Hypergraph-based Motion Generation with Multi-modal Interaction Relational Reasoning

Keshu Wu[a,b], Yang Zhou[b,*], Haotian Shi[c,*], Dominique Lord[b], Bin Ran[c], Xinyue Ye[a]

[a]*Center for Geospatial Sciences, Applications, and Technology and Department of Landscape of Architecture and Urban Planning, Texas A&M University, 788 Ross St, College Station, 77840, TX, United States*
[b]*Zachry Department of Civil and Environmental Engineering, Texas A&M University, 201 Dwight Look Engineering Building, College Station, 77843, TX, United States*
[c]*Department of Civil and Environmental Engineering, University of Wisconsin-Madison, 1415 Engineering Dr, Madison, 53706, WI, United States*

Abstract

The intricate nature of real-world driving environments, characterized by dynamic and diverse interactions among multiple vehicles and their possible future states, presents considerable challenges in accurately predicting the motion states of vehicles and handling the uncertainty inherent in the predictions. Addressing these challenges requires comprehensive modeling and reasoning to capture the implicit relations among vehicles and the corresponding diverse behaviors. This research introduces an integrated framework for autonomous vehicles (AVs) motion prediction to address these complexities, utilizing a novel **R**elational **H**ypergraph **I**nteraction-informed **N**eural m**O**tion generator (`RHINO`). `RHINO` leverages hypergraph-based relational reasoning by integrating a multi-scale hypergraph neural network to model group-wise interactions among multiple vehicles and their multi-modal driving behaviors, thereby enhancing motion prediction accuracy and reliability. Experimental validation using real-world datasets demonstrates the superior performance of this framework in improving predictive accuracy and fostering socially aware automated driving in dynamic traffic scenarios.

Keywords: Interaction representation, hypergraph, relational reasoning, multi-modal prediction, motion prediction, motion generation

1. Introduction

Understanding traffic interactions and the way they affect future vehicle trajectories is inherently complex [1]. In mixed traffic environments, where human-driven and automated vehicles coexist, this complexity is amplified, requiring precise interaction representation and behavior modeling for reliable motion prediction [2, 3, 4]. These scenarios often present dynamic interaction topologies with underlying relations, with interaction patterns as well topologies continuously evolving depending on the surrounding context as exampled by lane change maneuvers

*Corresponding authors:Yang Zhou (yangzhou295@tamu.edu), Haotian Shi (hshi84@wisc.edu);

[5, 6]. These relationships play a crucial role in guiding each vehicle's decision-making processes. Additionally, each vehicle can display multiple possible modalities in driving intentions and behaviors, including both longitudinal (e.g., acceleration and braking) and lateral (e.g., lane-changing and lane-keeping) maneuvers [7, 8]. Furthermore, collective behaviors, arising from interactions within a group of vehicles and encompassing both cooperative and competitive behaviors [9, 10, 12, 11], further complicate the understanding of these interactions[13, 14, 15]. Figure 1 describes the interaction among multi-vehicles and the corresponding multi-modal driving behaviors. Therefore, it is necessary to model the interaction and multi-modality and reason the interaction relation to accurately capture interactions and forecast their future behaviors [16, 17, 18].

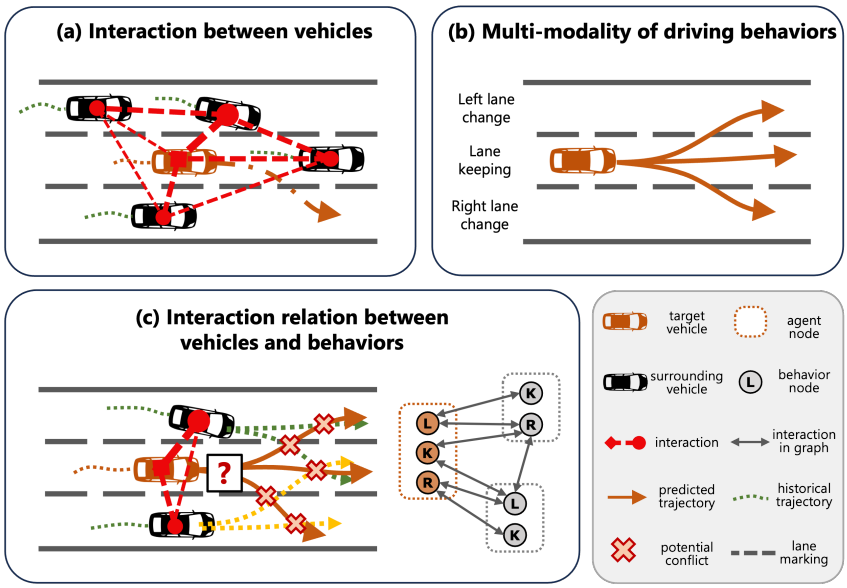

Figure 1: Major challenges: (a) vehicle interaction, (b) behavior multi-modality, and (c) interaction relational reasoning.

Efforts have been made to address the challenges of vehicle interactions and driving behavior multi-modality. Three primary approaches have been developed: social operation methods, attention-driven methods, and graph-based techniques. Social operations use pooling mechanisms to generate socially acceptable trajectories by capturing the influence of surrounding agents [20, 21]. Attention-driven approaches use attention mechanisms to dynamically weigh neighboring agents' information [22, 23, 24]. Graph-based methods leverage graph structures to model non-Euclidean spatial dependencies, effectively handling varying interaction topologies and predicting dynamic interactions [25, 26, 27, 28]. These complex interactions create uncertainty, complicating the accurate forecasting of a single future trajectory with high confidence due to varying driving behaviors in identical situations [29, 30], driven by individual driver characteristics and psychological factors. Addressing the multi-modality of driving behaviors often involves introducing latent variables, categorized into those with explicit semantics and those without. Models with explicit semantics use latent variables to clearly represent driving intentions, identifying

specific maneuvers and behaviors for multi-modal trajectory predictions [30, 31, 32]. Conversely, models without explicit semantics employ generative deep learning techniques, such as Variational Autoencoders (VAEs) [33] and Generative Adversarial Networks (GANs) [21, 34], to produce diverse trajectories by adding noise to encoded features. While these models generate a wide range of possible trajectories, they often struggle with issues related to interpretability and identifying the most effective strategies for sampling from the generated trajectories.

Despite the advances in modeling vehicle interactions and probabilistically forecasting multi-modal future trajectories, the inherent complexity of social dynamics in traffic systems continues to present significant challenges, with limitations arising from the complex nature of agent interactions in two key aspects: First, prevailing methods primarily focus on pair-wise interactions rather group-wise interactions. In multi-vehicle systems, dynamic interactions among vehicles often exhibit cooperative and competitive behaviors [9, 10, 35, 36], which have been rarely explored. Effectively capturing the collective influence of vehicle groups is vital for understanding complex social dynamics, especially in scenarios where vehicles engage in multi-modal behaviors such as lane changes, acceleration, and deceleration. In such contexts, vehicles need to make decisions based on the actions and intentions of several surrounding agents simultaneously, further complicating the task of accurate behavior prediction. For example, competitive behaviors, such as overtaking and lane-changing, can lead to conflicting objectives between vehicles. In the lane changing scenario as depicted in Figure 2, vehicle 1 attempts a right lane change (R) while vehicle 3 maintains lane-keeping (K) and vehicle 4 performs a left lane change (L). Conversely, cooperative interactions, such as car-following, are illustrated by vehicles 1, 3, and 4 forming a platoon, all maintaining lane-keeping while potentially responding to vehicle 2's left lane change. These examples highlight the need for models that can effectively capture the complexities of both individual and group dynamics in multi-modal traffic scenarios.

Second, while traditional graph-based methods such as [25, 26, 27, 37] excel at capturing pair-wise relationships, they are limited in representing the more intricate group-wise interactions. This limitation arises because graph-based approaches model interactions between pairs of agents with fixed topologies, making it difficult to capture the simultaneous influence of multiple agents on each other's behaviors. Unlike standard graphs, where each edge connects only two nodes, hypergraphs are capable of connecting an entire group of nodes within a shared context through a single hyperedge [38, 39]. As a result, a hypergraph, which is a generalization of a graph where hyperedges can connect multiple nodes to represent the higher-order relationships, offers a more accurate reflection of the group-wise interactions and allows for a more accurate representation of the collective influence of multiple agents in varying traffic conditions. This adaptability is essential for addressing the stochastic nature of human behavior, as hypergraph models can more effectively manage the uncertainty and variability inherent in the interactions resulting from collective behaviors, thereby facilitating more socially inspired automated driving [40, 41, 42].

To address the aforementioned challenges, this paper presents a novel hypergraph-based method for multi-modal trajectories prediction with relational reasoning. The proposed framework contains two parts: **G**raph-based **I**nteraction-awa**R**e **A**nticipative **F**easible **F**uture **E**stimator (`GIRAFFE`) and **R**elational **H**ypergraph **I**nteraction-informed **N**eural m**O**tion generator (`RHINO`). `GIRAFFE` enables multi-agent, multi-modal motion prediction of preliminary multi-vehicle trajectories, based on while `RHINO` framework, which utilizes an innovative Agent-Behavior Hypergraph to cap-

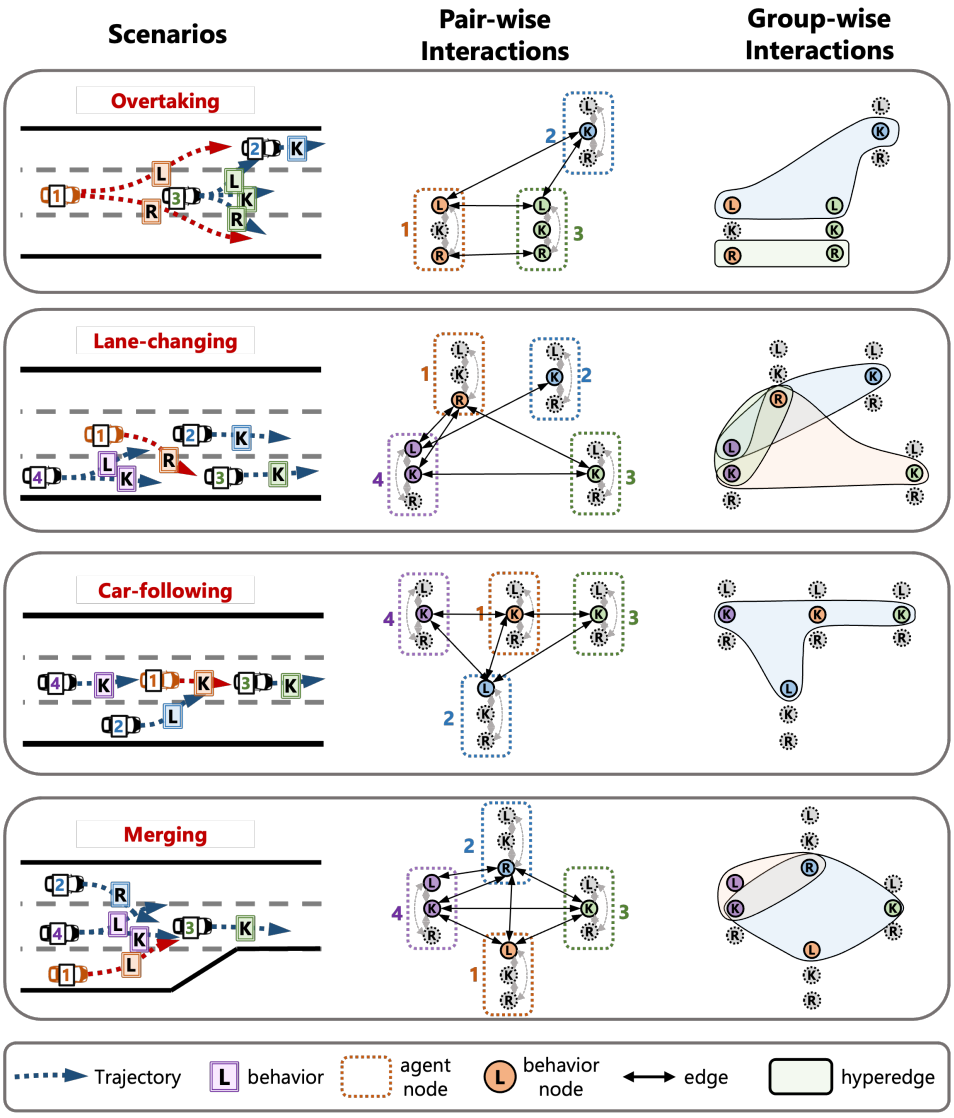

Figure 2: Pair-wise interaction and group-wise interaction in different scenarios.

ture group-wise interactions among various behavior modalities and motion states. Leveraging `GroupNet` [41] as its backbone, `RHINO` learns a multi-scale hypergraph topology in a data-driven manner to model group-wise interactions. Through neural message passing across the hypergraph, this approach integrates interaction representation learning and relational reasoning, enhancing the social dynamics of automated driving. Furthermore, a Conditional Variational Autoencoder (CVAE) framework is employed to generate diverse trajectory predictions by sampling from hid-

den feature distributions, effectively addressing the stochastic nature of agent behaviors.

To summarize, the key contributions of this work are as follows:

1. The framework adopts multi-scale hypergraphs to represent group-wise interactions among different modalities of driving behavior and the corresponding motion states of multiple agents in a flexible manner.
2. This framework incorporates interaction representation learning and relational reasoning to generate motions that are plausible and concurrently in a probabilistic manner depicted by the learned posterior distribution.

The remainder of this paper is structured as follows: Section 2 outlines the problem statement of this research. Section 3 introduces the methodology. Section 4 details the experimental setup and analysis of the results obtained. Section 5 concludes this paper.

2. Problem Statement

2.1. Problem Definition

The vehicle trajectory prediction in the dynamic realm of multi-vehicle interaction context of multi-lane highways involves determining the future movements of a target vehicle based on historical data and multi-modal predictions of its own state and the states of surrounding vehicles. This domain addresses two primary challenges: (i) multi-agent multi-modal trajectory prediction and (ii) prediction-guided motion generation after reasoning.

The objective of trajectory prediction is to estimate the future trajectories of the target vehicle and its surrounding vehicles, given their historical states. The historical states, spanning a time horizon $[1, \ldots, T]$, are represented as $\mathbf{X}_{1:T} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_T\} \in \mathbb{R}^{T\times N\times C_1}$, where N denotes the number of vehicles and C_1 denotes the number of features, including longitudinal and lateral positions and velocities. Each historical state $\mathbf{X}_t = \{\mathbf{x}_t^i | \forall i \in [1, N], \forall t \in [1, T]\} \in \mathbb{R}^{C_1}$ at time step t captures these details for each vehicle i. Notably, the superscript refers to vehicle indices with $i = 1$ representing the target vehicle, and the subscript to time steps, with $C_1 = 4$ for the input data.

The prediction model, $\mathbf{H}^{Pred}(\cdot)$, provides preliminary predictions of multi-modal trajectory candidates $\hat{\mathbf{X}}^M_{T+1:T+F} \in \mathbb{R}^{F\times N\times M\times C_2}$ for all the N vehicles over the future time horizon $[T+1, \ldots, T+F]$ with M modes of driving behaviors. This model takes historical data $\mathbf{X}_{1:T}$ as the input and outputs future longitudinal and lateral positions, where $C_2 = 2$. The forecasted states $\hat{\mathbf{X}}^M_f = \{\mathbf{x}_f^{m,i} | \forall m \in [1, M], \forall i \in [1, N], \forall f \in [1, F]\}$ aim to estimate each vehicle's future trajectory for each behavior mode m at time step $T + f$. This is mathematically formulated as:

$$\hat{\mathbf{X}}^M_{T+1:T+F} = \mathbf{H}^{Pred}(\mathbf{X}_{1:T}) \tag{1}$$

Based on that, the motion generation model $\mathbf{H}^{Gen}(\cdot)$ is further developed to generate plausible trajectories considering the implicit group-wise interactions, using both historical states $\mathbf{X}_{1:T}$ and preliminary multi-modal future trajectory candidates $\hat{\mathbf{X}}^M_{T+1:T+F}$ as the input. The generation model provides K plausible trajectory $\hat{\mathbf{Y}}^K_{T+1:T+F} = \{\hat{\mathbf{Y}}^K_{T+1}, \ldots, \hat{\mathbf{Y}}^K_{T+F}\} \in \mathbb{R}^{F\times N\times K\times C_2}$ for all the N vehicles for the next F time steps. Each generated state $\hat{Y}^K_{T+f} = \{\hat{\mathbf{y}}^{k,i}_{T+f} | \forall k \in [1, K], \forall i \in [1, N], \forall f \in [1, F]\} \in$

$\mathbb{R}^{C_2}$ represents the k-th generated longitudinal and lateral potisions of the i-th vehicle at time step $T + f$. The formulation for this generation problem is:

$$\hat{\mathbf{Y}}^{K}_{T+1:T+F} = \mathbf{H}^{Gen}(\mathbf{X}_{1:T}, \hat{\mathbf{X}}^{M}_{T+1:T+F}) \tag{2}$$

3. Methodology

Given the aforementioned problem, we first develop a customized framework architecture. Then, the vital components are further elaborated.

3.1. Framework Architecture

The proposed framework adopts an integrated architecture, as shown in Figure 3, which involves two major components:

- `GIRAFFE`: **G**raph-based **I**nteraction-awa**R**e **A**nticipative **F**easible **F**uture **E**stimator, which leverages graph representations to capture pair-wise interactions during both the historical and future time horizons, providing preliminary multi-modal trajectories prediction candidates for vehicles.

- `RHINO`: **R**elational **H**ypergraph **I**nteraction-informed **N**eural m**O**tion generator, which utilizes multi-scale hypergraph representations to model group-wise interactions and reason the interaction relations among the multi-modal behaviors. Built upon the preliminary multi-modal trajectories by GIRAFFE, RHINO will further generate plausible future trajectories for all vehicles in a probabilistic manner.

The subsequent sections will provide an in-depth explanation of the two principal frameworks.

3.2. GIRAFFE: Graph-based Motion Predictor

In our context, a graph representation $\mathcal{G}$ is adopted by modeling N vehicles as nodes $\mathcal{V} \in \mathbb{R}^N$ and the pair-wise interaction as the edges $\mathcal{E} \in \mathbb{R}^{|N\times N|}$. Further, The feature matrix $X \in \mathbb{R}^{N\times C}$ containing vehicle states (i.e., longitudinal and lateral position and speed) and the adjacency matrix $A \in \mathbb{R}^{N\times N}$ describing the interactions among nodes are further utilized to describe the graph. By that, we can define an Agent Graph as:

Definition 1 (Agent Graph). *Let $\mathcal{G}^a$ be a graph representing the motion states and interaction of N agents, with each agent represented as a node. $\mathcal{G}^a$ is expressed as*

$$\mathcal{G}^a = (\mathcal{V}^a, \mathcal{E}^a; X^a, A^a)$$

where $\mathcal{V}^a \in \mathbb{R}^N$ denotes the node set, $\mathcal{E}^a \in \mathbb{R}^{|N\times N|}$ denotes the edge set, $X^a \in \mathbb{R}^{N\times C}$ represents the feature tensor, $A^a \in \mathbb{R}^{N\times N}$ indicates the adjacency matrix.

To better represent the interaction and relations of the predicted multi-agent multi-modal trajectory candidates with graphs, we expand each agent node to multiple nodes of the number of behavior modes based on our previous work [25], which further renders an Agent-Behavior Graph.

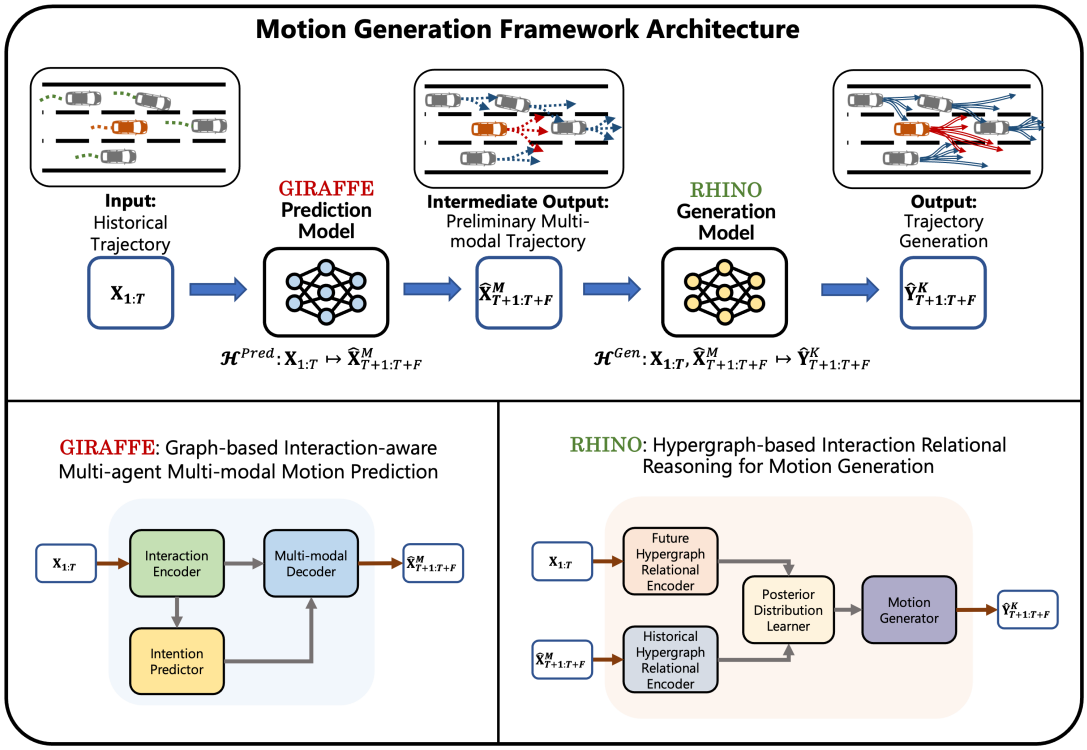

Figure 3: Framework architecture.

Definition 2 (Agent-Behavior Graph). *Let $\mathcal{G}^b$ be a graph representation of the multi-modal motion states of N agents, with each of M behavior modes for each agent represented as a node. $\mathcal{G}^b$ is expressed as*

$$\mathcal{G}^b = (\mathcal{V}^b, \mathcal{E}^b; X^b, A^b)$$

where $\mathcal{V}^b \in \mathbb{R}^{|MN|}$ denotes the node set, $\mathcal{E}^b \in \mathbb{R}^{|MN \times MN|}$ denotes the edge set, $X^b \in \mathbb{R}^{|MN| \times C}$ represents the feature tensor, $A^b \in \mathbb{R}^{|MN| \times |MN|}$ indicates the adjacency matrix.

The transition from an Agent Graph $\mathcal{G}^a$ to an Agent-Behavior Graph $\mathcal{G}^b$ is by an expansion function $\mathbf{F}^{expand}(\cdot)$ as:

$$\mathcal{G}^b = \mathbf{F}^{expand}(\mathcal{G}^a) \Leftrightarrow \begin{cases} \mathcal{V}^b = \bigcup_{i=1}^{N} \bigcup_{m=1}^{M} \left\{ v_{i,m}^b \right\} \\ \mathcal{E}^b = \bigcup_{i=1}^{N} \bigcup_{j=1}^{N} \left\{ e_{ij,mn}^b \mid A_{ij}^a \neq 0 \text{ and } \Lambda_{mn} \neq 0, \forall m, n \in \{1, \ldots, M\} \right\} \\ X^b = \bigcup_{i=1}^{N} \bigcup_{m=1}^{M} X_{i,m}^a \\ A^b = \bigcup_{i=1}^{N} \bigcup_{j=1}^{N} \left\{ A_{ij}^a \otimes \Lambda_{mn} \mid \forall m, n \in \{1, \ldots, M\} \right\} \end{cases} \tag{3}$$

As shown in Figure 4, in this process, each agent node $v_i^a \in \mathcal{V}^a$ in the Agent Graph is expanded into M behavior-specific nodes $v_i^b \in \mathcal{V}^b$, corresponding to the M potential behavioral modes of the vehicle. These newly generated behavior nodes, which may exhibit significant interdependencies, are interconnected through edges e_{ij}^b, forming a more complex interaction structure. Consequently, the adjacency matrix A^b is extended to accommodate the expanded node set, resulting in increased dimensionality. Here, Λ is a behavior correlation matrix, and each element Λ_{mn} in the matrix represents the correlation between behavior mode m of one agent and behavior mode n of another agent. The feature tensor X^b of the behavior nodes encodes the possible motion states under each behavioral mode, capturing the multi-modal nature of vehicle behavior.

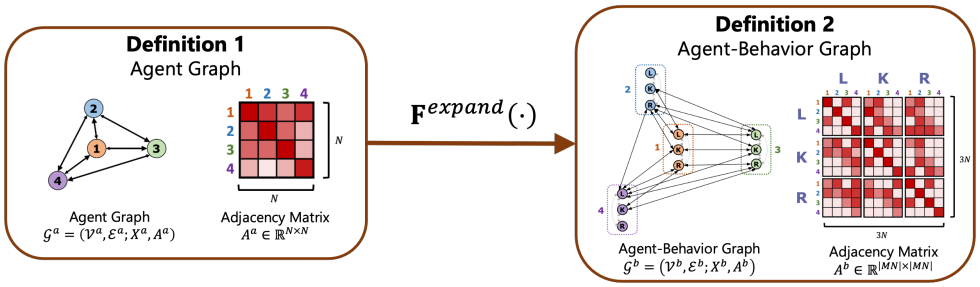

Figure 4: Definitions of graphs

Based on that, a deep neural network is designed to capture interactions between the target vehicle and surrounding vehicles by $\mathcal{G}^a$, and to represent the output, which consists of predicted multi-agent multi-modal trajectories, by $\mathcal{G}^b$, as illustrated in Figure 5. Three key modules of GIRAFFE $\mathbf{H}^{Pred}(\cdot)$ are introduced below:

Interaction Encoder. The Interaction Encoder utilizes a Diffusion Graph Convolution Network (DGCN) architecture to encode the dynamic graph embeddings, as described in Appendix B. The DGCN captures the bidirectional dependencies among vehicles by applying diffusion convolutions, which consider both forward and reverse processes to model the influence of surrounding

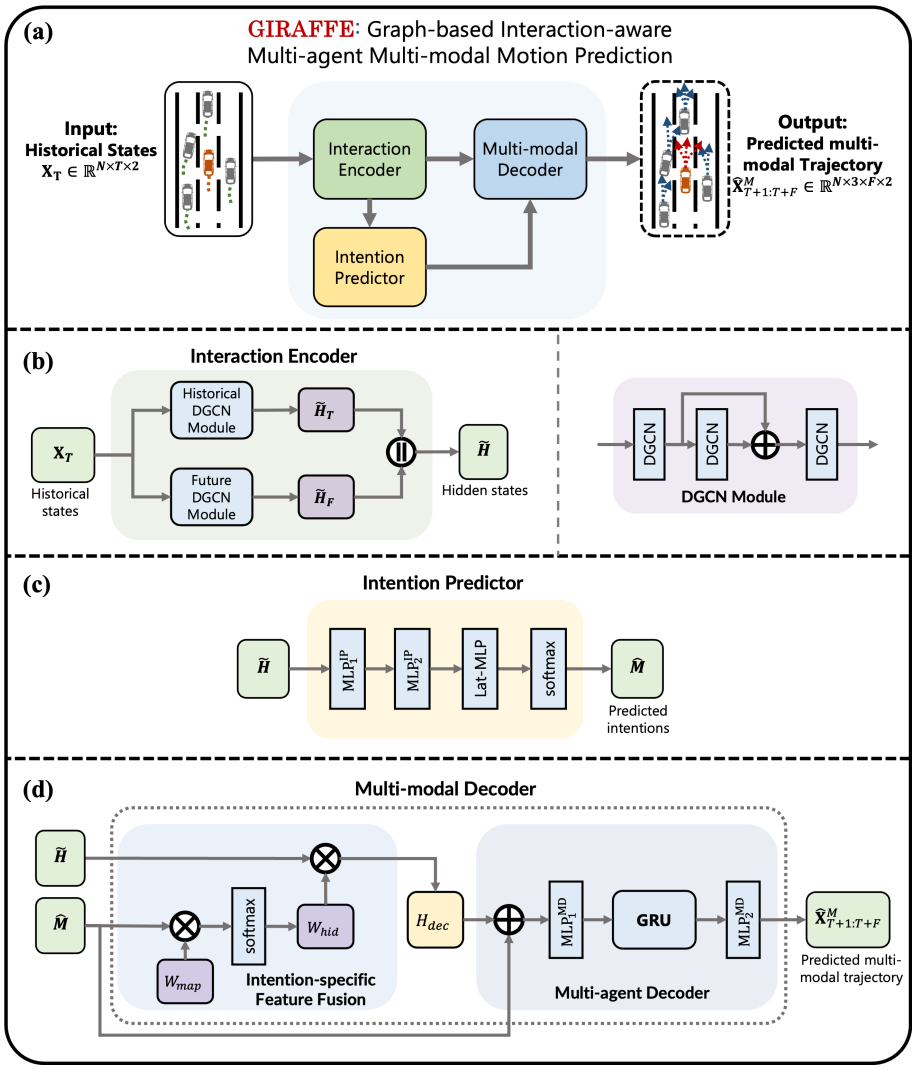

Figure 5: GIRAFFE Framework.

vehicles and the target vehicle's impact on them [43, 44]. This encoder adopts a $DGCN_H(\cdot)$ to generate graph embeddings for historical of and a $DGCN_F(\cdot)$ to generate future embeddings, merging them into a comprehensive representation that spans the entire time window of interest.

$$\tilde{H}_T = DGCN_H(X_T) \tag{4}$$

$$\tilde{H}_F = DGCN_F(X_T) \tag{5}$$

$$\tilde{H} = [\tilde{H}_T, \tilde{H}_F] \tag{6}$$

Intention Predictor. The Intention Predictor addresses the classification of future driving intentions, both laterally and longitudinally. Using the encoded graph representation, two MLP layers reduce the dimensions and encode the features into a latent space. The LatMLP layers with softmax activation then classify the lateral intentions over the future time horizon. These predictions help in understanding the potential maneuvers the vehicle might take, such as lane changes or speed adjustments.

$$\tilde{H}^{IP} = MLP(MLP(\tilde{H})) \tag{7}$$

$$\hat{m}^{lat} = \text{softmax}(LatMLP(\tilde{H}^{IP})) \tag{8}$$

Multi-modal Decoder. Finally, the Multi-modal Decoder fuses the predicted intentions of multiple agents with the latent space to produce multiple future trajectory distributions for each agent. This decoder uses a trainable weight matrix to combine features from distinct historical and future time steps, emphasizing the importance of sequential motion patterns. The GRU-based decoder ensures temporal continuity in the predicted trajectories, mapping the fused features to a bivariate Gaussian distribution representing the future vehicle positions. This approach allows the model to generate probabilistic predictions for multiple agents.

$$W_{hid} = \text{softmax}(W_{map} \otimes \hat{M}) \tag{9}$$

$$H_{dec} = W_{hid} \cdot \tilde{H} \tag{10}$$

$$\hat{\mathbf{X}}^{M}_{T+1:T+F} = MLP(GRU(MLP(H_{dec}, \hat{M}))) \tag{11}$$

3.3. RHINO: Hypergraph-based Motion Generator

Unlike traditional graph representations, which are confined to pair-wise relationships, hypergraphs offer a more sophisticated and comprehensive framework for representing group-wise interactions. By connecting multiple vehicles that exhibit strong correlations through hyperedges, hypergraphs enable a more robust analysis and optimization of the complex network of interactions. The concept of an Agent Hypergraph to represent agents and their group-wise interactions is introduced as follows:

Definition 3 (Agent Hypergraph). *Let $\mathcal{H}^b$ be a hypergraph representation of the motion states of N agents, with each agent represented as a node. The hypergraph $\mathcal{H}^a$ is expressed as*

$$\mathcal{H}^a = (\mathcal{V}^a, \mathcal{U}^a; X^a, H^a)$$

where $\mathcal{V}^a \in \mathbb{R}^N$ denotes the node set, $\mathcal{U}^a \in \mathbb{R}^L$ denotes the edge set, $X^a \in \mathbb{R}^{N\times C}$ represents the feature tensor, $H^a \in \mathbb{R}^{N\times L}$ indicates the incidence matrix, where H^a_{ij} indicates whether node v_i is part of the hyperedge u_j.

To convert an Agent Graph $\mathcal{G}^a$ into an Agent Hypergraph $\mathcal{H}^a$, we introduce a transformation function $\mathbf{F}^{transform}(\cdot)$. This function enables the shift from a pairwise interaction framework to a higher-order interaction model represented by the hypergraph. Formally, the transformation is expressed as:

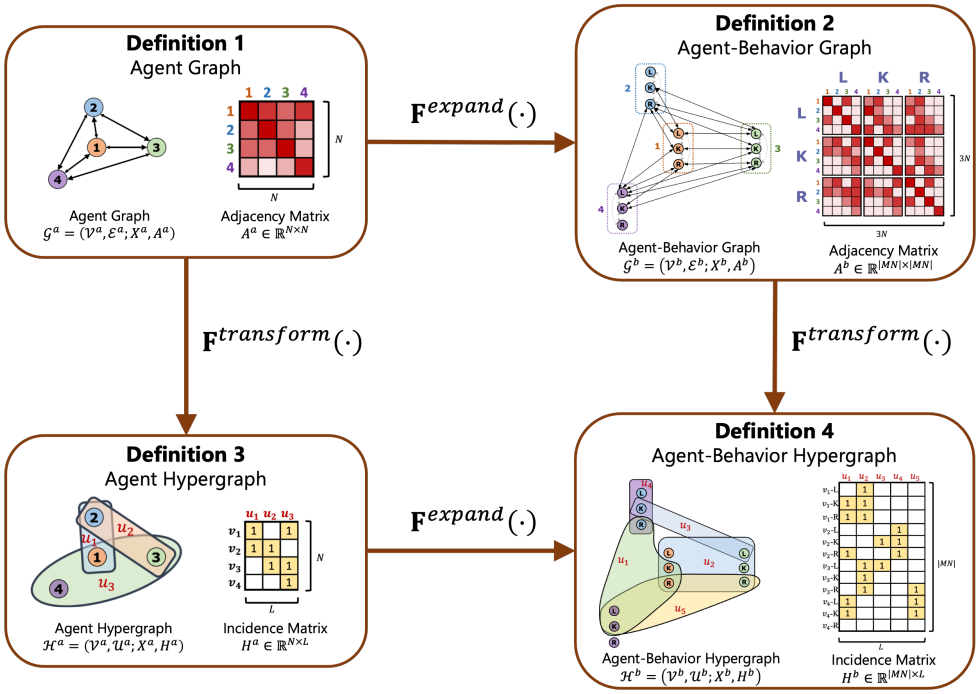

Figure 6: Definitions of graphs and hypergraphs

$$\mathcal{H}^a = \mathbf{F}^{transform}(\mathcal{G}^a) \Leftrightarrow \begin{cases} \mathcal{V}^a = \bigcup_{i=1}^{N} \{v_i^a\} \\ \mathcal{U}^a = \bigcup_{j=1}^{K} \left\{ u_j^a \mid u_j^a \subseteq \mathcal{V}^a, \Lambda_{u_j} \neq 0 \right\} \\ X^a = \bigcup_{i=1}^{N} X_i^a \\ H^a = \bigcup_{i=1}^{N} \bigcup_{j=1}^{K} \left\{ H_{ij}^a \mid v_i^a \in u_j^a \right\} \end{cases} \tag{12}$$

In this transformation, the node set $\mathcal{V}^a$ and the feature tensor X^a remain consistent between the graph and the hypergraph representations. However, the primary modification occurs in the edge formulation. The transformation replaces the pairwise edges of the original Agent Graph with hyperedges that can connect multiple nodes simultaneously. This redefinition of edges as hyperedges within the hypergraph $\mathcal{H}^a$ allows for the modeling of group-wise interactions, where a single hyperedge $u \in \mathcal{U}^a$ can link more than two nodes, capturing higher-order relationships among agents. The incidence matrix H^a is updated to reflect this change, where each entry H_{ij}^a

indicates whether agent v_i participates in hyperedge u_j. As a result, $\mathcal{H}^a$ can represent multi-agent interactions that involve multiple agents simultaneously, providing a richer and more flexible structure for modeling the dynamics of the system.

To enhance the understanding of complex group-wise interactions in multi-agent systems, it is essential to extend the traditional Agent Hypergraph model to account for the diverse behavioral modes of each agent. This is achieved by decomposing each agent node in the Agent Hypergraph $\mathcal{H}^a$ into multiple behavior-specific nodes, which correspond to the different modes of behavior each agent can exhibit. The result of this decomposition is the Agent-Behavior Hypergraph, denoted as $\mathcal{H}^b$.

Definition 4 (Agent-Behavior Hypergraph). *Let $\mathcal{H}^b$ be a hypergraph representation of the multi-modal motion states of N agents, with each of M behavior modes for each agent represented as a node. The hypergraph $\mathcal{H}^b$ is expressed as*

$$\mathcal{H}^b = (\mathcal{V}^b, \mathcal{U}^b; X^b, H^b)$$

where $\mathcal{V}^b \in \mathbb{R}^{|MN|}$ denotes the node set, $\mathcal{U}^b \in \mathbb{R}^L$ denotes the edge set, $X^b \in \mathbb{R}^{|MN| \times C}$ represents the feature tensor, $H^b \in \mathbb{R}^{|MN| \times L}$ indicates the incidence matrix, where H^b_{ij} indicates whether node v_i is part of the hyperedge u_j.

To formally describe the process of transitioning from an Agent Hypergraph $\mathcal{H}^a$ to an Agent-Behavior Hypergraph $\mathcal{H}^b$, the expansion function $\mathbf{F}^{expand}(\cdot)$ is applied. This function decomposes each agent node into multiple behavior-specific nodes and updates the hyperedge structure accordingly. The behavior-specific nodes correspond to the different behavior modes, while the hyperedges represent the higher-order interactions among the behavior modes of different agents.

$$\mathcal{H}^b = \mathbf{F}^{expand}\left(\mathcal{H}^a\right) \Leftrightarrow \begin{cases} \mathcal{V}^b = \bigcup_{i=1}^{N} \bigcup_{m=1}^{M} \left\{ v^b_{i,m} \right\} \\ \mathcal{U}^b = \bigcup_{j=1}^{L} \bigcup_{m=1}^{M} \bigcup_{n=1}^{M} \left\{ u^b_{j,mn} \mid u^a_j \neq 0 \text{ and } \Lambda_{mn} \neq 0 \right\} \\ X^b = \bigcup_{i=1}^{N} \bigcup_{m=1}^{M} X^a_{i,m} \\ H^b = \bigcup_{i=1}^{MN} \bigcup_{j=1}^{L} \left\{ H^b_{ij} \mid v^b_{i,m} \in u^b_{j,mn} \right\} \end{cases} \tag{13}$$

The node set $\mathcal{V}^b$ expands each agent v^b_i into multiple behavior-specific nodes $v^b_{i,m}$, where each m represents a different behavioral mode of the agent. The hyperedges $\mathcal{U}^b$ are formed between the behavior-specific nodes based on the group-wise interactions present in the original hypergraph, with the correlation between behavior modes captured by Λ_{mn}. The feature tensor X^b captures the state of each behavior node, inheriting the feature data from the original hypergraph.The incidence matrix H^b records whether a behavior-specific node $v^b_{i,m}$ is part of a hyperedge $u^b_{j,mn}$. In

this extended framework, hyperedges can represent these aforementioned complex group-wise interactions by connecting behaviors of multiple vehicles that are influenced simultaneously by a shared context, as illustrated in Figure 6. Therefore, an Agent-Behavior Hypergraph is defined to model the multi-agent, multi-modal system for reasoning about group-wise interaction relations.

In addition to the expansion process, the transformation function $\mathbf{F}^{transform}(\cdot)$ converts an Agent-Behavior Graph $\mathcal{G}^b$ into an Agent-Behavior Hypergraph $\mathcal{H}^b$. This transformation replaces the pairwise edges of the graph with hyperedges that capture higher-order interactions between behavior modes across multiple agents. The transformation is guided by the adjacency matrix A^b of the original graph and the behavior-mode correlation matrix Λ_{mn}. The transformation function $\mathbf{F}^{transform}(\cdot)$ is expressed as:

$$\mathcal{H}^b = \mathbf{F}^{transform}\left(\mathcal{G}^b\right) \Leftrightarrow \begin{cases} \mathcal{V}^b = \bigcup_{i=1}^{N} \bigcup_{m=1}^{M} \left\{v_{i,m}^b\right\} \\ \mathcal{U}^b = \bigcup_{j=1}^{L} \bigcup_{m=1}^{M} \bigcup_{n=1}^{M} \left\{u_{j,mn}^b \mid A_{ij}^b \neq 0 \text{ and } \Lambda_{mn} \neq 0\right\} \\ X^b = \bigcup_{i=1}^{N} X_i^b \\ H^b = \bigcup_{i=1}^{MN} \bigcup_{j=1}^{L} \left\{H_{ij}^b \mid v_{i,m}^b \in u_{j,mn}^b\right\} \end{cases} \tag{14}$$

This structure allows for the connection of behavior nodes across different agents, enabling the representation of interactions among diverse behaviors of multiple agents in a shared context. Hence, the Agent-Behavior Hypergraph not only captures the individual behavior of each agent but also models how these behaviors interact and influence one another within a multi-agent system.

3.3.1. RHINO Framework Architecture

The core of RHINO is to learn a multi-scale Agent-Behavior Hypergraph, where nodes represent the behaviors of agents and hyperedges capture their group-wise interactions. This hypergraph is then used to learn agent and interaction embeddings to better understand the underlying interaction relations. We also incorporate a basic multi-agent trajectory generation system based on the CVAE framework to handle the stochasticity of each agent's potential behaviors and motion states, generating plausible trajectories for each vehicle.

Thus, as illustrated in Figure 7, RHINO comprises the following modules:

- **Hypergraph Relational Encoder**, which transforms both the original historical states and predicted multi-agent multi-modal trajectories into hypergraphs, modeling and reasoning the underlying relation between the vehicles.

- **Posterior Distribution Learner**, which captures the posterior distribution of the future trajectory given the historical states and the predicted multi-modal future motion states of all the vehicles in the vehicle group.

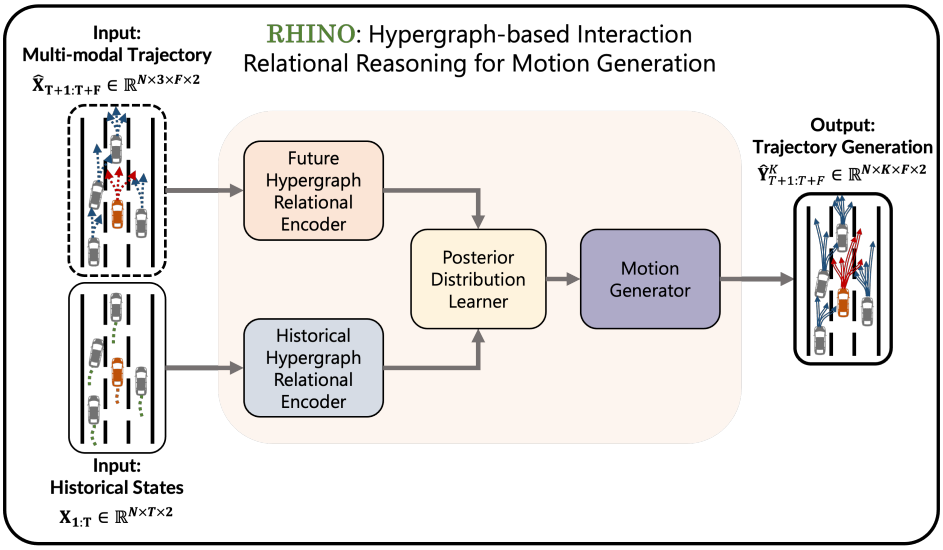

Figure 7: RHINO Framework.

- **Motion Generator**, which decodes the embeddings by concurrently reconstructing the historical states and generating the future trajectories.

3.3.2. Hypergraph Relational Encoder

We employ two Hypergraph Relational Encoder modules: a Historical Hypergraph Relational Encoder for handling historical states and a Future Hypergraph Relational Encoder for predicted multi-agent multi-modal trajectories from GIRAFFE. For the Historical Hypergraph Relational Encoder, the input historical states X_T form an Agent Hypergraph $\mathcal{H}^a_T$. For the Future Hypergraph Relational Encoder, the predicted multi-agent multi-modal trajectories $\hat{X}_{T+1:T+F}$ form an Agent-Behavior Hypergraph $\mathcal{H}^b_F$, where each agent node is expanded into three lateral behavior nodes with corresponding predicted future states. Both modules share the same structure regardless of the input hypergraph types.

Multi-scale Hypergraph Topology Inference. To comprehensively model group-wise interactions in the hypergraphs at multiple scales, we infer a multi-scale hypergraph to reflect interactions in groups with various sizes. Let $\mathcal{H} = \{\mathcal{H}^{(0)}, \mathcal{H}^{(1)}, \cdots, \mathcal{H}^{(S)}\}$ be a multi-scale hypergraph, and $\mathcal{V} = \{v_1, v_2, \cdots, v_N\}$ be a set of nodes. As shown in Figure 8, at any scale s, $\mathcal{H}^{(s)} = (\mathcal{V}, \mathcal{U}^{(s)})$ has a hyperedge set $\mathcal{U}^{(s)} = \{u^{(s)}_1, u^{(s)}_2, \cdots, u^{(s)}_J\}$ representing group-wise relations with J hyperedges. A larger s indicates a larger scale of agent groups, while $\mathcal{H}^{(0)} = (\mathcal{V}, \mathcal{U}^{(0)})$ models the finest pair-wise agent connections. The topology of each $\mathcal{H}^{(s)}$ is represented as an incidence matrix $H^{(s)}$.

To understand and quantify the dynamic interactions between agents within a given system, we adopt trajectory embedding to distill the motion states of agents into a compact and informative representation. To infer a multi-scale hypergraph, we construct hyperedges by grouping agents that have highly correlated trajectories, whose correlations could be measured by mapping the

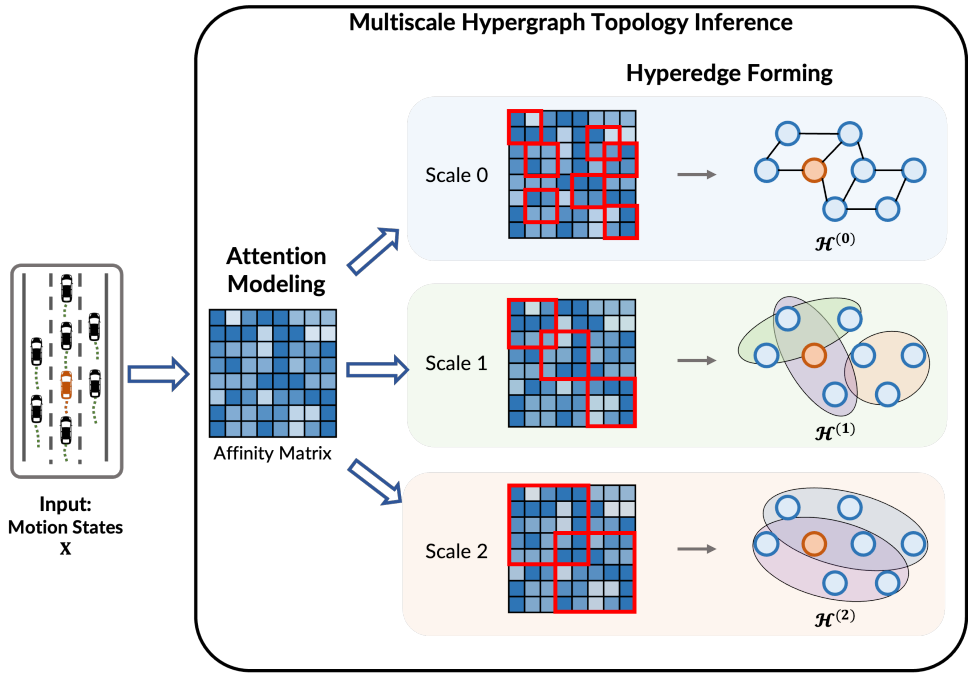

Figure 8: Hypergraph encoder.

trajectories as a high-dimensional feature vector. For the i-th agent in the system, the trajectory embedding is denoted as q_i. This embedding is a function of the agent's motion states, defined over a temporal window extending from time 1 to time T. The embedding function f_q, which is a trainable MLP, is responsible for transforming the motion states X^i into a vector $q_i \in \mathbb{R}^d$, where d is the dimensionality of the embedded space. Mathematically, the trajectory embedding is represented as:

$$q_i = f_q(X^i) \tag{15}$$

The affinity between agents is represented by an affinity matrix $\mathcal{A} \in \mathbb{R}^{N\times N}$, which contains the pairwise relational weights between all agents. The affinity matrix is defined as:

$$\mathcal{A} = \{\mathcal{A}_{ij} | \forall i, j = 1, \ldots, N\} \tag{16}$$

Each element $\mathcal{A}_{ij}$ is computed as the correlation between the trajectory embeddings of the i-th and j-th agents. The correlation is the normalized dot product of the two trajectory embeddings, expressed as:

$$\mathcal{A}_{ij} = \frac{q_i^\top q_j}{\|q_i\|_2 \|q_j\|_2} \tag{17}$$

Here, $\|\cdot\|_2$ denotes the L2 norm. The relational weight $\mathcal{A}_{ij}$ measures the strength of association between the trajectories of the i-th and j-th agents, capturing the degree to which their behaviors

are correlated. This enables the assessment of interaction patterns and can uncover underlying social or physical laws governing agent dynamics.

The formulation of a hypergraph necessitates the strategic formation of hyperedges that reflect the complex interaction between the nodes in the system. At the outset, the 0-th scale hypergraph $\mathcal{H}^{(0)}$ is considered, where the construction is based on pair-wise connections. Each node establishes a link with another node that has the highest affinity score with it.

As the complexity of the system is scaled up, beginning at scale $s \geq 1$, the methodology shifts towards group-wise connections. This shift is based on the intuition that agents within a particular group should display strong mutual correlations, suggesting a propensity for concerted action. To implement this, a sequence of increasing group sizes $\{J^{(s)}\}_{s=1}^{S}$ is established. For every node, denoted by v_i, the objective is to discern a group of agents that are highly correlated, ultimately leading to $J^{(s)}$ groups or hyperedges at each scale s. The hyperedge associated with a node v_i at a given scale s is indicated by $u_i^{(s)}$. The determination of the most correlated agents is framed as an optimization problem, aiming to link these agents into a hyperedge that accounts for group dynamics:

$$u_i^{(s)} = \arg\max_{\Gamma \subseteq \mathcal{V}} \left\| \mathcal{A}_{\Gamma,\Gamma} \right\|_1 \tag{18}$$

$$\text{s.t. } |\Gamma| = J^{(s)}; v_i \in \Gamma; i = 1, \ldots, N \tag{19}$$

The culmination of this hierarchical structuring is a multi-scale hypergraph, encapsulated by the set $\{\mathcal{H}^{(s)} \in \mathbb{R}^{N \times N}\}_{s=1}^{S}$, where each scale s embodies a distinct layer of abstraction in the representation of node relationships within the hypergraph.

Hypergraph Neural Message Passing. In order to discern the patterns of agent motion states from the inferred multi-scale hypergraph, we have tailored a multi-scale hypergraph neural message passing technique to construct the hypergraph topology. This method iteratively acquires the embeddings of vehicles and the corresponding interactions through node-to-hyperedge and hyperedge-to-node processes, as depicted in Figure 9.

The node-to-hyperedge mapping aggregates agent embeddings to generate interaction embeddings. Initially, for any given scale, the initial embedding for the ith agent, $v_i = q_i \in \mathbb{R}^d$. Each node v_j is associated with a hyperedge u_i, given that v_j is an element of u_i. This mapping facilitates the definition of the hyperedge interaction embedding. The hyperedge interaction embedding for a hyperedge u_i is defined as a function of the embeddings of the nodes contained within it, modulated by the neural interaction strength r_i and categorized through coefficients $c_{i,l}$. The per-category function $\mathcal{F}_l$ models the interaction process for each category, which is crucial for capturing the nuances of different interaction types. Each $\mathcal{F}_l$ is a trainable MLP, responsible for processing the aggregated node embeddings within the context of a specific interaction category. The mathematical formulation is:

$$u_i = r_i \sum_{l=1}^{L} c_{i,l} \mathcal{F}_l \left(\sum_{v_j \in u_i} v_j \right) \tag{20}$$

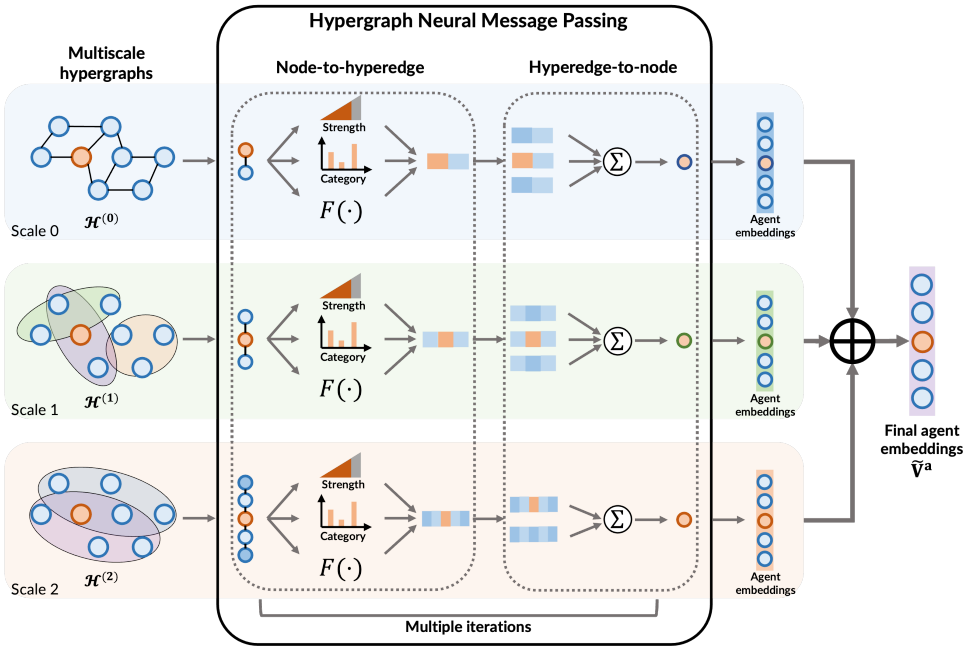

Figure 9: Hypergraph encoder.

The neural interaction strength r_i encapsulates the intensity of the interaction within the hyperedge and is obtained through a trainable model $\mathcal{F}_r$, applied to a collective embedding z_i with a sigmoid function σ as Eq.(21). This collective embedding z_i is represented as the weighted sum of the individual node embeddings within the hyperedge, signifying the aggregated information of agents in a group as Eq. (22). The weight w_j for each node is determined by a trainable MLP $\mathcal{F}_w$ as Eq.23.

$$r_i = \sigma(\mathcal{F}_r(z_i)) \tag{21}$$

$$z_i = \sum_{v_j \in u_i} w_j v_j \tag{22}$$

$$w_j = \mathcal{F}_w\left(v_j, \sum_{v_m \in u_i} v_m\right) \tag{23}$$

The neural interaction category coefficient c_i represent the model's reasoning about which type of the interaction is likely for hyperedge u_j, where $c_{i,l}$ denotes the probability of the l-th neural interaction category within L possible categories. These coefficients are computed using a softmax function applied to the output of another trainable MLP $\mathcal{F}_c$, which is further adjusted by an i.i.d. sample Gumbel distribution g as described in Appendix C, which add some random noise and a temperature parameter τ which controlls the smoothness of probability distribution [45]:

$$c_i = \text{softmax}\left(\frac{\mathcal{F}_c(z_i) + g}{\tau}\right) \tag{24}$$

These components, including neural interaction strength, interaction category coefficients, and per-category functions, provide a comprehensive mechanism for reasoning over complex, higher-order relationships, allowing the model to adapt its understanding of how agents collectively behave in diverse scenarios.

The process of hyperedge-to-node mapping is a pivotal step that allows for the update and refinement of agent embeddings within the hypergraph framework. Each hyperedge u_j is mapped back onto its constituent nodes v_i, assuming every v_i is included in u_j. The primary objective of this phase is to update the embedding of an agent. This is achieved through the function $\mathcal{F}_v$, which is a trainable MLP. The updated agent embedding $\tilde{v}_i$ is the result of the function applied to the concatenation of the agent's current embedding and the sum of the embeddings of all hyperedges that the agent is part of. Formally, the update rule for the agent embedding is represented as:

$$\tilde{v}_i \leftarrow \mathcal{F}_v\left(\left[\, v_i, \sum_{u_j \in \mathcal{U}_i} u_j \,\right]\right) \tag{25}$$

where $\mathcal{U}_i = \{u_j \,|\, v_i \in u_j\}$ denotes the set of hyperedges associated with the i-th node v_i, and $[\,\cdot,\cdot\,]$ symbolize the operation of embedding concatenation. This operation fuses the individual node embedding with the collective information conveyed by the associated hyperedges. This amalgamation is crucial as it encapsulates the influence exerted by the interactions within the hyperedges onto the individual agent.

The Hypergraph Relational Encoder applies the node-to-hyperedge and hyperedge-to-node phases iteratively, allowing agent embeddings to be refined and enriched as relationships within hyperedges evolve. Upon the completion of these iterations, the output is constructed as the concatenation of the agent embeddings across all scales. The final agent embedding matrix $\tilde{\mathbf{V}}^a$ is composed of the embeddings of all agents, where each agent embedding v_i is a concatenation of the embeddings from all scales, expressed as:

$$\tilde{\mathbf{V}}^a = [\tilde{v}_i] \in \mathbb{R}^{N \times |d(S+1)|}, \quad \forall i \in [1, \ldots, N] \tag{26}$$

where

$$\tilde{v}_i = [\tilde{v}_i^{(0)}, \tilde{v}_i^{(1)}, \ldots, \tilde{v}_i^{(S)}] \tag{27}$$

3.3.3. Posterior Distribution Learner

In our study, we incorporated multi-scale hypergraph embeddings into a multi-agent trajectory generation system using the CVAE framework [46] to address the stochastic nature of each agent's behavior, as shown in Figure 10. Here, we denote the historical trajectories $\mathbf{X}_{1:T}$ as $\mathbf{X}_T$, and denote the predicted future trajectories $\mathbf{X}_{T+1:T+F}$ as $\mathbf{X}_F$. Let $\log p(\mathbf{X}_F|\mathbf{X}_T)$ denote the log-likelihood of predicted future trajectories $\mathbf{X}_F$ given historical trajectories $\mathbf{X}_T$. The corresponding Evidence Lower Bound (ELBO) is defined as follows:

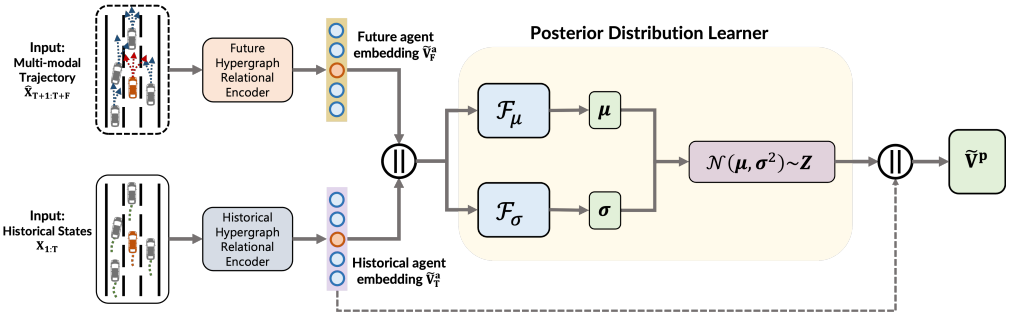

Figure 10: Posterior Distribution Learner.

$$\begin{aligned} \log p(X_F|X_T) \geq \mathbb{E}_{q(\mathbf{Z}|\mathbf{X}_F,\mathbf{X}_T)} \log p(\mathbf{X}_F|\mathbf{Z},\mathbf{X}_T) \\ - \mathrm{KL}(q(\mathbf{Z}|\mathbf{X}_F,\mathbf{X}_T) \parallel p(\mathbf{Z}|\mathbf{X}T)), \end{aligned} \tag{28}$$

where $\mathbf{Z} \in \mathbb{R}^{N \times d_z}$ represents the latent codes corresponding to all agents; $p(\mathbf{Z}|\mathbf{X}_T)$ is the conditional prior of $\mathbf{Z}$, modeled as a Gaussian distribution. KL represents the Kullback–Leibler divergence function. In this framework, $q(\mathbf{Z}|\mathbf{X}_F,\mathbf{X}_T)$ is implemented through an encoding process for embedding learning, and $p(\mathbf{X}_F|\mathbf{Z},\mathbf{X}_T)$ is realized via a decoding process that forecasts the future trajectories $\mathbf{X}_F$.

Thus, the goal of the Posterior Distribution Learner is to derive the Gaussian parameters for the approximate posterior distribution. This involves computing the mean μ_q and the variance σ_q based on the final output embeddings $\tilde{\mathbf{V}}_F^a$ and the target embeddings $\tilde{\mathbf{V}}_T^a$. These parameters are generated through two separate trainable MLPs, $\mathcal{F}_\mu$ and $\mathcal{F}_\sigma$, respectively. The latent code Z, representing possible trajectories, is then sampled from a Gaussian distribution parameterized by these means and variances. The final output embeddings $\tilde{\mathbf{V}}^p$ are a concatenation of the latent code $\mathbf{Z}$, the final output embeddings $\tilde{\mathbf{V}}_F^a$, and the target embeddings $\tilde{\mathbf{V}}_T^a$. The equations governing these processes are as follows:

$$\mu_q = \mathcal{F}_\mu(\tilde{\mathbf{V}}_F^a, \tilde{\mathbf{V}}_T^a) \tag{29}$$

$$\sigma_q = \mathcal{F}_\sigma(\tilde{\mathbf{V}}_F^a, \tilde{\mathbf{V}}_T^a) \tag{30}$$

$$\mathbf{Z} \sim \mathcal{N}(\mu_q, \mathrm{Diag}(\sigma_q^2)) \tag{31}$$

$$\tilde{\mathbf{V}}^p = [\mathbf{Z}, \tilde{\mathbf{V}}_T^a] \tag{32}$$

In these notations, μ_q and σ_q represent the mean and variance of the approximated posterior distribution. $\mathcal{F}_\mu$ and $\mathcal{F}_\sigma$ are the trainable MLPs that produce these parameters. $\mathbf{Z}$ denotes the latent code of possible trajectories, and $\tilde{\mathbf{V}}^p$ stands for the output embeddings, which fuses the latent code and the historical embeddings.

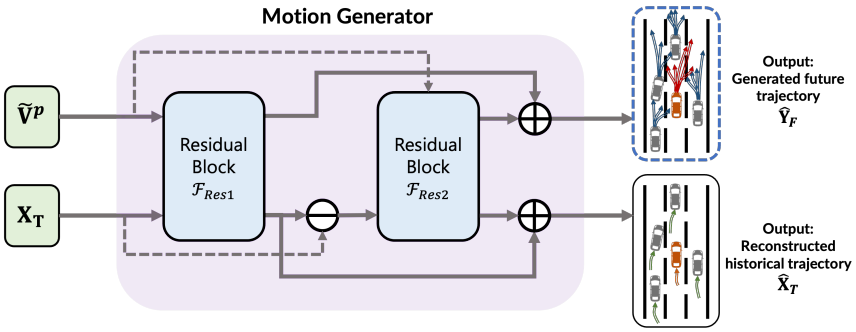

Figure 11: Motion Generator.

3.3.4. Motion Generator

The Motion Generator's objective is dual: to predict future trajectories and to reconstruct past trajectories from the given embeddings. The decoder accomplishes this by applying successive processing blocks, each contributing a residual that refines the trajectory estimates, as shown in Figure 11. The first processing block, $\mathcal{F}_{Res1}$, takes the output embeddings $\tilde{\mathbf{V}}^p$ and the target past trajectory $\mathbf{X}_T$ to generate initial estimates of the future and reconstructed past trajectories $\hat{\mathbf{X}}_{F,1}$ and $\hat{\mathbf{X}}_{T,1}$ respectively.

$$\hat{\mathbf{X}}_{F,1}, \hat{\mathbf{X}}_{T,1} = \mathcal{F}_{Res1}(\tilde{\mathbf{V}}^p, X_T) \tag{33}$$

Subsequently, the second block, $\mathcal{F}_{Res2}$, refines these estimates by considering the output embeddings and the residual of the past trajectory, which is the difference between the target past trajectory and the initial reconstructed past trajectory $\mathbf{X}_T - \hat{\mathbf{X}}_{T,1}$. This results in the second set of residuals $\hat{\mathbf{X}}_{F,2}$ and $\hat{\mathbf{X}}_{T,2}$:

$$\hat{\mathbf{X}}_{F,2}, \hat{\mathbf{X}}_{T,2} = \mathbf{X}_{Res2}(\tilde{\mathbf{V}}^p, \mathbf{X}_T - \hat{\mathbf{X}}_{T,1}) \tag{34}$$

Both $\mathcal{F}_{Res1}$ and $\mathcal{F}_{Res2}$ are composed of a GRU encoder for sequence encoding and two MLPs serving as the output header. The final predicted future trajectory $\hat{\mathbf{Y}}_F$ and the reconstructed past trajectory $\hat{\mathbf{X}}_T$ are obtained by summing the respective residuals from both processing blocks:

$$\hat{\mathbf{Y}}_F = \hat{\mathbf{X}}_{F,1} + \hat{\mathbf{X}}_{F,2} \tag{35}$$

$$\hat{\mathbf{X}}_T = \hat{\mathbf{X}}_{T,1} + \hat{\mathbf{X}}_{T,2} \tag{36}$$

This approach enables the model to iteratively refine its predictions and reconstructions, leveraging the capability of deep learning models to capture complex patterns in the data through a series of non-linear transformations.

4. Experiments and Results

4.1. Data Preparations

This research leverages two open-source datasets for the purpose of model training and validation: the Next Generation Simulation (NGSIM) dataset [47],[48] and the HighD dataset [49]. The NGSIM dataset provides a comprehensive collection of vehicle trajectory data, capturing activity from the eastbound I-80 in the San Francisco Bay area and the southbound US 101 in Los Angeles. This dataset encapsulates real-world highway scenarios through overhead camera recordings at a sampling rate of 10Hz. The HighD dataset originates from aerial drone recordings executed at a 25 Hz frequency between 2017 and 2018 in the vicinity of Cologne, Germany. Spanning approximately 420 meters of bidirectional roadways, it records the movements of approximately 110,000 vehicles, encompassing both cars and trucks, traversing an aggregate distance of 45,000 km. After data pre-processing, the NGSIM dataset encompasses 662 thousand rows of data, capturing 1,380 individual trajectories, while the HighD dataset comprises 1.09 million data entries, including 3,913 individual trajectories. For the purpose of training and evaluation of the model, the partition of the data allocates 70% to the training set and 30% to the test set. For the temporal parameters of the model, we adopt $T = 30$ frames to represent the historical horizon and $F = 50$ frames to signify the prediction horizon.

4.2. Training and Evaluation Metrics

Training loss of GIRAFFE. The training loss function for `GIRAFFE` is a summation of three terms:

$$\mathcal{L}^{PRED} = \mathcal{L}_{pred} + \mathcal{L}_{int} + \mathcal{L}_{fut} \tag{37}$$

The first component, $\mathcal{L}_{pred}$, is the mean squared error (MSE) between the fused predicted trajectory and the ground truth future trajectory:

$$\mathcal{L}_{pred} = \|\hat{\mathbf{Y}}_F - \mathbf{Y}_F\|_2^2 \tag{38}$$

The second component $\mathcal{L}_{int}$ represents the negative log-likelihood (NLL) of the predicted driving intentions, treating it as a classification task:

$$\mathcal{L}_{int} = NLL(\hat{M}; M) = -\sum_{m \in M} m \log P(\hat{m}|\mathbf{X}_T) \tag{39}$$

The third component $\mathcal{L}_{fut}$ is the loss associated with the inference of the future-guided graph feature matrix, which is an intermediate output:

$$\mathcal{L}_{fut} = \|\hat{H}_F - H_F\|_2^2 \tag{40}$$

Training loss of RHINO. The training loss function for RHINO is also a summation of three components:

$$\mathcal{L}^{GEN} = \mathcal{L}_{elbo} + \mathcal{L}_{recon} + \mathcal{L}_{var} \tag{41}$$

The first component, $\mathcal{L}_{elbo}$, corresponds to the ELBO loss [46] commonly used in variational autoencoders. It consists of a reconstruction loss term and a regularization term based on the Kullback-Leibler divergence between the learned distribution and a prior distribution:

$$\mathcal{L}_{elbo} = \alpha \|\hat{\mathbf{Y}}_F - \mathbf{Y}_F\|_2^2 + \beta \mathrm{KL}\left(\mathcal{N}\left(\mu_q, \mathrm{Diag}(\sigma_q^2)\right) \,\middle\|\, \mathcal{N}(0, \lambda I)\right) \tag{42}$$

The second component, $\mathcal{L}_{recon}$, represents the Historical Trajectory Reconstruction loss, which measures how accurately the reconstructed historical trajectories match the true historical data:

$$\mathcal{L}_{recon} = \lambda \|\hat{\mathbf{X}}_T - \mathbf{X}_T\|_2^2 \tag{43}$$

The final component, $\mathcal{L}_{var}$, is the Variety loss, inspired by Social-GAN [21]. This loss encourages diversity in the predicted future trajectories by minimizing the error across multiple sampled future trajectories:

$$\mathcal{L}_{var} = \min_k \|\hat{\mathbf{Y}}_F^{(k)} - \mathbf{Y}_F\|_2^2 \tag{44}$$

Table 1 presents the hyperparameter configurations used for the network architecture and the training process in the proposed framework.

Table 1: Hyperparameter Settings

Parameter	Value	Parameter	Value
T	30	decaying factor	0.6
F	50	α	1
neuron # of MLPs	128	β	0.8
learning rate	0.001	λ	0.5

Evaluation metrics To ascertain the predictive accuracy of the model, we employ the Root Mean Square Error (RMSE) as the evaluative criterion. This metric quantitatively measures the deviation between the predicted position, expressed as $(\hat{y}^l_{f,lat}, \hat{y}^l_{f,lon})$, and the ground truth position, indicated by $(y^l_{f,lat}, y^l_{f,lon})$ for all time step f within the predictive horizon prediction horizon $[T+1, T+F]$.

$$RMSE = \sqrt{\frac{1}{LF}\sum_{l=1}^{L}\sum_{f=T+1}^{T+F}\left((\hat{y}^l_{f,lat} - y^l_{f,lat})^2 + (\hat{y}^l_{f,lon} - y^l_{lon})^2\right)} \tag{45}$$

where the superscript l denotes the l-th test sample from the aggregate test sample set with length L.

4.3. *Results of Trajectory Generation*

The experimental results for trajectory generation of the K trajectories using the HighD dataset are presented in Figure 12. As can be found that, RHINO demonstrates strong generative capabilities, effectively producing plausible motion in a dynamic interactive traffic environment. To provide a more quantitative analysis, trajectory generation inaccuracies are illustrated in Figure 13. The generated longitudinal and lateral trajectories, along with the error box plots and heatmaps, are displayed. The box plot reveals that errors in both axes increase with the prediction time step by the natural of error propagation. However, the errors remain within an acceptable range, indicating decent model performance, which demonstrates high precision in trajectory generation. Notably, the model maintains a lower error margin for shorter prediction horizons, which is critical for short-term planning and reactive maneuvers in dynamic traffic environment.

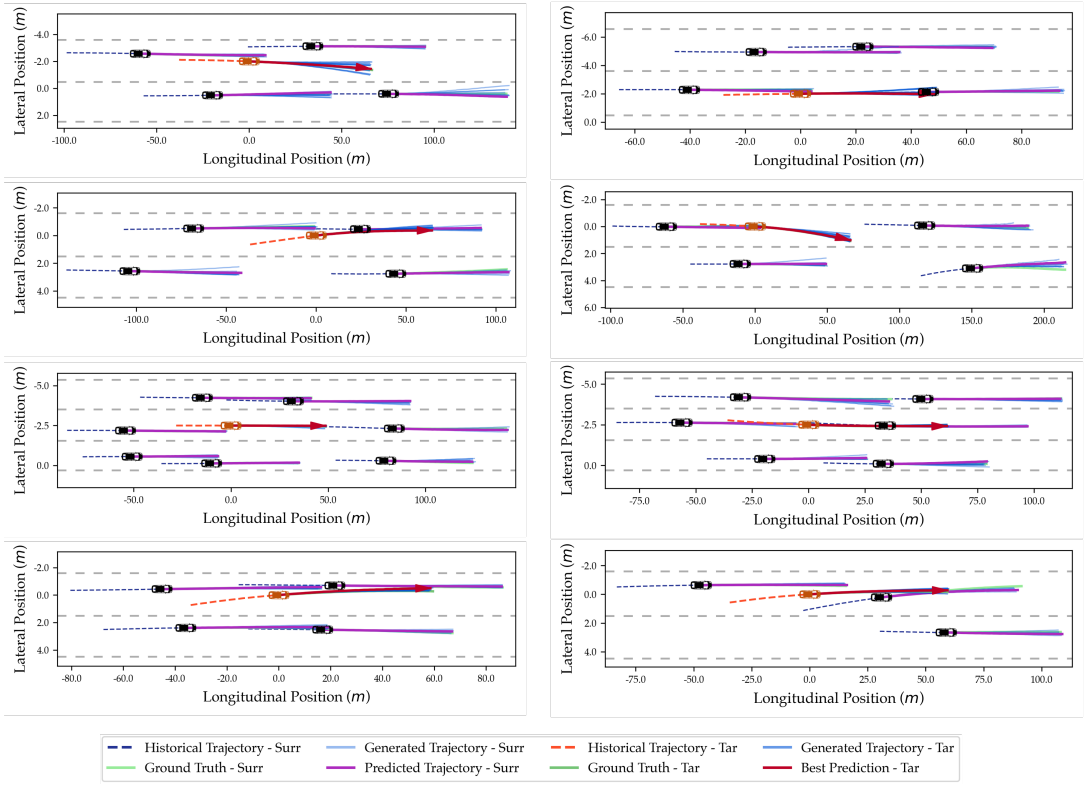

Figure 12: Trajectory generation results in highway scenarios.

The experiment focused on predicting vehicle trajectories within mixed traffic environments on highways by employing hypergraph inference to model group-based interactions. Through the application of hypergraph models at varying scales $s = 2, 3, 5$, the experiment captured the evolution of multi-vehicle interactions across both historical and future horizons. The figures depict these dynamics through three distinct columns: the first column presents vehicle trajectories

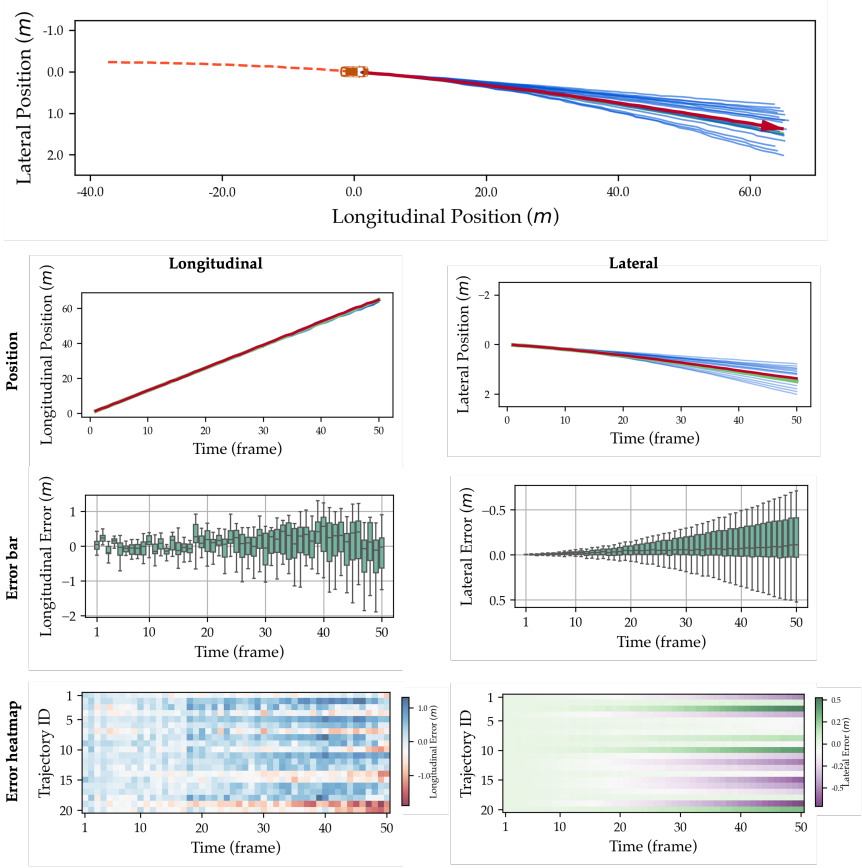

Figure 13: Longitudinal and lateral trajectory generation error analysis.

and the corresponding hyperedges, visualized as polygons that encapsulate groups of interacting vehicles. The second column illustrates the affinity matrix, where both rows and columns represent vehicles, and the strength of their relationships is indicated by the matrix values. The third column shows the incidence matrix, detailing the relationship between nodes and hyperedges, with each column representing a hyperedge and each vehicle's involvement in that hyperedge marked by a 1 in the corresponding row.

The hypergraph-based approach is particularly effective in modeling complex, higher-order interactions that are beyond the scope of traditional pairwise models. By forming hyperedges that encompass multiple vehicles, the model captures the collective influence that a group's behavior exerts on an individual vehicle. For instance, in the scenario shown in Figure 14, at scale $s = 5$, when the target vehicle TAR initiates a lane change, the hypergraph reflects the interaction not only with a single neighboring vehicle but also with multiple surrounding vehicles, such as following vehicle F, preceding vehicle P, and preceding vehicle RP in the right lane. More examples are illustrated in Appendix A. This capability to model group-wise interactions across different scales is

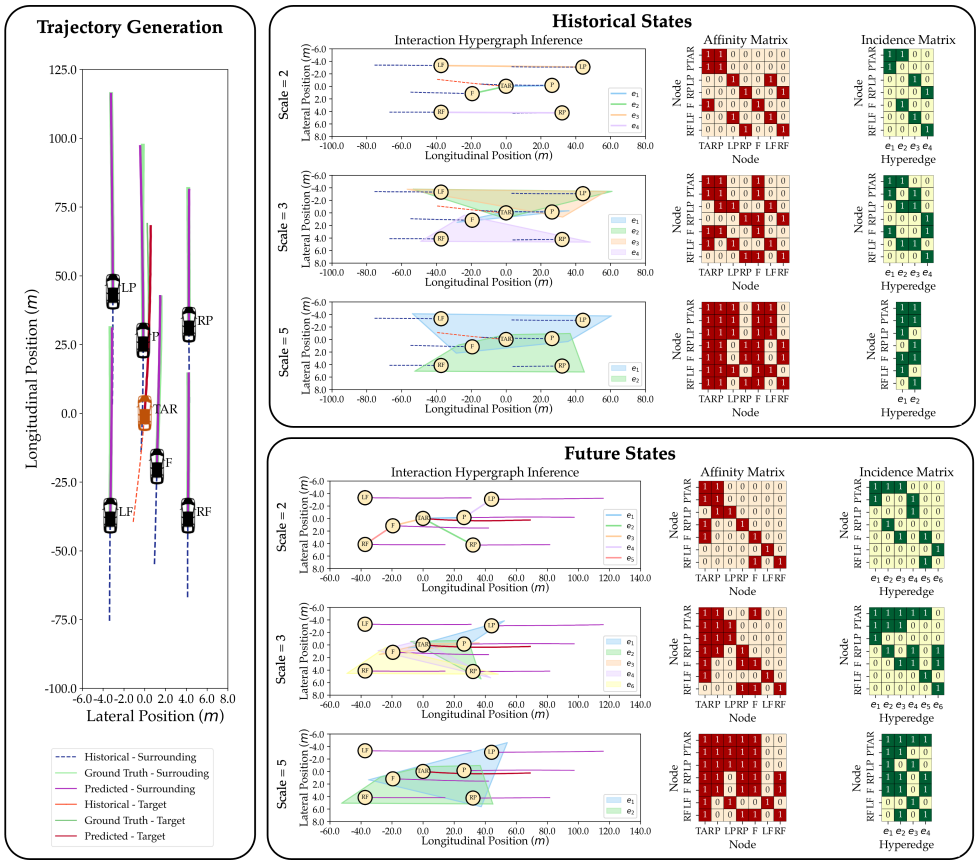

Figure 14: Trajectory generation with hypergraph inference.

essential for accurately predicting vehicle trajectories in congested highway environments, where the actions of one vehicle can trigger ripple effects that influence an entire group. The hypergraph's dynamic formation of hyperedges ensures that predicted trajectories remain adaptable and responsive to broader traffic conditions.

4.4. Comparisons and Ablation Study

To evaluate the privileges of our proposed method, the state of art methods (i.e., Social-LSTM (S-LSTM) [20], Convolutional Social-LSTM (CS-LSTM) [31], Planning-informed prediction (PiP) [50], Graph-based Interaction-aware Trajectory Prediction (GRIP) [26], Spatial-temporal dynamic attention network (STDAN) [22] are compared.

The compared results presented in Table 2 and Figure 15. As can be found that, the proposed framework demonstrates good performance with respect to the RMSE across a prediction horizon of 50 frames when compared with existing baseline models. It exhibits a reduced loss in comparison to C-LSTM, CS-LSTM, PiP, and GRIP. These outcomes suggest that the proposed model

effectively captures salient features pertinent to long-term predictions. In summary, the proposed framework outperforms baseline models on the HighD dataset and delivers commendable performance on the NGSIM dataset.

Since the RHINO adopts the GIRAFFE, we further compare the trajectory generation capability of RHINO with our previous work [25] and its enhanced version GIRAFFE. Both RHINO model and the enhanced GIRAFFE model consistently outperform the baseline models, demonstrating superior performance in various metrics. This suggests that our proposed approaches effectively address the limitations present in prevailing models by robustly capturing complex interactions.

Table 2: Prediction Error Obtained by Different Models in RMSE (m)

Dataset	Horizon (Frame)	S-LSTM	CS-LSTM	PiP	GRIP	STDAN	GIRAFFE	RHINO
NGSIM	10	0.65	0.61	0.55	0.37	0.42	0.38	**0.32**
	20	1.31	1.27	1.18	0.86	1.01	0.89	**0.78**
	30	2.16	2.08	1.94	1.45	1.69	1.45	**1.34**
	40	3.25	3.10	2.88	2.21	2.56	2.46	**2.17**
	50	4.55	4.37	4.04	3.16	3.67	3.24	**2.97**
HighD	10	0.22	0.22	**0.17**	0.29	0.19	0.19	0.19
	20	0.62	0.61	0.52	0.68	0.27	0.42	**0.26**
	30	1.27	1.24	1.05	1.17	0.48	0.81	**0.42**
	40	2.15	2.10	1.76	1.88	0.91	1.13	**0.65**
	50	3.41	3.27	2.63	2.76	1.66	1.56	**0.89**

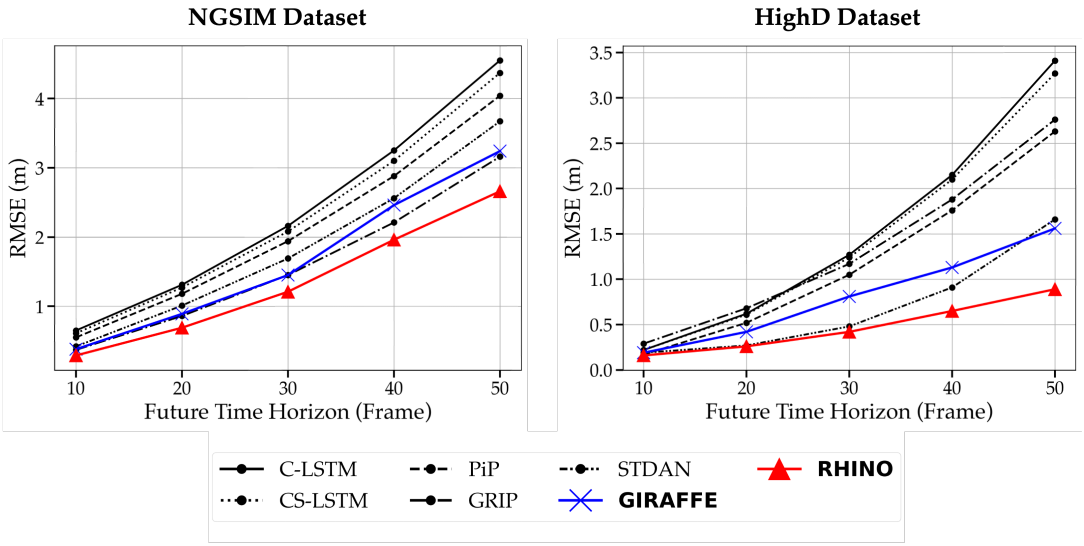

Figure 15: Prediction error obtained by different models in RMSE on NGSIM dataset (left) and HighD dataset (right).

Ablation study is conducted to provide more insights into the performance of our RHINO model, especially the impact of different components on the prediction performance by disabling the

Table 3: Ablation Test Results of RHINO in RMSE (m)

Horizon (Frame)	RHINO w/o HG	RHINO w/o MM	RHINO w/o PDL	RHINO
10	0.21	0.22	0.24	**0.19**
20	0.31	0.37	0.42	**0.26**
30	0.68	0.73	0.80	**0.42**
40	0.97	1.06	1.18	**0.65**
50	1.25	1.34	1.57	**0.89**

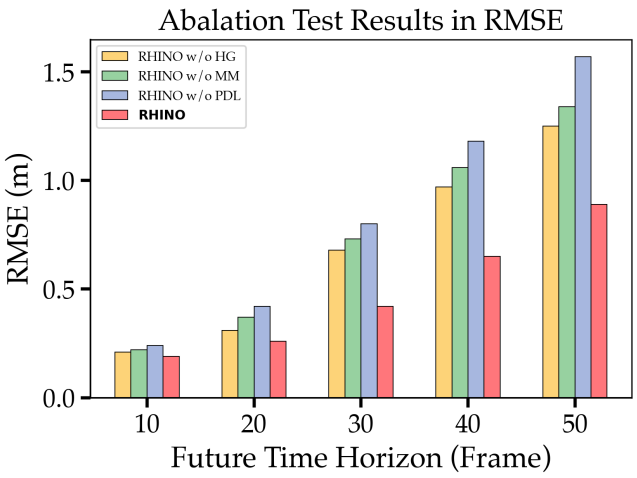

Figure 16: Ablation Study of RHINO.

corresponding component from the entire RHINO. In particular, we consider the following four variants:

- RHINO w/o HG (hypergraph) variant does not use the multi-scale hypergraphs representation but only adopts the pair-wise connected graph representations in the Hypergraph Relational Encoder.
- RHINO w/o MM (multi-modal) variant does not adopt the multi-agent multi-modal trajectory prediction results and only use the single predicted future states for each agent as the input of the RHINO.
- RHINO w/o PDL (posterior distribution learner) variant skips the Posterior Distribution Learner and directly input the graph embedding into the Motion Generator.

An investigation into the effects of model design variations, as presented in Table 3 and Figure 16. The removal of various components from RHINO invariably leads to performance degradation to varying degrees. Compared to the full RHINO, omitting the multi-scale hypergraphs results in

an evident increase in prediction error across the prediction horizon, indicating that modeling and reasoning group-wise interactions using hypergraphs, rather than solely pair-wise interactions, enhances prediction accuracy which underscores the necessity of hypergraphs. Further, excluding the multi-agent multi-modal trajectory prediction input leads to a more substantial degradation in performance, highlighting the importance of incorporating multi-modal motion states and discussing the corresponding group-wise interactions among multiple driving behaviors of multiple agents. Lastly, the absence of the Posterior Distribution Learner module emphasizes its critical role in handling the stochasticity of each agent's behavior. All these experiments justify the effectiveness of the full model.

5. Conclusions

In this study, we proposed a hypergraph enabled multi-modal probabilistic motion prediction framework with reasonings. This framework consists of two main components: `GIRAFFE` and `RHINO`. `GIRAFFE` focuses on predicting the interactive vehicular trajectories considering modalities. Based on that, `RHINO`, leveraging the flexibility and strengths on modeling the group-wise interactions, facilitate relational reasoning among vehicles and multi-modalities to render plausible vehicles trajectories. The framework extends traditional interaction models by introducing an agent-behavior hypergraph. This approach better aligns with traffic physics while being grounded in the mathematical rigor of hypergraph theory. Further, the approach employs representation learning to enable explicit interaction relational reasoning. This involves considering future relations and interactions and learning the posterior distribution to handle the stochasticity of behavior for each vehicle. As a result, the framework excels in capturing high-dimensional, group-wise interactions across various behavioral modalities.

The framework is tested using the NGSIM and HighD datasets. The results show that the proposed framework effectively models the interactions among groups of vehicles and their corresponding multi-modal behaviors. Comparative studies demonstrate that the framework outperforms prevailing algorithms in prediction accuracy. To further validate the effectiveness of each component, ablation studies were conducted, revealing that the full model performs best.

Several potential extensions of the framework include incorporating road geometries, vehicle types, and real-time weather data to improve trajectory prediction. By integrating weather information from sources like the OpenWeather API, the system could adjust predictions based on conditions such as temperature, wind, and precipitation, enhancing safety and route optimization [51]. Additional enhancements, like traffic signal integration, V2V and V2I communication, and human driver intent, could further improve accuracy and reliability in dynamic urban environments, minimizing disruptions and fostering safer, more informed autonomous driving.

Acknowledgement:

This research is funded by Federal Highway Administration (FHWA) Exploratory Advanced Research 693JJ323C000010. The results do not reflect FHWA's opinions.