG	DCG + DHG + SM	DCG + DHG + SM + SH	DCG + DHG + SM + SH + SP (Equal Attention)	DCG + DHG + SM + SH + SP	DCG + DHG + SM + SH + SP	
SDD	9.7 / 15.3	9.7 / 15.3	8.4 / 14.3	8.1 / 13.1	12.7 / 22.5	14.0 / 25.8	12.0 / 20.4	9.5 / 15.4	8.8 / 14.7	8.4 / 13.6	9.4 / 14.9	8.1 / 12.8		
NBA	1.65 / 2.66	1.53 / 2.45	1.46 / 2.19	1.49 / 2.30	1.69 / 2.73	1.80 / 3.04	1.56 / 2.32	1.37 / 2.00	1.28 / 1.81	1.23 / 1.73	1.31 / 1.87	1.16 / 1.64		

and maintain these velocities for a duration of Δt seconds. The position of an agent at any time is updated based on the velocities according to the update rule, which ensures realistic motion patterns for the robot and human agents.

The RL methods under different settings were trained with distinct reward functions according to their predictive capabilities. Specifically, RL methods with trajectory prediction use the reward function defined in Eq. (33) while those without trajectory prediction omit the r_{pred} component. We trained the model for 2×10^7 steps with an initial learning rate of 4×10^{-5} . We used 500 random, unseen test cases for evaluation. The intrusions into human personal spaces are determined based on the true future positions of pedestrians.

3) *Hardware*: All the experiments were run on an Ubuntu server equipped with an AMD Ryzen Threadripper® 3960X 48-Core CPU and an NVIDIA RTX® A6000 GPU.

E. Dynamic Relational Reasoning for Trajectory Prediction

1) *Quantitative Analysis*: The comprehensive comparisons of prediction results are provided in Table I-III. Note that our approach achieves state-of-the-art performance across all datasets consistently while the strongest baseline methods for each dataset are different.

a) *ETH-UCY and SDD datasets*: We adopted the standard setting for training and testing our model to predict trajectories in the future 4.8 seconds (12 time steps) based on the historical observations of 3.2 seconds (8 time steps). Our approach reduces minADE₂₀ / minFDE₂₀ by 14.3% / 16.7% and 20.0% / 22.7% on the ETH-UCY (overall) and ETH-UCY (difficult) datasets compared to the strongest baseline, respectively. Particularly, our method leads to a larger improvement for the difficult subset, indicating its advantage in modeling complex social interactions in dense scenarios. Our approach also achieves state-of-the-art performance on the SDD dataset.

b) *NBA dataset*: We predicted the trajectories in the future 4.0 seconds (10 time steps) based on the historical observations of 2.0 seconds (5 time steps). Our approach reduces minADE₂₀ / minFDE₂₀ by 20.5% / 25.1% on the NBA dataset compared to the strongest baseline.

2) *Ablative Analysis*: We conducted a comprehensive ablation study on the synthetic and benchmark datasets to demonstrate the effectiveness of model components in our approach from multiple aspects. We presented quantitative results of ablation settings along with our complete model in Fig. 6(a)(b) and Table I-III. The ablation settings are:

- SCG: This variant only enables *pairwise* static relational

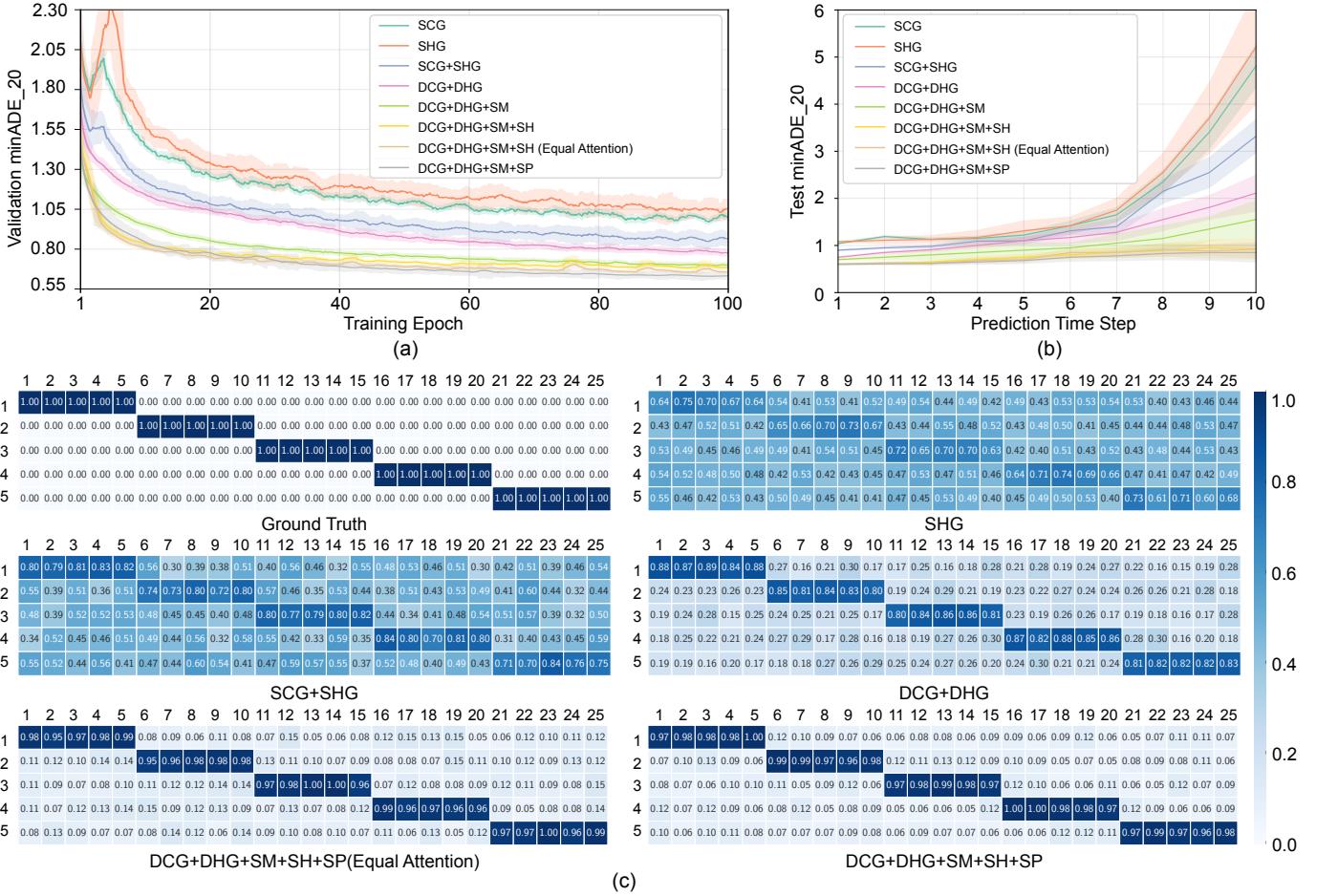


Fig. 6. The visualizations of results on the crowd simulation dataset. (a) minADE₂₀ vs. training epochs on the validation dataset. The solid lines indicate the average error and the shaded areas indicate variance. (b) minADE₂₀ vs. prediction horizon on the testing dataset. (c) The incidence matrix of the hypergraphs inferred by different model settings. Each row represents a hyperedge and each column represents a certain agent. The numbers indicate the probability of the agent belonging to the corresponding hyperedge (i.e., group). The hyperedges of SHG and SCG+SHG shown in the figure are inferred at the first prediction step since they only infer static relations while the others are inferred at a step after the group evolution. Best viewed in color.

reasoning and infers a static interaction graph.

- SHG: This variant only enables *group-wise* static relational reasoning and infers a static interaction hypergraph.
- SCG+SHG: This variant enables both *pairwise* and *group-wise* static relational reasoning.
- DCG+DHG: This variant enables both pairwise and group-wise *dynamic* relational reasoning.
- DCG+DHG+SM: This variant applies a *smoothness* regularization loss to the DCG+DHG model.
- DCG+DHG+SM+SH: This variant applies a *sharpness* regularization loss to the DCG+DHG+SM model.
- DCG+DHG+SM+SH+SP (Equal Attention): This variant is our method without the hypergraph attention.
- DCG+DHG+SM+SH+SP: This is our complete method, which additionally applies a *sparsity* regularization loss to the DCG+DHG+SM+SH model.

a) *Group-wise relational reasoning*: We show the effectiveness of group-wise relational reasoning by comparing the SCG, SHG, and SCG+SHG models. In all the experiments, SHG performs worse than SCG, which implies that group-wise relational reasoning through hypergraphs alone cannot

sufficiently model the interactions between agents. This is because pairwise interactions still dominate the multi-agent dynamics of the agents in either the same or different groups, which cannot be entirely ignored. Meanwhile, effective pairwise reasoning may help identify group behavior patterns. Among these model variants, SCG+SHG achieves the best performance, which implies the effectiveness of integrating pairwise and group-wise relational reasoning. SCG+SHG reduces minADE₂₀ by 4.2%, 6.7%, 5.8%, and 7.8% on the ETH-UCY (overall), ETH-UCY (difficult), SDD, and NBA datasets, respectively. As shown in Fig. 6(a) and 6(b), SCG+SHG also leads to a significant improvement in crowd simulation and the gap becomes larger as the prediction horizon increases.

b) *Dynamic Relational Reasoning*: We show the effectiveness of dynamic relational reasoning by comparing the SCG+SHG and DCG+DHG models. DCG+DHG performs better in all the experiments. In the crowd simulation where the group evolves significantly, the dynamic relational reasoning mechanism enhances model performance. Since SCG+SHG cannot predict the evolution of hyperedges, the model must infer an overall group topology for the whole prediction, which limits the flexibility of reasoning. DCG+DHG reduces

TABLE IV
MINADE₂₀ (METERS) VS. MAXIMUM NUMBER OF HYPEREDGES ON THE NBA DATASET.

Number	2	5	11	20	30	40	110
minADE ₂₀	1.22 ± 0.01	1.16 ± 0.02	1.28 ± 0.04	1.29 ± 0.03	1.31 ± 0.04	1.32 ± 0.05	1.34 ± 0.07

TABLE V
MINADE₂₀ (METERS) VS. NUMBER OF EDGE AND HYPEREDGE TYPES ON THE NBA DATASET.

Number	1	2	3	4	5	6	7
minADE ₂₀	1.27 ± 0.01	1.23 ± 0.02	1.16 ± 0.02	1.19 ± 0.02	1.23 ± 0.04	1.27 ± 0.05	1.29 ± 0.04

minADE₂₀ by 8.7%, 10.7%, 20.8%, and 12.2% on ETH-UCY (overall), ETH-UCY (difficult), SDD, and NBA datasets.

c) *Smoothness regularization:* We show the effectiveness of smoothness regularization by comparing the DCG+DHG and DCG+DHG+SM models. Fig. 6(a) and 6(b) show that DCG+DHG+SM achieves consistently better and more stable performance. The reason is that the smoothness regularization encourages smoother evolution of graphs and hypergraphs over time and restrains abrupt changes, which leads to a more stable training process and faster convergence. It also encourages the model to learn the temporal dependency between consecutively evolving relations. Applying the smoothness regularization loss during training reduces minADE₂₀ by 4.8%, 8.0%, 7.4%, and 6.6% on the ETH-UCY (overall), ETH-UCY (difficult), SDD, and NBA datasets, respectively.

d) *Sharpness regularization:* We show the effectiveness of the sharpness regularization by comparing DCG+DHG+SM and DCG+DHG+SM+SH models. Fig. 6(a) illustrates that the sharpness regularization further enhances the prediction performance in crowd simulation. Fig. 6(b) shows that DCG+DHG+SM+SH has a smaller variance in minADE₂₀ indicating that sharpness regularization also improves the model robustness to random initializations. Applying the sharpness regularization during training reduces minADE₂₀ by 5.0%, 4.3%, 4.5%, and 3.9% on the ETH-UCY (overall), ETH-UCY (difficult), SDD, and NBA datasets, respectively.

e) *Sparsity regularization:* DCG+DHG+SM+SH+SP has the smallest prediction error among all the model settings consistently. DCG+DHG+SM+SH+SP tends to infer sparser hypergraphs which learn more distinguishable group memberships for each agent. The model is encouraged to learn to recognize sparse pairwise relations and clear group assignments, which creates a strong inductive bias in the message passing process through the sparse relational structures. Applying sparsity regularization during training reduces minADE₂₀ by 5.3%, 9.1%, 3.6%, and 5.7% on the ETH-UCY (overall), ETH-UCY (difficult), SDD, and NBA datasets, respectively.

f) *Hyperedge numbers:* In Table IV, we compare the model performance with different hyperedge (i.e., group) numbers on the NBA dataset. It shows that setting 5 as the maximum number of hyperedges achieves the best performance based on a hyperparameter search. The model may not capture sufficient group-wise relations with only 2 hyperedges. At the same time, too many hyperedges (e.g., 11 and 110) may lead to overly dense group-wise relations that cannot capture clear group behaviors. In Table I-III, the reported numbers of

SHG show the best performance after doing a hyperparameter search. Based on the experimental results, we find that using a very large number of hyperedges in SHG leads to serious overfitting. Moreover, the standard deviation of the prediction error increases as the maximum hyperedge number increases because a larger number of hyperedges may lead to more local optima, which enlarges the randomness of the model performance. These observations imply that although pairwise relations are a special type of group-wise relations where only two agents are involved, it is not sufficient to just use a large number of hyperedges to cover all the relations, which motivates our design of the integrated pairwise and group-wise relational reasoning.

g) *Edge and hyperedge types:* In Table V, we compare the model performance with different numbers of edge and hyperedge types (i.e., relation types) on the NBA dataset. It shows that setting 3 as the edge and hyperedge type number achieves the best performance based on a hyperparameter search. The model may not capture a sufficient variety of pairwise or group-wise relations with only 2 edge and hyperedge types while too many types (e.g., 6) may lead to overfitting. We can see in Table V that the standard deviation of the prediction error increases as the type number increases because a larger variety of relations may lead to more local optima, which enlarges the randomness of the model performance.

h) *Attention mechanisms:* To validate the effectiveness of the attention mechanisms in Eq. (5) and (10), we replace the attention weights in DCG+DHG+SM+SH+SP with equal weights, which shows that adopting the attention mechanism reduces minADE₂₀ by 14.3%, 16.7%, 13.8%, and 11.5% on the ETH-UCY (overall), ETH-UCY (difficult), SDD, and NBA datasets, respectively. This implies the benefits of modeling different contributions of agents to group-wise relations and the influence of different groups on a certain agent.

3) *Qualitative Analysis:* We visualize the predicted trajectories and the group memberships inferred in the form of hyperedges for the crowd simulation, ETH-UCY, and NBA datasets in Fig. 6(c) and Fig. 7, respectively. For the crowd simulation, our approach infers accurate hyperedges with high confidence. First, SCG+SHG infers more clear group relations than SHG, which implies that explicitly learning pairwise relations enhances group identification. Second, SCG+SHG infers vague hyperedges while DCG+DHG can well capture the evolved group relations, indicating the benefits of dynamic reasoning. Third, with the proposed regularization mechanisms for relational learning, the model can infer group relations that

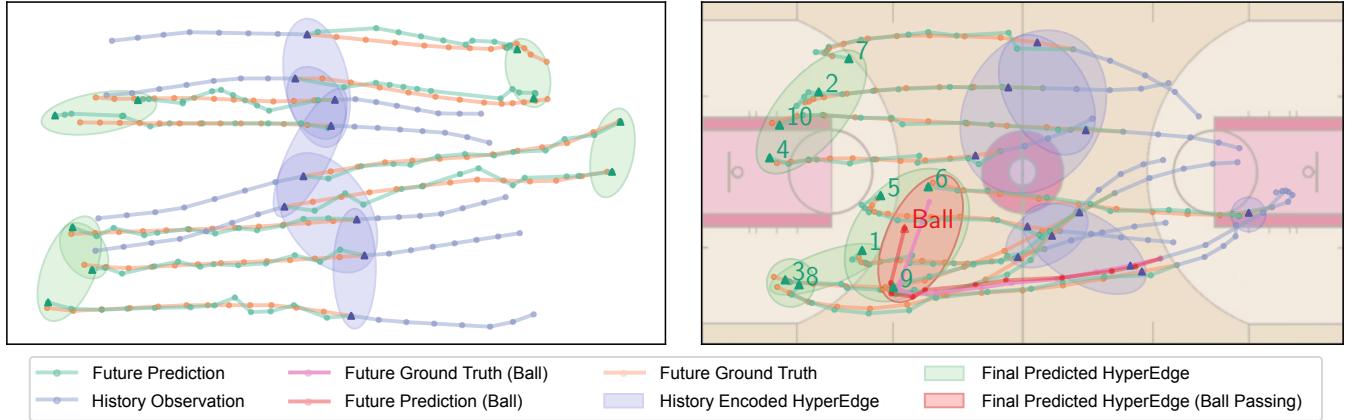


Fig. 7. The visualization of trajectory prediction and relational reasoning results on the ETH-UCY dataset (left) and NBA dataset (right). The purple ellipses represent the hyperedges inferred based on historical observations while the green ones represent the hyperedges inferred at the final prediction step after evolution. Our approach captures reasonable evolving patterns of human groups in both contexts. Best viewed in color.

TABLE VI
SOCIAL ROBOT NAVIGATION RESULTS

Method	SR↑	CR↓	TR↓	NT↓	PL↓	ITR↓	SD↑	GI↓
ORCA [67]	$80.77 \pm 1.93\%$	$18.59 \pm 1.82\%$	$0.64 \pm 0.10\%$	14.98 ± 0.09	20.71 ± 0.36	$18.28 \pm 0.82\%$	0.36 ± 0.02	$22.45 \pm 1.51\%$
DSOA [68]	$90.25 \pm 0.98\%$	$4.20 \pm 0.68\%$	$5.55 \pm 0.32\%$	16.02 ± 0.11	23.15 ± 0.42	$27.41 \pm 1.01\%$	0.18 ± 0.02	$38.41 \pm 2.12\%$
RL (no prediction)	$83.02 \pm 2.41\%$	$16.63 \pm 2.33\%$	$0.35 \pm 0.08\%$	16.39 ± 0.16	22.56 ± 0.21	$7.21 \pm 0.27\%$	0.32 ± 0.03	$9.91 \pm 0.20\%$
RL (constant velocity)	$84.11 \pm 2.55\%$	$15.63 \pm 1.31\%$	$0.26 \pm 0.05\%$	15.86 ± 0.18	21.71 ± 0.20	$4.01 \pm 0.11\%$	0.29 ± 0.02	$8.98 \pm 0.12\%$
Group-Aware [42]	$91.68 \pm 1.54\%$	$7.93 \pm 1.45\%$	$0.39 \pm 0.06\%$	15.27 ± 0.11	21.08 ± 0.19	$3.42 \pm 0.23\%$	0.33 ± 0.03	$8.79 \pm 0.21\%$
CrowdNav++ [13]	$93.21 \pm 1.31\%$	$6.66 \pm 1.30\%$	$0.13 \pm 0.03\%$	14.81 ± 0.07	20.37 ± 0.15	$3.11 \pm 0.10\%$	0.38 ± 0.02	$8.66 \pm 0.14\%$
Ours (w/o DHG+GR)	$94.57 \pm 1.21\%$	$5.33 \pm 1.33\%$	$0.10 \pm 0.03\%$	14.88 ± 0.08	20.39 ± 0.14	$3.12 \pm 0.10\%$	0.36 ± 0.03	$8.51 \pm 0.18\%$
Ours (w/o GR)	$96.08 \pm 0.61\%$	$3.81 \pm 0.55\%$	$0.11 \pm 0.03\%$	14.78 ± 0.10	20.33 ± 0.15	$3.09 \pm 0.10\%$	0.37 ± 0.02	$8.43 \pm 0.12\%$
Ours	$96.31 \pm 0.53\%$	$3.59 \pm 0.50\%$	$0.10 \pm 0.03\%$	14.92 ± 0.15	20.53 ± 0.13	$3.08 \pm 0.08\%$	0.40 ± 0.02	$5.21 \pm 0.11\%$
Oracle	$97.28 \pm 0.46\%$	$3.07 \pm 0.41\%$	$0.05 \pm 0.05\%$	14.57 ± 0.24	20.05 ± 0.13	$3.01 \pm 0.09\%$	0.42 ± 0.03	$4.29 \pm 0.04\%$

are almost identical to the ground truth.

Fig. 7 shows the predicted trajectories and hyperedges on the ETH-UCY and NBA datasets, where our approach generates trajectory hypotheses very similar to the ground truth in both scenarios based on the inferred dynamic relations. In particular, our model not only captures group relations where agents within the group share similar motion patterns but also identifies group relations between a set of agents who have potential conflicts or behave adversarially. For example, in the crowd simulation (left), our model identifies human groups (purple ellipses) who are walking in similar directions. Meanwhile, it captures the groups that are close and walking in opposite directions. These humans are inferred as another type of group, which involves interactions for collision avoidance. After the conflict period, the model infers a new set of reasonable group relations (green ellipses), which demonstrates the efficacy of dynamic reasoning. In the NBA scenario (right), the predicted hyperedge at the final step which contains agents 6, 9, and the basketball shows that our model captures the relation of ball passing between teammates.

F. Social Robot Navigation

1) *Quantitative Analysis:* To demonstrate the effectiveness of our social robot navigation method with dynamic relational

reasoning, we provide a comprehensive comparison between our method and baselines in terms of a series of standard evaluation metrics. We also include an Oracle method that employs the true future trajectories of humans for RL policy learning, which serves as a performance upper bound for our method. Note that this Oracle method is not practically deployable and only used for the comparative study. The experimental results are shown in Table VI.

By employing our trajectory prediction model with relational reasoning, the social navigation method significantly outperforms the rule-based method ORCA [67], the dynamical system-based obstacle avoidance (DSOA) algorithm [68], the non-predictive method RL (no prediction), the predictive baseline RL (constant velocity), Group-Aware [42], as well as CrowdNav++ [13] across all the evaluation metrics. The improvement is evident in achieving a higher success rate (SR), lower collision rate (CR) and timeout rate (TR), and shorter navigation time (NT) and path lengths (PL). Meanwhile, our method notably enlarges the social distance from humans and reduces improper intrusions into human personal or group spaces, as indicated by SD, ITR, and GI. This enables the robot to find safer navigation trajectories and minimize interference with pedestrians. Although the DSOA algorithm [68] effectively prevents collisions between robots and indi-



Fig. 8. The visualization of social robot navigation results of different methods in three representative test cases: (a) collision with pedestrians; (b) intrusion into pedestrians' future trajectory; and (c) instruction into pedestrian groups. In each case, we select three representative frames to show the evolution of the dynamic scenes. Best viewed in color.

vidual humans or small numbers of pedestrians, its disregard for group dynamics may cause the robot to inadvertently enter the center of a group. This positioning can lead to longer navigation paths as the robot must then navigate out of the group, potentially resulting in unavoidable collisions and unsafe navigation conditions. Moreover, the Group-Aware model [42], while considering group formation of pedestrians, exhibits limited performance when interacting with multiple surrounding dynamically evolving groups, which may lead to suboptimal actions. Our method achieves comparable performance to the Oracle method which has access to the ground truth human trajectories, showing the effectiveness of our prediction model.

2) Ablative Analysis: Here we show the benefits brought by group-wise relational reasoning and group-based reward.

a) Group-wise reasoning: Ours (w/o DHG+GR) in Table VI represents a variant of our method with only pairwise reasoning and hence it disables group-based reward. Due to the lack of understanding of group relations, this variant leads to a higher CR and lower SD because the prediction model may not predict group behaviors accurately, which degrades the safety of robot navigation. Meanwhile, it causes a higher ITR and GI during navigation, which may make the surrounding humans less comfortable. Although it achieves a slightly shorter NT and PL (i.e., higher efficiency), the improvement is brought by the cost of social appropriateness, which is not desired.

b) Group reward: Ours (w/o GR) in Table VI represents a variant of our method without applying the group-based reward for RL policy learning. It still uses group-wise reasoning for trajectory prediction. The results show that without explicitly considering group relations in the reward function design, the robot policy still leads to a higher CR, ITR, GI, and a lower SD, although it achieves a comparable SR to our complete method. The NT and PL of our method are slightly larger because the policy tends to choose a safer and socially compliant trajectory which may take detours during navigation, which is acceptable and even desired in practice.

3) Qualitative Analysis: We visualize several test cases and compare the robot trajectories planned by different variants of our methods and the state-of-the-art baseline (i.e., CrowdNav++ [13]) in Fig. 8. We mainly highlight the impact of group-wise relational reasoning and group-based reward on RL-based social robot navigation.

In these cases, Ours (w/o GR) predicts pedestrians' future trajectories more accurately than Ours (w/o DHG+GR) and CrowdNav++ consistently, which implies the effectiveness of group-wise reasoning for interactive behavior prediction. However, in Fig. 8(a), all the baseline methods cause a collision with pedestrians because they navigate the robot into a very dense area. In contrast, Ours plans a safer and more socially compliant trajectory thanks to the integration of group-based rewards for policy learning, which avoids getting into dense areas while maintaining good efficiency. Similarly, in Fig. 8(b), all the baseline methods lead to an intrusion into future trajectories of pedestrians, which may interfere with human intentions and violate social norms. In Fig. 8(c), all the baseline methods cause an intrusion into human groups while Ours accurately captures the groups and navigates the



Fig. 9. A screenshot of our experiment video showing that the robot equipped with our method can interact properly with dense crowds with natural moving patterns in real-world scenarios and generate safe paths through crowds.

robot through the interval between different groups safely. In all cases, our method considers group dynamics, which not only prevents collisions with individual pedestrians but also reduces intrusions into the human group as a whole.

4) Real-World Experiments: We deploy our model, trained in the simulation environments, directly on a RosMaster X3 robot equipped with four Mecanum wheels, which enables independent speed control in both horizontal and lateral directions. Different from ideal holonomic kinematics simulations, the RosMaster X3 robot cannot handle sudden changes in velocity. Therefore, we implement smoothness constraints to limit the maximum change in velocity, setting the maximum acceleration to 1 m/s^2 and the maximum velocity to 0.8 m/s . We apply direct clipping after receiving the velocities generated by our RL agent. We use a 2D LiDAR, RPLIDAR-A1, to detect human positions through a learning-based human detector, DR-SPAAM [69]. The robot navigates on a large outdoor terrace with natural pedestrians, which covers an area larger than $10 \text{ m} \times 12 \text{ m}$. In our environment, pedestrians walk at a natural speed with no constraints as shown in Fig. 9.

We select goal points for the robot close to human crowds to encourage maximum interactions. In our experiments, the robot is surrounded by up to 20 pedestrians in complex, dynamic scenarios. As shown in our experiment videos, and the robot successfully navigates to its goal while making appropriate decisions without collisions. In addition, we intentionally position humans to block the robot's path, and the robot responds promptly to avoid collisions, indicating that our navigation policy adapts well to various pedestrian motion patterns and interacts with real humans proactively despite noises from the perception module and robot dynamics.

VII. CONCLUSION

In this paper, we present a systematic dynamic relational reasoning framework that infers pairwise and group-wise relations, which proves to be effective for multi-agent trajectory prediction and social robot navigation. In addition to inferring the conventional graphs that model pairwise relations between interacting agents, we propose to infer relation hypergraphs to model group-wise relations that exist widely in real-world scenarios such as human crowds. Different hyperedge types represent distinct group-wise relations. We employ a dynamic

reasoning mechanism to capture the evolution of underlying relational structures. We propose effective mechanisms to regularize the smoothness of relation evolution, the sharpness of the learned relations, and the sparsity of the learned graphs and hypergraphs, which not only enhances training stability but also reduces prediction error. The proposed approach infers reasonable relations and achieves state-of-the-art performance on both synthetic crowd simulations and multiple benchmark datasets. Moreover, we present a deep reinforcement learning framework for social robot navigation that incorporates relational reasoning and trajectory prediction in a systematic way. It achieves state-of-the-art performance in terms of safety, efficiency, and social appropriateness compared with advanced baselines. A limitation of this work is that although our method outperforms the strongest baseline by a large margin, it still cannot achieve a zero collision rate in very dense scenarios. Therefore, in future work, we will investigate how to enable a provably safety guarantee.

REFERENCES

- [1] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *International Journal of Robotics Research*, vol. 39, no. 8, pp. 895–935, 2020.
- [2] P. Kothari, S. Kreiss, and A. Alahi, "Human trajectory forecasting in crowds: A deep learning perspective," *IEEE Transactions on Intelligent Transportation Systems*, 2021.
- [3] A. Pandey, S. Pandey, and D. Parhi, "Mobile robot navigation and obstacle avoidance techniques: A review," *Int Rob Auto J*, vol. 2, no. 3, p. 00022, 2017.
- [4] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel, "Neural relational inference for interacting systems," in *International Conference on Machine Learning (ICML)*. PMLR, 2018, pp. 2688–2697.
- [5] J. Li, F. Yang, M. Tomizuka, and C. Choi, "Evolvegraph: Multi-agent trajectory prediction with dynamic relational reasoning," *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 19 783–19 794, 2020.
- [6] C. Gruber and A. G. Schwing, "Dynamic neural relational inference," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, pp. 8510–8519.
- [7] J. Li, F. Yang, H. Ma, S. Malla, M. Tomizuka, and C. Choi, "Rain: Reinforced hybrid attention inference network for motion forecasting," in *International Conference on Computer Vision (ICCV)*, 2021, pp. 16 096–16 106.
- [8] L.-F. Wu, Q. Wang, M. Jian, Y. Qiao, and B.-X. Zhao, "A comprehensive review of group activity recognition in videos," *International Journal of Automation and Computing*, vol. 18, no. 3, pp. 334–350, 2021.
- [9] N. Bisagno, B. Zhang, and N. Conci, "Group lstm: Group trajectory prediction in crowded scenarios," in *European Conference on Computer Vision (ECCV) Workshops*, 2018.
- [10] R. Zhou, H. Gao, H. Zhou, M. Tomizuka, J. Li, and Z. Xu, "Grouptron: Dynamic multi-scale graph convolutional networks for group-aware dense crowd trajectory forecasting," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2022.
- [11] B. Zhou, X. Tang, and X. Wang, "Coherent filtering: Detecting coherent motions from crowd clutters," in *European Conference on Computer Vision (ECCV)*. Springer, 2012, pp. 857–871.
- [12] C. Xu, M. Li, Z. Ni, Y. Zhang, and S. Chen, "Groupnet: Multiscale hypergraph neural networks for trajectory prediction with relational reasoning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 6498–6507.
- [13] S. Liu, P. Chang, Z. Huang, N. Chakraborty, K. Hong, W. Liang, D. L. McPherson, J. Geng, and K. Driggs-Campbell, "Intention aware robot crowd navigation with attention-based interaction graph," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 12 015–12 021.
- [14] T. Gu, G. Chen, J. Li, C. Lin, Y. Rao, J. Zhou, and J. Lu, "Stochastic trajectory prediction via motion indeterminacy diffusion," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 17 113–17 122.
- [15] I. Bae and H.-G. Jeon, "A set of control points conditioned pedestrian trajectory prediction," in *AAAI Conference on Artificial Intelligence (AAAI)*, vol. 37, no. 5, 2023, pp. 6155–6165.
- [16] I. Bae, J. Oh, and H.-G. Jeon, "Eigentrajectory: Low-rank descriptors for multi-modal trajectory forecasting," in *International Conference on Computer Vision (ICCV)*, 2023, pp. 10 017–10 029.
- [17] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen, "A survey on trajectory-prediction methods for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, 2022.
- [18] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, no. 5, p. 4282, 1995.
- [19] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 935–942.
- [20] J. v. d. Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research*. Springer, 2011, pp. 3–19.
- [21] J. Nilsson, J. Fredriksson, and E. Coelingh, "Rule-based highway maneuver intention recognition," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2015, pp. 950–955.
- [22] X. Wang, K. T. Ma, G.-W. Ng, and W. E. L. Grimson, "Trajectory analysis and semantic region modeling using nonparametric hierarchical Bayesian models," *International Journal of Computer Vision*, vol. 95, no. 3, pp. 287–312, 2011.
- [23] J. Schulz, C. Hubmann, J. Löchner, and D. Burschka, "Multiple model unscented kalman filtering in dynamic Bayesian networks for intention estimation and trajectory prediction," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 1467–1474.
- [24] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 961–971.
- [25] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 683–700.
- [26] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2255–2264.
- [27] V. Kosaraju, A. Sadeghian, R. Martín-Martín, I. Reid, H. Rezatofighi, and S. Savarese, "Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
- [28] J. Li, H. Ma, and M. Tomizuka, "Conditional generative neural system for probabilistic trajectory prediction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6150–6156.
- [29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
- [30] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.
- [31] Y. Yuan, X. Weng, Y. Ou, and K. M. Kitani, "Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting," in *International Conference on Computer Vision (ICCV)*, 2021, pp. 9813–9823.
- [32] F. Giuliani, I. Hasan, M. Cristani, and F. Galasso, "Transformer networks for trajectory forecasting," in *International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 10 335–10 342.
- [33] J. Ngiam, V. Vasudevan, B. Caine, Z. Zhang, H.-T. L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal *et al.*, "Scene transformer: A unified architecture for predicting future trajectories of multiple agents," in *International Conference on Learning Representations (ICLR)*, 2021.
- [34] S. Shi, L. Jiang, D. Dai, and B. Schiele, "Motion transformer with global intention localization and local movement refinement," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 6531–6543, 2022.
- [35] A. Seff, B. Cera, D. Chen, M. Ng, A. Zhou, N. Nayakanti, K. S. Refaat, R. Al-Rfou, and B. Sapp, "Motionlm: Multi-agent motion forecasting as language modeling," in *International Conference on Computer Vision (ICCV)*, 2023, pp. 8579–8590.
- [36] C. Mavrogiannis, F. Baldini, A. Wang, D. Zhao, P. Trautman, A. Steinfeld, and J. Oh, "Core challenges of social robot navigation: A survey," *ACM Trans. on Human-Robot Interaction*, vol. 12, no. 3, pp. 1–39, 2023.

- [37] A. Turnwald and D. Wollherr, "Human-like motion planning based on game theoretic decision making," *International Journal of Social Robotics*, vol. 11, pp. 151–170, 2019.
- [38] D. Fridovich-Keil, E. Ratner, L. Peters, A. D. Dragan, and C. J. Tomlin, "Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1475–1481.
- [39] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1343–1350.
- [40] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 285–292.
- [41] L. Liu, D. Dugas, G. Cesari, R. Siegwart, and R. Dubé, "Robot navigation in crowded environments using deep reinforcement learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5671–5677.
- [42] K. Katyal, Y. Gao, J. Markowitz, S. Pohland, C. Rivera, I.-J. Wang, and C.-M. Huang, "Learning a group-aware policy for robot navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 11328–11335.
- [43] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. Battaglia, "Learning to simulate complex physics with graph networks," in *International Conference on Machine Learning (ICML)*, 2020, pp. 8459–8468.
- [44] P. Battaglia, R. Pascanu, M. Lai, D. Jimenez Rezende, and K. Kavukcuoglu, "Interaction networks for learning about objects, relations and physics," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.
- [45] C. Gao, J. Xu, Y. Zou, and J.-B. Huang, "DRG: Dual relation graph for human-object interaction detection," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 696–712.
- [46] M. Perez, J. Liu, and A. C. Kot, "Skeleton-based relational reasoning for group activity analysis," *Pattern Recognition*, vol. 122, p. 108360, 2022.
- [47] X. Su, S. Xue, F. Liu, J. Wu, J. Yang, C. Zhou, W. Hu, C. Paris, S. Nepal, D. Jin *et al.*, "A comprehensive survey on community detection with deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
- [48] D. Koller, N. Friedman, S. Džeroski, C. Sutton, A. McCallum, A. Pfeffer, P. Abbeel, M.-F. Wong, C. Meek, J. Neville *et al.*, *Introduction to Statistical Relational Learning*. MIT Press, 2007.
- [49] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1468–1476.
- [50] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, 2021.
- [51] A. Vemula, K. Muelling, and J. Oh, "Social attention: Modeling attention in human crowds," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 4601–4607.
- [52] C. Yu, X. Ma, J. Ren, H. Zhao, and S. Yi, "Spatio-temporal graph transformer networks for pedestrian trajectory prediction," in *European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 507–523.
- [53] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zamabaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner *et al.*, "Relational inductive biases, deep learning, and graph networks," *arXiv preprint arXiv:1806.01261*, 2018.
- [54] K. Li, M. Shan, K. Narula, S. Worrall, and E. Nebot, "Socially aware crowd navigation with multimodal pedestrian trajectory prediction for autonomous vehicles," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–8.
- [55] A. J. Sathyamoorthy, J. Liang, U. Patel, T. Guan, R. Chandra, and D. Manocha, "Densecavoid: Real-time navigation in dense crowds using anticipatory behaviors," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 11345–11352.
- [56] K. D. Katyal, G. D. Hager, and C.-M. Huang, "Intent-aware pedestrian prediction for adaptive crowd navigation," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3277–3283.
- [57] A. Vemula, K. Muelling, and J. Oh, "Modeling cooperative navigation in dense human crowds," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1685–1692.
- [58] R. Bellman, "A markovian decision process," *Journal of mathematics and mechanics*, pp. 679–684, 1957.
- [59] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.
- [60] C. Maddison, A. Mnih, and Y. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," in *International Conference on Learning Representations (ICLR)*, 2017.
- [61] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnns encoder–decoder for statistical machine translation," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2014, p. 1724.
- [62] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software (TOMS)*, vol. 22, no. 4, pp. 469–483, 1996.
- [63] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
- [64] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *International Conference on Computer Vision (ICCV)*. IEEE, 2009, pp. 261–268.
- [65] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," in *Computer Graphics Forum*, vol. 26, no. 3, 2007, pp. 655–664.
- [66] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 549–565.
- [67] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *International Symposium Robotics Research (ISRR)*. Springer, 2011, pp. 3–19.
- [68] L. Huber, J.-J. Slotine, and A. Billard, "Avoiding dense and dynamic obstacles in enclosed spaces: Application to moving in crowds," *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 3113–3132, 2022.
- [69] D. Jia, A. Hermans, and B. Leibe, "Dr-spaam: A spatial-attention and auto-regressive model for person detection in 2d range data," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10270–10277.



Jiachen Li (Member, IEEE) is currently an assistant professor in the Department of Electrical and Computer Engineering and the Department of Computer Science and Engineering (by courtesy) at the University of California, Riverside, USA. He is the director of the Trustworthy Autonomous Systems Laboratory (TASL). Prior to joining the faculty, he was a postdoctoral scholar at Stanford University. He received his Ph.D. from the University of California, Berkeley in 2021. He received a B.E. degree from Harbin Institute of Technology, China in 2016. His research interest lies at the broad intersection of robotics, trustworthy AI, reinforcement learning, control and optimization, and their applications to intelligent autonomous systems, particularly in human-robot interactions and multi-agent systems. Dr. Li was recognized as an RSS Robotics Pioneer in 2022 and an ASME DSCD Rising Star in 2023. He serves as an associate editor or a reviewer for multiple journals and conferences. He has organized multiple workshops on robotics, machine learning, computer vision, and intelligent transportation systems.



Chuanbo Hua received his B.S. degree in Mathematics from the Harbin Institute of Technology, China, in 2020. He was a Computer Science Engineering exchange researcher at Pohang University of Science and Technology (POSTECH), Republic of Korea, in 2019. He then completed his M.S. degree in Industrial and Systems Engineering at the Korea Advanced Institute of Science and Technology (KAIST), Republic of Korea, in 2022. Currently, he is a Ph.D. candidate in the System Intelligence Laboratory, part of the Department of Industrial and Systems Engineering at KAIST. His research interests are primarily focused on combinatorial optimization, reinforcement learning, and dynamic physical process simulation.



Jianpeng Yao received his B.S. and M.S. degrees in Electrical Engineering from Huazhong University of Science and Technology in 2019 and 2022, respectively. He worked as an algorithm engineer in the autonomous driving industry, where he focused on learning-based trajectory prediction and generation. Currently, he is a Ph.D. student at the University of California, Riverside. His research interests include reinforcement learning, trajectory planning, and trustworthy AI, with applications to robotics and autonomous systems.



Hengbo Ma received his Ph.D. degree from the Department of Mechanical Engineering at the University of California, Berkeley in 2023. During his Ph.D. study, Hengbo Ma conducted research on human behavior understanding for robotics and autonomous driving. Before that, he received a B.E. degree from the Department of Control Science and Engineering, Harbin Institute of Technology, China in 2017. His research interest lies in the intersection of control, decision making, optimization, machine learning and artificial intelligence with their applications in robotics and autonomous systems.



Jinkyoo Park is currently an associate professor of Industrial and Systems Engineering and adjoint professor of Graduate School of Artificial Intelligence at Korea Advanced Institute of Science and Technology (KAIST), Republic of Korea. He received his B.S. degree in Civil and Architectural Engineering from Seoul National University in 2009, an M.S. degree in Civil, Architectural and Environmental Engineering from the University of Texas Austin in 2011, an M.S. degree in Electrical Engineering from Stanford University in 2015, and a Ph.D. degree in Civil and Environmental Engineering from Stanford University in 2016. His research goal is to explore the potential of the various machine-learning approaches for improving complex decision-making methods in optimization, optimal control, and game theory.



Victoria M. Dax, currently pursuing her Ph.D. at Stanford Intelligent Systems Laboratory, is focused on integrating niche advancements from other deep learning methods, such as Reinforcement Learning and Variational Autoencoders, into Graph Neural Networks. Before starting her doctoral studies, Victoria worked in the autonomous driving industry, where she developed a practical understanding of applying intelligent systems. Victoria began her academic path at ETH Zurich, earning a Bachelor of Science in Mechanical Engineering in 2016. She

furthered her education at Stanford University, where she completed a Master's degree in Aeronautics and Astronautics in 2018.



Mykel J. Kochenderfer (Senior Member, IEEE) is an Associate Professor of Aeronautics and Astronautics at Stanford University. He is the director of the Stanford Intelligent Systems Laboratory (SISL), conducting research on advanced algorithms and analytical methods for the design of robust decision making systems. Prior to joining the faculty in 2013, he was at MIT Lincoln Laboratory where he worked on aircraft collision avoidance for manned and unmanned aircraft. He received his Ph.D. from the University of Edinburgh in 2006. He received B.S. and M.S. degrees in computer science from Stanford University in 2003. He is an author of the textbooks *Decision Making under Uncertainty: Theory and Application* (MIT Press, 2015), *Algorithms for Optimization* (MIT Press, 2019), and *Algorithms for Decision Making* (MIT Press, 2022).