This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

KITNet: A Region-Attention-Activated Trajectory Predictor With Hierarchical Graph Neural Network in Dynamic-Mutant Multi-Agent System

Zeyu Zhou, Shanqing Wang, Anmin Huang, Jin Lou, Wei Tang, *Member, IEEE*, and David Navarro-Alarcon, *Senior Member, IEEE*

***Abstract*—Safe and efficient operation of Autonomous Delivery Vehicles (ADVs) in dynamic multi-agent environments, such as university campuses and industrial parks, necessitates accurate trajectory prediction of interacting agents. Conventional autonomous navigation systems, however, rely heavily on reactive real-time perception and often fail to predict complex spatio-temporal interactions among heterogeneous agents. This limitation frequently leads to suboptimal motion planning outcomes and operational inefficiencies, including deadlock situations, in congested scenarios. This paper introduces Knowledge-Interaction-Temporal Network (KITNet), a novel trajectory prediction framework specifically designed for ADVs operating in such complex, dynamic settings. KITNet employs a hierarchical Graph Neural Network (GNN) architecture to model intricate interaction dynamics, incorporating a novel attention mechanism based on set theory for enhanced spatio-temporal feature extraction and prediction of diverse behavior patterns. We evaluate KITNet on several trajectory prediction benchmarks according to the different tailored behavior modes under the defined mode space, including the ETH/UCY pedestrian dataset, the NGSIM highway driving dataset, and the Argoverse 2 urban driving dataset. Our results demonstrate state-of-the-art prediction accuracy, outperforming or matching existing graph-based and recurrent approaches. Furthermore, we discuss the integration of KITNet's predictive outputs into local motion planning modules, showing potential for significantly reducing conflict scenarios and optimizing trajectory execution for ADVs. These findings establish KITNet as a highly effective trajectory predictor for autonomous transport systems, critically advancing predictive navigation and bridging the gap between perception and robust intelligent decision-making in complex urban and campus environments.**

***Index Terms*—Spatiotemporal prediction, regional set theory, modeling of shape mapping, hierarchical graph neural network, temporal neural network.**

Received 29 December 2024; revised 21 May 2025 and 4 October 2025; accepted 11 November 2025. This work was supported in part by the Key Research and Development Program of Shaanxi under Grant 2024GX-ZDCYL-02-06 and Grant 2024CY2-GJHX-91. The Associate Editor for this article was Z. Li. *(Corresponding authors: Wei Tang; David Navarro-Alarcon.)*

Zeyu Zhou is with the School of Automation, Northwestern Polytechnical University, Xi'an 710129, China, and also with the Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hong Kong.

Shanqing Wang is with the School of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China.

Anmin Huang, Jin Lou, and Wei Tang are with the School of Automation, Northwestern Polytechnical University, Xi'an 710129, China (e-mail: tangwei@nwpu.edu.cn).

David Navarro-Alarcon is with the Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hong Kong (e-mail: david.navarro-alarcon@polyu.edu.hk).

This article has supplementary downloadable material available at https://doi.org/10.1109/TITS.2025.3635237, provided by the authors.

Digital Object Identifier 10.1109/TITS.2025.3635237

I. Introduction

AUTONOMOUS Delivery Vehicles (ADVs), functioning as Automated Guided Vehicles (AGVs) within logistics systems, are increasingly deployed in complex, dynamic multi-agent environments such as university campuses and industrial parks. Safe and efficient navigation in these domains necessitates robust capabilities to handle both static obstacles and the intricate interactions with dynamic agents, including pedestrians, cyclists, and other vehicles. Conventional autonomous navigation stacks, exemplified by standard Robot Operating System (ROS) or other implementations, predominantly rely on reactive collision avoidance. These systems leverage real-time perception data (e.g., from LiDAR or cameras) to detect immediate obstacles and execute reactive maneuvers like speed adjustments or path deviations. While effective at preventing imminent collisions, purely reactive strategies (e.g., Dynamic Window Approach (DWA), Velocity Obstacles (VO)) inherently lack the ability to anticipate future interactions and behaviors of surrounding agents. This limitation frequently leads to conservative, inefficient, or unstable operational outcomes, particularly in congested and constrained spaces. For instance, in a narrow pathway with mixed traffic, reactive responses among an ADV, pedestrians, bicycles and vehicles can result in late-stage maneuvers and operational deadlocks, severely degrading throughput. Consequently, a critical need exists for integrating predictive capabilities into the navigation pipeline. By accurately forecasting the future states and intended trajectories of dynamic agents, an ADV can execute proactive motion planning, preemptively identifying and resolving potential conflicts, optimizing its path and timing, and ultimately enhancing both safety and operational efficiency. This proactive approach is essential, especially considering the non-holonomic constraints and inertia inherent in vehicles, which preclude instantaneous reactions to unexpected environmental changes.

In the realm of trajectory prediction for vehicles and pedestrians, various research directions have been explored. Early approaches relied on complex kinematic models to forecast trajectories, utilizing tools like Kalman filters and context-switching linear dynamic systems [1], [2]. These models typically assumed that non-cooperative targets followed fixed behavior patterns. However, in reality, these targets are influenced not only by their own dynamic constraints but also by interactions with their surroundings. The computational intensity and lack of flexibility of kinematic models hinder

1558-0016 © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on December 10,2025 at 03:30:12 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

their ability to fully capture the interaction between vehicles and pedestrians.

To address these limitations, recent years have seen a surge in data-driven trajectory prediction research. With bypassing the cumbersome kinematic modeling process [3], [4], [5], [6], [7], [8], these methods can implicitly learn pedestrian's movement patterns and interactions from large datasets of known trajectories. In 1995, Helbing and Molnar [9] pioneered the study of pedestrian interaction behavior to predict future trajectories. However, these methods aimed to predict smooth trajectories, but real-world trajectories are often disrupted by sharp turns and sudden stops, necessitating models that capture more intricate features. This led to the advent of deep learning models. One of the pioneering works in this domain was the Social-LSTM model developed by Alahi [10], which applied deep learning to pedestrian trajectory prediction. Social-LSTM uses Long Short-Term Memory (LSTM) networks as encoders to extract feature states from pedestrian historical trajectories and employs a pooling mechanism to aggregate these features from neighboring pedestrians. The aggregated features are then decoded by LSTM decoders to generate future trajectories. The Social-affinity LSTM [11] improved on this by assigning different weights to encoded feature states based on the relative distances between pedestrians and their neighbors. In Social-GAN [12], the generator module follows the Social-LSTM's pooling mechanism for initial trajectory prediction, while adversarial training between the generator and discriminator modules refines the accuracy of the generated trajectories.

However, the aforementioned methods, based on Recurrent Neural Networks (RNN) [13], inherently lack the ability to capture high-level spatial-temporal structures, making it challenging to accurately model spatial features in temporal data. Using graph topologies to represent real-world relationships allows for the reflection of spatial interactions among agents through node interactions. In recent years, Graph Neural Network (GNN) and their variants [14], [15], [16], [17], including Graph Convolutional Network (GCN) and Graph Attention Network (GAT) with spatiotemporal characteristics, have been applied to pedestrian trajectory prediction, demonstrating competitive results. Zhang designed a directed social graph [18] based on real-time position and velocity direction to capture social behaviors within groups. Huang proposed the Spatial-Temporal Graph Attention (STGAT) network [19] to predict future pedestrian trajectories. STGAT employs a graph attention mechanism to capture spatial interactions encoded by LSTM at each time step, and an additional LSTM encodes the temporal correlations of these interactions. Social-BiGAT [20], similar to STGAT, uses a GAN to train the model. The latest model, PPT [21], further refines this attention mechanism, distinguishing short-term perturbations and long-term dependencies among agents more effectively. Unlike these models, Social-STGCNN [22] directly models pedestrian trajectories as a graph and designs a kernel function to represent interactions with surrounding pedestrians, followed by a decoder to generate predicted trajectories. Recent research on GNNs has provided powerful tools for representing complex relationships. For example, Li et al. employ a framelet transform in multimodal GNNs to capture multi-scale conversational cues (FrameERC) [23], and Bai et al. propose quantum kernel-based graph representations for classification (HAQJSK) [24]. The broad impact of deep neural networks on graph-structured data across domains is highlighted by Li et al. in Guest Editorial [25]. These advances inspire our graph-based trajectory predictor, which addresses unique challenges in traffic scenes.

Although these models have seen some application, they mainly address short-term, limited multi-agent interaction scenarios. In real-life situations, traditional vehicles or pedestrians often exhibit behaviors such as moving against traffic, sudden lane changes, sharp turns, or abrupt stops, leading to highly complex interactions [26], [27], [28], [29], [30]. These phenomena are crucial for ensuring the stable and safe operation of multi-agent systems. Therefore, it is essential not only to gradually characterize these interactions but also to find ways to simplify and highlight them, making them more discernible and easier for AI models to learn effectively.

To address this, we propose a hierarchical GNN architecture tailored for dynamic-mutant multi-agent scenarios. This architecture comprises a self-trajectory sub-network, multiple interaction sub-networks, and an end-point interaction sub-network. Outputs from the self-trajectory sub-network are fed into the interaction sub-networks, and outputs from these networks are further processed by a temporal convolutional network. For all instances within the first and third types of networks, we establish two methods for calculating the attention weights of the edges between network nodes. Utilizing region and set theory, we derive and present criteria for selecting between these calculation methods. The first method, a conventional approach, considers only the relative distance and velocity between agents. The second, a specialized method, addresses 18 specific branches corresponding to six types of abrupt behavior patterns, factoring in regional indicators, relative distance, speed, and angular offset between agents, setting attention functions via scatter fitting. This effectively resolves the current challenges in modeling the complex and varied potential interactions among multiple agents.

The primary innovations and contributions of this research can be summarized as follows:

1. Hierarchical GNN Architecture for Abrupt Behavior Patterns: We categorize abrupt behaviors into six distinct patterns and design a hierarchical GNN architecture for each. The interaction network is split into terminal and non-terminal networks, revealing that the terminal interaction patterns possess unique latent features needing extraction. This approach is rare in interaction networks, distinguishing and reorganizing sampled points at different timestamps.
2. Novel Attention Mechanism via Region and Set Theory: We introduce a groundbreaking method using region and set theory to trigger attention conditions between trajectory points. Five types of regional expansion mappings are established, including models for two special mappings. The intersections of these high-dimensional mapped regions serve as the condition for triggering new

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on December 10,2025 at 03:30:12 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

attention weight calculation functions between GNN nodes.

3. Edge Attention Weights Using Region and Speed-Angle Indicators: We propose using regional and speed-angle indicators to represent edge attention weights in GNNs, bridging these concepts through spatiotemporal optimization and relational inference strategies. To our knowledge, this is the first mature application of set theory in modeling GNN attention functions.
4. Enhanced Predictive Performance with KITNet: Unlike models such as Social-LSTM, Social-GAN, and Social-STGCNN that integrate extensive historical data with social behavioral analysis for trajectory prediction, our KITNet model focuses on the agent's current state and interactions. It maximally leverages limited present information, achieving more timely and effective predictions. Consequently, KITNet does not require extensive multimodal historical data, significantly improving its practical applicability and extending its utility to other data predictions involving complex motivations and behaviors.

II. Preliminaries

In this study, we integrate region theory with set theory to analyze agent interactions. Here, a "region" represents a high-dimensional domain derived from an agent's current position, mapped through lines or curves, while "sets" characterize the intersections between these high-dimensional domains.

To formalize this framework, we define an n-dimensional space ($\mathbb{R}^n$) equipped with corresponding regional extension mappings. By analyzing the intersections of these mapped regions among agents, we establish criteria to identify and trigger abrupt behavioral changes. This approach offers a systematic perspective for modeling and predicting complex multi-agent dynamics.

1) Agent Representation in n-Dimensional Space:

a) Position vector: Each agent A_i is located at:

$$\mathbf{p}_i = (x_{i1}, x_{i2}, \cdots, x_{in})^T \in \mathbb{R}^n \tag{1}$$

b) Heading direction: A unit vector

$$\mathbf{u}_i = (u_{i1}, u_{i2}, \cdots, u_{in})^T \in \mathbb{R}^n \tag{2}$$

where $\|\mathbf{u}_i\| = 1$.

c) Velocity vector: $\mathbf{v}_i = v_i \mathbf{u}_i$, with speed $v_i \geq 0$.

Orthogonal Complement: The set

$$\mathbf{u}_i^{\perp} = \left\{ \mathbf{w} \in \mathbb{R}^n \,\middle|\, \mathbf{u}_i^T \mathbf{w} = 0 \right\} \tag{3}$$

2) Extended Mappings:

a) Row extension mapping:

$$(A_i \to G_r) = \left\{ \mathbf{p} = \mathbf{p}_i + s\mathbf{u}_i \,|\, s \in [0, D_L(v_i)] \right\} \tag{4}$$

where, $D_L(v_i)$ is the length-dimensional safety distance.

b) Column extension mapping ($G_c(A_i)$):

$$(A_i \to G_c) = \left\{ \mathbf{p} = \mathbf{p}_i + \mathbf{w} \,\middle|\, \mathbf{w} \in \mathbf{u}_i^{\perp}, \|\mathbf{w}\| \leq D_W(v_i) \right\} \tag{5}$$

where, $D_W(v_i)$ is the width-dimensional safety distance.

c) Diagonal extension mapping ($G_d(A_i)$):

$$(A_i \to G_d) = \left\{ \mathbf{p} = \mathbf{p}_i + s\mathbf{w} \,\middle|\, s \in [0, D_D(v_i)], \mathbf{w} \in S_\phi \right\} \tag{6}$$

where, $S_\phi = \left\{ \mathbf{w} \in \mathbb{R}^n \,\middle|\, \|\mathbf{w}\| = 1, \mathbf{w}^T \mathbf{u}_i = \cos\phi \right\}$, $D_D(v_i)$ is the diagonal safety distance.

d) Sine curve extension mapping ($G_s(A_i)$):

$$(A_i \to G_s) = \left\{ \mathbf{p}(t) = \mathbf{p}_i + v_i t \mathbf{u}_i + A \sin(\omega t) \mathbf{v}_i \,|\, t \in [0, T_s] \right\} \tag{7}$$

where, $\mathbf{v}_i \in \mathbf{u}_i^{\perp}, \|\mathbf{v}_i\| = 1$, A is the amplitude, ω is the angular velocity, T_s is the duration of the maneuver.

e) Bézier curve extension mapping ($G_b(A_i)$):

$$(A_i \to G_b) = \left\{ \mathbf{B}(t) = \sum_{k=0}^{m} B_{k,m}(t) \mathbf{P}_k \,|\, t \in [0, 1] \right\} \tag{8}$$

where, $B_{k,m}(t) = \binom{m}{k} (1-t)^{m-k} t^k$, $\mathbf{P}_k \in \mathbb{R}^n$ is the control points.

3) Regions and Metrics:

a) Longitudinal distance:

$$\Delta s_{ij} = \left(\mathbf{p}_j - \mathbf{p}_i\right)^T \mathbf{u}_i \tag{9}$$

b) Lateral distance:

$$\Delta l_{ij} = \left\| \left(\mathbf{p}_j - \mathbf{p}_i\right) - \Delta s_{ij} \mathbf{u}_i \right\| = \left\| \left(\mathbf{p}_j - \mathbf{p}_i\right)^{\perp} \right\| \tag{10}$$

c) Relative velocity:

$$\Delta v_{ij} = v_i - v_j \tag{11}$$

d) Heading angle difference:

$$\Delta \theta_{ij} = \arccos\left(\mathbf{u}_i^T \mathbf{u}_j\right) \tag{12}$$

III. Overall Network Structure

A. Process of the Proposed Trajectory Predictor

In the initial phases of trajectory prediction research for targets, conventional methods primarily entail independently modelling each target while depicting target interactions through pooling mechanisms. Nonetheless, this approach encounters challenges in accurately and efficiently capturing the intricate interaction dynamics among targets in real-world settings. It entails constructing spatiotemporal graphs among target trajectories to extract historical trajectory features essential for predicting future trajectories. The specific operational sequence of the predictor is delineated in Figure 1.

As depicted in Figure 1, when considering both single-agent and multi-agent interactions, the sampled position information is input into both the self GNN and the interaction GNN. In the self GNN, nodes represent the trajectory information of a single agent at specific timestamps. The interaction GNN is divided into multiple trajectory interaction sub-networks and an end-point interaction sub-network, where nodes represent all but the last sampled trajectory points and the last sampled trajectory point, respectively. After feature extraction in the self-network, the new features of non-terminal and terminal points are fed into the nodes of the trajectory interaction sub-network and the terminal interaction sub-network. In the GNN, black directed edges represent normal attention modeling modules, while orange directed edges represent special attention

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on December 10,2025 at 03:30:12 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

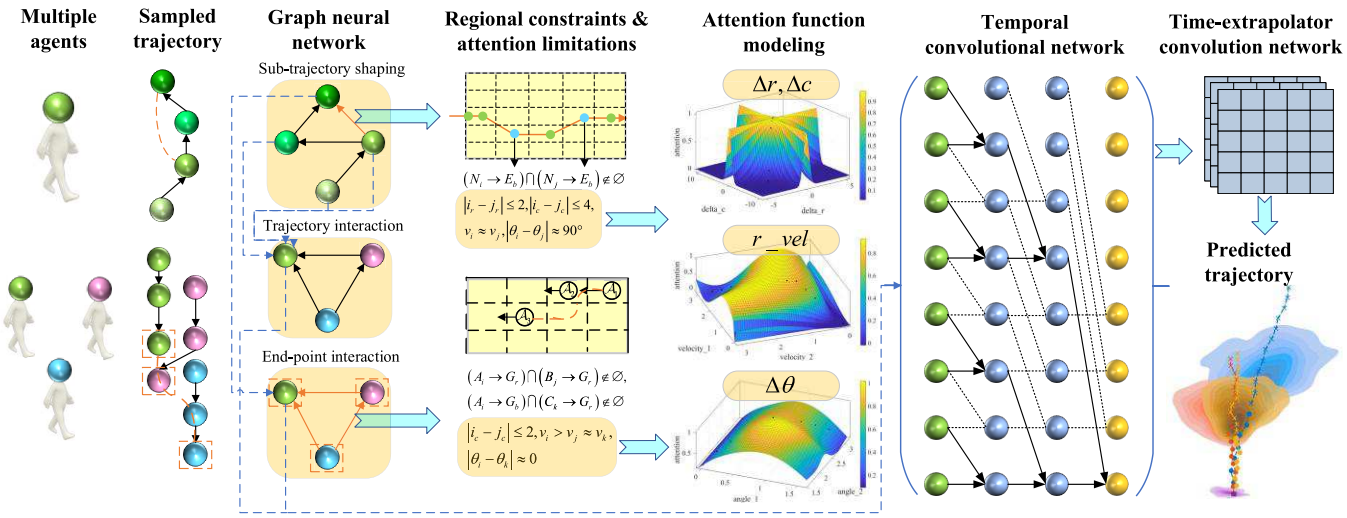


Fig. 1. Overall process of the proposed trajectory predictor.

modeling modules that consider abrupt behavior patterns. The direction of an edge from A to B indicates that A has an attention weight towards B, but not vice versa.

In the self-network, the most influential trajectory point may not be the one from the previous timestamp. For example, after an agent changes lanes and then returns, the trajectory point during the return is influenced mainly by the point at which the lane change occurred. In the interaction network, the final sampled trajectory point reflects the potential or imminent interaction behaviors among agents, necessitating a special attention function. A novel condition is established to determine whether to use a special or general attention function. It's based on whether the intersection of the expanded mappings of different trajectory points is empty. The special attention function is further calculated by combining regional indicators and speed-angle indicators, providing high attention under specific constraints for each abrupt behavior sub-scenario.

By fitting scatter points under particular indices, we set up the special attention functions. Three GNNs utilize GCN to extract spatial features from the spatiotemporal graph data, embedding the mutual influences between targets into the spatial feature data. Subsequently, on the spatiotemporal graph with embedded spatial features, the temporal dependencies between the same target across adjacent data frames are extracted to accomplish the final spatiotemporal feature extraction. Finally, the spatiotemporal features of each trajectory point are fed into a temporal convolutional extrapolation network to predict future trajectories.

The role of three sub-networks named Self-Trajectory Shaping (STS) network, Trajectory Interaction Network (TIN), and End-Point Interaction (EPI) network can be described as follows.

- STS handles individual agent motion encoding (ensuring each agent's own dynamics are captured);
- TIN handles pairwise/multi-agent past interactions through the graph (adjusting trajectories due to neighbors);
- EPI resolves final real-time destination conflicts and long-term interaction effects (identifying the behavioral interactions between agents and guiding them to make more stable predictions over relatively long-time periods).

While KITNet's architecture is multi-stage, we have kept each component lightweight, and the overall parameter count remains moderate, and the graph computations are sparse (we established a minimum threshold such that any attention value between agent nodes below 0.1 is treated as zero). Consequently, this results in no inter-node connections between agents in many scenarios. Compared to a non-hierarchical single-network model, KITNet introduces some overhead by having multiple sub-networks, but this is largely offset by the sparsity of the graph interactions. The modular design also allows parallelization: STS runs for all agents in parallel; graph operations are highly parallelizable on GPU. We observed that KITNet's inference speed is sufficient for real-time use (e.g., ~80 ms per frame for a scene with 5 agents, which is well within the planning reaction time for autonomous vehicles). In contrast, a hypothetical "simpler" model without our hierarchical decomposition might need a deeper or wider single network to capture the same phenomena, potentially with more parameters and less interpretability. Our approach thus strikes a balance between complexity and performance, achieving better accuracy with a structured model that is still computationally feasible.

B. Attention Update

We update the attention weights between utilizing region extension and attention modelling. Let R_i represent the set of spatial locations (or states) occupied by agent i over a short time window. We denote the set of neighboring agents for i as N_i. The region-extension mapping for agent i can be defined as a union of sets: $R_i^{ext} = R_i \cup \bigcup_{j \in N_i} R_j$, for all j satisfying a proximity criterion.

Algorithm 1 outlines the procedure for region extension and attention modelling in a clear, algorithmic format, making the process much more understandable.

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on December 10,2025 at 03:30:12 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Algorithm 1 Pseudo-code of Region Extension Mapping and Attention Modelling

Input: A set of agents S with their current states (positions, velocities, etc.).

Output: Extended regions $\{R_i^{ext}\}$ and attention- weighted features $\{h_i\}$ for all agents.

1. **Initialize** an interaction graph:
- Set $V = S$
- Set $E = \emptyset$
- Define the graph as $G = (V, E)$.
2. **For each** agent $i \in V$:

a. Compute initial region R_i (agent i's occupied space or trajectory segment).

b. Identify neighbor set:$N_i =, j \in V \setminus \{i\} \mid$ **distance**$(i, j) < d_0$ **or meets interaction criterion**

c. **For each** neighbor $j \in N_i$:
- Add $edge\,(i, j)$ to E
- Extend region of agent i by taking the set union: $R_i^{ext} \leftarrow R_i^{ext} \cup R_j$.

d. **For each** neighbor $j \in N_i$, compute the raw attention score:

$$e_{ij} = f_{att}\,(feat(i), feat(j), rel(i, j))$$

where:
- $feat(i)$ and $feat(j)$ are the encoded feature vectors of agents i and j;
- $rel(i, j)$ represents relative features (e.g., distance, bearing);
- $f_{att}(\cdot)$ is the function needs to be fitted.

e. **Normalize the attention weights** over all neighbors $j \in N_i$:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in N_i} \exp(e_{ik})}$$

f. **Aggregate neighbor features** using the computed attention weights:

$$h_i = \sum_{j \in N_i} \alpha_{ij} \cdot feat(j)$$

This produces an attention-weighted feature vector h_i for agent i.

IV. Attention Modeling Under Mutant Behavior

A. Classification of Mutation Behavior Patterns

In typical scenarios, agents generally move at a constant speed in a straight line. However, agents often exhibit behavioral changes due to temporary interactions or long-term cooperation with other agents. These changes can be categorized into minor adjustments and major abrupt changes. For minor adjustments, the mutual influence between agents can be modeled using general attention functions. Major abrupt changes involve temporary interactions, such as avoiding path conflicts, and long-term cooperative behaviors, such as following or walking in parallel among familiar agents.

Based on these two primary behavior categories, we propose specific abrupt behavior design schemes:

1. Replication Mode (R): When agents have similar speeds and are in close proximity, if they encounter an obstacle that narrows their path, they are predicted to follow each other in a single file. Otherwise, they are predicted to walk side by side.
2. Speed Control Mode (S): For agents traveling in the same lane and close proximity, if the agent's speed is non-zero, they are predicted to decelerate. Otherwise, they are predicted to stop, likely due to a traffic signal.
3. Lane Changing Mode (L): For agents with significantly different speeds but close proximity in the same lane, if no other close agents are in the adjacent lane, they are predicted to change lanes. Otherwise, they are predicted to overtake and then return to the original lane.
4. Interweaving Mode (I): For agents with a speed angle difference of approximately 90° and close vertical distance, they are predicted to interweave, where one agent reduces its speed to almost zero to let the other pass.
5. Turning Mode (T): For agents with direction-opposite and similar speeds in edge lanes, if their speeds are low, they are predicted to turn or make a U-turn, changing their speed angle by 90° or 180° to enter a new path.
6. Fallback Mode (F): For agents surrounded by a group of stationary agents, they are predicted to fallback, indicating a likely need to park.

We discovered that using only speed and position metrics is insufficient to systematically summarize all scenarios involving the six identified behaviors. Thus, we established a set of mapping models based on set theory, applying them to the high-dimensional extension of sampled trajectory points. The intersection of these high-dimensional extended sets, whether involving different sampled trajectory points of a single agent or the last sampled trajectory points of different agents, indicates whether the agents are experiencing or are about to experience one of the six abrupt behavior changes. Concurrently, new metrics were developed to construct attention functions for each behavior, laying the groundwork for a comprehensive representation of the internal interactions inherent in these six behavior patterns.

Using the mapping principles, we defined the following mapping models: row extension mapping (G_r), column extension mapping (G_c), Bézier curve extension mapping (G_b), diagonal extension mapping (G_d), and sine curve extension mapping (G_s). Each concrete mapping way can be seen in the Appendix A (see the supplementary material). The established metrics include row region difference (Δr), column region difference (Δc), velocity ratio (r_vel), and angle difference ($\Delta\theta$).

B. Multi-Agent Attention Condition Triggering and Weight Calculation Method

In the absence of abrupt behavior changes, the interaction patterns between an agent's internal trajectory points, as well as with other agents, are relatively straightforward, primarily determined by their relative speed and distance. For two trajectory points i and j, we calculate the relative time by dividing the relative position by the displacement component of the

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on December 10,2025 at 03:30:12 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

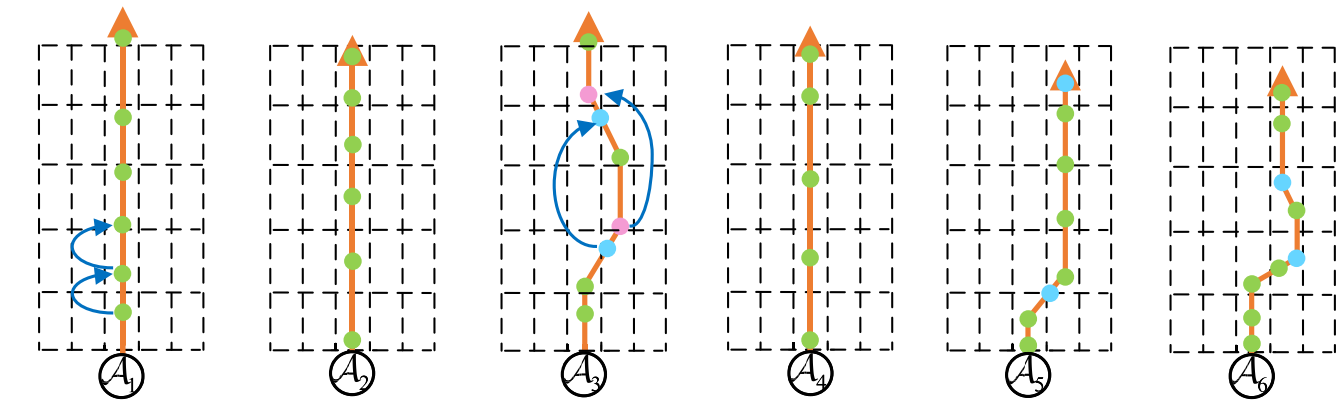


Fig. 2. Regular scenarios of one's own trajectory under six mutation behavior patterns. The trajectories of $\mathcal{A}1-\mathcal{A}6$ correspond to the behavior patterns of R, S, L, I, T, and F, respectively. As illustrated by the blue arrow, the position of each green trajectory point is all highly determined by the last one, presenting a progressive impact, the position later blue/pink trajectory point is mainly determined by the earlier one. For instance, in scenario (a), even when speed changes occur, the stable interaction pattern results in a smaller rate of speed change compared to scenario (b). Whereas (b) represents a more deliberate action prepared over a longer duration than the temporary avoidance maneuver seen in (d); consequently, its rate of speed change is less variable. Scenario (e) involves moving towards the right lane and eventually decelerating, which indicates a tendency to turn right. Finally, (f) shows a slight correction after turning, a common maneuver preceding potential future parallel or reverse parking.

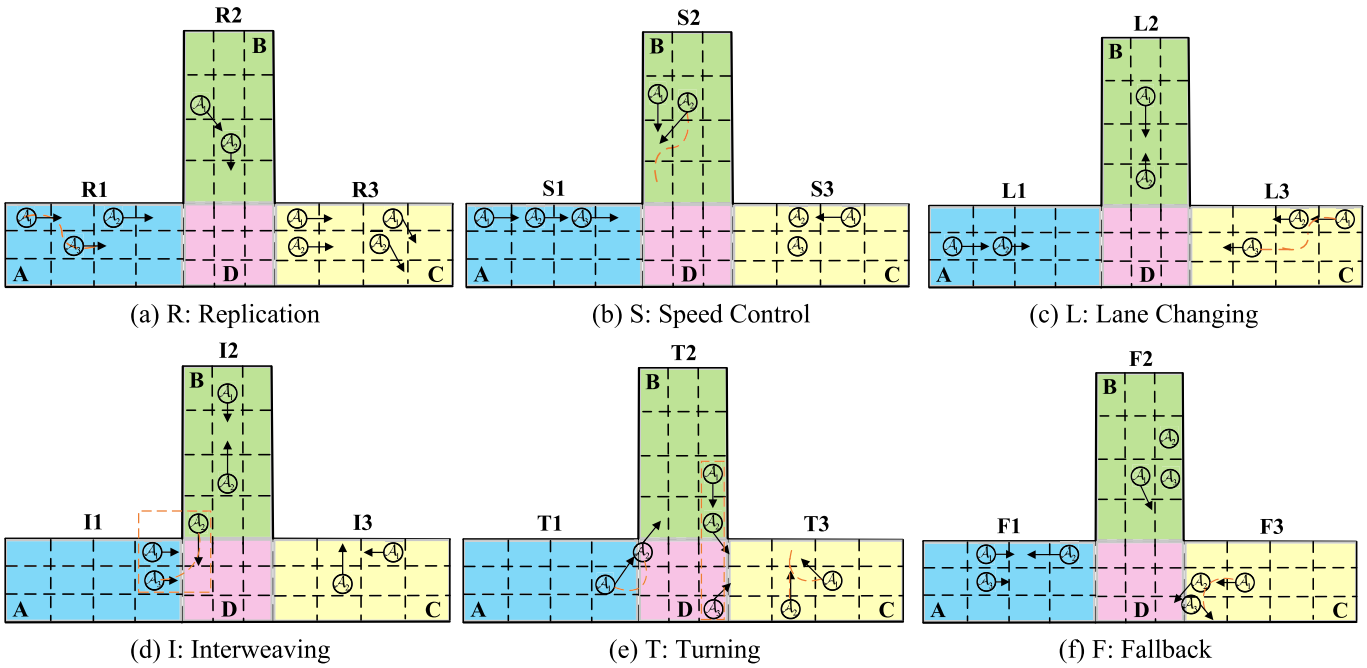


Fig. 3. Regular scenarios of endpoint interaction. The sub behavior refers to the actions taken by agent $\mathcal{A}_1$ in each sub scenario.

relative velocity. This is followed by a Softmax operation to normalize the weights, ensuring that closer targets receive higher weights. The detailed calculation for general attention function is shown in the following equation:

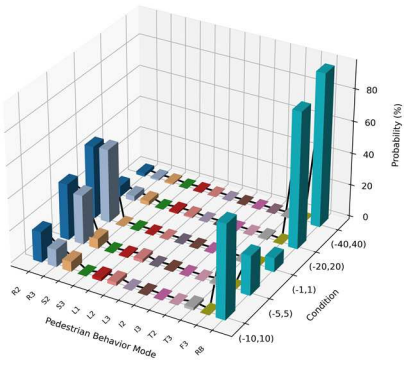
$$\begin{aligned} d_t^{ij} &= (p_t^i - p_t^j)^2 / \|(p_t^i - p_t^j) * (v_t^i - v_t^j)\| \\ A_t(i,j) &= a_t^{ij} = \exp(-d_t^{ij}) / \sum_{k \in N \setminus \{i\}} \exp(-d_t^{ik}) \end{aligned} \tag{13}$$

where, p_t^i, p_t^j represent the positional information of trajectory points at time t, while v_t^i, v_t^j represent the velocity information of trajectory points at time t.

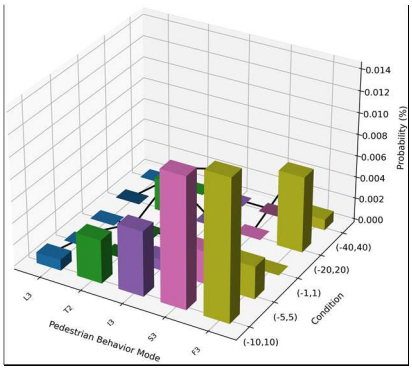
For the six abrupt behavior modes, mathematical modeling is performed using the variables and mappings established in the previous section. This allows us to set trigger conditions for each behavior mode and provides a reference for designing specific attention functions for each mode. The six behavior modes are denoted as R, S, L, I, T, and F, with each mode corresponding to different trajectory shapes as shown in Figure 2. For endpoint interaction modeling, each of the six behavior modes is further divided into three sub-behaviors, named R1~R3, etc., resulting in a total of 18 sub-behaviors. These sub-behaviors are represented in three regions, A, B, and C, as illustrated in Figure 3. According to Figure 4, The meaning, explanation and choosing method on different types

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on December 10,2025 at 03:30:12 UTC from IEEE Xplore. Restrictions apply.

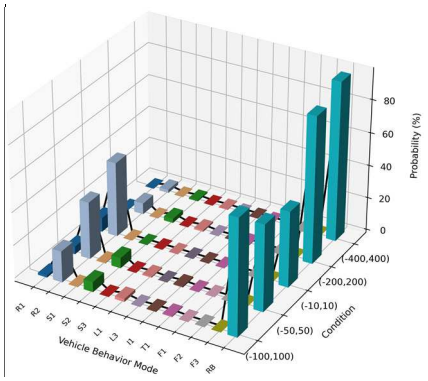
This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



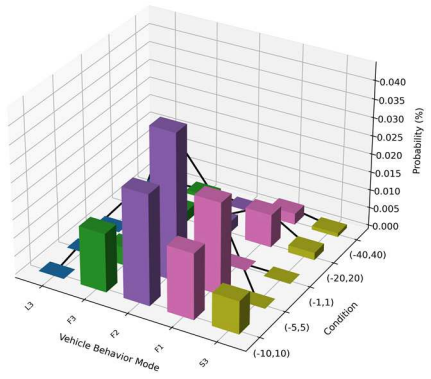
(a) Probability of each pedestrian behavior



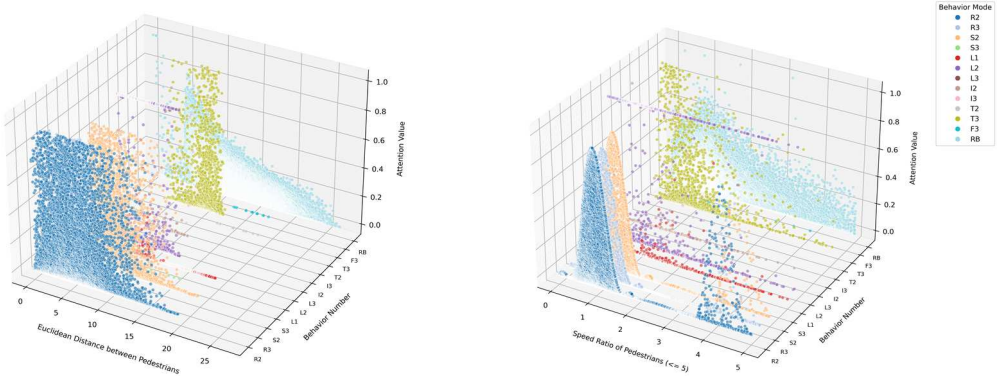
(b) Low probability of 5 pedestrian behaviors



(c) Probability of each vehicle behavior

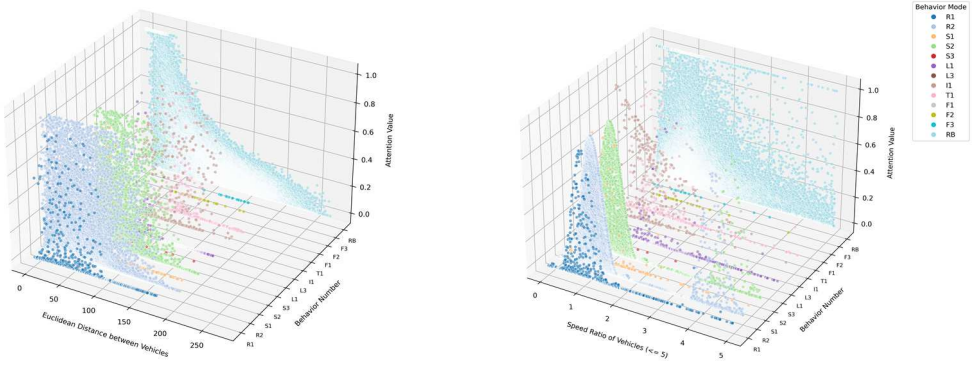


(d) Low probability of 5 vehicle behaviors



(e) Each attention value under pedestrian Euclidean distance

(f) Each attention value under pedestrian velocity ratio



(g) Each attention value under vehicle Euclidean distance

(h) Each attention value under vehicle velocity ratio

Fig. 4. Probability and attention value under general and mutant behaviors.

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on December 10,2025 at 03:30:12 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

of objects (pedestrian/vehicle) for each sub behavior on agent $\mathcal{A}_1$ is shown in Appendix C (see the supplementary material).

Regarding the region division rules, the lateral direction is defined as parallel to the road surface, while the longitudinal direction is defined as tangential to it. Specifically, the length of each delineated unit rectangular region is set to half of the safety distance maintained by agent $\mathcal{A}_1$ in the lateral direction. Furthermore, the width of each unit area corresponds to the standard width of a lane. It's worth noting that on sidewalks, a similar concept applies: the 'lane' width is approximated by the average human shoulder width (around 0.5 meters), and these 'lanes' are demarcated every 0.5 meters moving rightward from the sidewalk's leftmost boundary.

The calculation for safety distance is as follows:

$$D_{safe}(v, \mu, t_r) = v \cdot t_r + \frac{v^2}{2\mu g} \tag{14}$$

where v is the current agent's speed, μ is the road adhesion coefficient, t_r is the reaction time, g is the gravitational acceleration. In this experiment, given the scenario of a dry asphalt pavement, the parameters were determined empirically, with the μ set to 0.7 and t_r to 0.5 seconds.

Bézier curves and sine curves represent two distinct patterns, each characterizing lane changes and turns between agents, respectively. These curves are integral to the six types of behavioral transitions studied. Specifically, Bézier curves are employed in the first three behavior modes, while sine curves are used in the latter three. Consequently, we should model these curves accurately and determine their corresponding parameters. The modeling results are detailed in Appendix D (see the supplementary material).

For multi-agent attention condition triggering via region-extension mapping, which can be contrasted with other interaction modeling approaches:

(1) Distance-based cutoffs: A common approach is to consider any agent within a fixed radius or within a certain field-of-view of agent i as interacting. This is effectively a simple set inclusion criterion where $\mathrm{N_i}$ contains all agents within distance $\mathrm{d_0}$ (an ε-ball neighborhood). While efficient, a fixed radius may ignore relevant farther agents (e.g., a fast oncoming vehicle) or include irrelevant closer agents (if separated by a barrier). TP2Net uses a vehicle-of-interest (VOI) selection based on predefined spatial positions (e.g., left-front, rear-right) and encodes these neighbors using convolutional operations. This fixed layout corresponds to an implicit region mapping, but the patterns are fixed and limited, also, it can only predict the trajectory of one vehicle at a time, which may lead to time complexities under different traffic densities.

(2) Grid-based occupancy pooling: Prior works (e.g., Social LSTM) divide the space into grid cells around an agent and mark cells as occupied or not. This is analogous to defining on a discrete grid and combining occupancy sets. However, grid-based methods can be coarse and miss continuous interaction strengths; our set-union mapping preserves exact occupied regions and can be extended continuously.

(3) Graph-learned attention: Modern graph attention networks (e.g., GAT, MHA, GNN pooling) allow the model to learn soft interaction weights between agents. In this view, region-extension corresponds to a hard binary relation (included/excluded), whereas attention learns a continuous-valued influence α_{ij}. KITNet builds on this idea by combining both symbolic set-based inclusion and region-aware attentional gating, allowing richer physical reasoning.

In contrast, KITNet's region-extension model constructs explicit, adaptive interaction fields using set-theoretic union operations on multiple agents. These fields naturally vary with agent velocities, directions, and historical intent, allowing KITNet to dynamically reason about feasible mutual occupancy zones and incorporate them into the graph construction stage.

C. Derivation of Triggering Condition and Indicator Requirement

Based on the insights presented in Figures 2 and 3, it is essential to derive the triggering conditions and indicator requirements for the 18 sub-behaviors using high-dimensional spatiotemporal theory and region-set theory. Triggering conditions define the circumstances under which a specific sub-behavior emerges between agents. In such cases, the attention functions between the involved agents must be specially configured.

Indicator requirements specify the range of parameters to be considered when setting these attention functions. When the metric values fall within the defined range, the attention function gains higher weight, indicating a stronger likelihood of corresponding interactive behaviors between the agents.

To illustrate this process, we focus on deriving the triggering conditions and indicator requirements for sub-behavior I1, a highly interactive behavior. The same methodological framework can be extended to the remaining sub-behaviors.

We denote the state of agent R_1, R_2, R_3 involved in the scenario as A_i, B_j, C_k,

Since A_i is approaching the intersection along a straight path, we can represent its path using the Row Extension Mapping (G_r):

$$(A_i \to G_r) = \left\{\mathbf{p} = \mathbf{p}_i + s\mathbf{u}_i \,|\, s \in [0, D_L(v_i)]\right\} \tag{15}$$

where, $\mathbf{p}_i$ is the initial position of A_i, s is the scalar parameter along the heading direction, $D_L(v_i)$ is the longitudinal safety distance for A_i.

As B_j is approaching tangentially and moving faster, its path can be represented using the Sine Curve Extension Mapping (G_s) to model any curvature:

$$\left(B_j \to G_s\right) = \left\{\mathbf{p} = \mathbf{p}_j(t) \,\middle|\, t \in \left[t_0, t_f\right]\right\} \tag{16}$$

where, $\mathbf{p}_j(t) = \mathbf{p}_j + v_j t\mathbf{u}_j + A\sin(\omega t)\, v_j^{\perp}$, A is the amplitude of the sine curve (models lateral deviation), ω is the angular frequency, $v_j^{\perp}$ is the unit vector perpendicular to $\mathbf{u}_j$, t_0, t_f is the start and end times of the maneuver.

C_k is moving nearly parallel to A_i, so its path is also represented by G_r:

$$(C_k \to G_r) = \left\{\mathbf{p} = \mathbf{p}_k + s'\mathbf{u}_k \,\middle|\, s' \in [0, D_L(v_k)]\right\} \tag{17}$$

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on December 10,2025 at 03:30:12 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

where, $\mathbf{p}_k$ is the initial position of C_k, s' is the scalar parameter along the heading direction, $D_L(v_k)$ is the longitudinal safety distance for C_k.

1) Analyze Agent A_i's Inability to Change Lanes: The angle between A_i and C_k is:

$$\Delta\theta_{ik} = \arccos\left(\mathbf{u}_i^T \mathbf{u}_k\right) \tag{18}$$

Given $\mathbf{u}_k$ is nearly parallel to $\mathbf{u}_i$:

$$\Delta\theta_{ik} \leq \varepsilon_\theta \tag{19}$$

The Relative position can be expressed as:

$$\mathbf{p}_k(t) - \mathbf{p}_i(t) = \Delta s_{ik}\mathbf{u}_i + \Delta l_{ik}\mathbf{v}_i^{\perp} \tag{20}$$

The lateral distance is:

$$\Delta l_{ik} = \left\| \left(\mathbf{p}_k(t) - \mathbf{p}_i(t)\right)^{\perp} \right\| \tag{21}$$

Due to small$\Delta\theta_{ik}$, the lateral displacement over distance Δs_{ik} is:

$$\delta l_{ik} = \Delta s_{ik} \tan(\Delta\theta_{ik}) \approx \Delta s_{ik}\Delta\theta_{ik} \tag{22}$$

So, the total lateral distance is:

$$\Delta l_{ik} = \Delta l_0 + \delta l_{ik} = \Delta l_0 + \Delta s_{ik}\Delta\theta_{ik} \tag{23}$$

where Δl_0 is the initial lateral separation at $t = 0$.

Since $\Delta\theta_{ik}$ is small, it is highly likely that $\Delta l_{ik} \leq D_W(v_i)$. As a result, A_i cannot change lanes due to C_k's proximity, which can be expressed as:

$$(A_i \to G_c) \cap (C_k \to G_c) \notin \emptyset \tag{24}$$

2) Analyze Potential Collision at Intersection: The relative position between A_i and B_j is:

$$\Delta\mathbf{p}_{ij}(t) = \left[\mathbf{p}_j(0) + v_j t\mathbf{u}_j + A_j \sin\left(\omega_j t\mathbf{v}_j^{\perp}\right)\right] - \left[\mathbf{p}_i(0) + v_i t\mathbf{u}_i\right] \tag{25}$$

Since B_j is moving faster ($v_j > v_i$), they will reach the intersection at similar times, causing a condition which needs to prevent potential collision:

$$\left\|\Delta\mathbf{p}_{ij}(t)\right\| \leq D_{safe} \tag{26}$$

where, D_{safe} is the minimum safe distance to avoid collision.

Thus, we can define the collision probability at time t:

$$P_{collision}(t) = \exp\left(-\left(\frac{\left\|\Delta\mathbf{p}_{ij}(t)\right\|}{D_{safe}}\right)\right)^2 \tag{27}$$

Because $\left\|\Delta\mathbf{p}_{ij}(t)\right\| \leq D_{safe}$, $P_{collision}(t)$ is significant.

There is a high risk of collision between A_i and B_j at the intersection. Therefore, we can derive the following triggering condition, which means that A_i's path overlaps with B_j's path within their safety regions, indicating a potential collision if no action is taken:

$$(A_i \to G_r) \cap \left(B_j \to G_s\right) \notin \emptyset \tag{28}$$

The direction of A_i and C_k is parallel, so similarly, B_j and C_k also meet this collision avoidance condition:

$$\left(B_j \to G_s\right) \cap (C_k \to G_r) \notin \emptyset \tag{29}$$

As long as one of the two conditions is met, it can be considered that there is a potential collision risk at this intersection:

$$\left[(A_i \to G_r) \cap \left(B_j \to G_s\right)\right] \cup \left[\left(B_j \to G_s\right) \cap (C_k \to G_r)\right] \notin \emptyset \tag{30}$$

3) Deriving the Spatial Proximity Condition: Since agent A_i and agent B_j are approaching the intersection tangentially, they must be in nearby columns. The columns represent lanes or pathways. Therefore:

$$|i_c - j_c| \leq n \tag{31}$$

where, n is a small integer representing allowable proximity.

Given the context of an intersection and practical considerations, we set $n = 2$. This condition ensures that A_i and B_j are close enough spatially for their paths to potentially intersect at the intersection.

4) Deriving the Speed Condition: Since A_i cannot change lanes due to C_k's proximity and cannot overtake C_k, it implies:

$$v_k < v_i \tag{32}$$

Combined with the previous information:

$$v_k < v_i < v_j \tag{33}$$

This speed relationship contributes to the decision of A_i to wait for B_j to cross.

5) Deriving the Heading Angle Difference: As mentioned before, A_i and B_j are approaching the intersection tangentially. So:

$$\left|\theta_i - \theta_j\right| \approx 90^\circ \tag{34}$$

D. Final Trajectory Distribution Prediction

The temporal convolutional neural network mainly comprises two convolutional designs: causal convolution and dilated convolution. Among these, the concept of causal convolution is similar to that of a Markov chain, meaning the convolution at time t can only infer from the elements before that time t, denoted by $\{x_1, x_2 \cdots, x_t\}$, and is independent of future data. Its computational formula is expressed as follows:

$$F(x_t) = \sum_{k=1}^{K} f_k x_{t-K+k} \tag{35}$$

where, $F(x_t)$ represents the convolution calculation result at x_t, where K denotes the size of the one-dimensional convolutional kernel, and f_k represents the specific parameter value of the kernel at position k. The convolution result at x_t is only related to the data before it and is independent of future data. Meanwhile, convolution can be understood as a sliding window that continuously moves along the sequence data to perform convolution operations.

After the spatiotemporal graph convolutional network processes and extracts features from the spatiotemporal graph, the resulting feature data $V \in R^{c_{out} \times T_{obs} \times N}$ is used as the input for the time extrapolation convolutional network (TXP-CNN). Here, the parameter c_{out} represents the dimension of the output for each node, indicating the five parameters $(\hat{\mu}x, \hat{\mu}y, \hat{\sigma}x, \hat{\sigma}y, \hat{\rho})$ that represent the bivariate Gaussian distribution of the predicted trajectory. Therefore, its value is set to 5 in this context. The feature data V has a length of T_{obs} along the time dimension, while the predicted trajectory has a length of T_{pre} along the time dimension. Hence, the time extrapolation convolutional neural network needs to extend the feature data V along the time dimension.

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on December 10,2025 at 03:30:12 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I

THE TRIGGERING CONDITIONS AND APPROXIMATE INDICATOR REQUIREMENTS FOR EACH SAMPLED TRAJECTORY POINT IN ONE AGENT AMONG THE THREE TYPES OF MUTATION BEHAVIORS

Behavior pattern	Triggering condition	Approximate indicator requirement
Replication (R)	$(N_i \to E_b) \bigcap (N_j \to E_b) \not\in \varnothing$	$\lvert i_r - j_r \rvert \le 2, \lvert i_c - j_c \rvert \le 4, v_i \approx v_j, \lvert \theta_i - \theta_j \rvert \approx 90°$
Turning (T)	$(N_i \to E_b) \bigcap (N_j \to E_r) \not\in \varnothing$	$\lvert i_r - j_r \rvert \le 2, \lvert i_c - j_c \rvert \le 4, v_i > v_j, \lvert \theta_i - \theta_j \rvert \approx 45°$
Fallback (F)	$(N_i \to E_s) \bigcap (N_j \to E_s) \not\in \varnothing$	$\lvert i_r - j_r \rvert \le 1, \lvert i_c - j_c \rvert \le 2, v_i \approx v_j, \lvert \theta_i - \theta_j \rvert \approx 90°$

TABLE II

THE TRIGGERING CONDITIONS AND INDICATOR REQUIREMENTS FOR EACH SAMPLED TRAJECTORY END-POINT IN MULTIPLE AGENT AMONG THE THREE TYPES OF MUTATION BEHAVIORS

Sub behavior	Triggering condition	Indicator requirement
R1	$(A_i \to G_r) \bigcap (B_j \to G_r) \not\in \varnothing, (A_i \to G_b) \bigcap (C_k \to G_b) \not\in \varnothing$	$\lvert i_c - j_c \rvert \le 2, v_i \approx v_j, \lvert \theta_i - \theta_j \rvert \approx 0$
R2	$(A_i \to G_b) \bigcap (B_j \to G_r) \not\in \varnothing$	$\lvert i_r - j_r \rvert \le 1, \lvert i_c - j_c \rvert \le 2, v_i \approx v_j, \lvert \theta_i - \theta_j \rvert \approx 45°$
R3	$\left[(A_i \to G_c) \bigcap (B_j \to G_c)\right] \bigcup \left[(A_i \to G_d) \bigcap (B_j \to G_d)\right] \not\in \varnothing$	$\lvert i_r - j_r \rvert \le 2, \lvert i_c - j_c \rvert = 0, v_i \approx v_j, \lvert \theta_i - \theta_j \rvert \approx 0$
S1	$(A_i \to G_r) \bigcap (B_j \to G_b) \bigcap (C_k \to G_r) \not\in \varnothing$	$\lvert i_c - j_c \rvert \le 2, v_i > v_j \approx v_k, \lvert \theta_j - \theta_k \rvert \approx 0$
S2	$(A_i \to G_r) \bigcap (B_j \to G_b) \not\in \varnothing$	$\lvert i_r - j_r \rvert \le 1, \lvert i_c - j_c \rvert \le 1, v_i \approx v_j, \lvert \theta_i - \theta_j \rvert \approx 45°$
S3	$(A_i \to G_r) \bigcap (B_j \to G_r) \not\in \varnothing, (A_i \to G_b) \bigcap (C_k \to G_r) \not\in \varnothing$	$\lvert i_c - j_c \rvert \le 2, v_j, v_k \approx 0, \lvert \theta_i - \theta_j \rvert \approx 0$
L1	$(A_i \to G_r) \bigcap (B_j \to G_r) \not\in \varnothing$	$\lvert i_c - j_c \rvert \le 2, v_i > v_j, \lvert \theta_i - \theta_j \rvert \approx 0$
L2	$(A_i \to G_c) \bigcap (B_j \to G_s) \not\in \varnothing$	$\lvert i_c - j_c \rvert \le 4, v_i > v_j, \lvert \theta_i - \theta_j \rvert \approx 180°$
L3	$(A_i \to G_d) \bigcap (B_j \to G_s) \not\in \varnothing, (A_i \to G_b) \bigcap (C_k \to G_r) \not\in \varnothing$	$\lvert i_c - j_c \rvert \le 2, v_i > v_j \approx v_k, \lvert \theta_i - \theta_k \rvert \approx 0$
I1	$\left[(A_i \to G_r) \bigcap (B_j \to G_s)\right] \bigcup \left[(B_j \to G_s) \bigcap (C_k \to G_r)\right] \not\in \varnothing$, $(A_i \to G_c) \bigcap (C_k \to G_c) \not\in \varnothing$	$\lvert i_c - j_c \rvert \le 2, v_k < v_i < v_j, \lvert \theta_i - \theta_j \rvert \approx 90°$
I2	$(A_i \to G_s) \bigcap (B_j \to G_d) \not\in \varnothing$	$\lvert i_c - j_c \rvert \le 4, v_i < v_j, \lvert \theta_i - \theta_j \rvert \approx 180°$
I3	$(A_i \to G_d) \bigcap (B_j \to G_s) \not\in \varnothing$	$\lvert i_r - j_r \rvert \le 2, \lvert i_c - j_c \rvert \le 2, v_i < v_j, \lvert \theta_i - \theta_j \rvert \approx 90°$
T1	$(A_i \to G_s) \bigcap (B_j \to G_s) \not\in \varnothing$	$\lvert i_r - j_r \rvert \le 1, \lvert i_c - j_c \rvert \le 1, v_i \approx v_j, \lvert \theta_i - \theta_j \rvert \approx 0$
T2	$(A_i \to G_r) \bigcap (B_j \to G_r) \bigcap (C_k \to G_r) \not\in \varnothing$	$\lvert i_c - j_c \rvert \le 2, v_i \approx v_j \approx v_k, \lvert \theta_i - \theta_j \rvert \approx 45°, \lvert \theta_i - \theta_k \rvert \approx 135°$
T3	$(A_i \to G_s) \bigcap (B_j \to G_c) \not\in \varnothing$	$\lvert i_r - j_r \rvert \le 2, \lvert i_c - j_c \rvert \le 2, v_i < v_j, \lvert \theta_i - \theta_j \rvert \approx 45°$
F1	$(A_i \to G_b) \bigcap (B_j \to G_s) \not\in \varnothing, (A_i \to G_b) \bigcap (B_j \to G_c) \not\in \varnothing$	$\lvert i_r - k_r \rvert \le 1, v_k < v_i < v_j, \lvert \theta_i - \theta_j \rvert \approx 180°, \lvert \theta_i - \theta_k \rvert \approx 0$
F2	$\left[(A_i \to G_d) \bigcap (B_j \to G_d)\right] \bigcup \left[(A_i \to G_d) \bigcap (C_k \to G_d)\right] \not\in \varnothing$, $(B_j \to G_r) \bigcap (C_j \to G_r) \not\in \varnothing$	$\lvert i_r - j_r \rvert \le 2, v_j, v_k \approx 0, \lvert \theta_i - \theta_k \rvert \approx 45°$
F3	$(A_i \to G_s) \bigcap (B_j \to G_b) \not\in \varnothing, (A_i \to G_s) \bigcap (C_k \to G_d) \not\in \varnothing$	$\lvert i_c - j_c \rvert \le 2, v_i \approx v_j < v_k, \lvert \theta_i - \theta_j \rvert \approx 45°, \lvert \theta_i - \theta_k \rvert \approx 135°$

V. EXPERIMENT RESULTS

A. The Triggering Probability and Attention Value of General and Mutant Behaviors

For scenarios with abrupt behaviors, the specific behavior modes potentially associated with each type are as follows:

1) Pedestrians (also extensible to non-motorized vehicles): R2, R3, S2, S3, L1, L2, L3, I2, I3, T2, T3, F3;

2) Vehicles (specifically motorized vehicles): R1, R2, S1, S2, S3, L1, L3, I1, T1, F1, F2, F3.

To evaluate the validity of our behavioral classifications and triggering conditions, we first tested the occurrence probabilities of one regular behavior and 12 abrupt behaviors each on the above. The agent positions were uniformly sampled within five ranges: [−10,10] m, [−5,5] m, [−1,1] m, [−20,20] m, and [−40,40] m, resulting in 100,000 simulated scenarios. Figures 4(a), (b), and (c) reveal that apart from the regular mode, the Replication Behavior Mode dominated, as avoidance scenarios were relatively rare, aligning with real-world observations where agents often travel independently or in recognized groups. Also, it is observable that the frequency of various interaction modes differs significantly between the two distinct types of agents, pedestrians and vehicles. For instance, the S3 behavior mode exhibits a notably higher

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on December 10,2025 at 03:30:12 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE III
COMPARISONS WITH THE CURRENT STATE-OF-THE-ART METHODS ON THE ETH/UCY DATASET IN MINADE/MINFDE (M) METRIC. TEXT IN BOLD DENOTES THE BEST RESULTS

Prediction Model	Dataset						
	ETH	HOTEL	UNIV	ZARA1	ZARA2	Argoverse 2	NGSIM
Social-LSTM [10]	1.09/2.35	0.79/1.76	0.67/1.40	0.47/1.00	0.56/1.17	1.8/3.89	
Social-GAN [12]	0.81/1.52	0.72/1.61	0.60/1.26	0.34/0.69	0.42/0.84		
GAT [19]	0.68/1.29	0.68/1.40	0.57/1.29	0.29/0.60	0.37/0.75		
Social-BiGAT [20]	0.69/1.29	0.49/1.01	0.55/1.32	0.30/0.62	0.36/0.75		
Social-STGCNN [22]	0.64/1.11	0.49/0.85	0.44/0.79	0.34/0.53	0.30/0.48		
Attention-GCNN [31]	0.68/1.22	0.31/0.41	0.39/0.69	0.34/0.55	0.28/0.44		
STAR [32]	0.56/1.11	0.26/0.50	0.52/1.15	0.41/0.90	0.31/0.71		
PECNet [33]	0.54/0.87	0.18/0.24	0.35/0.60	0.22/0.39	0.17/0.30		
Trajectorn++ [34]	0.39/0.83	0.12/0.21	0.20/0.44	**0.15**/0.33	**0.11**/0.25		
Agentformer [35]	0.45/0.75	0.14/0.22	0.25/0.45	0.18/0.30	0.14/0.24		
MemoNet [36]	0.40/0.61	**0.11**/**0.17**	0.24/0.43	0.18/0.32	0.14/0.24		
TGR [37]						0.99/1.60	
Deep-GAN [38]						0.72/1.21	
QCNet [39]						0.62/1.19	
CS-LSTM [40]							2.29/3.34
DeepTrack [41]							2.01/3.25
STA-LSTM [42]							1.89/3.16
PishGu [43]							0.88/1.96
KITNet (w/o HS & AM)	0.92/1.68	0.70/1.58	0.61/1.30	0.31/0.58	0.40/0.86	0.86/1.58	1.78/3.20
KITNet (w/o HS)	0.55/0.84	0.29/0.53	0.37/0.62	0.22/0.35	0.23/0.32	0.67/1.21	0.97/2.06
KITNet (w/o AM)	0.59/0.90	0.28/0.55	0.40/0.68	0.20/0.35	0.27/0.38	0.75/1.34	0.95/2.12
KITNet (Full)	**0.37**/**0.55**	0.13/**0.17**	**0.18**/**0.37**	**0.14**/**0.24**	0.15/**0.23**	**0.49**/**0.88**	**0.51**/**1.04**

probability of occurrence in vehicle interaction scenarios, primarily because traffic congestion, leading to lead agents having zero speed, is more likely on roadways. Conversely, retreat modes, which are considerably less common in real-world scenarios, consistently rank among the least frequent behavior modes observed. Collectively, these findings offer preliminary validation for the rationality underlying the defined region set mapping. When testing with broader positional ranges, the likelihood of strong interactions decreased due to increased inter-agent distances, resulting in more regular behavior occurrences, further validating the rationality of the triggering conditions.

Next, we analyzed inter-agent attention in these 100,000 scenarios. Figures 4(d) and (e) highlight how attention values decrease with increasing Euclidean distance, ultimately converging to zero (visually represented as a straight line). This supports the correctness of our attention function design. Additionally, no matter pedestrian or vehicle, as Euclidean distance and speed ratios increase, strong interactions diminish while regular modes dominate. Notably, the rate of decline in the attention function is less pronounced in the speed-ratio dimension, with more scattered data points. This is attributed to the greater influence of distance over speed in determining interaction likelihood; agents farther apart are less likely to engage, even with similar speeds.

B. Comparison Under General and Mutant Behaviors

To provide a clear illustration of the trajectory outputs from the bivariate Gaussian probability model, we sampled 100 predicted trajectories using the model's bivariate Gaussian parameters. This allowed us to visualize the distribution of future trajectory predictions. The figure below demonstrates the effectiveness of trajectory predictions under general behaviors, comparing the output distributions of Social-GAN, Social-STGCNN, and our method in the HOTEL scenario. The red circles in the scene indicate the targets of the trajectories to be predicted. The subsequent three columns display the prediction results of each method, with different colors representing different pedestrian targets. Dotted lines denote the observed trajectories over the initial 8 frames, while cross-marked lines indicate the true future trajectories over the subsequent 12 frames. The colored areas represent the predicted trajectory distributions, visualizing the means and variances from the bivariate Gaussian parameter predictions.

Then, 11 representative scenes in two datasets were selected for the five datasets to compare prediction outcomes. Figure 5 shows the trajectory prediction distributions as output by Social-GAN, Social-STGCNN, and our proposed method with three ablative versions when dealing with changing intention. In the figure, red circles indicate the target trajectories to be predicted. We have clearly specified the particular datasets (including sub-datasets within ETH/UCY) and the corresponding frame numbers, facilitating readers in cross-referencing with the original video streams or tables provided by the dataset. The six columns respectively display the prediction outcomes of each method, with different colors representing different pedestrian targets. Dotted lines indicate the observed trajectories over the first 8 frames, while crossed lines show the true future trajectories over the subsequent 12 frames. The

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on December 10,2025 at 03:30:12 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

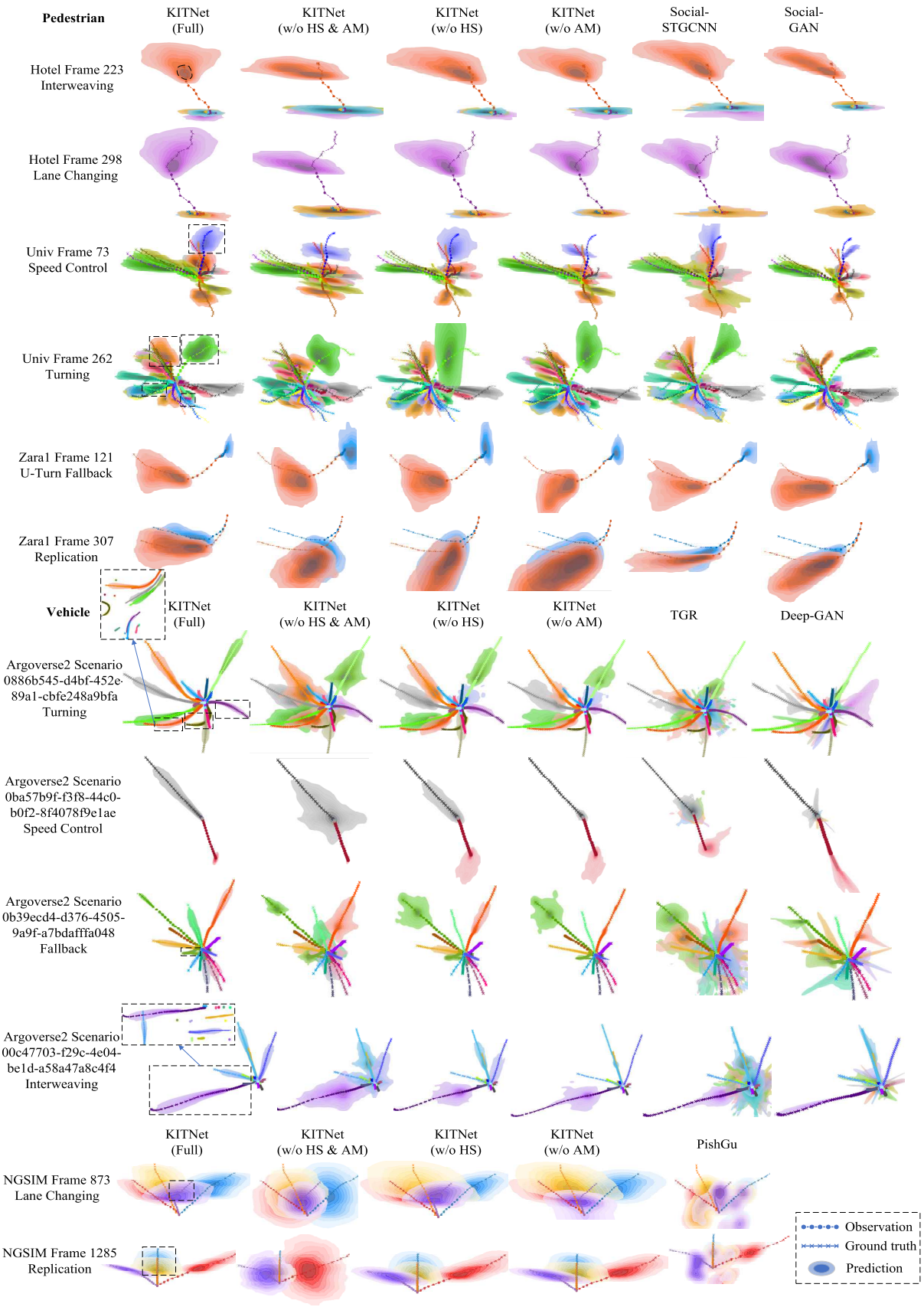


Fig. 5. Trajectory prediction effect for ETH/UCY, Argoverse 2 and NGSIM dataset under 18 mutant behavior patterns.

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on December 10,2025 at 03:30:12 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

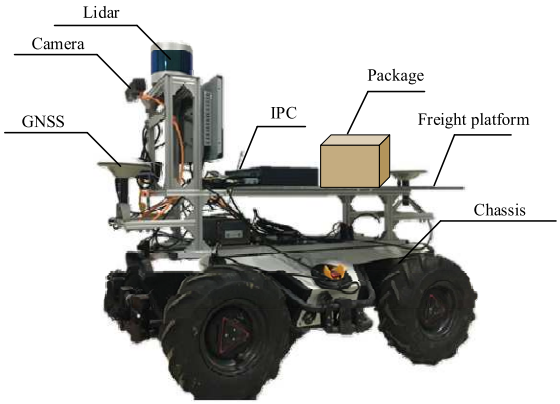


Fig. 6. Hardware platform for ADV.

colored areas represent the predicted trajectory distributions, visualized through the means and variances of the bivariate Gaussian parameter predictions. The information for three ablative models is shown below:

- "KITNET (w/o HS)" for removing the hierarchical structure (no three-stage processing, combine the interaction graph for self-trajectory and multi-trajectory together, all attention weight is calculated using attention mechanism),
- "KITNET (w/o AM)" for removing the attention modelling (still dividing the graph part into three subnetwork and process the data in order),
- "KITNET (w/o HS & AM)" for removing both (no three-stage processing, only setting up attention weights based on relative distance and speed).

The results demonstrate that the KITNET (w/o HS) model, even while retaining attention function modeling, occasionally predicts future behavior patterns correctly. However, its predictions suffer from increased noise, primarily because it lacks the crucial differentiation between self-trajectory evolution patterns, prior inter-agent interaction patterns, and current inter-agent interaction patterns. This deficiency can lead to outdated interaction patterns being misidentified as currently persistent modes, consequently resulting in a larger predicted Gaussian distribution area compared to the KITNET (Full) model. Conversely, the KITNET (w/o AM) model demonstrates the opposite effect. Illustrative examples include the predicted blue Gaussian distribution in Univ Frame 73, the red Gaussian distribution in Zara1 Frame 121, and the red Gaussian distribution in NGSIM Frame 597 and Argoverse2 '0886b545-d4bf-452e-89a1-cbfe248a9bfa'. Moreover, the KITNET (w/o HS & AM) model frequently exhibits both of the aforementioned undesirable effects simultaneously, thereby clearly demonstrating the necessity of both modules.

Also, the results clearly demonstrate that our method significantly outperforms Social-STGCNN, Social-GAN, TGR, Deep-GAN and PishGu models in predicting trajectories influenced by different interaction-induced abrupt behaviors. Social-STGCNN and Social-GAN rely more heavily on training data patterns, thus failing to accurately capture the complex interactions and potential behavior changes among pedestrians. As for trajectory prediction towards vehicle, PishGu sometimes identifies incorrect interaction modes, causing its predicted distribution to deviate from the ground truth (e.g., the purple distribution in Frame 1285, which mistakenly identified a slight left turn as a U-turn interaction behavior), and occasionally exhibits unconcentrated distributions, which can, to some extent, affect the judgment of subsequent decision systems in physical platforms. With respect to the interleaving and fallback behaviors unique to urban driving scenarios, models such as TGR and Deep-GAN lack sufficient prediction accuracy, stemming from their difficulty in discerning the interactive relationships between the target vehicle and surrounding agents. In a specific scene, such as the one identified by UUID '00c47703-f29c-4e04-be1d-a58a47a8c4f4', our method correctly anticipates that the vehicle with the blue trajectory will first accelerate and subsequently engage in an interleaving maneuver with the purple-trajectory vehicle. Consequently, the predicted blue Gaussian distribution does not intersect with the purple distribution, as the model infers that the blue vehicle will decelerate to wait for the other to pass. In actuality, while the final ground-truth trajectory of the blue vehicle does slightly cross that of the purple one, this represents the ultimate action taken after the purple vehicle has passed; the antecedent behavior of accelerating and then decelerating to wait—which our model correctly identified—did indeed precede this. The other models, in contrast, fail to predict such complex interaction phenomena.

C. Comparison of Overall Indicators

In the trajectory prediction task using this public dataset, the performance is evaluated using two metrics: Average Distance Error (ADE) and Final Distance Error (FDE). ADE measures the average Euclidean distance between the predicted and the true trajectories across all prediction steps, while FDE assesses the Euclidean distance at the final prediction time step. The calculation formula for both is as follows:

$$ADE = \frac{\sum_{i=1}^{N}\sum_{t=T_{obs}+1}^{T_{obs}+T_{pre}} \|Y_t^i - \hat{Y}_t^i\|_2}{N \times T_{pre}} \tag{36}$$

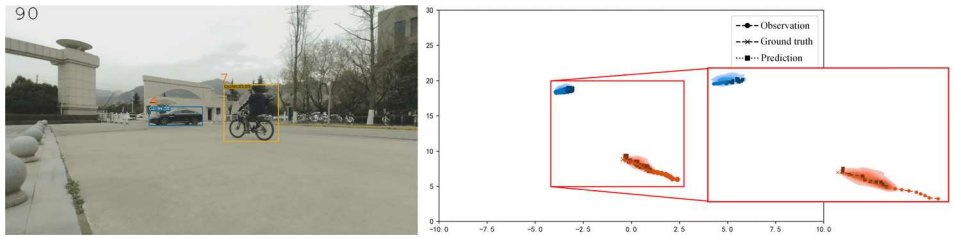
$$FDE = \frac{\sum_{i=1}^{N} \|Y_{T_{obs}+T_{pre}}^i - \hat{Y}_{T_{obs}+T_{pre}}^i\|_2}{N} \tag{37}$$

In this equation, Y_t^i and $\hat{Y}_t^i$ represent the true position and the predicted position of the i^{th} pedestrian at time step t, respectively. The ADE is computed by averaging the Euclidean distance error over $T_{pre} = 12$ frame across the predicted trajectory, while the FDE calculates the Euclidean distance error only at the final time step of the prediction.

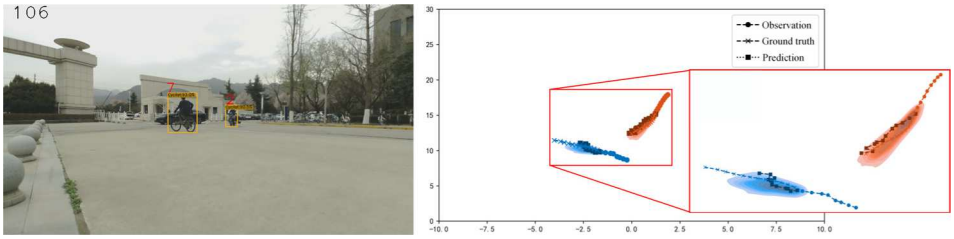
On the ETH/UCY pedestrian trajectory dataset, our approach, KITNet, was compared with other Gaussian distribution-based probabilistic models. Each model's output was sampled 6/20 times according to the standard (20 for ETH/UCY and NGSIM, and 6 for Argoverse 2), and the minimum ADE and FDE from these samples were recorded as the evaluation metrics. These metrics represent the best possible performance the models could achieve. The detailed results are presented in Table III. From the prediction results,

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on December 10,2025 at 03:30:12 UTC from IEEE Xplore. Restrictions apply.

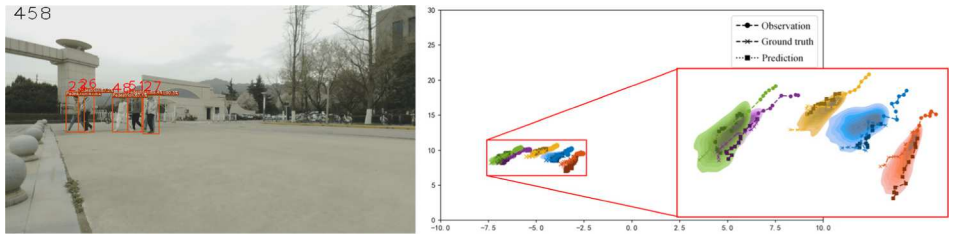
This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



(a) Heterogeneous targets



(b) Homogeneous targets



(c) Multiple targets

Fig. 7. Comparison result of actual data for predictor inference.

KITNet achieved the best performance in 80% of the evaluated scenarios across the five scenes. Notably, KITNet showed superior performance in predicting the final trajectory point (FDE) in 100% of the cases, indicating its stronger ability to forecast the long-term trajectory trend.

D. Real-World Deployment

For physical experiment, we utilize the hardware platform and camera calibration in Figure 6. The system primarily consists of a wheeled robotic chassis, which receives control signals to manage the robot's movement. It is equipped with an Industrial Personal Computer (IPC) responsible for processing and executing various computational and analytical tasks. Additionally, a suite of sensors is integrated to gather environmental data surrounding the robot. These sensors include a monocular camera, a 32-line LiDAR, and GNSS equipment, among others. To validate the performance of the proposed predictor in real-world environments, data from actual scenes using this platform were recorded, and the target's historical trajectories were obtained through object tracking methods. The data were played frame by frame, and targets with complete 20-frame trajectory data were extracted. The first 8 frames of each trajectory were utilized as observation data, while the subsequent 12 frames served as ground truth. The predictor estimated the bivariate Gaussian parameters for predicting the target's velocity over the next 12 frames. From these parameters, 100 sample trajectories were drawn to visualize the predicted trajectory distribution. For a clearer comparison, the mean trajectory derived from the predicted parameters was plotted alongside the true trajectory, as shown in Figure 7.

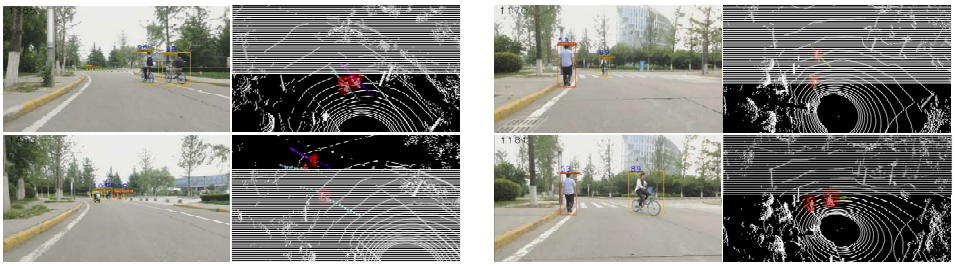
In Figure 7, the left panel displays the corresponding camera image frame, while the right panel shows a top-down view comparing the predicted and actual trajectories. In the top-down view, the robot is positioned at the origin, with each color-coded trajectory representing a distinct target, and the shaded areas depicting the predicted trajectory distributions.

Frame 90 illustrates the prediction for heterogeneous targets (cars and cyclists). The top-down view shows that the predicted trajectories closely match the true trajectories, with the predicted distribution encompassing the actual movement. In Frame 106, KITNet predicts the trajectories of two cyclists, accurately capturing their movement states and minimizing the prediction error. Frame 458 showcases KITNet's ability to handle multiple pedestrian trajectories, demonstrating the effectiveness of the model in predicting group behaviors.

To validate the performance benefits of our trajectory prediction approach within a practical navigation and planning context for transport robots, the developed framework—integrating detection, prediction, and decision-making modules—was deployed and evaluated on an actual ADV's onboard computing platform. Operating as a cohesive processing pipeline, sensor inputs from the ADV's LiDAR and depth camera are first processed by the detection module to localize dynamic agents. These detected states are then passed

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on December 10,2025 at 03:30:12 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



(a) Homogeneous target, opposite direction (b) Heterogeneous target, opposite direction

Fig. 8. Effect of joint deployment with robots in static scenes.

to the prediction module, which forecasts the agents' future trajectories over a predefined horizon. Finally, the decision-making module utilizes both current states and predicted trajectories to generate appropriate motion commands and update the ADV's planned path.

The system's real-time operational output is visually represented in Figure 8, generated within the Rviz environment. The top-down view displays raw LiDAR point clouds (white), cylindrical bounding boxes (red contours) for detected dynamic agents, and overlaid trajectory data for each tracked agent (unique colors). Dimmer points indicate historical trajectories (past 8 frames), while brighter points represent the predicted trajectories (subsequent 12 frames) produced by our prediction module. This visualization effectively illustrates the framework's ability to perceive current states and anticipate future movements of surrounding agents.

In Figure 8(a), showing same-type targets moving in opposite directions, the point cloud view on the right reveals that the predictor effectively handles trajectory prediction in such scenarios. In Frame 1385, two cyclists move together, and since the adjacency matrix focuses on nearby targets' movement states, it predicts nearly identical trajectories for the pair. In Frame 1433, where the same-type targets move in opposite directions, the predictor uses their historical trajectory data to infer their future movements accurately.

In Figure 8(b), depicting different-type targets moving in opposite directions, both the detector and tracker function correctly, identifying the positions and historical trajectories of pedestrian #53 and cyclist #89 across two frames. Despite their differing motion characteristics, the predictor accurately forecasts their future trajectories. In Frame 53, the predictor captures the interaction between the targets, predicting that the pedestrian (#53) will avoid a potential collision by shifting left. In Frame 1181, the predictor successfully identifies the cyclist's (#89) intention to turn based on their historical trajectory, demonstrating the model's capacity to handle scenarios where target movement patterns differ significantly.

To rigorously assess the computational feasibility for real-world mobile robotic deployment, we conducted a systematic evaluation of the processing latency for each core module (detector, predictor, decider). Timing commenced upon receipt of input data at each module's interface and concluded upon completion of data processing and output encoding. The measured runtime statistics for each component are summarized in Figure 9.

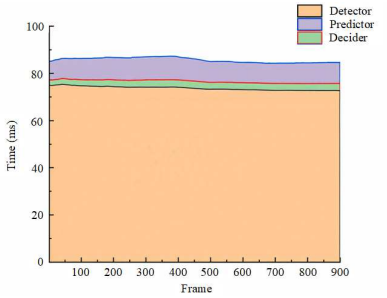


Fig. 9. Time consuming on each frame for ADV platform.

As shown in Figure 9, the sequentially integrated pipeline exhibits an average end-to-end processing latency of 84.59 milliseconds. This latency is well within the typical 100 ms (10 Hz) update rate of the ADV's onboard sensors, confirming that the proposed predictive navigation framework meets the real-time requirements for practical deployment in the considered operational scenarios. These results demonstrate strong potential for integrating KITNet-based predictive capabilities into real-time intelligent transportation systems.

VI. Conclusion

This study addresses the challenge of predicting future trajectories in multi-agent systems where complex and abrupt interactions significantly alter intended paths. We propose a novel trajectory prediction method with following key findings:

1. Sinusoidal curves can effectively model the trajectories of agents during turns, while Bezier curves are suitable for representing lane-changing behaviors.

2. Dividing the scene into sub-regions allows for a more stable assessment and computation of multi-agent interactions under complex behaviors. The resulting attention functions, reflecting interaction dynamics, achieved high degrees of fit.

3. A hierarchical GNN can more effectively capture the intrinsic features of agents, allowing for greater flexibility in setting edge attention weights. Leveraging mappings and metrics derived from regional set theory enables a more robust extraction of interaction characteristics among agents.

As for limitation, this inductive approach to classifying behavioral modes inherently struggles to provide exhaustive coverage for all possible scenarios, particularly long-tail cases arising from complex interactions and hybrid cases resulting from long-term temporal sequences. Under certain exceptional

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on December 10,2025 at 03:30:12 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

circumstances, the predicted behavior mode may be erroneous, leading to deviations in trajectory forecasting. Given that the current model primarily focuses on relatively superficial interaction modes such as collision avoidance, waiting, and following, we plan to address these limitations by incorporating a Vision-Language Model (VLM) driven by a Large Language Model (LLM). This will enable a deeper, more semantic interpretation of inter-agent intentions—for instance, to decipher more advanced, long-horizon interaction patterns such as: 'Pedestrian A appears to intend to cross the road towards the shop opposite, yet they seem unhurried and are decelerating. Vehicle B, on a tangentially intersecting path with Pedestrian A, would likely perceive this nuance and consequently accelerate to pass through.' By integrating such capabilities, we can achieve broader coverage of long-tail and hybrid scenarios, thus rendering the prediction model substantially more comprehensive.

REFERENCES

[1] D. Vashishtha and M. Panda, "Maximum likelihood multiple model filtering for path prediction in intelligent transportation systems," *Proc. Comput. Sci.*, vol. 143, pp. 635–644, 2018.

[2] E. Rehder and H. Kloeden, "Goal-directed pedestrian prediction," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 139–147.

[3] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg, "Who are you with and where are you going?," in *Proc. CVPR*, Jun. 2011, pp. 1345–1352.

[4] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 261–268.

[5] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," *Comput. Graph. Forum*, vol. 26, no. 3, pp. 655–664, Sep. 2007.

[6] D. Helbing, L. Buzna, A. Johansson, and T. Werner, "Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions," *Transp. Sci.*, vol. 39, no. 1, pp. 1–24, Feb. 2005.

[7] G. Ferrer, A. Garrell, and A. Sanfeliu, "Robot companion: A social-force based approach with human awareness-navigation in crowded environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 1688–1694.

[8] W. Choi and S. Savarese, "A unified framework for multi-target tracking and collective activity recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 215–230.

[9] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 51, no. 5, pp. 4282–4286, May 1995.

[10] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 961–971.

[11] Z. Pei, X. Qi, Y. Zhang, M. Ma, and Y.-H. Yang, "Human trajectory prediction in crowded scene using social-affinity long short-term memory," *Pattern Recognit.*, vol. 93, pp. 273–282, Sep. 2019.

[12] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Oct. 2018, pp. 2255–2264.

[13] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal LSTM with trust gates for 3D human action recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 816–833.

[14] Z. Fang, Q. Long, G. Song, and K. Xie, "Spatial–temporal graph ODE networks for traffic flow forecasting," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 364–373.

[15] L. Zhao et al., "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.

[16] H. Hu, Q. Wang, M. Cheng, and Z. Gao, "Trajectory prediction neural network and model interpretation based on temporal pattern attention," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 3, pp. 2746–2759, Mar. 2023.

[17] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2020, pp. 753–763.

[18] L. Zhang, Q. She, and P. Guo, "Stochastic trajectory prediction with social graph network," 2019, *arXiv:1907.10233*.

[19] Y. Huang, H. Bi, Z. Li, T. Mao, and Z. Wang, "STGAT: Modeling spatial–temporal interactions for human trajectory prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6271–6280.

[20] V. Kosaraju, A. Sadeghian, R. Martín-Martín, I. Reid, S. H. Rezatofighi, and S. Savarese, "Social-BiGAT: Multimodal trajectory forecasting using bicycle-GAN and graph attention networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 137–146.

[21] X. Lin, T. Liang, J. Lai, and J.-F. Hu, "Progressive pretext task learning for human trajectory prediction," 2024, *arXiv:2407.11588*.

[22] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14412–14420.

[23] M. Li et al., "FrameERC: Framelet transform based multimodal graph neural networks for emotion recognition in conversation," *Pattern Recognit.*, vol. 161, May 2025, Art. no. 111340.

[24] L. Bai et al., "HAQJSK: Hierarchical-aligned quantum Jensen-Shannon kernels for graph classification," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 11, pp. 6370–6384, Nov. 2024.

[25] M. Li et al., "Guest editorial: Deep neural networks for graphs: Theory, models, algorithms, and applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 4, pp. 4367–4372, Apr. 2024.

[26] S. Qiao, D. Shen, X. Wang, N. Han, and W. Zhu, "A self-adaptive parameter selection trajectory prediction approach via hidden Markov models," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 284–296, Feb. 2015.

[27] Z. Zhou et al., "A novel cooperative path planning method based on UCR-FCE and behavior regulation for large-scale multi-robot system," *Appl. Intell.*, vol. 53, pp. 30706–30745, Aug. 2023.

[28] P. Zhang, W. Ouyang, P. Zhang, J. Xue, and N. Zheng, "SR-LSTM: State refinement for LSTM towards pedestrian trajectory prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Aug. 2019, pp. 12085–12094.

[29] W. Tang, A. Huang, E. Liu, J. Wu, and R. Zhang, "A reliable robot localization method using LiDAR and GNSS fusion based on a two-step particle adjustment strategy," *IEEE Sensors J.*, vol. 24, no. 22, pp. 37846–37858, Nov. 2024.

[30] X. Mo, H. Liu, Z. Huang, X. Li, and C. Lv, "Map-adaptive multimodal trajectory prediction via intention-aware unimodal trajectory predictors," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 6, pp. 5651–5663, Jun. 2024.

[31] K. Li, S. Eiffert, M. Shan, F. Gomez-Donoso, S. Worrall, and E. Nebot, "Attentional-GCNN: Adaptive pedestrian trajectory prediction towards generic autonomous vehicle use cases," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 14241–14247.

[32] C. Yu, X. Ma, J. Ren, H. Zhao, and S. Yi, "Spatio-temporal graph transformer networks for pedestrian trajectory prediction," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 507–523.

[33] K. Mangalam et al., "It is not the journey but the destination: Endpoint conditioned trajectory prediction," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 759–776.

[34] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 1–12.

[35] Y. Yuan, X. Weng, Y. Ou, and K. Kitani, "AgentFormer: Agent-aware transformers for socio-temporal multi-agent forecasting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9793–9803.

[36] C. Xu, W. Mao, W. Zhang, and S. Chen, "Remember intentions: Retrospective-memory-based trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 6478–6487.

[37] Y. Liu, J. Zhang, L. Fang, Q. Jiang, and B. Zhou, "Multimodal motion prediction with stacked transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, Aug. 2021, pp. 7573–7582.

[38] J. Amirian, J.-B. Hayet, and J. Pettré, "Social ways: Learning multi-modal distributions of pedestrian trajectories with GANs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, Sep. 2019, pp. 1726–1735.

[39] Z. Zhou, J. Wang, Y.-H. Li, and Y.-K. Huang, "Query-centric trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, Jun. 2023, pp. 17863–17873.

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on December 10,2025 at 03:30:12 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[40] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1468–1476.
[41] V. Katariya, M. Baharani, N. Morris, O. Shoghli, and H. Tabkhi, "DeepTrack: Lightweight deep learning for vehicle trajectory prediction in highways," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 18927–18936, Oct. 2022.
[42] L. Lin, W. Li, H. Bi, and L. Qin, "Vehicle trajectory prediction using LSTMs with spatial–temporal attention mechanisms," *IEEE Intell. Transp. Syst. Mag.*, vol. 14, no. 2, pp. 197–208, Mar. 2022.
[43] G. A. Noghre, V. Katariya, A. D. Pazho, C. Neff, and H. Tabkhi, "Pishgu: Universal path prediction network architecture for real-time cyber-physical edge systems," in *Proc. ACM/IEEE 14th Int. Conf. Cyber-Physical Syst. (CPS-IoT Week)*, May 2023, pp. 88–97.

Zeyu Zhou received the B.S. degree in mechanical design, manufacturing and automation from Xi'an Technological University, Xi'an, China, in 2019, and the M.S. degree in control engineering from Northwestern Polytechnical University, Shaanxi, China, in 2022, where he is currently pursuing the Ph.D. degree in control science and engineering and the Ph.D. degree in mechnical engineering with The Hong Kong Polytechnic University, Hong Kong. His research interests include multi-robot path planning, trajectory prediction and decision making, deep learning, and reinforcement learning. He won the title of Suzhou Fencai Leading Talent in 2022, Shaanxi Excellent Graduate, and the Red Dot China Good Design Award.

Shanqing Wang received the B.S. degree in robotics engineering from Northwestern Polytechnical University, Xi'an, China, in 2024. He is currently pursuing the M.S. degree in electrical and electronic engineering with Zhejiang University, Hangzhou, China. His current research interests include trajectory prediction, control systems, artificial intelligence, and digital twins.

Anmin Huang received the B.S. degree in mechanical design, manufacturing and automation from Harbin Engineering University, Harbin, China, in 2020, and the M.S. degree in mechanical engineering from Northwestern Polytechnical University, Xi'an, China, in 2023, where he is currently pursuing the Ph.D. degree in control engineering. His current research interests include perception and multisensor fusion localization of mobile robots.

Jin Lou received the B.S. degree in automation from Hebei University of Engineering, Handan, China, in 2021, and the M.S. degree in control engineering from Northwestern Polytechnical University, Xi'an, China, in 2024. His current research interests include trajectory prediction, target recognition and tracking, and embodied AI.

Wei Tang (Member, IEEE) received the B.S. and M.S. degrees in mechanical design, manufacturing and automation and the Ph.D. degree in control science and engineering from Northwestern Polytechnical University, Xi'an, China. His research interests include autonomous mobile robot, vibration control, and engine intelligent control. He is a Distinguished Patent Examination Expert of China National Intellectual Property Administration and a Review Expert of the National Natural Science Foundation of China.

David Navarro-Alarcon (Senior Member, IEEE) received the Ph.D. degree in mechanical and automation engineering from The Chinese University of Hong Kong (CUHK), Hong Kong, in 2014. From 2015 to 2017, he was a Research Assistant Professor with the CUHK T Stone Robotics Institute. Since 2017, he has been with The Hong Kong Polytechnic University (PolyU), Hong Kong, where he is currently an Associate Professor with the Department of Mechanical Engineering. His current research interests include perceptual robotics and control theory. He currently serves as an Associate Editor for IEEE TRANSACTIONS ON ROBOTICS.

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on December 10,2025 at 03:30:12 UTC from IEEE Xplore. Restrictions apply.