Spatial-Temporal Graph Representation Learning for Multi-Agent Trajectory Prediction

RUOCHEN LI

Spatial-Temporal Graph Representation Learning for Multi-Agent Trajectory Prediction

Ruochen Li

A thesis presented for the degree of
Doctor of Philosophy at Durham University

Department of Computer Science
Durham University
United Kingdom
2026-01-19

Abstract

Trajectory prediction entails the forecasting of future movement trajectories of traffic agents derived from their historical observed behaviors. This sophisticated technique is essential for various real-world applications such as path planning and collision avoidance for autonomous driving systems, and anomaly detection within video surveillance technologies.

However, trajectory prediction for multi-agent scenarios presents significant challenges due to the complex interaction dynamics across diverse traffic environments. These environments can range from homogeneous settings dominated by similar agents (*e.g.*, pedestrians in crowds) to heterogeneous scenes with mixed agent types (*e.g.*, pedestrians, vehicles, cyclists, *etc.*). To tackle these challenges, an integrated understanding of agent behaviors across diverse contexts is essential. Agents continuously adjust their movements based on surrounding entities, creating complex interaction patterns that vary between homogeneous pedestrian crowds and heterogeneous traffic scenarios. Capturing these nuanced spatial–temporal inter-dependencies demands sophisticated models that represent both individual and collective dynamics while accommodating distinct agent behaviors. The primary aim of this research is to develop robust and accurate trajectory prediction frameworks capable of bridging this gap and operating effectively across both homogeneous and heterogeneous contexts. To achieve this aim, this dissertation pursues three core objectives: **(1)** Analyzing dynamics and spatial–temporal interactions in homogeneous pedestrian crowds. **(2)** Understanding interaction patterns for heterogeneous traffic environments with diverse agent types. **(3)** Developing a unified framework that integrates insights from both heterogeneous and homogeneous settings for improved and robust trajectory prediction.

The motivation of this research stems from the limitations of existing trajectory prediction methods across different settings. In homogeneous pedestrian scenarios, highly interactive and collective behaviors pose challenges for modeling high-order spatial–temporal dependencies. In heterogeneous environments, diverse agent types such as pedestrians, cyclists, and vehicles exhibit asymmetric dynamics that remain difficult to capture with current approaches. Moreover, most methods treat these contexts in isolation, lacking robustness and generalization in real-world environments. Addressing these gaps calls for unified graph-based frameworks that can integrate insights from both domains while advancing spatial–temporal modeling to represent complex interactions and long-range dependencies more effectively.

This research introduces a series of novel frameworks designed to enhance the robustness and accuracy of trajectory prediction under different settings. We begin by addressing the challenges of homogeneous pedestrian trajectory prediction, where the highly interactive nature of pedestrians and their collective behaviors demand precise modeling of spatial–temporal relationships. To this end, we propose **UniEdge**, a dual-graph–inspired unified spatial–temporal edge-enhanced graph network that effectively captures both high-order cross-time interactions and complex

influence patterns between pedestrians, providing more accurate and socially aware predictions in homogeneous settings.

We then extend our investigation to heterogeneous environments featuring multiple interacting agent types. For this purpose, we propose **Multiclass-SGCN**, a sparse graph-based trajectory prediction network with agent class embedding that models the unique dynamics among heterogeneous agents such as pedestrians, vehicles, and cyclists. By integrating semantic agent-class information with motion features, Multiclass-SGCN explicitly represents cross-type interaction dynamics while maintaining computational efficiency.

Building on the insights gained from both homogeneous and heterogeneous contexts, and recognizing the need for a more broadly applicable solution, we propose a behavioral pseudo-label informed sparse graph convolution network (**BP-SGCN**) for trajectory prediction across both settings. It introduces the novel concept of behavioral pseudo-labels to represent different movement patterns of traffic agents without requiring additional annotations. Through a cascaded training scheme that optimizes clustering and trajectory prediction in tandem, BP-SGCN effectively captures both inter-class and intra-class behavioral variations, offering a robust, unified framework for trajectory prediction across diverse environments.

Our extensive experimental evaluations and qualitative analyses across multiple benchmark datasets consistently demonstrate that the proposed frameworks outperform state-of-the-art methods in trajectory prediction, validating the effectiveness of our progressive research approach from homogeneous to heterogeneous to unified prediction systems.

Declaration

The work in this thesis is based on research carried out at the Department of Computer Science, Durham University, United Kingdom. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Professor Hubert P. H. Shum, for his unwavering guidance, patience, and support throughout my doctoral journey at Durham. His ability to identify potential in my early, unpolished ideas and transform them into meaningful research has been invaluable to my academic growth. Beyond providing direction, he has been a constant source of encouragement, ensuring that I had the resources and confidence needed to overcome challenges and stay the course. I am sincerely thankful for his trust, mentorship, and the many opportunities he has created for me—contributions that have profoundly shaped both my research and my professional development.

I would also like to extend my sincere gratitude to my co-supervisor, Dr Stamos Katsigiannis, whose patience, openness, and insightful guidance have greatly enriched my PhD journey. He has always been willing to engage in thoughtful discussions on a wide range of ideas, encouraging me to explore different perspectives and refine my research directions. His constructive feedback and steady encouragement have been instrumental in helping me navigate complex problems with clarity and confidence, making him an invaluable source of support throughout this work.

I am deeply grateful to my dear friends in our research group — Tanqiu Qiao, Shuang Chen, Li Li, Haozheng Zhang, Manli Zhu, Xiatian Zhang, Ziyi Chang, Xiaotang Zhang, and Ruisheng Han — for bringing joy, encouragement, and camaraderie into my life. Your friendship has been a constant source of motivation, providing light-hearted moments and welcome distractions that have helped me stay balanced amidst the demands of research. I am also grateful to Shuang Chen and Jiacheng Yao for infusing my life beyond research with joy and camaraderie. From countless Dota 2 battles to late-night conversations, we have shared over a thousand hours of laughter, friendly rivalry, and much-needed escapes from the pressures of academic life.

To my partner, Lunette, my constant source of encouragement and unwavering support throughout this journey. You have been beside me in moments of uncertainty, offering comfort, patience, and understanding. Your belief in me has been a steady light, guiding me through challenges and reminding me of the bigger picture beyond deadlines and experiments.

To my parents, the unwavering foundation and guiding compass of my life—you are the roots that keep me steady and the strength that propels me forward. Your boundless love and steadfast support have carried me across continents in pursuit of my dreams, giving me the courage to face uncertainty and the resilience to overcome setbacks. The sacrifices you have made, often quietly and without recognition, have laid the path for every opportunity I have been given. The wisdom you have imparted has illuminated my way through challenge and change, and your belief in me has been the constant light in moments of doubt. I carry your strength, values, and lessons in all that I do, knowing that every milestone I reach is built upon the foundation you have given me.

Dedication

To my parents.

Contents

List of Figures

List of Tables

CHAPTER 1

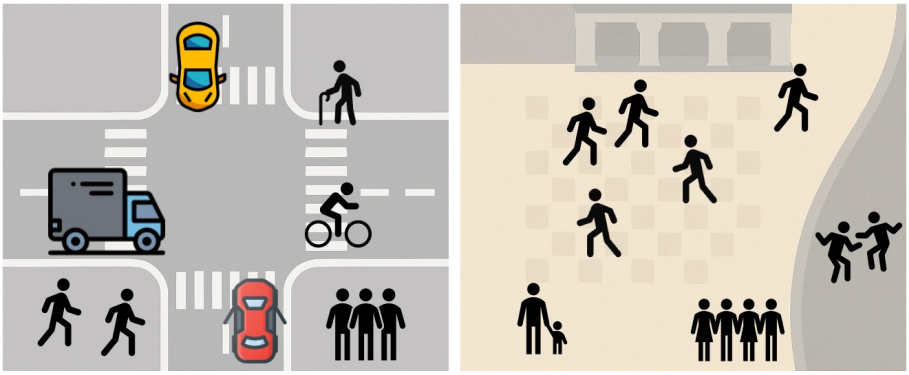
Introduction

Trajectory prediction, which forecasts the future movement paths of traffic agents based on their historical behaviors, is a critical technology widely applied in modern applications such as autonomous driving for collision avoidance systems, emergency braking systems [8–11] and video surveillance technologies for identifying suspicious activities [12–15]. For example, Fig. 1.1(a) shows an overhead surveillance camera capturing pedestrian flows in a public area, Fig. 1.1(b) depicts a sidewalk robot navigating in a pedestrian-rich environment. Both scenarios require accurate trajectory prediction to ensure safety and support reliable decision-making.

(a) (b)

Figure 1.1: Examples of real-world scenarios that rely on trajectory prediction. (a) Video surveillance of public area [1]; (b) Delivery robot operating in urban environment [2].

In multi-agent crowd environments, trajectory prediction refers to the task of forecasting future trajectories of multiple interacting agents over a specific time horizon, given their observed past movements. Unlike traditional time-series forecasting tasks, which typically analyze data with strong periodicity, identifiable trends, and a stable number of variates [16–18], trajectory prediction presents unique complexities. Agent movements often lack predictable cycles, the number of interacting entities fluctuates dynamically, and crucially, future paths are dominated by complex, emergent interactions between agents rather than simply extrapolating past individual behavior. This complexity manifests in both the spatial domain, where agents navigate shared physical spaces with varying constraints, and the temporal domain, where movement decisions evolve dynamically based on changing contexts [19–23]. Notably, the nature and intensity of these challenges can vary considerably depending on the composition and structure of the environment.



(a) (b)

Figure 1.2: Examples of real-world scenarios that rely on trajectory prediction. (a) Heterogeneous traffic scenarios; (b) Homogeneous pedestrian scenarios.

Trajectory prediction tasks are commonly categorized based on the environment's composition into two main types: homogeneous pedestrian scenarios and heterogeneous scenarios. The former involves interactions exclusively among agents of the same type—pedestrians—while the latter includes diverse agent types such as pedestrians, vehicles, and cyclists, etc (Fig. 1.2). In homogeneous pedestrian scenarios, the primary challenge lies in modeling behaviors that are both highly stochastic and socially driven. Although pedestrians belong to a single agent class, their motion patterns vary widely due to latent factors such as intent, personality, and social affiliations [24–26]. These

behaviors unfold in open environments without explicit traffic rules or physical constraints, leading to significant freedom of movement and unpredictability [21, 27–30]. As a result, effective forecasting requires models capable of inferring subtle social cues, anticipating complex group dynamics, and capturing implicit high-order interactions [31] that emerge from indirect and non-obvious dependencies among agents. In contrast, heterogeneous scenarios present additional challenges arising from the coexistence of agents with fundamentally different dynamics and interaction patterns [3, 32–35]. For instance, pedestrians exhibit flexible and reactive behaviors, whereas vehicles and cyclists operate under stricter kinematic constraints and traffic regulations. These disparities give rise to asymmetric interactions that are difficult to model using conventional approaches [19, 36], thereby requiring reasoning mechanisms capable of capturing agent-specific behaviors and cross-type influences [19, 34, 37].

To address the distinct challenges posed by different environment types, this thesis investigates trajectory prediction in both homogeneous pedestrian scenarios and heterogeneous multi-agent scenarios, each presenting unique modeling difficulties. In homogeneous settings, we focus on capturing latent social dynamics and high behavioral stochasticity, while in heterogeneous environments, we emphasize agent-type-specific reasoning and asymmetric cross-agent interactions. Building on insights from both domains, we further explore a unified modeling framework that integrates the strengths of each approach—aiming to generalize across diverse scenarios and effectively capture both implicit social cues and explicit inter-agent heterogeneity. While our goal is to develop a unified framework that addresses both homogeneous and heterogeneous challenges, doing so requires confronting several fundamental limitations in existing modeling approaches.

1.1 Motivations

While earlier approaches to trajectory prediction relied on rule-based models and probabilistic frameworks, recent progress has shifted the focus toward deep learning, owing to its superior capacity to model complex and non-linear agent interactions. Despite strong performance on standard benchmarks, accurately forecasting agent trajectories

in real-world, multi-agent environments remains a significant challenge. This difficulty stems not only from the diversity of agent behaviors and the stochastic nature of human and vehicular motion, but also from the limitations of current modeling paradigms in capturing the underlying relational and temporal dependencies. Graph-based representation learning has emerged as a promising direction, offering a natural way to encode agent-to-agent relationships through structured message passing. However, existing graph-based methods still face technical limitations in modeling dynamic spatial-temporal interactions, such as insufficient representation of high-order dependencies, limited generalization across heterogeneous environments, and the inability to adaptively reason over varying interaction complexities. Addressing these challenges calls for more expressive, flexible, and context-aware modeling frameworks.

A key modeling challenge in trajectory prediction lies in accurately capturing complex agent interactions under varying contextual and structural constraints. In homogeneous pedestrian scenarios, the difficulty arises not from agent-type diversity but from the need to model fine-grained social cues, group dynamics, and inherently stochastic behaviors. While agent semantics are uniform, many existing methods adopt a decoupled spatial-temporal modeling strategy—first encoding spatial interactions frame by frame, then learning temporal dependencies separately [27–31, 38, 39]. This separation limits the ability to capture higher-order temporal dependencies and often disrupts the temporal consistency of social interactions in dynamic crowds. Meanwhile, heterogeneous environments introduce additional complexity due to the coexistence of multiple agent types. Conventional methods [20, 28, 29, 40, 41] often struggle to handle asymmetric interactions and semantic distinctions across agent classes, leading to oversimplified or overly dense graph structures that obscure the relative importance of cross-type relations. Although recent approaches have attempted to incorporate semantic labels to differentiate agent types [6, 34, 42], they typically rely on densely connected spatial graphs, introducing redundant links that dilute informative signals and hinder model efficiency and interpretability.

Beyond context-specific limitations, a broader challenge lies in the generalizability of learned interaction representations in multi-agent trajectory prediction. Many existing models are tightly coupled with particular scene characteristics or training distributions,

resulting in limited effectiveness when applied to novel or unseen environments. Addressing this issue requires frameworks that can extract transferable behavioral patterns without relying heavily on manual annotations or dataset-specific assumptions.

This thesis is motivated by core challenges in trajectory prediction that arise from both practical complexity and technical limitations. Real-world environments are increasingly dynamic and diverse—ranging from unstructured homogeneous pedestrian crowds to structured heterogeneous scenarios with multiple agent types—creating demand for models that can capture nuanced social dynamics, reason over heterogeneous agent semantics, and generalize beyond narrow training distributions. To address these needs, this thesis develops unified, graph-based frameworks that systematically improve semantic reasoning, spatiotemporal representation, and generalization.

1.2 Research Aims

The primary aim of this thesis is to advance multi-agent trajectory prediction by developing novel, robust, and generalizable graph-based frameworks. These frameworks are designed to enhance semantic expressiveness, spatial-temporal modeling accuracy, and adaptability in complex interaction scenarios. Traditional methods often face critical limitations, including poor scalability to heterogeneous agent types, reliance on costly manual class annotations, limited capacity to capture high-order spatial-temporal dependencies, inefficient information propagation due to multi-step aggregation, and the neglect of implicit interaction patterns encoded in edge features. This research seeks to address these challenges through the following objectives:

1. **To develop semantic-aware, adaptive graph-based models for heterogeneous trajectory prediction**: This research aims to design novel graph architectures capable of effectively integrating semantic label information and selectively modeling interactions between different agent classes. By doing so, it seeks to eliminate redundant connections, enhance the representation of asymmetric relational cues, and ultimately improve the accuracy and efficiency of trajectory prediction in complex heterogeneous traffic scenarios.

2. **To establish unified spatial-temporal modeling frameworks for homogeneous pedestrian environments**: This research aims to unify spatial and temporal interactions into coherent modeling frameworks that accurately capture higher-order interaction patterns and the evolution of social dynamics over time. Special emphasis is placed on leveraging edge-level relational information to better represent implicit social influences and temporal dependencies within dynamic pedestrian crowds.

3. **To develop unsupervised and generalizable behavior representation learning approaches**: Recognizing the limitations of annotation-intensive supervised methods, this research seeks to investigate unsupervised or weakly-supervised strategies for learning transferable behavioral representations. The ultimate goal is to enhance model adaptability and robustness, enabling trajectory prediction frameworks to generalize effectively across both homogeneous pedestrian scenarios and heterogeneous multi-agent environments, thus reducing reliance on extensive manual labeling and supporting deployment in diverse real-world settings.

Together, these research objectives directly address key limitations of conventional deep learning models for trajectory prediction. By incorporating semantic information and adaptive graph sparsification, the proposed methods improve interaction modeling in heterogeneous scenarios, while unified spatial-temporal representations deepen understanding of complex social dynamics in pedestrian crowds. Furthermore, exploring unsupervised and generalizable behavior representations enhances adaptability to unseen environments, reducing the need for costly annotations. Collectively, these aims lay the foundation for robust, interpretable, and transferable trajectory prediction models applicable to domains such as autonomous navigation, intelligent surveillance, and crowd management.

1.3 Contributions

This thesis makes several key contributions towards developing more robust, adaptable, and accurate graph-based spatial-temporal modeling techniques for multi-agent trajectory prediction, as summarized below:

- We propose **Multiclass-SGCN** (Chapter 3), a semantic-aware sparse graph convolutional framework for heterogeneous trajectory prediction. The model embeds agent-type semantics and motion cues through velocity–label representations and constructs adaptive sparse interaction graphs via an attention-based masking strategy. This design reduces redundant cross-type connections, captures asymmetric relational patterns more effectively, and significantly improves prediction accuracy in heterogeneous traffic scenes with pedestrians, cyclists, and vehicles

- We present **UniEdge** (Chapter 4), a unified spatial-temporal graph network for homogeneous pedestrian trajectory prediction. It models high-order cross-time interactions through a patch-based spatial-temporal formulation and introduces a dual-graph convolutional module to jointly capture node- and edge-level dependencies. Coupled with a Transformer encoder-based predictor, this framework achieves strong performance on multiple public pedestrian datasets by modeling both fine-grained social dynamics and long-range temporal correlations.

- We present **BP-SGCN** (Chapter 5), an unsupervised framework that learns structured behavioral representations via pseudo-label–guided deep clustering with cross-scale structural consistency. This approach enables robust trajectory forecasting without manual annotations and generalizes effectively across both heterogeneous and homogeneous pedestrian environments, facilitating the discovery of transferable motion patterns in diverse real-world scenarios.

1.4 Publications

The research related to this thesis has been previously published in the following peer-reviewed publications:

- **Li, R.**, Katsigiannis, S., & Shum, H. P. H., "Multiclass-SGCN: Sparse Graph-based Trajectory Prediction with Agent Class Embedding." In *Proceedings of the IEEE International Conference on Image Processing* (ICIP), 2022. . . . . (Chapter 3)

- **Li, R.**, Qiao, T., Katsigiannis, S., Zhu, Z., & Shum, H. P. H., "Unified Spatial-Temporal

Edge-Enhanced Graph Networks for Pedestrian Trajectory Prediction." *IEEE Transactions on Circuits and Systems for Video Technology* (TCSVT), 2025. . (Chapter 4)

- **Li, R.**, Katsigiannis, S., Kim, T.-K., & Shum, H. P. H., "BP-SGCN: Behavioral Pseudo-Label Informed Sparse Graph Convolution Network for Pedestrian and Heterogeneous Trajectory Prediction." *IEEE Transactions on Neural Networks and Learning Systems* (TNNLS), 2025. . . . . (Chapter 5)

In addition to the listed publications above, there are other peer-reviewed publications that have not been included in this thesis:

- **Li, R.**, Zhu, Z., Qiao, T., & Shum, H. P. H., "ViTE: Virtual Graph Trajectory Expert Router for Pedestrian Trajectory Prediction." *under review at AAAI*, 2026.
- Qiao, T., **Li, R.**, Li, F. W. B., & Shum, H. P. H., "From Category to Scenery: An End-to-End Framework for Multi-Person Human-Object Interaction Recognition in Videos." *International Conference on Pattern Recognition* (ICPR), pp. 262–277, 2024.
- Qiao, T., **Li, R.**, Li, F. W. B., Kubotani, Y., Morishima, S., & Shum, H. P. H., "Geometric Visual Fusion Graph Neural Networks for Multi-Person Human-Object Interaction Recognition in Videos." *Expert Systems with Applications* (ESWA), vol. 290, p. 128344, 2025.

1.5 Thesis Structure

This thesis advances trajectory prediction by leveraging graph-based representation learning in both heterogeneous and homogeneous multi-agent environments. The chapters are structured to lead the reader from motivation and theoretical foundations through methodological innovations to empirical validation, ensuring a coherent and progressive narrative.

Chapter 1 outlines the importance of trajectory prediction in real-world applications such as autonomous driving and video surveillance. It highlights the central challenges of modeling complex spatial-temporal interactions and heterogeneous agent semantics, and clearly states the research motivation, objectives, and key contributions of the thesis.

Chapter 2 presents a comprehensive literature review covering four key areas: trajectory prediction in both homogeneous pedestrian crowds and heterogeneous multi-agent environments; advances in spatial-temporal graph representation learning; approaches to unsupervised behavior clustering; and standard evaluation metrics for trajectory forecasting. The chapter systematically identifies the limitations of existing methods and positions the proposed frameworks within the broader research landscape.

Chapter 3 introduces **Multiclass-SGCN**, a sparse graph convolutional model for heterogeneous trajectory prediction. This chapter presents the velocity–label embedding, adaptive interaction masking, and the model's ability to selectively capture meaningful spatial-temporal interactions among diverse agent types.

Chapter 4 presents **UniEdge**, a unified spatial-temporal graph network for homogeneous pedestrian scenarios. It employs a patch-based spatial-temporal formulation and a dual-graph reasoning mechanism (E2E-N2N-GCN) to jointly model node- and edge-level interactions, further enhanced with a Transformer-based predictor for long-range temporal reasoning.

Chapter 5 introduces **BP-SGCN**, an unsupervised framework for transferable trajectory representation learning. This chapter proposes a pseudo-label–guided clustering strategy with cross-scale consistency training, enabling generalization across both heterogeneous and homogeneous scenes without manual annotations.

Finally, Chapter 6 summarizes the main contributions of the thesis, reflects on the methodological advances in graph-based trajectory prediction, and outlines future directions including multimodal and contextual integration, adaptive continual learning, and closed-loop evaluation in high-fidelity simulation.

CHAPTER 2

Literature Review

Trajectory prediction in multi-agent environments represents a fundamental challenge in computer vision and robotics, requiring the forecasting of future movement patterns based on observed historical behaviors while considering complex inter-agent interactions and environmental constraints. This capability is essential for numerous applications including autonomous navigation systems [9–11], intelligent surveillance [12–15], and crowd management [43, 44], where understanding and anticipating agent movements ensures safety and enables proactive decision-making.

In this chapter, we present a comprehensive review of trajectory prediction research, systematically examining the evolution from classical approaches to state-of-the-art graph-based methods. Section 2.1 establishes the foundational concepts by delineating the distinct challenges posed by homogeneous pedestrian scenarios versus complex heterogeneous environments. Section 2.2 delves into the critical aspects of spatial-temporal modeling, examining how graph-based spatial interactions and temporal dependencies are captured and integrated to enhance prediction accuracy. Section 2.3 reviews clustering-based approaches for modeling behavioral patterns in trajectory prediction, highlighting the evolution from traditional distance-based methods to recent deep embedding techniques that enable more expressive and adaptive behavior representation. Finally,

Section 2.4 showcases the evaluation metrics used in trajectory prediction. Throughout this review, we identify key limitations in existing approaches and highlight research gaps that motivate our proposed frameworks, demonstrating how our contributions advance the state-of-the-art in graph-based trajectory prediction across both heterogeneous and homogeneous settings.

2.1 Multi-Agent Trajectory Prediction

This section provides an overview of multi-agent trajectory prediction across diverse environmental contexts, organized into two fundamental categories: homogeneous pedestrian trajectory prediction (Section 2.1.1) and heterogeneous trajectory prediction (Section 2.1.2). The former focuses on predicting future positions of pedestrians in structured, single-agent-type environments, while the latter addresses more complex scenes involving diverse agent types such as vehicles, cyclists, and pedestrians.

2.1.1 Trajectory Prediction in Homogeneous Pedestrian Crowds

The field of homogeneous pedestrian trajectory prediction is dedicated to forecasting the future trajectories of pedestrians. In contrast to heterogeneous scenarios, this setting presents distinct challenges due to the high stochasticity and intricate social dynamics of pedestrian behavior. Despite the apparent homogeneity, individual pedestrians exhibit significant diversity in their underlying behaviors, influenced by latent factors like personality, intentions, and social affiliations [45, 46].

Traditional rule-based approaches

Pedestrian trajectory prediction has long been a topic of interest, well before the advent of deep learning. Before the rise of data-driven techniques, early pedestrian motion modeling predominantly relied on physically inspired or rule-based frameworks, which emphasized interpretability and leveraged handcrafted domain knowledge to describe agent behaviors. These approaches offer strong interpretability and often reflect intuitive or domain-specific behaviors. Among them, the Social Force Model (SFM) [47] stands out as a foundational framework, modeling pedestrians as particles subjected to attractive

forces toward their goals and repulsive forces from other agents and obstacles. These typically include an attractive force drawing the pedestrian toward their intended goal, and repulsive forces generated by nearby agents and static obstacles. According to Newton's second law, the net force determines the acceleration of the pedestrian [48], forming a continuous-time dynamic system. When considering only the attractive component, the model serves as a basic goal-directed motion generator. This model has inspired numerous extensions, including the Generalized Centrifugal Force Model (GCFM) [49], which incorporates anisotropic sensitivity zones and velocity-adaptive forces for better realism in dense crowds. As reviewed in [50–52], SFM has also been integrated with heuristic decision layers or game-theoretic components to capture complex interactions like yielding or negotiation with vehicles.

Another prominent line of rule-based models includes Cellular Automata (CA) approaches [53, 54], which discretize the spatial domain and evolve agent states using local transition rules. Models such as the Floor Field Model [55] simulate attractive potentials toward exits and congestion-based repulsive fields, enabling efficient simulation of crowd flow and evacuation scenarios. Some recent CA-based works extend their applicability to pedestrian-vehicle interactions in semi-structured zones like drop-off areas [56]. In addition to force-based models, some early approaches adopted velocity-based heuristics to model motion trends. For example, the Velocity Obstacle (VO) paradigm [57] predicts collisions based on extrapolated velocities and defines avoidance maneuvers through geometric rules. Variants like Reciprocal Velocity Obstacles (RVO) [58] account for mutual adaptation, making them suitable for real-time multi-agent planning. These models are commonly used in robot navigation and multi-agent systems due to their low computational cost and real-time applicability.

Despite their conceptual simplicity and computational efficiency, traditional models are limited by manually crafted assumptions and struggle to generalize to unstructured environments or capture long-range dependencies. Their deterministic structure and constrained expressiveness have prompted a shift toward data-driven approaches, which learn complex interaction patterns from real-world trajectories while retaining the ability to integrate physical priors.

Deep learning–based approaches

While traditional models offer interpretable, physics-based frameworks, their limitations in handling uncertainty and complex human behaviors have driven a shift toward data-driven deep learning approaches. Early deep learning models for pedestrian trajectory prediction primarily relied on recurrent neural networks (RNNs) to capture temporal dependencies. Social-LSTM [28] introduced a social pooling layer to encode interactions among pedestrians, enabling context-aware forecasting. SS-LSTM [59] further enhanced this architecture by incorporating occupancy grid–based representations, improving interaction awareness in dense scenes. However, these deterministic models struggle to capture the intrinsic uncertainty and multimodality of human motion, prompting a shift toward generative approaches.

To address the inherently multimodal nature of pedestrian futures, generative models such as Generative Adversarial Networks (GANs) [60] and conditional VAEs (CVAEs) have been widely adopted. Social-GAN [21] employs adversarial training to generate socially plausible, diverse trajectories, using a pooling module to encode interactions. In parallel, CVAE-based methods offer a probabilistic framework by learning a latent distribution over future intentions. Representative works include DESIRE [40], which combines CVAE with inverse reinforcement learning for intent inference; Trajectron++ [61], which models multiple agents in a dynamic probabilistic graphical model; and Y-Net [5], which introduces a goal-conditioned decoder to better structure the latent space. Further enhancements, such as SocialVAE [22], incorporate scene and agent priors to guide sampling and improve diversity. While CVAE frameworks provide interpretability and structured uncertainty, they often require careful regularization to avoid mode collapse or blurred predictions.

Graph-based approaches have gained traction due to their ability to naturally model spatial interactions among agents. In this paradigm, each pedestrian is represented as a node, and their interactions are encoded via dynamic edges that evolve over time. Social-STGCNN [29] introduces a spatial-temporal graph convolution framework that jointly captures spatial dependencies and temporal evolution. STGAT [38] and Social-BiGAT [62] apply attention mechanisms on graphs to dynamically weigh neighbors based on relevance. These methods demonstrate improved generalization in complex

scenes, especially in crowded or structured environments. Recent works have begun to explore high-level graph frameworks that combine edge and node features to capture richer relational information. GroupNet [63] pioneered this direction by introducing interaction strength and category features to enhance edge significance beyond simple connections. Following this trend, GC-VRNN [64], HEAT [65], and MFAN [66] further advance graph modeling by integrating edge features into node embeddings, enhancing relational reasoning capabilities.

More recent methods adopt Transformer-based architectures to model long-range dependencies across both spatial and temporal domains. Transformer-based methods [67] employ attention mechanisms to learn pairwise relationships directly, offering greater flexibility in multi-agent reasoning. Examples include AgentFormer [68], which encodes joint trajectories with cross-agent attention, and TUTR [69], which introduces temporal uncertainty modeling. MultiModalTransformer [70] further extends this by combining visual and semantic features for scene-aware prediction.

In parallel, diffusion-based methods have emerged as powerful tools for modeling uncertainty. These models, such as MID [71] and LED [72], formulate trajectory prediction as a denoising process from a Gaussian noise prior. By progressively refining sampled trajectories, diffusion models achieve high diversity while maintaining physical plausibility, outperforming traditional GAN/CVAE baselines in recent benchmarks.

Despite these advances, two critical challenges remain in homogeneous pedestrian settings: (i) effectively capturing high-order, cross-time dependencies and implicit edge-to-edge influences without disrupting temporal consistency, and (ii) learning transferable behavioral structures that can generalize to unseen crowd scenarios. This thesis addresses these gaps through two complementary frameworks: UniEdge (Chapter 4), which unifies spatial–temporal reasoning via a dual-graph, edge-enhanced architecture to model high-order and edge-centric dependencies, and BP-SGCN (Chapter 5), which integrates unsupervised behavioral clustering with sparse graph convolution to capture transferable motion patterns without manual annotation, thereby enhancing cross-scenario adaptability.

For homogeneous pedestrian trajectory prediction, both traditional rule-based approaches and deep learning–based methods reviewed in this section are evaluated on

pedestrian-only benchmarks, including ETH/UCY [1, 73] and the pedestrian subset of the Stanford Drone Dataset [74]. These datasets cover a range of crowd scenarios with varying densities, motion patterns, and interaction complexities. To ensure fair comparison between different modeling paradigms, all methods are evaluated using the same displacement-based metrics. The evaluation protocol is detailed in Section 2.4, while dataset descriptions and experimental setups are provided in Chapter 4 and Chapter 5.

While substantial progress has been made in modeling pedestrian dynamics, real-world applications demand extending these approaches to more diverse traffic scenarios involving heterogeneous agents. This necessitates a deeper examination of models designed for pedestrian–vehicle or multi-class interactions, as discussed in Section 2.1.2.

2.1.2 Trajectory Prediction in Heterogeneous Environments

Traditional Rule-Based Approaches

In heterogeneous environments involving both pedestrians and vehicles, traditional rule-based methods have been extended to model the asymmetric and multi-agent nature of interactions. Rather than focusing solely on pedestrian-pedestrian dynamics, these methods aim to explicitly encode vehicle influence, often characterized by larger physical size, higher speed, and non-holonomic constraints. A number of studies extend the SFM to heterogeneous settings by introducing repulsive forces from vehicles, often shaped by anisotropic or speed-adaptive distance functions to reflect asymmetric danger zones [75, 76]. To better capture negotiation behaviors in shared spaces, game-theoretic layers have been added on top of SFM, treating pedestrian-vehicle interactions as sequential decision games [77, 78]. Other works employ heuristic utilities that balance safety and goal-seeking, with strategies guided by collision risk indicators such as time-to-collision or projected motion overlap [79].

Deep Learning-Based Approaches

Compared to homogeneous pedestrian settings, heterogeneous trajectory prediction introduces additional challenges due to the presence of agents with varying dynamics, such as vehicles, cyclists, and pedestrians. These agents differ in their speed profiles,

interaction behaviors, and scene constraints, requiring models to reason across types and capture asymmetric multi-agent interactions [19, 34, 80]. As discussed in the previous section, modeling pedestrian interactions provides valuable insights, but does not generalize well to real-world traffic scenes that exhibit diverse agent types and behaviors.

Early methods for predicting heterogeneous trajectories frequently employed spatial reasoning and fusion techniques to model agent dynamics. The Multi-Agent Tensor Fusion (MATF) framework [81], for example, represents spatial and contextual connections using a convolutional fusion module, which processed a tensor to maintain spatial alignment between agents and scene elements. JPKT [82] considers vehicles as rigid particles, applying kinematics to non-particle entities, and separately models vehicles and pedestrians using distinct long short-term memory (LSTM) [83] layers. Proposal-based approaches such as CoverNet [84] generates predefined multimodal trajectory anchors from observations of both vehicles and pedestrians. DATF [85] models agent-to-agent and agent-to-scene interactions through the attention mechanism and proposes a new approach to estimate the trajectory distribution. Furthermore, models based on the Social Force [47] paradigm have advanced the field of heterogeneous trajectory prediction. By using physically-inspired forces, they explicitly model the complex interactions and collision-avoidance behaviors between different classes of agents, such as vehicles and pedestrians [42, 86]. To further refine trajectory realism, methods like the Knowledge Correction framework [87] fuses domain knowledge with deep networks, balancing prediction accuracy with adherence to traffic semantics. Meanwhile, diffusion-based approaches such as ParkDiffusion [80] apply stochastic modeling to forecast multimodal outcomes within structurally constrained parking environments for heterogeneous agents.

Graph representation possesses powerful capabilities for relational reasoning and representation. Recent studies leverage GNNs to capture intricate spatial and semantic dependencies among diverse traffic agents, as well as their interactions with lanes and the surrounding environment. For example, Grimm *et al.* [88] proposed a heterogeneous graph structure that integrates both road-bound and non-road-bound agents via semantic anchor paths, enabling more valid and multimodal trajectory predictions. The UNIN framework [34] constructs a large-scale, category-aware interaction graph with hierarchical attention mechanisms to model cross-category interactions within

an unbounded neighborhood. Building on this direction, HRG+HSG [35] introduces a risk-aware graph design that incorporates safety constraints to facilitate interpretable and risk-sensitive forecasting. Other works such as SSS [89] and MVHGN [90] further address agent heterogeneity by employing adaptive graph structures—using selective state spaces or multi-view hierarchical message passing—to jointly capture semantic, spatial, and type-aware relationships. To enhance the expressive power of graph-based models, a number of approaches [6, 33, 89, 91] explicitly incorporate class labels into interaction graphs, thereby allowing the model to distinguish agent types and tailor message passing accordingly.

Despite substantial progress in heterogeneous trajectory prediction, two key challenges remain: (i) scalability in dense traffic scenes, where complex interaction modeling can incur prohibitive computational costs, and (ii) reliance on costly, manually annotated class labels to distinguish agent types, which is often impractical in real-world deployments. This thesis addresses these gaps through two complementary frameworks: Multiclass-SGCN (Chapter 3), which incorporates agent-type semantics and motion cues into a sparse interaction architecture to efficiently capture asymmetric, cross-type dependencies, and BP-SGCN (Chapter 5), which replaces manual class labels with unsupervised behavioral pseudo-labels, enabling scalable and label-free modeling of heterogeneous interactions while maintaining high predictive accuracy.

For heterogeneous trajectory prediction, models are typically evaluated on multi-agent traffic datasets such as Argoverse [92] and heterogeneous subsets of Stanford Drone Dataset [74], which include multiple agent types and complex interaction dynamics. In this thesis, both traditional and deep learning–based heterogeneous approaches are evaluated under consistent evaluation criteria to enable fair comparison. The evaluation framework is detailed in Section 2.4, and comprehensive descriptions of the datasets and experimental settings are provided in Chapter 3 and Chapter 5.

2.1.3 Literature Surveys on Multi-Agent Trajectory Prediction

Several survey and review papers have systematically summarized multi-agent trajectory prediction from complementary perspectives. Existing surveys provide structured overviews of modeling paradigms, ranging from classical physics-based approaches to

deep learning and hybrid methods, and analyze how interactions, uncertainty, and multimodality are handled across different model families [46, 93]. These works organize the literature according to architectural choices, interaction representations, learning objectives, and commonly used datasets and evaluation protocols.

From the perspective of autonomous driving and mixed traffic scenarios, other surveys focus specifically on pedestrian–vehicle interactions in heterogeneous environments [94, 95]. They examine interaction modeling strategies for unstructured or shared spaces, review datasets involving multiple agent types, and discuss challenges related to safety, scalability, and real-world deployment.

Building on these surveys, this thesis does not replicate their detailed taxonomies. Instead, it focuses on underexplored aspects of graph-based trajectory prediction, with particular emphasis on interaction-aware graph representation design, including sparse graph construction for efficient interaction modeling, semantic abstraction derived without manual labels, and edge-centric relational representations that explicitly model inter-agent relationships.

2.2 Spatial-Temporal Graph Representation Learning

Understanding and forecasting the motion of multiple agents in dynamic environments requires capturing both their spatial interactions and temporal evolution. In recent years, graph-based methods have emerged as a powerful paradigm for trajectory prediction, offering a natural way to represent agent interactions through nodes and edges. This section reviews spatial-temporal graph representation learning approaches that model complex multi-agent behaviors. We begin by introducing graph-based interaction modeling strategies in Section 2.2.1, which focus on how to construct and encode the relationships between agents using various forms of graphs. Following this, we discuss spatial-temporal fusion mechanisms in Section 2.2.2, which describe how spatial and temporal information is integrated within graph-based architectures to allow trajectory prediction.

2.2.1 Graph-Based Spatial Interaction Modeling

A primary challenge in trajectory prediction is to effectively model complex spatial interactions among traffic agents. While early deep learning methods propose to use spatial pooling mechanisms [21, 28] and grid-based mechanisms [59] to aggregate neighborhood information by summarizing nearby agents' hidden states within predefined spatial regions or grids, these approaches often assume a fixed interaction range and struggle to capture more complex, long-range dependencies.

To overcome these limitations, graph architectures have become a dominant paradigm in the field, as they offer a natural and flexible framework to represent agents as nodes and their relationships. This allows for explicit modeling of the interaction topology, which is crucial for understanding social behaviors. This subsection presents an overview of graph-based spatial interaction modeling techniques, focusing on how inter-agent relationships are represented and utilized for learning spatial dependencies.

Graph Representation for Homogeneous Pedestrians

In the trajectory prediction field, graph-based spatial interaction modeling has evolved beyond simple proximity-based topologies, giving rise to diverse designs that capture complex agent-agent relations. A widely used and foundational approach is the distance-based graph representation [29, 96], in which edges are constructed based on fixed spatial thresholds or K-nearest neighbors (KNN) computed at each time step. These graphs offer clear geometric interpretability and computational efficiency. However, they typically treat all connected neighbors equally, with fixed edge weights that fail to reflect the varying importance of different interactions. To address this, attention-based graph representations [38, 97–100] have been introduced to assign learnable context-dependent weights to edges based on the attention mechanism [67, 101]. These methods enable the model to dynamically evaluate the relative importance of neighboring agents, allowing it to capture asymmetric interactions that are crucial for accurate and interpretable spatial interaction representations.

Beyond specific architectural choices, sparsity has been increasingly recognized as a fundamental inductive bias in graph-based interaction modeling. In real-world pedestrian scenes, interactions are inherently local and asymmetric, and not all nearby agents exert

meaningful influence at every time step. Constructing densely connected graphs therefore introduces redundant or noisy interactions, which can dilute salient relational signals and hinder effective message passing. From a computational perspective, fully connected interaction graphs incur quadratic complexity with respect to the number of agents, limiting scalability in crowded scenes. More importantly, recent studies have shown that selectively sparsifying interaction graphs based on geometric constraints, motion consistency, or learned relevance can improve both predictive accuracy and robustness by mitigating over-smoothing and over-squashing effects during graph propagation [102]. Empirical evidence across multiple benchmarks suggests that sparse interaction graphs can match or outperform dense counterparts while significantly reducing computational cost, particularly in dynamic or cluttered environments [27, 33, 103].

Building on this principle, methods such as SGCN [27] and SDAGCN [103] reduce superfluous or irrelevant connections by constructing sparse, often directed, graphs. These models use criteria such as field-of-view constraints, relative motion direction, or learned attention scores to prune the graph, resulting in more efficient and interpretable interaction modeling. By focusing only on the most salient agent relationships, these sparse graphs not only reduce computational overhead but also improve the overall prediction performance.

In recent years, several studies have also proposed novel graph construction paradigms to better reflect social structures and behavioral dynamics among agents. Group-based graphs [63, 104, 105] segment agents into latent groups or clusters based on motion coherence, social affinity, or spatial proximity, and then model inter-group and intra-group interactions as separate subgraphs. This hierarchical formulation enables more structured and scalable representation of multi-agent interactions, reduces noise from irrelevant or weakly correlated agents, and captures collective behaviors. Besides, HighGraph [31] proposes a high-order graph convolution operator that goes beyond conventional pairwise message passing by aggregating information from higher-order node combinations, such as triplets or cliques. This design allows the model to capture indirect interactions and complex multi-agent dependencies that are difficult to represent using standard edge-based graph convolutions. Together, these diverse graph formulations reflect a growing recognition that spatial interactions in pedestrian dynamics are structured, context-

dependent, and often asymmetric—properties that naïve or uniform graph designs cannot fully capture.

In Chapter 4, we further extend the concept of high-order graphs by introducing a unified graph structure that transforms complex high-order cross-time interactions into simplified first-order relationships, enabling efficient and expressive modeling of long-range dependencies across both space and time while preserving the underlying relational dynamics. Furthermore, Chapter 5 complements this in homogeneous pedestrian settings where agent semantics are uniform by constructing sparse, semantically informed interaction graphs guided by unsupervised behavioral pseudo-labels, thereby enabling the learned graph topology to adapt across varying crowd scenarios without manual annotation.

Graph Representation for Heterogeneous Environments

Recent advances in trajectory prediction for dense and mixed traffic environments have increasingly leveraged graph neural networks to model the complex and dynamic interactions among heterogeneous agents such as vehicles, pedestrians, and cyclists. In such settings, the diversity of agent kinematics, motion constraints, and interaction semantics introduces additional modeling challenges compared to homogeneous scenarios. To address these complexities, a growing body of research has explored constructing heterogeneous graphs in which nodes explicitly represent different agent types and edges encode their interaction patterns. These edges can incorporate various relational cues, including spatial proximity, relative velocity, and semantic role, enabling the model to reason about both intra-class behaviors, such as coordinated pedestrian movement, and inter-class interactions, such as pedestrian–vehicle negotiation in shared spaces. By leveraging this structured representation, heterogeneous graph frameworks aim to capture asymmetric influences between agents, account for type-specific motion dynamics, and improve prediction robustness in diverse, real-world traffic scenes.

For example, HTFNet [106] and VNAGT [91] employ a heterogeneous graph network combined with a transformer-based attention mechanism that uses relation-dependent parameters to distinguish the influence between different types of agents. HEAT [65] and NLNI [34] introduce a type-specific heterogeneous graph attention encoder net-

work for capturing both intra- and inter-class interactions, enabling simultaneous and accurate trajectory prediction for multiple agent types in complex traffic scenarios. To enhance the computation efficiency of graph message passing, SMGCN [33] proposes a sparse graph architecture to capture important heterogeneous interactions. MVHGN [90] further enhances prediction by combining multi-view logical correlations and adaptive spatial topology networks, allowing the model to mine both micro-level and macro-level logical-physical features of heterogeneous traffic agents. Additionally, models like TraG-CAN [107] and HDGT [108] extend the heterogeneous graph paradigm by integrating spatial attention mechanisms and scene encoding, respectively, to better capture the diverse semantic relationships and context-dependent interactions among agents of different types. These innovations demonstrate that heterogeneous graph neural networks, equipped with specialized encoders, attention mechanisms, and context integration, are highly effective for modeling the nuanced behaviors and interactions of diverse agents, leading to significant improvements in trajectory prediction accuracy and robustness in real-world traffic environments. While these approaches have significantly improved prediction accuracy, challenges remain in designing graph structures that are both computationally efficient and semantically expressive, which motivates the methods proposed in this thesis.

In heterogeneous trajectory prediction, constructing a graph representation that is both computationally efficient and semantically expressive enough to capture diverse inter- and intra-class behaviors remains a significant challenge. In Chapter 3, we address this by integrating explicit agent-class semantics with an adaptive sparse graph architecture, enabling efficient modeling of asymmetric dynamics between different agent types. In Chapter 5, we present a complementary bottom-up framework that leverages unsupervised deep clustering to derive behavioral pseudo-labels directly from motion data, uncovering nuanced motion patterns without manual annotation. Together, these approaches advance semantic and structural graph representation learning, achieving state-of-the-art performance in heterogeneous trajectory prediction.

2.2.2 Spatial-Temporal Fusion for Trajectory Prediction

While graph-based models are effective at capturing spatial interactions at individual time steps, trajectory prediction is inherently a sequential task that requires modeling how these interactions evolve over time. Consequently, a key component of modern trajectory prediction frameworks is the mechanism for fusing spatial and temporal information. A widely adopted strategy in the literature is the decoupled spatial–temporal fusion paradigm, wherein spatial features are first extracted independently at each time step, and a dedicated temporal modeling module subsequently processes the resulting sequence of spatial embeddings to learn the underlying sequential dynamics [109].

The choice of the temporal modeling module has evolved. Early works, and many strong baselines to this day, adopt Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) [83] and Gated Recurrent Units (GRU) [110], to process the temporally ordered spatial graph features [111–115]. RNNs are naturally suited for capturing temporal dependencies but suffer from limited ability to model long-term dependencies due to gradient vanishing and their inherently sequential nature, which restricts parallelization during training. To alleviate these issues, Temporal Convolutional Networks (TCNs) [116] have been increasingly adopted. TCNs use 1D convolutions over time to capture fixed-size receptive fields while allowing for parallel processing. Unlike RNNs, they can model long-range dependencies via dilation and deeper layers. Methods such as Social-STGCNN [29] and SGCN [27] are representative of this line of work, combining spatial GCNs with TCN backbones for more scalable and temporally expressive modeling. More recently, Transformer-based architectures have emerged as a powerful alternative for temporal modeling in trajectory prediction. Transformers leverage global self-attention to model dependencies across all time steps simultaneously and have shown strong performance in capturing complex temporal patterns, especially in multimodal or highly dynamic environments [68, 69]. They overcome key limitations of RNNs (sequential computation) and TCNs (limited context windows) by attending to the entire trajectory history in a data-driven and content-aware manner.

Despite the widespread adoption and success of this decoupled spatial-temporal paradigm, it has a fundamental limitation: the separation of spatial and temporal processing can disrupt the natural inter-dependencies within spatial-temporal representa-

Encoding information	**Node-level encoding**	**Edge-level encoding**	**Representative works**
Trajectory geometry	Absolute positions or displacements	Relative distance / relative velocity relations	Social-LSTM [28]; STGCNN [29]; STGAT [38]; Trajectron++ [61]; AgentFormer [68]
Agent semantics / type (supervised)	Explicit class or role embeddings	Type-aware or asymmetric relations	UNIN [34]; HSG [35]; **Multiclass-SGCN (Chapter 3)**
Agent semantics / type (label-free)	Behavioral or motion-based pseudo-labels	Implicit type-dependent interactions	**BP-SGCN (Chapter 5)**
Relational enrichment (node-centric)	Implicit edge-to-node aggregation	Edge-enhanced but node-updated relations	GC-VRNN [64]; HEAT [65]; MFAN [66]
Relational enrichment (edge-centric)	Node representations conditioned on edge-centric modeling	Explicit edge representations with independent states	**UniEdge (Chapter 4)**

Table 2.1: Design space of node- and edge-level encodings in graph-based trajectory prediction. The table summarizes how geometric and semantic information is distributed across nodes and edges in existing methods, and highlights distinct relational modeling paradigms explored in this thesis.

tions [117,118]. By first encoding interactions frame-by-frame and then learning temporal dependencies, these models often fail to capture high-order cross-time interactions—for example, how an agent's position at time $t-2$ directly influences a neighbor's behavior at time t This multi-step aggregation process can lead to information dilution and a phenomenon known as "under-reaching" [119] where important long-range spatial-temporal cues are weakened or lost before they can inform the final prediction. This limitation hinders the model's ability to reason about complex, evolving social dynamics, particularly in scenarios that require immediate and nuanced responses to environmental changes.

2.2.3 Representation Design for Graph Construction

Table 2.1 summarizes the representation design space of graph-based trajectory prediction methods from the perspective of node- and edge-level encodings. Specifically, it categorizes how geometric information, agent semantics, and relational structures have been encoded either at nodes or edges in existing literature, together with representative works adopting each design choice.

As shown in the table, most existing approaches primarily encode trajectory geometry and agent semantics at the node level, while edge representations are commonly limited to relative geometric relations or attention-based weights. Supervised semantic information is typically incorporated via explicit class or role embeddings, whereas label-free semantic abstractions are far less explored, particularly in heterogeneous trajectory prediction where such labels are costly or unavailable. This highlights a semantic representation gap, where existing methods rely heavily on manually annotated agent types, limiting their scalability and practical applicability. Importantly, Table 2.1 also reveals a clear gap in relational modeling: although several methods incorporate edge features to assist node updates, these approaches remain fundamentally node-centric, with edges lacking independent representations or temporal dynamics. Explicit edge-centric modeling, where relations are treated as first-class entities with their own states, remains largely underexplored prior to this thesis.

2.3 Unsupervised Behavior Clustering

The clustering of temporal trajectory patterns allows modeling the behavioral groups for better trajectory prediction [111, 120]. Early works focus on the raw trajectory represented as 2D coordinates. Support vector clustering is introduced as a closed-loop method on motion vectors for motion behavior representations [121]. K-means on trajectory vectors or sequence key points obtain cluster centers to enhance trajectory prediction [111, 122]. DBSCAN is proposed to avoid manually specifying cluster numbers, adding more flexibility and interpretability to behavior patterns [120]. GP-Graph directly uses the absolute distance among pedestrians to determine the division of group [25]. The recent PCCSNet leverages BiLSTM network to encode coordinates prior to K-means

clustering, identifying behavioral modalities [123]. In addition to modalities, FEND further applies 1D CNN and LSTM for trajectory encoding and employs the K-means for long-tail trajectory clustering to distinguish trajectory patterns [124].

However, most existing methods rely on shallow trajectory representations, limiting their ability to capture nuanced, evolving behaviors. Additionally, distance-based clustering approaches often struggle with complex motion patterns. To address these issues, we propose a cascaded optimization scheme featuring an end-to-end Deep Embedded Clustering (DEC) [125] module, which iteratively refines cluster assignments using a KL-divergence objective. This dynamic adaptation yields richer latent representations, enabling a more data-driven and expressive approach to modeling agent behaviors.

2.4 Evaluation and Metric

To quantitatively assess the performance of the trajectory prediction models presented in this thesis, we employ two of the most widely adopted metrics in the field: the Average Displacement Error (ADE) and the Final Displacement Error (FDE). These metrics evaluate the pixel or real-world coordinate distance between the predicted path and the ground-truth path.

2.4.1 Average Displacement Error (ADE)

The Average Displacement Error measures the average L2 distance between the predicted trajectory points and the ground-truth points over the entire prediction horizon. It provides a comprehensive assessment of the overall prediction accuracy across all future time steps. The ADE is calculated as:

$$\text{ADE} = \frac{1}{N \times T_{pred}} \sum_{i=1}^{N} \sum_{t=1}^{T_{pred}} \|\hat{p}_t^i - p_t^i\|_2 \tag{2.1}$$

where N is the total number of agents in the scene, T_{pred} is the length of the prediction horizon, $\hat{p}_t^i = (\hat{x}_t^i, \hat{y}_t^i)$ represents the predicted 2D coordinates for agent i at future time step t, and $p_t^i = (x_t^i, y_t^i)$ are the corresponding ground-truth coordinates. The operator $\|\cdot\|_2$ denotes the Euclidean (L2) norm.

2.4.2 Final Displacement Error (FDE)

The Final Displacement Error specifically evaluates the accuracy at the end of the prediction horizon. It is defined as the L2 distance between the predicted final destination and the ground-truth final destination at time step T_{pred}. This metric is particularly important for assessing a model's ability to forecast long-term intentions and final goals. The FDE is calculated as:

$$\text{FDE} = \frac{1}{N}\sum_{i=1}^{N} \|\hat{p}^i_{T_{pred}} - p^i_{T_{pred}}\|_2 \tag{2.2}$$

where the variables are defined identically to those in the ADE calculation.

2.4.3 Evaluation of Multimodal Predictions

Since pedestrians' future movements are inherently multimodal, modern trajectory prediction frameworks, including those developed in this thesis, typically generate multiple plausible future trajectories to capture this uncertainty [21, 27, 28]. In line with standard evaluation practice, our models generate K (e.g., $K = 20$) trajectory samples for each agent. To enable fair comparison with other state-of-the-art generative models, evaluation metrics are computed on the single trajectory sample that achieves the minimum displacement error relative to the ground truth. While this best-of-K evaluation strategy is widely adopted in the literature, alternative evaluation choices or dataset-specific conventions may exist across different benchmarks; unless otherwise stated, this thesis adheres to the standard evaluation protocols associated with each benchmark.

CHAPTER 3

Semantic-Aware Sparse Graph Modeling for Heterogeneous Trajectory Prediction

Portions of this chapter have previously been published in the following peer-reviewed publication [32]:

- **Li, R.**, Katsigiannis, S., & Shum, H. P. H., "Multiclass-SGCN: Sparse Graph-based Trajectory Prediction with Agent Class Embedding." In *Proceedings of the IEEE International Conference on Image Processing* (ICIP), 2022.

Trajectory prediction of road users in real-world scenarios is challenging because their movement patterns are stochastic and complex. Previous pedestrian-oriented works have been successful in modelling the complex interactions among pedestrians, but fail in predicting trajectories when other types of road users are involved (e.g., cars, cyclists, etc.), because they ignore user types. Although a few recent works construct densely connected graphs with user label information, they suffer from superfluous spatial interactions and temporal dependencies. To address these issues, we propose Multiclass-SGCN, a sparse graph convolution network based approach for multi-class trajectory prediction

that takes into consideration velocity and agent label information and uses a novel interaction mask to adaptively decide the spatial and temporal connections of agents based on their interaction scores. The proposed approach significantly outperformed state-of-the-art approaches on the Stanford Drone Dataset, providing more realistic and plausible trajectory predictions.

3.1 Introduction

Trajectory prediction has drawn considerable attention with the development of autonomous vehicles in recent years. Specifically, models take the observed trajectories of different agents in real-world scenes to predict their future movement patterns, benefiting self-driving cars for collision avoidance [9], as well as anomalous movement flow detection [126]. To tackle the challenge of modeling the complex and stochastic nature of social interaction patterns, methods focusing on spatial interaction modeling and temporal dependency capturing are proposed. Social-LSTM [28] uses pooling windows for interaction modeling and recurrent architecture for temporal capturing, whereas Social-STGCNN [29] uses relative distance to measure interactions between agents and temporal convolution networks (TCN) [116] to handle temporal dependencies. STAR [127] and TF [128] propose transformer-based [67] architectures for both spatial and temporal aspects, achieving impressive performance. As densely connected graphs may generate superfluous interactions, leading to impractical computational costs, Sparse Graph Convolution Network (SGCN) [27] proposes a self-attention based sparse graph architecture to mitigate these problems.

The main challenge of trajectory prediction is to consider the different movement behaviors of different classes of agents. The aforementioned research only focuses on pedestrians and does not consider other classes of agents, such as cars and cyclists, which have a significant effect on trajectory prediction. Intuitively speaking, even if two agents have a similar velocity, human instincts would force us to pay more attention to the movements of the larger agents, such as considering car over bicycle. To address this issue, Semantics-STGCNN [3, 129] considered class labels for multi-class trajectory prediction by embedding agent-label features into the velocity representations [130], ensuring that

the upcoming GCN [131] aggregates both features. Nevertheless, Semantics-STGCNN still suffers from the superfluous interactions problem as it uses a densely connected graph. It also lacks a separate modeling of temporal dependencies, thus suffering from long-term predictions.

In this paper, we propose Multiclass Sparse Graph Convolution Network (*Multiclass-SGCN*), an attention-based sparse GCN for multi-class trajectory prediction that models interactions and temporal dependencies among multi-class agents in real scenes. We introduce a novel method to embed the correlated agent label and velocity features to build the velocity-label graph (VLG) representation, with particular care to learn the optimal embedding for each feature separately. In the sparse graph learning module, we designed a novel adaptive interaction mask to spatially and temporally evaluate attention patterns and generate plausible sparse adjacency matrices, enabling each agent to focus only on explicit neighbours and important time steps. Finally, GCN [131] and TCN [116] layers are employed for the final trajectory prediction.

Performance was evaluated on the Stanford Drone Dataset (SDD) [74] against state-of-the-art approaches, showing that our proposed model outperforms all existing methods for all the examined evaluation metrics by a significant margin.

The contributions of this work are: **(1)** We present *Multiclass-SGCN*, a GCN for predicting multi-class agent trajectories, which outperforms state-of-the-art methods. **(2)** To effectively model the different patterns of multi-class agent trajectories, we propose a novel algorithm to separately embed the correlated features of class label and velocity, resulting in an optimal embedding for different natures of input features. **(3)** To create sparse attention of neighbors from different classes, we propose an adaptive interaction mask that adaptively filters neighbors of lower influence.

3.2 Multiclass-SGCN

Given a series of T video frames with N agents, the corresponding 2-D trajectory coordinates (x_t^i, y_t^i), velocity $\mathcal{V}_t^i = (x_t^i - x_{t-1}^i, y_t^i - y_{t-1}^i)$, and one-hot encoded semantic labels $\mathcal{L}_t^i$, $\forall\, t \in [1, T]$ and $\forall\, i \in [1, N]$, the goal of multi-class trajectory prediction is to predict the future trajectory coordinates of each agent $(x_t^i, y_t^i)\ \forall\, t \in [T+1, T']$. An overview of the

proposed Multiclass-SGCN for trajectory prediction is provided in Figure 3.1. We employ SGCN [27] as our backbone as it introduces a self-attention mechanism to enhance the spatial and temporal sparsity of the neighbour graph. The two key components of our network are the velocity label graph embedding that separately embeds the velocity and class labels for an optimal representation, and the enhanced sparse graph learning that adaptively determines the neighbour graph for each agent based on its attention preferences.

3.2.1 Velocity-Label Graph (VLG) Embedding

We observe that the two important factors that affect the movement of an agent are the classes and velocity of neighbours. Class labels, $\mathcal{L}_t^i$, can indicate how different classes of agents, such as pedestrian, car, cyclist, have different influences [3]. Velocity, $\mathcal{V}_t^i$, enhances the ability of a model to capture the geometric features of agents [29]. As velocity and classes are highly correlated, such as a car would have a higher speed, it would be advantageous to model them together. At the same time, as they are two different features, it would be better to embed them separately.

To encode the spatial and temporal features, we construct a spatial VLG (SVLG) and a temporal VLG (TVLG). SVLG contains the features of all the agents at time step t, with $G_{svlg} = (X_t, A_t)$, $X_t = \{\mathbf{x}_t^i \mid i = 1, ..., N\}$, while TVLG contains the features of each individual agent over all time steps, such that $G_{tvlg} = (X^i, A^i)$, $X^i = \{\mathbf{x}_t^i \mid t = 1, ..., T\}$. X is the concatenation of $\mathcal{V}_t^i$ and $\mathcal{L}_t^i$, and A_t and A^i are adjacency matrices that represent the edges of the SVLG and TVLG respectively, indicating whether the nodes are connected (denoted as 1) or not (denoted as 0). Following [27], A^i is initialised as 1 and A_t as an upper triangular matrix filled with 1.

We propose a *velocity-label graph (VLG) embedding* that combines the advantages of velocity and class label, while learning an optimal embedding for each of them. The graph embedding of VLG is computed by combining the embeddings of velocities and

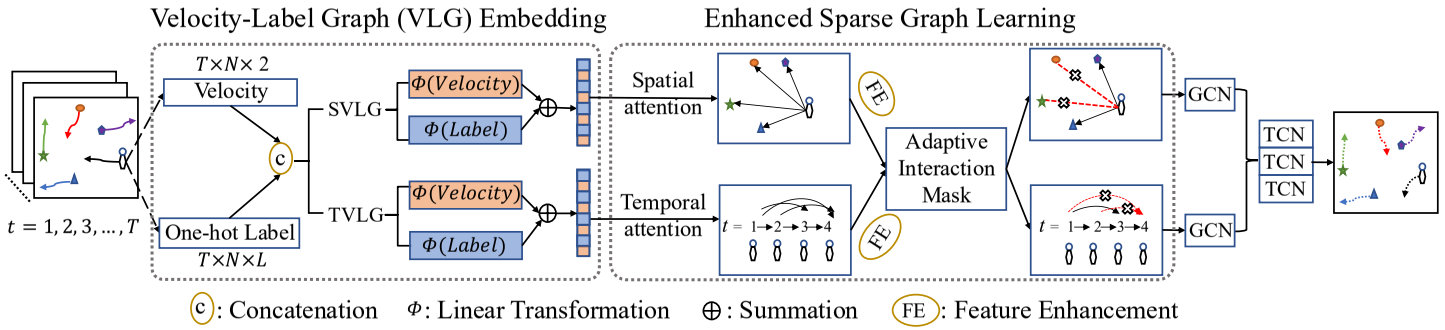


Figure 3.1: The network structure of Multiclass-SGCN. Given a sequence of T frames including N agents, we extract the velocity and label features to build spatial and temporal velocity-label graph (SVLG and TVLG). The embedded VLG features are passed into enhanced sparse graph learning with the proposed adaptive interaction mask to construct meaningful sparse attention adjacency matrices. Graph convolution networks (GCN) and TCN are employed to aggregate and make predictions.

one-hot encoded class labels of agents:

$$\begin{aligned} E_{vlg} &= E^{\mathcal{V}}_{vlg} + E^{\mathcal{L}}_{vlg} \\ E^{\mathcal{V}}_{vlg} &= \phi(G^{\mathcal{V}}_{vlg}, W_{E^{\mathcal{V}}_{vlg}}) \\ E^{\mathcal{L}}_{vlg} &= \phi(G^{\mathcal{L}}_{vlg}, W_{E^{\mathcal{L}}_{vlg}}) \end{aligned} \tag{3.1}$$

where $G^{\mathcal{V}}_{vlg}$ and $G^{\mathcal{L}}_{vlg}$ are subgraphs of VLG corresponding to the velocity and label features respectively, $\phi(\cdot,\cdot)$ a linear transformation, $W_{E^{\mathcal{V}}_{vlg}} \in \mathbb{R}^{2\times D_{E_{vlg}}}$ and $W_{E^{\mathcal{L}}_{vlg}} \in \mathbb{R}^{L\times D_{E_{vlg}}}$ the weights of the linear transformation, L the length of encoded one-hot labels, and $D_{E_{vlg}}$ the embedding size.

3.2.2 Enhanced Sparse Graph Learning

We enhance the sparse graph learning module of SGCN [27] to better model the multi-class nature of the problem. This module is constructed from the numerical interaction scores calculated by the self-attention module. It then extracts high-level spatial-temporal interaction features and uses an interaction mask with a fixed threshold of 0.5 to optimise the sparsity of graph representations by pruning weak connections with lower attention relevance. We argue that the interaction mask threshold should be adaptively adjusted through the learning process of each individual agent.

Given the embedded SVLG and TVLG, E_{svlg} and E_{tvlg}, a self-attention module [67] is implemented to calculate the attention scores $\mathcal{A}$ between each node pairs:

$$\begin{aligned} Q_{vlg} = \phi(E_{vlg}, W^{vlg}_{Q}), \quad K_{vlg} = \phi(E_{vlg}, W^{vlg}_{K}) \\ \mathcal{A}_{vlg} = \text{Softmax}(\frac{Q_{vlg} \times K^{T}_{vlg}}{\sqrt{d_{vlg}}}) \end{aligned} \tag{3.2}$$

where $\phi(\cdot,\cdot)$ denotes a linear transformation, W^{vlg}_{Q} and W^{vlg}_{K} are learnable weight matrices, $\sqrt{d_{vlg}}$ is the scaled factor for numerical stability. The output spatial and temporal attention matrices, $\mathcal{A}_{svlg}$ and $\mathcal{A}_{tvlg}$, are of size $T \times N \times N$ and $N \times T \times T$, respectively. Following [27], we implement a feature enhancement module using a series of asymmetric convolution layers [132] to extract high-level interaction features, and using one-by-one convolutions on the spatial attention scores to capture the temporal dependencies, thus

creating the high-level interaction attention features F_{svlg} and F_{tvlg}.

To sparsify the high-level interaction attention matrix, we propose an *adaptive interaction mask (AIM)* to extract the set of neighbors in SVLG and TVLG. Manually-set fixed interaction thresholds, as used by SGCN [27], cannot fully describe the patterns of spatial interactions and temporal dependencies of each agent. We propose an average operator to adaptively calculate a threshold and remove the influence of less important neighbors, allowing the system to adapt according to the interactions of various types of agents, thus being more suitable for more complex scenes compared to the global threshold approach of SGCN [27]. In particular, the (i, j)-th element of the adaptive sparse interaction mask M_{vlg} is computed as:

$$M_{vlg}[i,j] = \begin{cases} 1, \ \sigma(F_{vlg}[i,j]) > \frac{\sum_{j=1}^{N} \sigma(F_{vlg}[i,j])}{N} \\ 0, \ \text{otherwise} \end{cases} \tag{3.3}$$

where σ indicates the Sigmoid function. Using the adaptive interaction mask, we construct a sparse adjacency matrix for graph convolution, and because of the removal of superfluous connections, the sparse graph enables the GCN model to learn from influential neighbors, thus improving both training speed and prediction accuracy.

Similarly to [27], we apply two separate branches of the GCN [131] to fuse the sparse spatial VLG and sparse temporal VLG. The two GCN branches differ in the order of their input, as the first is fed the spatial VLG before the temporal VLG, whereas the second is fed in the reverse order. Then, the last outputs of these two GCN branches are summed to provide the final trajectory representation H. Finally, temporal convolution networks (TCN) [116] are used on the temporal dimension, assuming that the coordinates (x_t^i, y_t^i) of agent i at frame t follow the bi-variate Gaussian distribution as $N(\mu_t^i, \sigma_t^i, \rho_t^i)$, a cascade of TCN layers can be used to predict parameters in the bi-variate Gaussian distribution. To train the proposed network, we minimize the negative log-likelihood loss function to estimate the trained parameters following [29].

3.3 Experimental Results

The proposed model was trained and validated on the Stanford Drone Dataset (SDD) [74]. SDD has class labels for six different types of agents, including *pedestrian*, *cyclist*, *cart*, *car*, *skater*, and *bus*. Data is captured from bird's-eye view by flying a drone over Stanford University's campus. We follow existing works [27], [20] that apply 8 observed frames (3.2 seconds) to predict the next 12 frames (4.8 seconds), then 20 samples are derived from the learnt multivariate distribution. The model was evaluated in terms of the Minimum Average Displacement Error (mADE) and the Minimum Final Displacement Error (mFDE) as in [29], as well as in terms of the Average ADE (aADE) and the Average FDE (aFDE) proposed by [3] who argued that aADE and aFDE evaluate the models more holistically. The Adam [133] optimizer was used for training, with a 0.0001 learning rate and a batch size of 256. To compare with Semantics-STGCNN [3], we also normalized and denormalized the input trajectory data with a scaling factor of 10. Training typically converged in around 35-45 epochs.

3.3.1 Quantitative Results

The proposed method was compared to 8 models in total, including the baseline Linear model, energy function based behavioral model (SF [134]), Social-LSTM [28], Social-GAN [20], CAR-Net [135], DESIRE [40], Social-STGCNN [29] and Semantics-STGCNN [3], the existing state-of-the-art model for multi-class trajectory prediction. Notably, the results of Semantics-STGCNN were evaluated using the published source code, whereas other results were provided by [3]. Results are presented in Table 3.1 in terms of mADE and mFDE. It is evident that the proposed model outperformed all other models, including the latest Semantics-STGCNN [3] with a 3.76 decrease in mADE and 3.71 decrease in mFDE, indicating the importance of considering label information and velocity in complex trajectory prediction tasks, as well as of using an adaptive interaction mask. Furthermore, as discussed in [3], common minimum-based metrics (mADE and mFDE) focus only on the best sampled sample, which is not comprehensive in real-world scenarios, while average-based metrics (aADE and aFDE) can be more plausible and high level. To this end, we compared the proposed Multiclass-SGCN with Semantic-STGCNN using aADE

Table 3.1: Performance comparison with the state-of-the-arts.

Model	mADE	mFDE
Linear	37.11	63.51
SF [134]	36.48	58.14
Social-LSTM [28]	31.19	56.97
Social-GAN [20]	27.25	41.44
CAR-Net [135]	25.72	51.80
DESIRE [40]	19.25	34.05
Social-STGCNN [29]	26.46	42.71
Semantics-STGCNN [3]	18.12	29.70
Multiclass-SGCN (ours)	**14.36**	**25.99**

Table 3.2: Performance comparison with Semantics-STGCNN.

Model	mADE	mFDE	aADE	aFDE
Semantics-STGCNN [3]	18.12	29.70	33.14	61.14
Multiclass-SGCN (ours)	**14.36**	**25.99**	**22.87**	**45.30**

and aADE (Table 3.2), demonstrating a significant improvement of more than minus 10 for both metrics.

To further validate the contribution of class labels (CL), separate embedding (SE) of the VLG, and adaptive interaction mask (AIM), we conducted three ablation experiments by evaluating three variants of the proposed method: i) Mutliclass-SGCN w/o SE denotes that the embedding of the input graph was computed from the whole feature matrix, instead of separately for velocity and labels (Section 3.2.1); ii) To evaluate the effectiveness of our sparsification design. Mutliclass-SGCN w/o AIM denotes that a manually set interaction threshold ($\xi = 0.5$) was used for all agents to measure the existence of their neighbors, as in SGCN [27], instead of our proposed adaptive interaction mask (Section 3.2.2); iii) Mutliclass-SGCN w/o CL denotes that the embedding of the input graph was computed only for velocity, instead of both velocity and class labels. Results in Table 3.3 show that the proposed use of class labels and of the SE and AIM modules is important for boosting the performance of the model, especially AIM, which led to a 43.3% reduction in aADE and a 41.2% reduction in aFDE, indicating the importance of adaptively modeling the interaction patterns of each agent, because agents of different classes may have different

Table 3.3: Ablation study results.

Model	mADE	mFDE	aADE	aFDE
Multiclass-SGCN w/o SE	14.77	**25.44**	24.74	48.42
Multiclass-SGCN w/o CL	15.32	26.39	26.29	50.30
Multiclass-SGCN w/o AIM	22.05	29.53	40.33	76.99
Multiclass-SGCN (ours)	**14.36**	25.99	**22.87**	**45.30**

attention preferences.

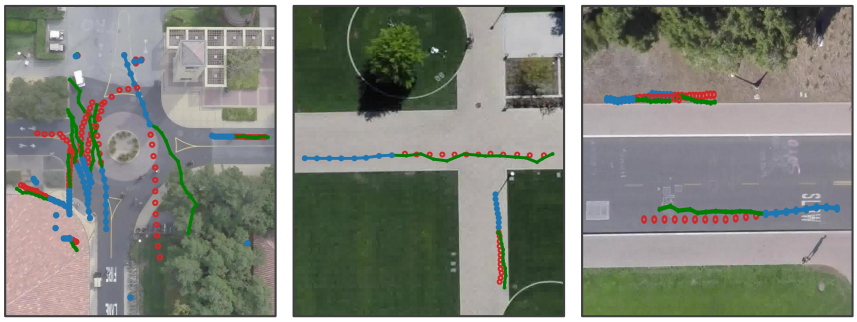
3.3.2 Qualitative Results

Predicted trajectories by the proposed Multiclass-SGCN and Semantics-STGCNN [3] for one frame from three scenarios are shown in Figure 3.2, demonstrating that our proposed model can make more realistic and consistent trajectory predictions. Specifically, in the complex circular scenario (left-most images in Figure 3.2), which contains too many agents, both methods failed to converge to the ground-truth, especially when agents are turning or moving at high speeds, but the prediction results of our Multiclass-SGCN exhibit less divergence and are better aligned with the ground-truth trajectories. Moreover, for some static agents, Semantics-STGCNN generates abnormal predictions, while our model does not. As for the middle images in Figure 3.2, it is clear that Semantics-STGCNN totally diverges from the ground-truth, whereas our results match the ground-truth considerably. Furthermore, for the right-most images in Figure 3.2, both methods are close to the ground-truth, but Multiclass-SGCN presents more stable trajectories with lower amplitude oscillations.

To summarize, Semantics-STGCNN underperforms because the densely connected graph inherently introduces superfluous interactions that disrupt normal trajectories, and the lack of separate modeling of temporal dependencies results in unstable movements, even when no social interactions occur. In contrast, Multiclss-SGCN overcomes these issues by modeling both spatial interactions and temporal dependencies with velocity-label graph embedding and enhanced sparse graph learning modules, leading to better predictions.

Moreover, we present ablation visualizations in Figure 3.3 to qualitatively assess the

(a) Semantics-STGCNN [3]



(b) Muticlass-SGCN (ours)

Figure 3.2: Comparisons between our method and Semantics-STGCNN. Blue filled circles are observed trajectories, red hollow circles are ground-truth, purple lines in (a) are predicted results by [3], green lines in (b) are predicted results by the proposed Multiclass-SGCN.

contribution of each component to the prediction performance. In all three scenes, the full Multiclass-SGCN produces smoother trajectories that stay closer to the ground truth, especially around turning points and interaction areas, where deviations are significantly reduced. In contrast, removing AIM or SP leads to clearly larger prediction errors and overly linear trajectories in regions with long-term forecasting and dense interactions, demonstrating that both modules are essential for capturing fine-grained multi-class interactions and scene constraints.

3.4 Summary

This chapter presented **Multiclass-SGCN**, a sparse graph-based trajectory prediction framework tailored for heterogeneous traffic environments involving multiple agent types. By integrating semantic agent-class information with motion features through a velocity–label graph and employing an adaptive interaction mask to filter low-relevance connections, the framework effectively captures asymmetric cross-type interactions while

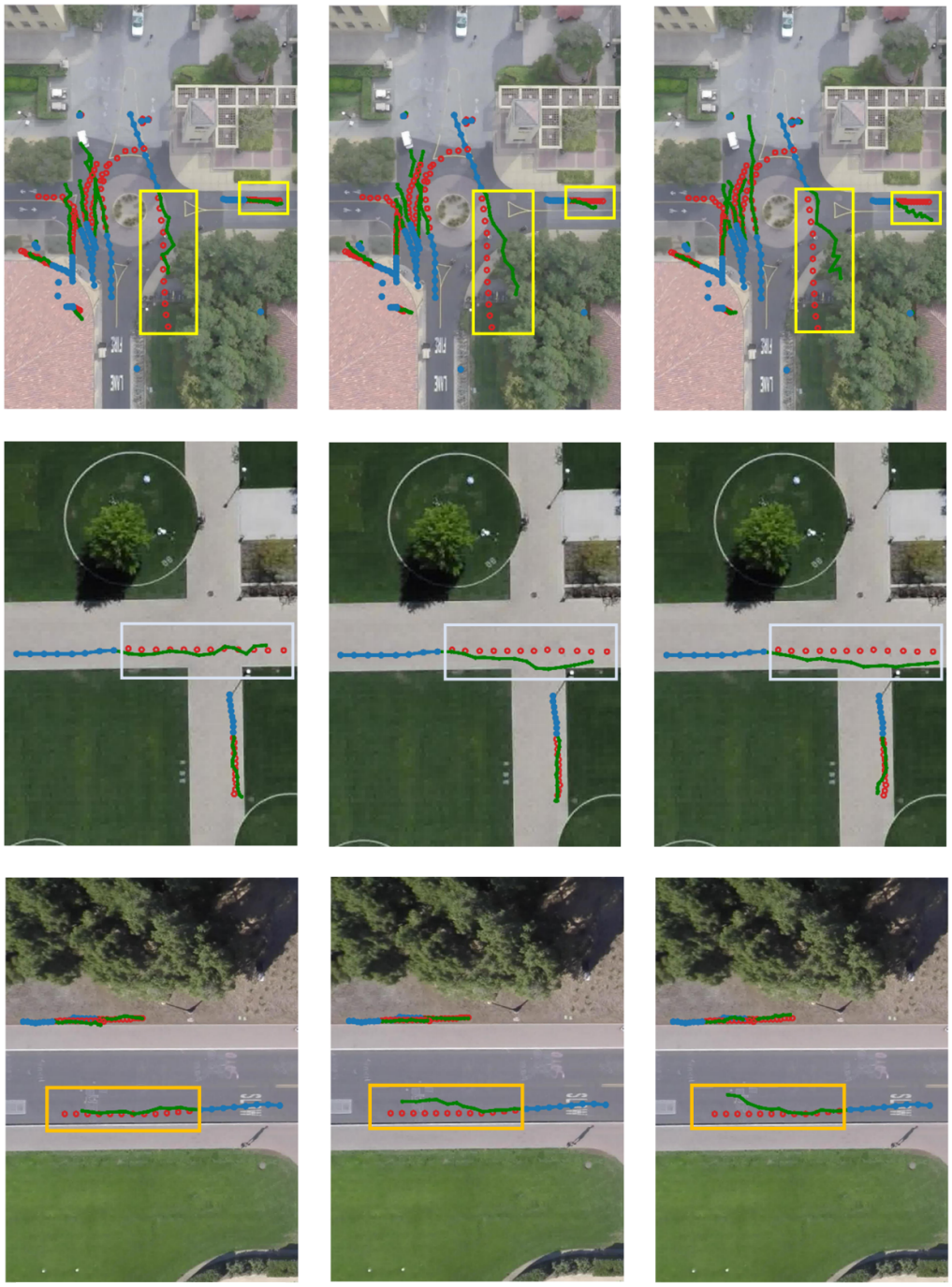


Figure 3.3: Multiclass-SGCN vs. Multiclass-SGCN (w/o AIM) vs. Multiclass-SGCN (w/o SP) in three different scenes. Blue filled circles are observed trajectories, red hollow circles are ground-truth, green lines are predicted results. Sample trajectories with significant differences are highlighted in the box.

maintaining computational efficiency.

While heterogeneous settings pose unique challenges due to agent diversity, homoge-

neous pedestrian scenarios present a different set of difficulties. In such environments, the absence of class distinctions shifts the emphasis toward capturing subtle, high-order dependencies and edge-level relational dynamics that emerge purely from motion and spatial context. The next chapter addresses these challenges with the UniEdge framework, which unifies spatial–temporal reasoning into a single high-order graph formulation, enabling efficient and expressive modeling of pedestrian interactions.

CHAPTER 4

Unified Spatial-Temporal Graph Reasoning in Homogeneous Pedestrian Trajectory Forecasting

Portions of this chapter have previously been published in the following peer-reviewed publication [136]:

- **Li, R.**, Qiao, T., Katsigiannis, S., Zhu, Z., & Shum, H. P. H., "Unified Spatial-Temporal Edge-Enhanced Graph Networks for Pedestrian Trajectory Prediction." *IEEE Transactions on Circuits and Systems for Video Technology* (TCSVT), 2025.

Pedestrian trajectory prediction aims to forecast future movements based on historical paths. Spatial-temporal methods often separately model spatial interactions among pedestrians and temporal dependencies of individuals. They overlook the direct impacts of interactions among different pedestrians across various time steps (i.e., high-order cross-time interactions). This limits their ability to capture spatial-temporal inter-dependencies and hinders prediction performance. To address these limitations, we propose UniEdge with three major designs. Firstly, we introduce a unified spatial-temporal graph data structure that simplifies high-order cross-time interactions into first-order relationships,

enabling the learning of spatial-temporal inter-dependencies in a single step. This avoids the information loss caused by multi-step aggregation. Secondly, traditional GNNs focus on aggregating pedestrian node features, neglecting the propagation of implicit interaction patterns encoded in edge features. We propose the Edge-to-Edge-Node-to-Node Graph Convolution (E2E-N2N-GCN), a novel dual-graph network that jointly models explicit N2N social interactions among pedestrians and implicit E2E influence propagation across these interaction patterns. Finally, to overcome the limited receptive fields and challenges in capturing long-range dependencies of auto-regressive architectures, we introduce a transformer encoder-based predictor that enables global modeling of temporal correlation. UniEdge outperforms state-of-the-arts on multiple datasets, including ETH, UCY, and SDD.

4.1 Introduction

The aim of pedestrian trajectory prediction is to forecast future paths based on observed movements (Figure 4.1(a)). High-precision prediction systems are crucial for applications like self-driving vehicles [8, 137] and video surveillance [138]. Specifically, in intelligent surveillance systems, especially at accident-prone intersections, early detection of pedestrian crossing intentions within a few seconds enables timely warnings to approaching vehicles through Vehicle-to-Everything (V2X) communication between vehicles, infrastructure and pedestrians, providing sufficient time for vehicles to react and reduce accident risks [139].

Predicting pedestrian trajectory is inherently challenging, primarily due to the complexity of interactions in which pedestrians continuously adjust their movements based on the evolving dynamics of others over multiple time steps. Spatial-temporal graph architectures (Figure 4.1(b)) are widely used to analyze human motions [140, 141] and pedestrian trajectories [7, 25, 27, 29, 30, 38, 39, 62], capturing spatial interactions within each frame and temporal dependencies over time.

This challenge is particularly severe when modeling **high-order cross-time interactions**, i.e., complex interactions among pedestrians across multiple time steps. Traditional spatial-temporal graph architectures require multiple steps to capture these interactions,

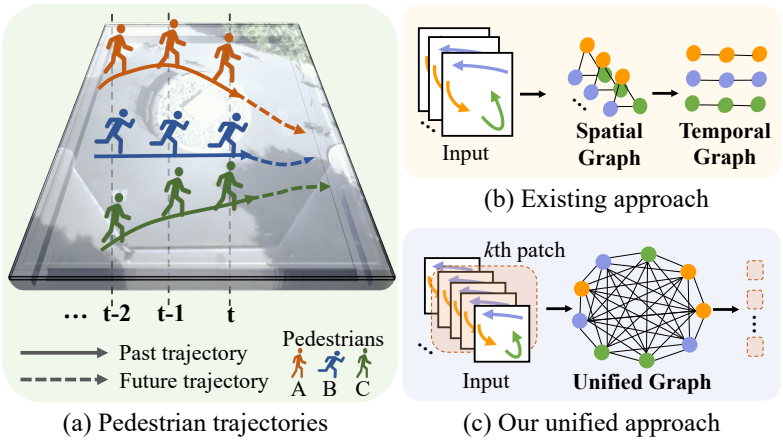


Figure 4.1: Motivation Illustration. **(a) Real-world pedestrian trajectories** over multiple time frames. **(b) Existing spatial-temporal approaches** separately model the spatial interactions among pedestrians and temporal dependencies of individuals. **(c) Our unified spatial-temporal graph** integrates spatial-temporal inter-dependencies and simplifies high-order cross-time interactions into first-order relationships.

where each node first aggregates spatial information at individual time steps and then addresses temporal dependencies through temporal networks. STGAT [38] combines graph attention [101] with Long Short-Term Memory (LSTM) [83] for sequential temporal modeling, while Social-STGCNN [29] and SGCN [27] advance to integrating Graph Convolutional Network (GCN) [131] with Temporal Convolutional Network (TCN) [116] for parallel processing. This paradigm has two key disadvantages: (1) when processing high-order interactions among pedestrians, this multi-step aggregation paradigm leads to potential under-reaching [119] due to increased effective resistance [102], where important interaction patterns are diluted and compressed with the increase of aggregation steps; and (2) the separation of spatial and temporal processing can disrupt the natural unified spatial-temporal inter-dependencies observed in real-world scenarios [117, 118], particularly in situations requiring immediate response to dynamic changes.

Another challenge lies in modeling the implicit influence propagation through edges in pedestrian social interactions. While Graph Neural Networks (GNNs) are widely adopted for modeling pedestrian interactions [25, 29, 38], existing approaches primarily focus on **Node-to-Node** (N2N) interactions (Figure 4.2(a)) through GNNs, e.g., using inverse distance [29] or attention-based [27, 38] weighting. Recent works like GroupNet [63] and HEAT [65] advance to **Edge-to-Node** (E2N) interactions (Figure 4.2(b)) by

incorporating edge features into node representations, enhancing the relation reasoning ability of the system. However, both N2N and E2N focus on the training of node features, while neglecting the crucial **Edge-to-Edge** (E2E) patterns [142, 143]. This fundamental limitation restricts GNNs' ability to capture the full spectrum of interaction dynamics in pedestrian behaviors, particularly in complex spatial-temporal scenarios where one pedestrian's behavior can implicitly influence others through cascade effects [142].

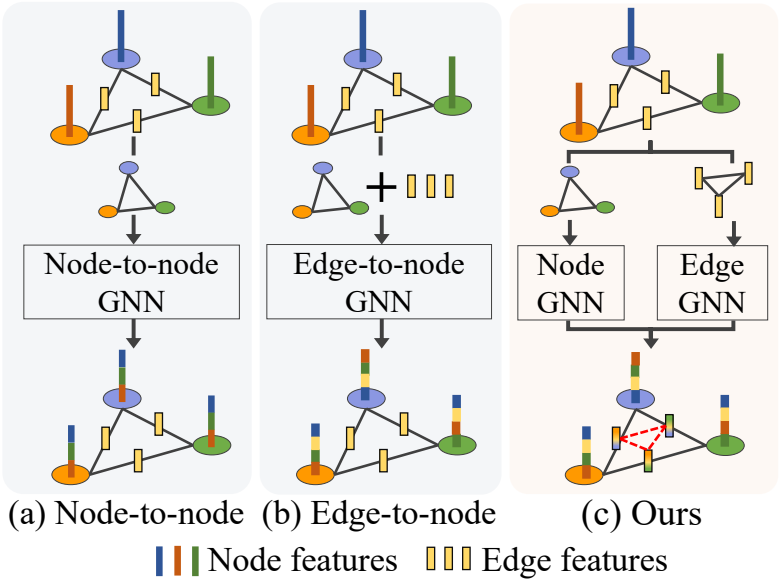


Figure 4.2: Illustration of graph learning procedures. (a) Node-to-Node (N2N), (b) Edge-to-Node (E2N), and (c) Our novel dual-graph introduces the combination of N2N and Edge-to-Edge (E2E) paradigm.

In this paper, we introduce the Unified Spatial-Temporal Edge-enhanced Graph Network (UniEdge) for pedestrian trajectory prediction. To address the first challenge, our unified spatial-temporal graph segments input trajectories into patch-based structures (Figure 4.1 (c)), simplifying high-order cross-time interactions into first-order relationships. This approach reduces effective resistance [102] and mitigates the under-reaching problem [119], preventing information dilution during propagation. By processing spatial-temporal information jointly in a single step, each unified patch maintains natural spatial-temporal inter-dependencies, enabling immediate responses to dynamic changes while preserving multi-step interaction patterns.

To tackle the second challenge, we introduce Edge-to-Edge-Node-to-Node Graph Convolution (E2E-N2N-GCN), a dual-graph network that jointly processes both node

and edge patterns, as depicted in Figure 4.2 (c). Dual-graph design provides a deeper understanding of graph topology in various domains [143, 144]. Our dual-graph architecture consists of two complementary graphs: a node-level graph that models explicit N2N social interactions among pedestrians, and an edge-level graph that captures the implicit E2E influence propagation across these interaction patterns. Specifically, we employ a first-order boundary operator [145] to construct edge graphs that reveal how interaction patterns influence each other through connected edges. This approach enables nuanced analysis of both individual behaviors and collective dynamics, essential for predictive accuracy in crowded environments.

Finally, we introduce a Transformer encoder-based predictor to overcome the limited receptive fields and long-range dependency challenges of auto-regressive architectures. Our predictor leverages attention mechanisms [67] to enable global modeling of temporal correlations through learnable placeholders, substantially improving the prediction capability.

Our approach outperforms state-of-the-art methods on commonly used pedestrian trajectory prediction datasets, including ETH [1], UCY [73] and Stanford Drone Dataset (SDD) [74]. The source code for UniEge is openly released on `https://github.com/Carrotsniper/UniEdge`.

Our contributions can be summarized as follows:

- We propose a unified spatial-temporal graph data structure that simplifies high-order cross-time interactions into first-order relationships. This enables direct learning of spatial-temporal inter-dependencies in a single step, avoiding information loss caused by multi-step aggregation while preserving critical interaction patterns.

- We introduce the Edge-to-Edge-Node-to-Node Graph Convolution (E2E-N2N-GCN), a novel dual-graph architecture that jointly captures both explicit N2N social interactions among pedestrians and implicit E2E influence propagation across interaction patterns through first-order boundary operators. This enables more comprehensive modeling of complex pedestrian behaviors.

- We introduce a transformer-based predictor that overcomes the limited receptive

fields and challenges associated with capturing long-range dependencies inherent in auto-regressive architectures. This enables global modeling of temporal correlations, substantially improving prediction performance.

4.2 Methodology

4.2.1 Problem Formulation and Feature Initialization

The goal of pedestrian trajectory prediction is to estimate the possible future trajectories of a pedestrian based on observed trajectories and nearby neighbors. Mathematically, consider a multi-pedestrian scenario containing N pedestrians in T_{obs} time steps. The observed trajectories of each pedestrian $i \in [1, \ldots, N]$ can be represented as $X_i = \{(x_t^i, y_t^i) \mid t \in [-T_{obs}+1, \ldots, 0]\}$ and its ground-truth future trajectories can be defined as $Y_i = \{(x_t^i, y_t^i) \mid t \in [1, \ldots, T_{pred}]\}$. For N pedestrians, the observed and ground-truth future trajectories are $\mathbf{X} = [X_1, X_2, \ldots, X_N] \in \mathbb{R}^{N \times T_{obs} \times 2}$ and $\mathbf{Y} = [Y_1, Y_2, \ldots, Y_N] \in \mathbb{R}^{N \times T_{pred} \times 2}$ respectively, where 2 denotes the 2D coordinates. Our proposed UniEdge aims to learn a prediction function $\mathcal{F}_{pred}(\cdot)$ that minimizes the differences between the predicted trajectories $\hat{\mathbf{Y}} = \mathcal{F}_{pred}(X)$ and the ground-truth future trajectories $\mathbf{Y}$. Instead of directly predicting absolute coordinates, we follow [25, 27, 29, 30] that predict relative coordinates of each pedestrian to ensure the robustness and generalization ability of the system across different scenarios.

For trajectory feature initialization, our model takes inputs consisting of pedestrian velocities $\mathbf{v}$, velocity norms $\rho = \|\mathbf{v}\|_2$, and pedestrian movement angles $\theta = \text{angle}(\mathbf{v})$, where $\|\cdot\|_2$ denotes the vector 2-norm and $\text{angle}(\cdot)$ is the function that computes the angle of the velocity vectors. We follow [146] that subtract each historical $\mathbf{v}_t, t \in [-T_{obs}, 0]$ by the corresponding endpoint $\mathbf{v}_{T_{pred}}$ as the pre-process step. These motion dynamic features are embedded and then concatenated to obtain the final geometric feature representation as follows:

$$\mathcal{X} = \text{CONCAT}(f(\mathbf{v}, W_v), f(\rho, W_{norm}), f(\theta, W_{angle})),$$

where $\mathcal{X} \in \mathbb{R}^{N \times T_{obs} \times D}$, N and T_{obs} represent the total number of pedestrians and time steps, respectively, and D denotes the embedded feature dimension. Here, $f(\cdot, \cdot)$ rep-

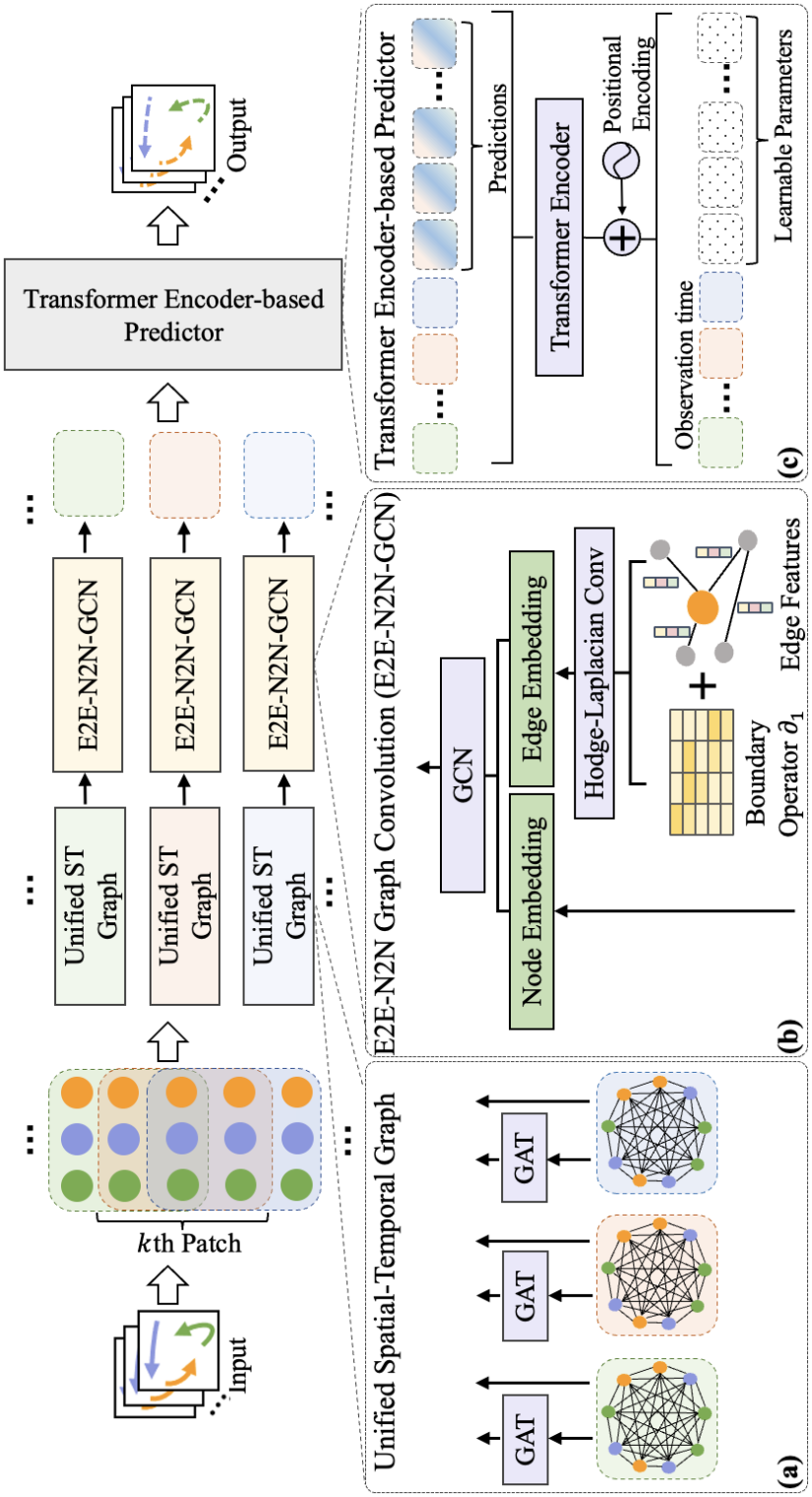


Figure 4.3: Overview of the proposed UniEdge. (a) Construction of patch-based unified spatial-temporal graphs that simplify cross-time interactions into first-order relationships, (b) Edge-to-Edge-Node-to-Node Graph Convolution (E2E-N2N-GCN) that jointly processes N2N interactions and E2E influence propagation, and (c) Transformer Encoder-based trajectory predictor.

resents Multi-Layer Perceptron (MLP) for feature embedding, and W represents the corresponding weights.

4.2.2 Unified Spatial-temporal Graph

Previous trajectory prediction methods often adopt a two-step approach, separately modeling pedestrian spatial interactions and individual temporal dependencies [27–29]. This approach is limited in capturing high-order cross-time interactions, which require multi-step aggregation. Such multi-step processing increases the effective resistance - a measurement of graph connectivity that quantifies the efficiency of information flow between nodes [102, 147]. High effective resistance impedes graph message-passing, leading to under-reaching problem [119], where message flows from distant nodes are diluted and compressed.

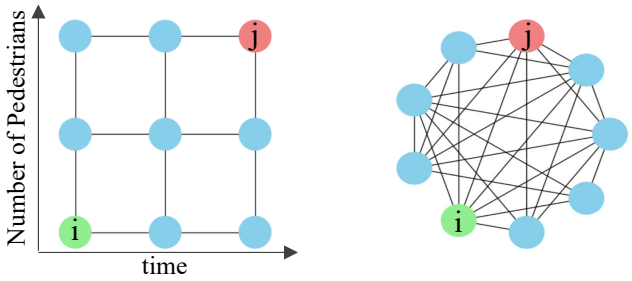


Figure 4.4: Comparison of effective resistance (R_{ij}) between traditional spatial-temporal approach (left, $R_{ij} = 1.50$) and our unified spatial-temporal graph (right, $R_{ij} = 0.27$). Lower R_{ij} indicates better message propagation efficiency.

To address these challenges, we propose a unified spatial-temporal graph to simplify high-order cross-time interactions among pedestrians into first-order relationships, enabling direct learning of spatial-temporal inter-dependencies, and preserving high-order interactions without information dilution. This design significantly reduces the effective resistance during message passing, improving information flow efficiency [102, 147] and alleviating the risk of under-reaching [119]. Figure 4.4 illustrates the difference in effective R between the message-passing paradigms of traditional spatial-temporal approach and

our unified approach:

$$R_{ij} = (e_i - e_j)^T L^+ (e_i - e_j) \tag{4.1}$$

where L^+ denotes the Moore-Penrose pseudoinverse of the graph Laplacian matrix representing the graph connectivity [148], and e_i, e_j are standard basis vectors corresponding to nodes i and j. Lower R_{ij} values indicate better message propagation efficiency between nodes.

To reduce computational overhead in processing entire sequences and to better capture fine-grained pedestrian dynamics, we adopt a patch-based strategy akin to the local receptive fields used in convolution kernel for image processing. [149]. Specifically, to construct the unified spatial-temporal graph depicted in Figure 4.3 (a), the input features are segmented into K overlapping patches across the temporal dimension T_{obs}. These patches are defined by a length L and a stride $\mathcal{S}$, yielding $K = \left\lfloor \frac{T_{obs}-L}{\mathcal{S}} \right\rfloor + 1$. For each patch k, ranging from 1 to K, a graph $\mathcal{G}^k_{node} = (\mathcal{Z}^k, \mathcal{A}^k_{node})$ is constructed. Here, $\mathcal{Z}^k \in \mathbb{R}^{NL \times D}$ represents the node features, and $\mathcal{A}^k_{node} \in \mathbb{R}^{NL \times NL}$ denotes the node adjacency matrix, which encapsulates the node connections. This configuration further benefits subsequent trajectory prediction phases by reducing the number of input tokens from T_{obs} to K, which is crucial when using the transformer encoder model. It leads to a quadratic reduction in memory usage and computational complexity for the attention map, by a factor of $\left(\frac{T_{obs}}{K}\right)^2$.

We then apply GAT [38, 62, 150] to initialize interaction strengths for the kth graph $\mathcal{G}^k$ as:

$$\mathcal{H}^k_{node} = \text{GAT}(\mathcal{Z}^k, \mathcal{A}^k_{node}), \tag{4.2}$$

where each node $\mathcal{H}^k_{node,i}$ is embedded as:

$$\mathcal{H}^k_{node,i} = \sigma\left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \alpha^k_{i,j} \mathbf{\Theta} \mathcal{Z}^k_j\right), \tag{4.3}$$

$$\alpha^k_{i,j} = \frac{\exp\left(\mathbf{a}^\top \Gamma\left(\mathbf{\Theta}[\mathcal{Z}^k_i \parallel \mathcal{Z}^k_j]\right)\right)}{\sum_{j \in \mathcal{N}(i) \cup \{i\}} \exp\left(\mathbf{a}^\top \Gamma\left(\mathbf{\Theta}[\mathcal{Z}^k_i \parallel \mathcal{Z}^k_j]\right)\right)}, \tag{4.4}$$

where $\mathbf{\Theta}(\cdot)$ is transformation function, $\Gamma(\cdot)$ and $\sigma(\cdot)$ denote activation functions, $\mathcal{N}(\cdot)$ is the neighbor set of node i and $\mathbf{a}^\top$ represents learnable parameters. Attention coefficient

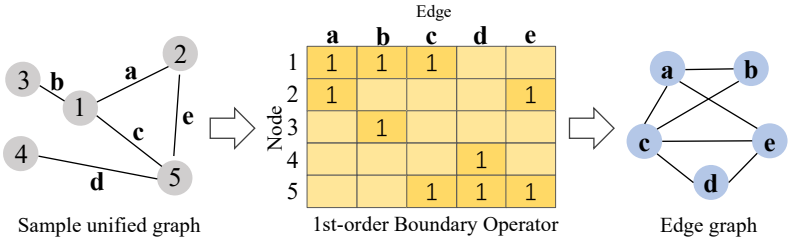


Figure 4.5: Illustration of edge graph construction from a unified spatial-temporal graph using the first-order boundary operator $\mathcal{B}_1$. Nodes are represented by numbers, and edges connecting these nodes are labeled with letters. Applying the first-order boundary operator transforms each edge into a node in the edge graph, with connections formed based on shared nodes in the original graph.

$\alpha_{i,j}^k$ represents the weights between two nodes. During training, these weight coefficients are dynamically updated to reflect the importance of each node's contribution to its neighbors.

4.2.3 E2E-N2N Graph Convolution (E2E-N2N-GCN)

Previous pedestrian trajectory models typically adopt node-centric approaches, such as N2N [25, 27, 29, 30, 151] and E2N [63, 65] paradigms to understand and capture node dependencies. However, these methods overlook crucial E2E patterns, limiting their ability to capture the full spectrum of interaction dynamics. This oversight may result in a partial understanding of pedestrian behaviors, especially in complex scenarios where interaction patterns influence each other.

To address this limitation, we propose a novel Edge-to-Edge-Node-to-Node Graph Convolution (E2E-N2N-GCN) module (Figure 4.3 (b)), a dual-graph architecture that leverages the first-order boundary operator to construct edge graphs. By jointly modeling both explicit N2N social interactions among pedestrians and implicit E2E influence propagation across interaction patterns, our approach enables more comprehensive modeling of complex pedestrian behaviors. This dual-graph design allows each unified spatial-temporal graph to capture how interaction patterns evolve and influence each other through connected edges, leading to more accurate trajectory predictions.

To construct the edge graph, we apply the first-order boundary operator $\mathcal{B}_1$ to transform it into its corresponding undirected edge graph $\mathcal{G}_{edge}^k = (\mathcal{E}^k, \mathcal{A}_{edge}^k)$, where $\mathcal{E}^k$

represents the node features in the edge graph, and $\mathcal{A}^k_{edge}$ indicates the new adjacency relations. This operator reinterprets the connections between nodes (edges in the original graph) as nodes in the new graph, creating edges between these new nodes if they share a common node in the original graph. Figure 4.5 illustrates this process, effectively showing how relationships are redefined to highlight deeper interaction dynamics.

To analyze and update the feature propagation of each edge graph, we employ the first-order Hodge Laplacian [142, 143] to analyze and learn the dynamics within these edge graphs:

$$\mathcal{L}_1 = \mathcal{B}_1^\top \mathcal{B}_1 + \mathcal{B}_2^\top \mathcal{B}_2, \tag{4.5}$$

where $\mathcal{L}_1$ represents first-order Hodge Laplacian operator, and $\mathcal{B}_1^\top$ captures and enhances edge relationships, focusing on direct interactions. $\mathcal{B}_2$ is typically relevant for higher-dimensional structures and not a primary focus here. We perform edge convolution by adapting the Hodge-Laplacian Laguerre Convolution (HLLConv) [142, 143] to obtain the high-level edge embedding $\mathcal{H}^k_{edge}$ for each edge graph k:

$$\begin{aligned} \mathcal{H}^k_{edge} &= HLLConv(\mathcal{E}^k, \mathcal{A}^k_{edge}) \\ &= \hbar_1 * \mathcal{E}^k \\ &= \sum_{j=0}^{J-1} \theta_j \Gamma_j(\mathcal{L}_1)\mathcal{E}^k, \end{aligned} \tag{4.6}$$

where $\hbar_1$ is a spectral filter based on $\mathcal{L}_1$ applied to update edge features $\mathcal{E}^k$, with θ_j representing learnable parameters, and $\Gamma_j(\cdot)$ indicates the Laguerre polynomial functions. Detailed explanations of spectral filter $\hbar_1$ are shown in Algorithm 1.

Finally, after obtaining the embedded node features $\mathcal{H}^k_{node}$ and edge features $\mathcal{H}^k_{edge}$ for the kth unified spatial-temporal graph, we leverage a fusion GCN to integrate node and edge embeddings, enhancing the understanding of graph dynamics. Specifically, we incorporate normalized edge embedding as weights into the aggregation process of GCN:

$$\mathcal{H}^k = GCN(\mathcal{H}^k_{node}, \mathcal{H}^k_{edge}, \mathcal{A}^k_{node}), \tag{4.7}$$

Algorithm 1: Hodge-Laplacian Laguerre Convolution

Input: First-order Hodge Laplacian $\mathcal{L}_1 = \mathcal{B}_1^\top \mathcal{B}_1 + \mathcal{B}_2^\top \mathcal{B}_2$
Output: Spectral filter $\hbar_1$
Step 1: Perform eigen-decomposition on $\mathcal{L}_1$:

$$\mathcal{L}_1 \phi_1^i = \lambda_1^i \phi_1^i$$

to obtain the orthonormal bases ϕ_1^i for $i \in [0, 1, 2, \cdots, \infty]$.
The spectral filter $\hbar$ of the 1-st order HL can be represented as:

$$\hbar_1(\cdot, \cdot) = \sum_{i=0}^{\infty} \hbar_1(\lambda_1^i) \phi_1^i(\cdot) \phi_1^i(\cdot)$$

Step 2: Approximate the spectral filter $\hbar_1(\lambda_1)$ by Laguerre polynomial functions:

$$\hbar_1(\lambda_1) = \sum_{j=0}^{J-1} \theta_j \Gamma_j(\lambda_1)$$

where θ_j is the jth expansion coefficient with jth Laguerre polynomial, and $\Gamma_j(\cdot)$ is written in a recurrence format as:

$$\Gamma_{j+1}(\lambda_1) = \frac{(2j + 1 - \lambda_1)\Gamma_j(\lambda_1) - j\Gamma_{j-1}(\lambda_1)}{j + 1}$$

with base cases defined as:

$$\Gamma_0(\lambda_1) = 1, \quad \Gamma_1(\lambda_1) = 1 - \lambda_1$$

and each node i in the graph is embedded as:

$$\mathcal{H}_i^k = \sigma\left(\mathbf{\Theta}(\mathcal{H}_{node,i}^k) + \sum_{j \in \mathcal{N}(i)} \Phi(\mathcal{H}_{edge,ij}^k) \mathbf{\Theta}(\mathcal{H}_{node,j}^k)\right), \tag{4.8}$$

where $\mathbf{\Theta}(\cdot)$ and $\Phi(\cdot)$ are linear transformations for node and edge features [142], with $\sigma(\cdot)$ as the activation function.

4.2.4 Transformer Encoder Predictor

Temporal dependency modeling in trajectory prediction has evolved through various architectures. RNNs [21,28] and TCNs [27,29] have been widely adopted, they suffer from limited receptive fields and struggle to capture long-range dependencies. Although Transformer encoder-decoder architectures [67,69,137] address the long-range dependency

issue, it introduces extra computation costs.

In this work, we design a Transformer encoder-based predictor for trajectory prediction. As shown in Figure 4.3 (c), by encoding future trajectories as learnable parameters and concatenating them with historical trajectories, our approach enables unified modeling of both past and future information, allowing the model to fully leverage global temporal dependencies [152] for more accurate predictions. We simply stack the graph embeddings $\mathcal{H}^k$ output by E2E-N2N-GCN across all patches to obtain the integrated feature representations $\mathbf{H}$:

$$\mathbf{H} = \text{STACK}(\mathcal{H}^1, \mathcal{H}^2, \cdots, \mathcal{H}^K) \in \mathbb{R}^{K\times(NL)\times D}. \tag{4.9}$$

We perform temporal average pooling across the L channel, and the output $\mathbf{H} \in \mathbb{R}^{N\times K\times D}$ is served as the historical input tokens. We then initialize a learnable placeholder to form the padded future tokens as $\mathbf{F} \in \mathbb{R}^{N\times T_{pred}\times D}$. The temporal channel of these tokens, T_{pred}, is tailored to match our prediction horizon. This setup aligns with the requirements of the Transformer encoder architecture [67, 153], which necessitates uniform sequence lengths for both inputs and outputs to enable synchronous processing. This design allows our model to directly produce trajectories of the required length. Throughout the training process, these placeholders are incrementally refined to represent the predicted trajectories, thereby enhancing the prediction capabilities.

Finally, the input tokens for the Transformer encoder are formed by concatenating the learned historical input tokens $\mathbf{H}$ and padded future tokens $\mathbf{F}$, resulting in the concatenated feature representation $\hat{\mathbf{H}}_{in} \in \mathbb{R}^{N\times(K+T_{pred})\times D}$. We further enhance these tokens with a learnable additive position embedding $\mathbf{P} \in \mathbb{R}^{N\times(K+T_{pred})\times D}$ [67] that is applied to the entire concatenated sequence to preserve the temporal order information. The Transformer encoder then processes these augmented inputs to produce the predicted sequence representations $\hat{\mathbf{Y}} \in \mathbb{R}^{N\times(K+T_{pred})\times D}$:

$$\begin{aligned} \hat{\mathbf{Y}} &= \mathit{Encoder}(\hat{\mathbf{H}}_{in} + \mathbf{P}), \\ \hat{\mathbf{H}}_{in} &= [\mathbf{H} \parallel \mathbf{F}], \end{aligned} \tag{4.10}$$

where $[\cdot \parallel \cdot]$ denotes the concatenation operation along the temporal dimension. Note

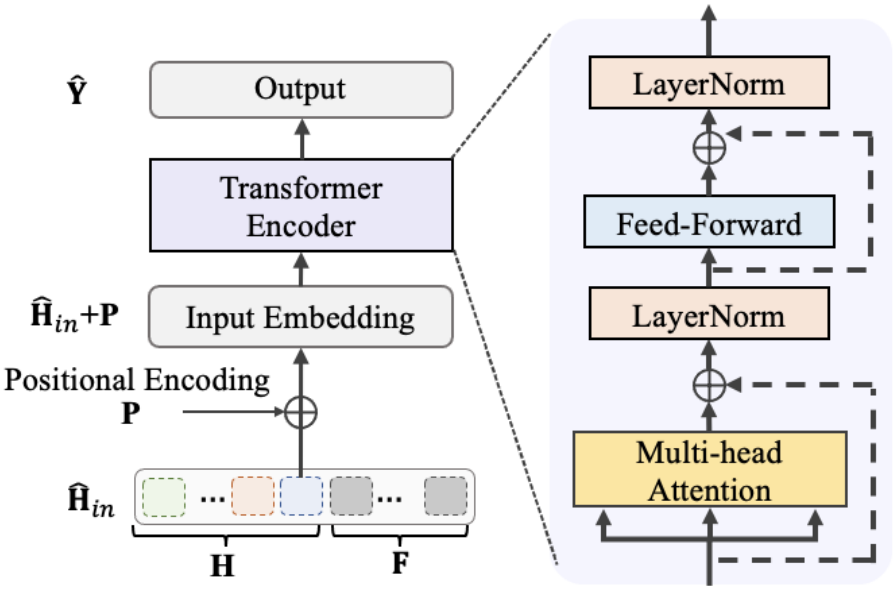


Figure 4.6: Illustration of the Transformer encoder-based predictor.

that $\hat{\mathbf{Y}}$ represents the complete output of the encoder with length $K + T_{pred}$, only the last T_{pred} time steps are used as the predicted trajectory representations, corresponding to the padded future tokens $\mathbf{F}$. The architecture of the Transformer encoder and the learning process are shown in Figure 4.6. Similarly to [7, 27, 29], we employ the bi-variate Gaussian loss function $\mathcal{L}_{prediction}$ to optimize the trajectory prediction:

$$\mathcal{L}_{prediction} = -\sum_{t=1}^{T_{pred}} \log \mathcal{P}((x_t, y_t)|\hat{\mu}_t, \hat{\sigma}_t, \hat{\rho}_t), \tag{4.11}$$

where $\hat{\mu}$ and $\hat{\sigma}$ are the mean and variance of bi-variate Gaussian distribution, and $\hat{\rho}$ represents the correlation coefficient.

4.2.5 Implementation Details

The UniEdge framework, developed using PyTorch, is trained end-to-end on an NVIDIA TITAN XP GPU. We use a consistent batch size of 128 across all datasets, with initial learning rates set at 0.001 for the ETH/UCY datasets and 0.01 for the SDD datasets. The learning rate is adjusted every 50 epochs by a factor of 0.5. The AdamW optimizer is employed to train the model. The architecture for learning graph employs single-layer GAT, HLLConv, and GCN components. Node and edge embedding dimensions are set to

128. The Transformer encoder-based predictor is configured with a hidden dimension of 256 with 4 attention heads.

4.3 Experiments

4.3.1 Experimental Setup

We evaluate the proposed UniEdge on multiple benchmark datasets, including ETH [1], UCY [73], and Stanford Drone Dataset (SDD) [74]. The ETH dataset contains two subsets (ETH and HOTEL) and the UCY dataset contains three subsets (UNIV, ZARA1, ZARA2), with the total number of pedestrians captured in these 5 subsets being 1,536. SDD is a benchmark dataset for pedestrian trajectories captured by a drone with a bird's eye viewing of university campus scenes and it contains 5,232 pedestrians across 8 different scenes.

We follow the experimental setup of [27, 28, 154], using 3.2 seconds (8 frames) of observation trajectories to predict the next 4.8 seconds (12 frames). For ETH and UCY datasets, we follow existing works [21, 25, 27, 29, 30, 69] and use the leave-one-out strategy for training and evaluation. For SDD, we follow the existing train-test split [25, 30, 39] to train and test our proposed method. During training, we employ data augmentation following [154] to diversify and enrich our training datasets. This strategy is pivotal in enhancing the model's generalization capabilities.

During testing, we follow the standard protocol [21, 28] and sampling strategy [25] that generates 20 predictions from the predicted distributions; the best sample is used to compute the evaluation metrics. Average Displacement Error (ADE) and Final Displacement Error (FDE) [21, 27–29] are used as evaluation metrics:

$$\begin{aligned} \text{ADE} &= \frac{1}{N \times T_{pred}} \sum_{i=1}^{N} \sum_{t=1}^{T_{pred}} \sqrt{(x_t^i - \hat{x}_t^i)^2 + (y_t^i - \hat{y}_t^i)^2}, \\ \text{FDE} &= \frac{1}{N} \sum_{i=1}^{N} \sqrt{(x_{T_{pred}}^i - \hat{x}_{T_{pred}}^i)^2 + (y_{T_{pred}}^i - \hat{y}_{T_{pred}}^i)^2}, \end{aligned} \tag{4.12}$$

where $(\hat{x}_t^i, \hat{y}_t^i)$ and (x_t^i, y_t^i) represent the predicted trajectory coordinates and ground-truth trajectory coordinate for the i-th pedestrian at time step t.

Table 4.1: Results on The ETH (ETH, HOTEL) and UCY (UNIV, ZARA1, ZARA2) Datasets for Pedestrian Trajectory Prediction

Method	Venue/Year	ADE(↓) / FDE(↓)					
		ETH	HOTEL	UNIV	ZARA1	ZARA2	**AVG**
Social GAN [21]	CVPR'18	0.87/1.62	0.67/1.37	0.76/1.52	0.35/0.68	0.42/0.84	0.61/1.21
Social-STGCNN [29]	CVPR'20	0.64/1.11	0.49/0.85	0.44/0.79	0.34/0.53	0.30/0.48	0.44/0.75
SGCN [27]	CVPR'21	0.63/1.03	0.32/0.55	0.37/0.70	0.29/0.53	0.25/0.45	0.37/0.65
GP-Graph [25]	ECCV'22	0.43/0.63	0.18/0.30	0.24/0.42	0.17/0.31	0.15/0.29	0.23/0.39
Social-VAE [22]	ECCV'22	0.41/0.58	0.13/0.19	0.21/0.36	0.17/0.29	0.13/0.22	0.21/0.33
MemoNet [155]	CVPR'22	0.40/0.61	**0.11**/**0.17**	0.24/0.43	0.18/0.32	0.14/0.24	0.21/0.35
GroupNet [63]	CVPR'22	0.46/0.73	0.15/0.25	0.26/0.49	0.21/0.39	0.17/0.33	0.25/0.44
Graph-TERN [39]	AAAI'23	0.42/0.58	0.14/0.23	0.26/0.45	0.21/0.37	0.17/0.29	0.24/0.38
MSRL [151]	AAAI'23	0.28/0.47	0.14/0.22	0.24/0.43	0.17/0.30	0.14/0.23	0.19/0.33
LED [72]	CVPR'23	0.39/0.58	**0.11**/**0.17**	0.26/0.43	0.18/0.26	0.13/0.22	0.21/0.33
EqMotion [26]	CVPR'23	0.40/0.61	0.12/0.18	0.23/0.43	0.18/0.32	0.13/0.23	0.21/0.35
EigenTrajectory [30]	ICCV'23	0.36/0.57	0.13/0.21	0.24/0.43	0.20/0.35	0.15/0.26	0.22/0.36
TUTR [69]	ICCV'23	0.40/0.61	**0.11**/0.18	0.23/0.42	0.18/0.34	0.13/0.25	0.21/0.36
SMEMO [156]	TPAMI'24	0.39/0.59	0.14/0.20	0.23/0.41	0.19/0.32	0.15/0.26	0.22/0.35
MFAN [66]	PR'24	0.48/0.62	0.17/0.21	0.26/0.41	0.23/0.36	0.21/0.33	0.27/0.39
DDL [146]	ICRA'24	**0.26**/0.50	0.15/0.35	0.29/0.58	0.16/0.29	0.13/0.22	0.20/0.39
ATP-VAE [157]	TCSVT'24	0.48/0.76	0.14/0.20	0.26/0.44	0.28/0.48	0.20/0.35	0.27/0.45
MRGTraj [158]	TCSVT'24	0.28/0.47	0.21/0.39	0.33/0.60	0.24/0.44	0.22/0.41	0.26/0.46
SingularTrajectory [159]	CVPR'24	0.35/**0.42**	0.13/0.19	0.25/0.44	0.19/0.32	0.15/0.25	0.21/0.32
HighGraph [31]	CVPR'24	0.40/0.55	0.13/**0.17**	0.20/0.33	0.17/0.27	**0.11**/0.21	0.20/0.30
UniEdge (Ours)	-	0.36/0.46	**0.11**/**0.17**	**0.19**/**0.28**	**0.14**/**0.20**	**0.11**/**0.16**	**0.18**/**0.25**

4.3.2 Baseline Methods

We compare the proposed UniEdge framework with the following previous state-of-the-art methods:

Graph-based methods: Social-STGCNN [29]: an approach that models spatial-temporal pedestrian interactions through graphs; SGCN [28]: an approach that models spatial-temporal interactions through sparse directed spatial graph and sparse directed temporal graph; GP-Graph [25]: an approach that considers group-based pedestrian behaviors; Graph-TERN [39]: an approach that integrates multi-relational graph and control endpoint for trajectory prediction; EigenTrajectory(+SGCN) [30]: a model that learns trajectories in eigenspaces and graph representations. MFAN [66]: an approach that models spatial-temporal interactions for both edges and nodes. HighGraph [31]: a plug-and-play module that captures high-order dynamics of pedestrians - we use the HighGraph(+Social-VAE) variant for comparisons.

Generative-based methods: Social GAN [21]: a method that uses pooling window module with Generative Adversarial Network (GAN) to generate diverse trajectories; Social-VAE [22]: a method that employs timewise variational autoencoder(VAE) and

attention mechanism to generate trajectories; GroupNet [63]: a method that introduces multiscale hypergraph with edge strength, utilizing conditional-VAE (CVAE) to generate trajectories; MSRL [151]: a method that models multi-stream interactions for trajectory prediction based on CVAE; MRGTraj [158]: a method based on CVAE and non-auto-regressive transformer encoder to generate diverse trajectories; ATP-VAE [157]: an attention-based VAE architecture for trajectory prediction; LED [72]: a multi-modal framework based on diffusion for prediction; SingularTrajectory [159]: a diffusion framework based on singular projection and adaptive anchor to generate trajectories.

Other methods: MemoNet [155]: an approach based on the retrospective-memory bank for trajectory representations; EqMotion [26]: an approach that models trajectories via equivariant dynamics and invariant interaction; TUTR [69]: a transformer-based framework; SMEMO [156]: an approach that models trajectories through social memory modules; DDL [146]: goal-based transformer for trajectory prediction.

4.3.3 Quantitative Comparison

ETH and UCY Datasets

Table 4.1 presents the quantitative comparisons of our UniEdge model against existing methods under ADE and FDE metrics. Compared to the previous state-of-the-art (SOTA) generative-based method MSRL, our UniEdge demonstrates improvements of 5.3% in average ADE and 24.2% in average FDE. Unlike MSRL, which is a two-stage framework requiring separate training for the CVAE model and the trajectory decoder, UniEdge operates in an end-to-end manner, improving the overall performance while maintaining model parameter efficiency. Compared to the best graph-based method HighGraph, our UniEdge shows significant improvements of 10.0% in average ADE and 16.7% in average FDE. Although HighGraph introduces high-order interaction modeling, it operates only on individual time steps, rather than cross-time interactions, which limits its effectiveness in capturing dynamic changes over time. Contrasted to these graph-based methods, our UniEdge comprehensively models edge information flow and cross-time interactions, which can be the key to performance gain. Compared to DDL, which uses similar data pre-processing techniques, our UniEdge surpasses it by 10.0% in ADE and 35.9% in FDE,

Table 4.2: Results on The Stanford Drone Dataset (SDD) for Pedestrian Trajectory Prediction

Method	Venue/Year	ADE(↓) / FDE(↓) SDD
Social GAN [21]	CVPR'18	27.23/41.44
Social-STGCNN [29]	CVPR'20	26.46/42.71
GroupNet [63]	CVPR'22	9.31/16.11
MemoNet [155]	CVPR'22	8.56/12.66
GP-Graph [25]	ECCV'22	9.10/13.80
MSRL [151]	AAAI'23	8.22/13.39
Graph-TERN [39]	AAAI'23	8.43/14.26
LED [72]	CVPR'23	8.48/11.66
EigenTrajectory [30]	ICCV'23	8.05/13.25
TUTR [69]	ICCV'23	7.76/12.69
SMEMO [156]	TPAMI'24	8.11/13.06
MFAN [66]	PR'24	9.69/14.51
HighGraph [31]	CVPR'24	7.98/11.42
UniEdge (Ours)	-	**7.51**/**10.89**

demonstrating enhanced prediction performance. While our UniEdge model demonstrates state-of-the-art (SOTA) performance on four subsets (HOTEL, UNIV, ZARA1, and ZARA2), particularly in environments with rich pedestrian interactions such as UNIV, it faces challenges similar to the graph-based SOTA method HighGraph on the ETH subset. This limitation of graph-based methods is mainly caused by the sparsity of the ETH subset, where fewer pedestrians and limited interactions constrain the expressive power of graph representations.

SDD Dataset

Table 4.2 presents the quantitative comparison results of our model against various previous methods on SDD dataset. Unlike the ETH and UCY datasets, the SDD is a larger dataset featuring more complex pedestrian interactions. Compared to generative-based methods, UniEdge improves 8.6% in ADE compared to MSRL and 6.6% in FDE compared to LED. As a graph-based approach, our UniEdge outperforms the best graph-based HighGraph model by 5.9% in ADE and 4.6% in FDE. Compared to SOTA methods, UniEdge shows an improvement of 3.0% in ADE over TUTR. These results further highlight the effectiveness of our proposed UniEdge model in handling complex social scenarios.

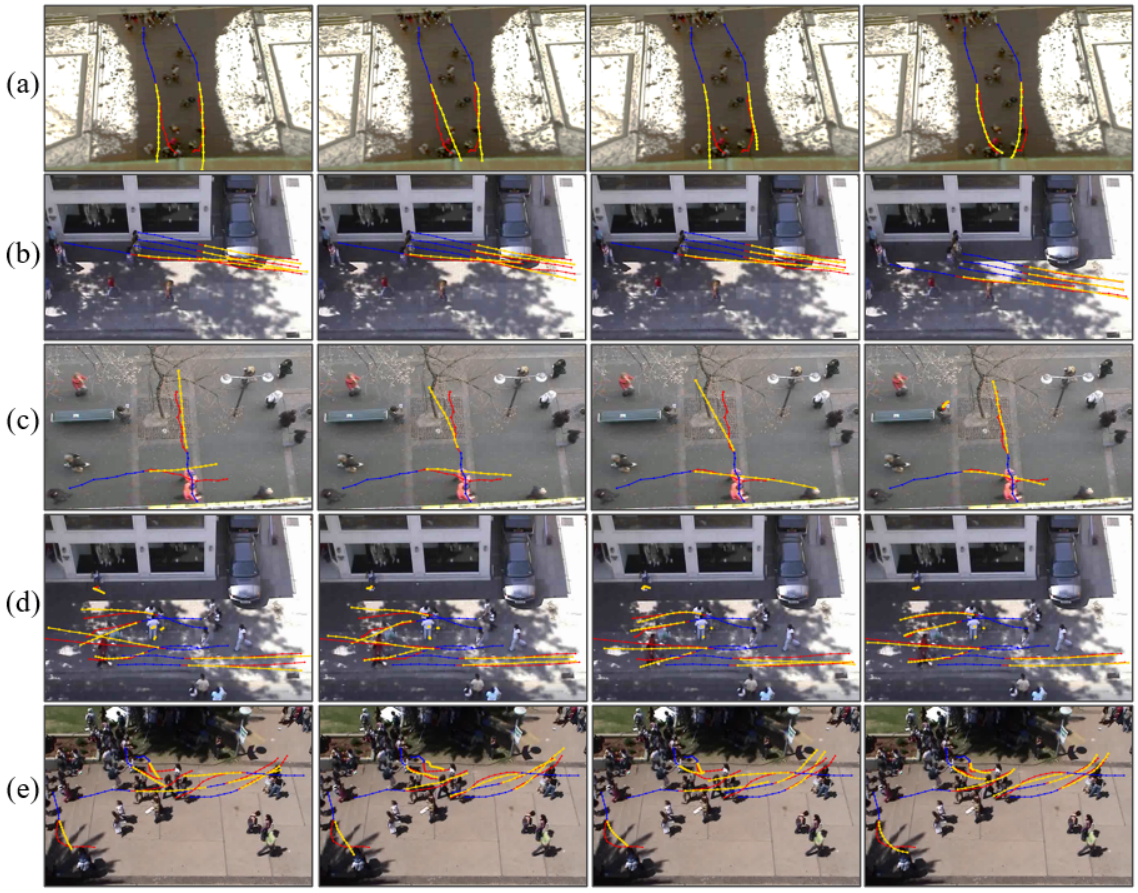


Figure 4.7: Visualization of predicted trajectories on the ETH and UCY datasets. Historical trajectories are in blue, ground-truth trajectories are in red, and predicted trajectories are in yellow. Scenario (a) shows two pedestrians walking in parallel and meet; Scenario (b) illustrates a group of pedestrians walking in parallel; (c) shows pedestrians meeting each other; (d) depicts several groups walking in opposing directions; and (e) presents a more complex scenario that pedestrian movements are stochastic.

4.3.4 Qualitative Comparison

Trajectory Visualization Comparison

In this section, we compare the most likely predictions between our UniEdge and previous graph-based methods, GP-Graph [25], Graph-TERN [39] and EigenTrajectory [30] on the ETH and UCY datasets.

As shown in Figure 4.7, our prediction results are significantly closer to the ground-truth trajectories compared to other methods in all scenarios. **Scenario (a)** depicts two pedestrians walking and eventually meeting, where our predictions successfully capture their gradual convergence even in sparse environments. **Scenario (b)** shows

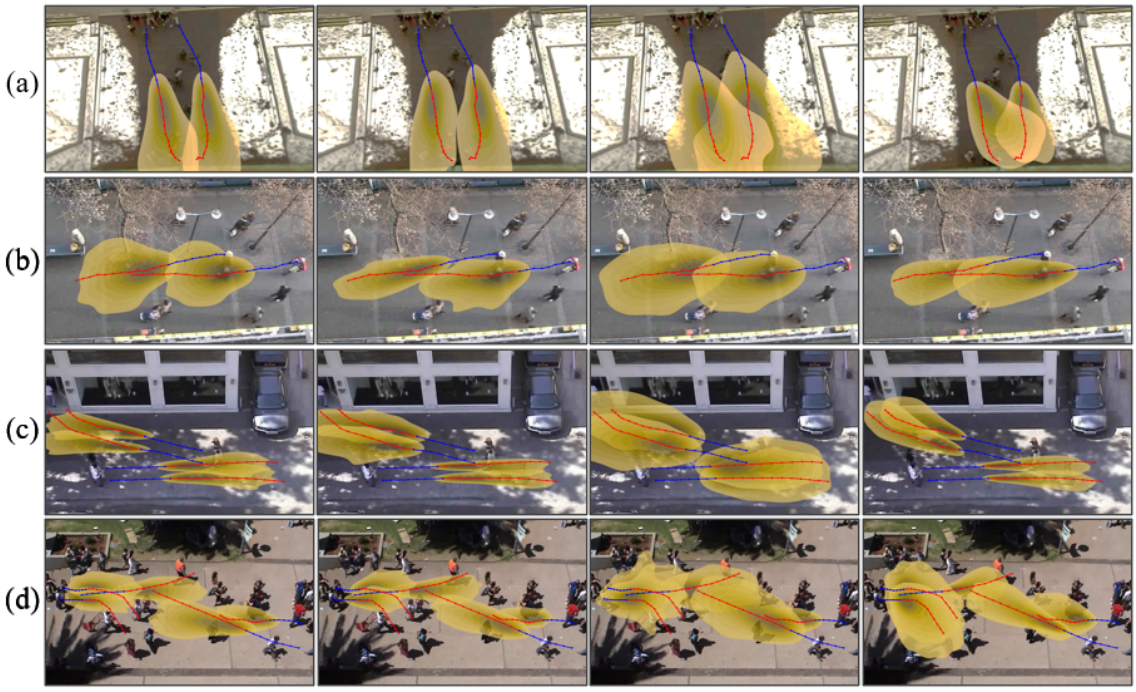


Figure 4.8: Visualization of predicted distributions on the ETH and UCY datasets. Historical trajectories are in blue, ground-truth trajectories are in red, and predicted trajectories are in yellow. Scenario (a) and (b) show two pedestrians walking in parallel with convergence; (c) presents two groups of pedestrians walking in opposing directions; (d) illustrates random walking behaviors.

pedestrians moving in parallel, where our approach achieves better alignment with ground-truth and avoids collisions compared to other methods. **Scenario (c)** presents two pedestrians meeting, where GP-Graph and EigenTrajectory fail to capture non-linear collision avoidance patterns. While Graph-TERN provides plausible predictions, our method better aligns with ground-truth by effectively modeling cross-time interactions. **Scenario (d)** presents a complex scenario in which several groups of pedestrians walk in opposing directions. In this case, GP-Graph and EigenTrajectory significantly suffer pedestrian collision issues. Our UniEdge demonstrates superior capability in capturing nonlinear movements, showcasing enhanced predictive accuracy in dynamically complex pedestrian interactions compared to previous methods. Finally, **scenario (e)** features complex non-linear trajectories with abrupt changes, where our method better captures overall movement trends despite shared challenges with certain trajectories.

Distribution Visualization Comparisons

In this section, we further compare the predicted distributions of UniEdge with GP-Graph [25], Graph-TERN [39] and EigenTrajectory [30] on the ETH and UCY datasets. As shown in Figure 4.5, our method generates more accurate and plausible distributions. In **scenario (a)**, while other methods' distributions cover the ground-truth, they fail to capture the pedestrian convergence trend that our method successfully predicts. In **scenarios (b)** and **(c)**, GP-Graph and Graph-TERN generate either too narrow or broad distributions, failing to capture non-linear trajectories. EigenTrajectory covers ground-truth but produces overly broad, overlapping distributions that lead to collision issues. Our method achieves comprehensive coverage with fewer collision predictions. In **scenario (d)** with random walking patterns, our approach better captures both non-linear and linear trajectories.

4.3.5 Ablation Study and Model Analysis

Model Component Analysis

To verify the influence of each module incorporated in our UniEdge, we conduct ablation studies on the ETH and UCY datasets, which contain five different social scenarios. The results of these studies are detailed in Table 4.3. In our experiments, variant (1) corresponds to the model excluding node-level embedding (NN), i.e., the model eliminates node-level GAT for capturing N2N interactions. Variant (2) represents the model without edge-level embedding (EE), meaning that edge information is not integrated into the model's architecture, neglecting implicit edge feature propagation. Lastly, variant (3) describes the modeling process without learning edge graphs through Hodge-Laplacian Laguerre Convolution (HC). Specifically, node-level embedding provides an overall picture of pedestrians' interaction intentions to capture initial N2N interactions, the overall performance dropped 11.1% in ADE and 24.0% in FDE without N2N interactions. Variant (2) shows that without the modeling of implicit E2E influence propagation, the performance dropped 16.7% in ADE and 20.0% in FDE. Variant (3) demonstrate the effectiveness of the proposed edge-level reasoning, without Hodge-Laplacian Laguerre Convolutions, the overall performance dropped 16.7% in ADE and 16.0% in FDE, respectively. Notably,

Table 4.3: Ablation Analysis of UniEdge on The ETH and UCY Datasets. NN = Node-level Embedding, EE = Edge-level Embedding, HC = Hodge-Laplacian Laguerre Convolution

Variant	NN	EE	HC	ADE(↓) / FDE(↓)					
				ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
(1)	×	✓	✓	0.40/0.63	0.13/0.20	0.22/0.32	0.15/0.23	0.12/0.19	0.20/0.31
(2)	✓	×	✓	0.39/0.54	0.14/0.18	0.23/0.35	0.16/0.24	0.13/0.19	0.21/0.30
(3)	✓	✓	×	0.39/0.47	0.12/0.18	0.24/0.38	0.17/0.22	0.14/0.18	0.21/0.29
Ours	✓	✓	✓	**0.36/0.46**	**0.11/0.17**	**0.19/0.28**	**0.14/0.20**	**0.11/0.16**	**0.18/0.25**

Table 4.4: Feature Embedding Analysis on The ETH and UCY Datasets

Method	ADE(↓) / FDE(↓)					
	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
w/ GCN [131]	0.39/0.57	0.15/0.19	0.22/0.34	0.17/0.25	0.13/0.18	0.21/0.31
w/ GraphSage [161]	0.38/0.52	0.12/0.19	0.21/0.30	**0.14**/0.22	0.12/0.17	0.19/0.28
Ours	**0.36/0.44**	**0.11/0.17**	**0.19/0.28**	**0.14/0.20**	**0.11/0.16**	**0.18/0.25**

the UNIV subset, which contains the most pedestrians and the most complex interactions [160], shows a decrease of 26.3% in ADE and 35.7% in FDE without edge graph learning, underscoring the importance of Hodge-Laplacian Laguerre convolution in managing the propagation of complex interactions. These findings underscore the importance of each module to the comprehensive functionality of our UniEdge model in trajectory prediction.

To investigate the effectiveness of different node embedding approaches in our framework, we evaluate several graph neural networks as alternatives to our GAT-based N2N module, as shown in Table 4.4. The baseline GCN [131] exhibits limited performance due to its uniform neighborhood aggregation strategy. GraphSage [161] achieves improved results through its sampling-based aggregation strategy. Compared to GCN and GraphSage, GAT-based approach demonstrates superior performance through its attention mechanism, which enables dynamic weighting of pedestrian interactions while providing better interpretability through attention weights.

Table 4.5: Edge Feature Analysis on The ETH and UCY Datasets

Edge Feature	ADE(↓) / FDE(↓)					
	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
Reciprocal distance	0.40/0.55	0.14/0.21	0.21/0.31	0.16/0.23	0.13/0.20	0.21/0.30
Gaussian Kernel	0.38/0.52	0.13/0.19	0.20/0.30	0.16/0.23	0.13/0.19	0.20/0.29
Ours	**0.36/0.46**	**0.11/0.17**	**0.19/0.28**	**0.14/0.20**	**0.11/0.16**	**0.18/0.25**

Table 4.6: Trajectory Predictor Analysis on The ETH and UCY Datasets. PE = Positional Encoding, Attn. Head = Attention Head, LN = Layer Normalization

Trajectory Predictor	ADE(↓) / FDE(↓)					
	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
w/o PE	0.45/0.51	0.13/0.19	0.29/0.42	0.20/0.28	0.16/0.22	0.25/0.32
w/o Attn. Head	0.37/0.47	0.12/0.19	0.23/0.35	0.17/0.24	0.13/0.19	0.20/0.29
w/o LN	0.38/0.47	0.13/0.18	0.21/0.31	0.15/0.23	0.13/0.18	0.20/0.27
Ours	**0.36/0.44**	**0.11/0.17**	**0.19/0.28**	**0.14/0.20**	**0.11/0.16**	**0.18/0.25**

Edge Feature Analysis

To assess the impact of edge features in our UniEdge model, we conduct experiments focusing on their incorporation into edge graphs. As detailed in Table 4.5, we examine three edge feature types: a Gaussian kernel $\mathcal{E}_{i,j} = \exp\left(-\frac{d_{i,j}}{2\sigma^2}\right)$, which captures spatial relationships through the distance $d_{i,j}$ between nodes i and j, and the standard deviation σ; a reciprocal distance kernel $\mathcal{E}_{i,j} = \frac{1}{d_{i,j}+\epsilon}$, highlighting inverse distance to represent pedestrian interactions; and a Euclidean distance kernel $\mathcal{E}_{i,j} = d_{i,j}$, quantifying node relationships based on direct distance. Results in Table 4.5 show that the Euclidean distance (ours) kernel outperforms other features on the ETH and UCY datasets. We think this is because the Euclidean distance kernel directly and accurately measures distances between pedestrians, providing a more intuitive representation of pedestrian interactions.

Trajectory Predictor Analysis

To evaluate the effectiveness of the core modules in our Transformer encoder-based predictor and the corresponding padding approaches, we conduct extensive experiments

Table 4.7: Trajectory Predictor Comparison Analysis on The ETH and UCY Datasets

Trajectory Predictor	ADE(↓) / FDE(↓)					
	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
RNN-based [83]	0.84/1.18	0.18/0.30	0.40/0.66	0.62/1.13	0.24/0.41	0.46/0.74
TCN-based [116]	**0.34**/0.48	0.13/0.19	0.25/0.35	0.17/0.26	0.14/0.19	0.21/0.29
Ours	0.36/**0.44**	**0.11**/**0.17**	**0.19**/**0.28**	**0.14**/**0.20**	**0.11**/**0.16**	**0.18**/**0.25**

on the predictor design. The results are presented in Table 4.6. We analyze three predictor variants: one without positional encoding (w/o PE), one without attention heads (w/o Attn. Head), and one without layer normalization (w/o LN). The experimental results demonstrate that the absence of any of these modules leads to degraded performance. Notably, the elimination of positional encoding has the most significant impact, resulting in performance degradation of 38.9% in ADE and 28.0% in FDE compared to the complete model. This substantial performance drop demonstrates the crucial role of positional encoding in preserving temporal ordering information of trajectory sequences, which is essential for understanding the temporal evolution of pedestrian motion patterns. Furthermore, the removal of attention heads leads to particularly inferior performance on the UNIV and ZARA1 subsets, which contain group activities with rich interactions, highlighting the importance of attention mechanisms in capturing temporal dependencies.

To evaluate the performance on different predictor architectures, we conduct experiments on the ETH and UCY datasets, as shown in Table 4.7. The RNN-based [83] predictor shows limited performance due to its constrained receptive field and auto-regressive nature. The TCN-based predictor [116] achieves strong performance on the ETH dataset due to its relatively large receptive field. However, its performance is limited on other datasets where temporal dependencies are more complex. Our Transformer Encoder-based predictor achieves superior performance by effectively capturing long-term dependencies through its non-local attention mechanism [67, 153].

Unified Spatial-temporal Graph Analysis

In this section, we analyze the effectiveness and impact of our proposed unified spatial-temporal graph data structure while keeping other components fixed. The construction

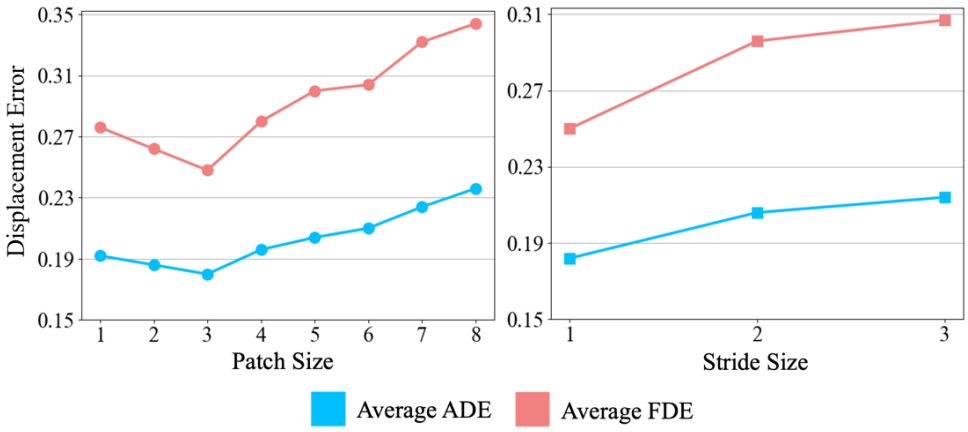


Figure 4.9: Impact analysis of unified spatial-temporal graph through patch size and stride size parameters on the ETH and UCY datasets.

of this data structure is controlled by two key parameters: patch size L and stride size $\mathcal{S}$. We conduct experiments on the ETH and UCY datasets to thoroughly analyze how these parameters affect the model's ability to capture spatial-temporal inter-dependencies.

As shown in Figure 4.9 (**left**), we evaluate how patch size affects unified spatial-temporal graph construction. A patch size of 1 reduces our model to traditional two-stage spatial-temporal approaches [27,29,30,38], where cross-time interactions are not explicitly modeled. The model achieves optimal performance with a patch size of 3, effectively capturing local spatial-temporal dependencies. Larger patch sizes, despite capturing more context information, may introduce redundant connections that degrade performance.

Second, we analyze the impact of stride size as shown in Figure 4.9 (**right**). The stride size determines the number of unified spatial-temporal graphs and the overlap between adjacent patches. A larger stride size reduces the overlap between patches during graph construction, which in turn decreases the total number of unified spatial-temporal graphs. A stride size of 1 yields the best performance in both ADE and FDE metrics, as it enables the capture of more fine-grained cross-time interactions through increased number of unified spatial-temporal graphs. The increased number of unified spatial-temporal graphs enables the transformer encoder-based predictor to leverage more spatial-temporal contexts for enhanced performance.

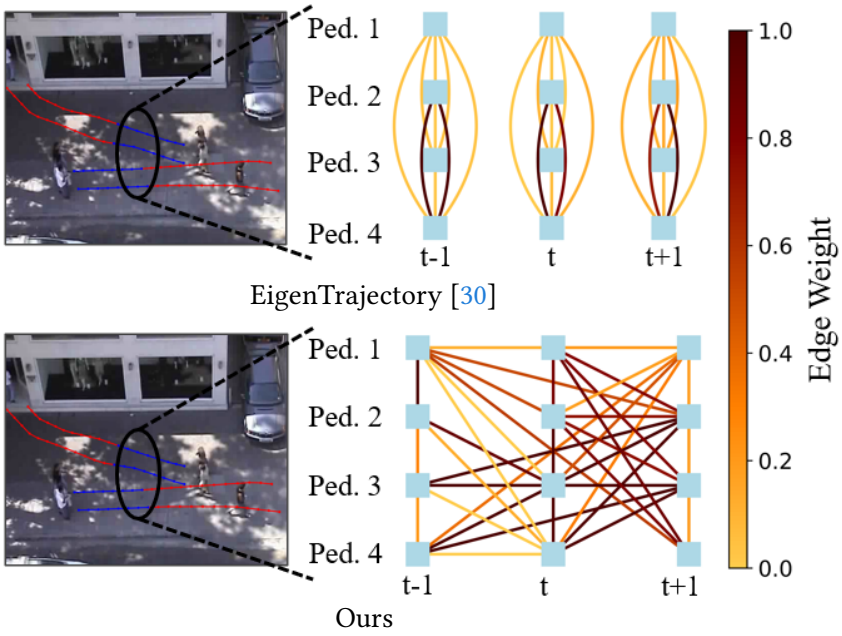


Figure 4.10: Edge weight visualization of traditional two-stage spatial-temporal approach EigenTrajectory and our UniEdge. Historical trajectories are in blue and ground-truth trajectories are in red.

Edge Weight Visualization

To provide qualitative insights into the differences between our UniEdge model and conventional spatial-temporal architecture, we visualize the edge weights of our unified spatial-temporal graph and EigenTrajectory [30]. Figure 4.10 illustrates a representative scenario where two groups of pedestrians approach each other across consecutive frames. While EigenTrajectory constructs independent spatial graphs for each frame, limiting its ability to capture high-order temporal dependencies, our unified spatial-temporal graph architecture explicitly models cross-temporal interactions across all three frames. The visualization demonstrates how our model captures extended temporal dynamics, revealing interaction patterns that conventional spatial-temporal frameworks may overlook.

Predictor Attention Weight Visualization

This section visualizes the attention weights of the Transformer encoder–based predictor to provide insight into how temporal information and relational cues are utilized during trajectory forecasting. In particular, we analyze how the model distributes attention between historical trajectory tokens and the learnable placeholder padding introduced to support unified spatial–temporal reasoning.

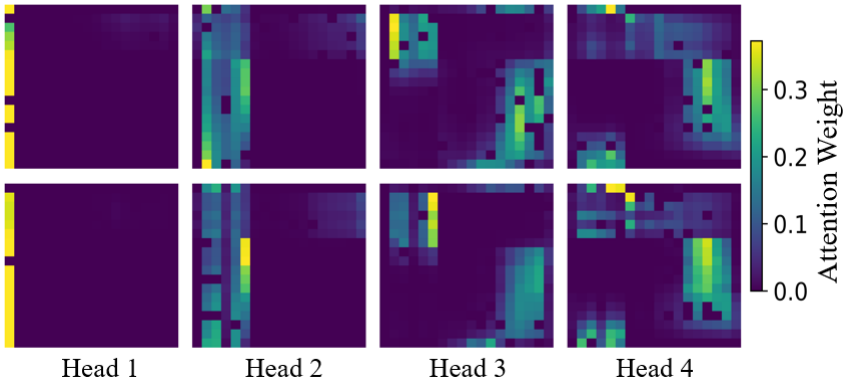


Figure 4.11: Predictor attention weight visualization. Four attention heads are configured in our experiments to analyze their impacts.

As shown in Figure 4.11, different attention heads exhibit clear and complementary specialization patterns. Heads 1 and 2 primarily attend to temporally adjacent historical states, indicating that they focus on modeling local temporal continuity and motion dynamics within observed trajectories. In contrast, Heads 3 and 4 assign higher attention weights to interactions between the learnable padding tokens and selected historical contexts, suggesting that these heads are responsible for aggregating global or cross-time relational information into the padding representations.

This behavior is consistent with the design motivation of UniEdge. Instead of uniformly mixing temporal and relational information, the model learns specialized attention pathways, where some heads focus on preserving temporal motion consistency while others use learnable padding tokens as relational anchors to aggregate salient interaction patterns across time. These attention distributions provide interpretable evidence that UniEdge effectively decouples and coordinates temporal encoding and relational reasoning within a unified Transformer framework.

Complexity and Efficiency Analysis

To evaluate the efficiency and computational complexity of UniEdge, Table 4.8 presents a comprehensive analysis of model complexity and computational efficiency among mainstream frameworks. We categorize the methods based on their temporal modeling paradigm into non-transformer and transformer-based temporal modeling methods. Compared to non-transformer temporal modeling methods such as EigenTrajectory [30], although UniEdge contains more parameters, it maintains competitive inference time

while achieving significant improvements in prediction accuracy (18.2% in ADE and 30.6% in FDE). For common real-world trajectory prediction scenarios such as traffic collision avoidance and anomaly detection, we believe this trade-off is justified as prediction accuracy takes precedence over computational complexity, especially since higher accuracy in these applications can significantly reduce the risk of severe outcomes. Compared to transformer-based temporal modeling methods like TUTR [69] and MRGTraj [158], UniEdge demonstrates superior efficiency with significantly lower parameters and FLOPs. Although TUTR achieves the fastest inference time, UniEdge maintains comparable computational speed while delivering substantially better prediction accuracy. Results demonstrate the effectiveness of our architecture in balancing computational efficiency and accuracy.

Table 4.8: Complexity and Inference Time Analysis. All Models Are Evaluated on NVIDIA RTX3080 GPU

Methods	Param $\times 10^6$	FLOPs (M)	Infer. Time (ms)	ADE(↓)/FDE(↓)
Non-Transformer Temporal Modeling				
Social-VAE [22]	2.15	292.95	40.27	**0.21**/**0.33**
Graph-TERN [39]	0.05	22.59	40.15	0.24/0.38
EqMotion [26]	3.02	7.75	35.92	**0.21**/0.35
EigenTrajectory [30]	**0.02**	**1.36**	**22.26**	0.22/0.36
Transformer-based Temporal Modeling				
TUTR [69]	0.44	64.54	**20.21**	0.21/0.36
MRGTraj [158]	4.35	580.38	26.51	0.26/0.46
UniEdge (Ours)	**0.34**	**26.49**	27.02	**0.18**/**0.25**

4.3.6 Discussion

In this section, we discuss potential reasons for the relatively lower performance of graph-based trajectory prediction approaches [30, 31, 39, 66] on the ETH subset, as compared to other scenarios. As indicated in Table 4.9, the test set for the ETH subset averages only 2.59 pedestrians per sample, significantly less than other subsets, particularly the UNIV subset, which averages 25.70 pedestrians per sample. This stark variation in pedestrian density impacts the efficacy of graph-based methods, which rely on graph structures to model social interactions [27, 154]. The relatively sparse graph connectivity in the

Table 4.9: Dataset Statistics on The ETH and UCY Datasets

Dataset	ETH	HOTEL	UNIV	ZARA1	ZARA2
Total Test Samples	70	301	947	602	921
Avg. Pedestrians	2.59	3.50	25.70	3.74	6.33

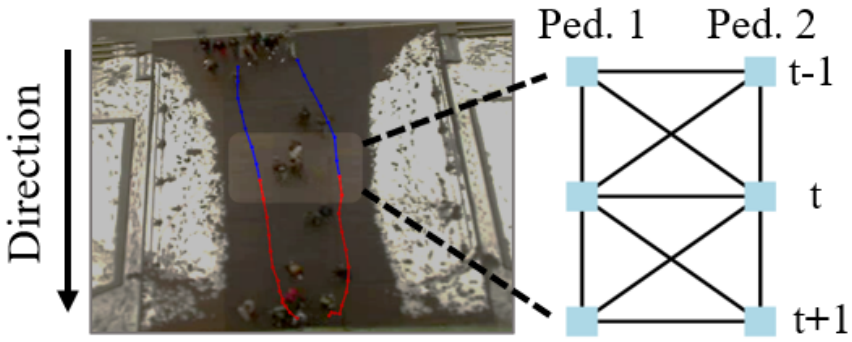


Figure 4.12: Sample scenario in ETH dataset. Historical trajectories are in blue, ground-truth trajectories are in red.

ETH scenario may impair message passing, potentially limiting the model's ability to effectively propagate and refine contextual information across nodes, which could hinder accurate representation of complex social interactions of graph-based approaches. In contrast, UniEdge demonstrates enhanced performance in scenarios with dense social interactions (HOTEL, UNIV, ZARA1, and ZARA2) by effectively capturing the more intricate social dynamics.

To further illustrate these challenges, we visualize a representative case from the ETH dataset in Figure 4.12. The example shows how UniEdge constructs a unified spatial-temporal graph between Ped.1 and Ped.2, even though their trajectories are relatively stable with minimal interaction, potentially introducing unnecessary modeling bias. Additionally, while the scene contains multiple pedestrians, only a few trajectories are annotated, hindering the model's ability to capture comprehensive interaction patterns. To address these challenges, one promising direction is to develop dynamic graph optimization strategies [162] that adapt connectivity based on scene characteristics. Such adaptive approaches would reduce redundant connections in sparse scenarios while preserving rich interaction modeling in dense scenarios, improving the prediction performance.

Additionally, we identify several promising directions to enhance our model's performance and adaptability. **First**, we aim to refine the model with an adaptive patch

segmentation technique that dynamically adjusts patch sizes based on scene complexity metrics such as pedestrian density and interaction frequency [163], addressing the limitations of our current fixed patch size strategy and potentially improving prediction accuracy in varying crowd scenarios. **Second**, we plan to incorporate multimodal data sources, particularly environmental contextual images [138, 164], to enhance our model's awareness of physical constraints and scene semantics, enabling more precise predictions in complex urban environments while reducing prediction errors caused by environmental factors. **Finally**, we will explore hardware optimization strategies for the transformer architecture [165, 166] to improve deployment efficiency in real-time applications, reducing computation latency while maintaining prediction accuracy.

4.4 Summary

This chapter presented **UniEdge**, a unified spatial–temporal graph framework designed to address key limitations in homogeneous pedestrian trajectory prediction. To capture complex, high-order cross-time interactions among agents, we introduced a patch-based unified spatial–temporal graph structure that transforms high-order dependencies into simplified first-order relationships. This design improves message propagation efficiency and alleviates under-reaching by reducing reliance on multi-step aggregation. To jointly capture individual motion patterns and collective influence dynamics, we proposed a dual-graph convolutional architecture—Edge-to-Edge and Node-to-Node Graph Convolution (E2E–N2N–GCN)—that reasons over both node-level social interactions and edge-level propagation patterns, enriching the representation of implicit behavioral influences. A Transformer-based trajectory predictor was further incorporated to model global temporal dependencies, enhancing the ability to forecast long-range behaviors.

Together, these components form a unified and flexible framework for modeling homogeneous pedestrian environments with improved accuracy and social awareness. Building on the insights gained here, the next chapter expands the scope to both homogeneous and heterogeneous settings. In particular, we introduce BP-SGCN, a behavioral pseudo-label informed sparse graph convolutional network that discovers latent motion patterns in an unsupervised manner. By leveraging these learned behavioral representations, BP-SGCN

enhances spatial–temporal interaction modeling and improves generalization across diverse traffic scenarios without requiring manual annotations.

CHAPTER 5

Unsupervised Behavior Structure Learning for Generalizable Trajectory Prediction

Portions of this chapter have previously been published in the following peer-reviewed publication [19]:

- **Li, R.**, Katsigiannis, S., Kim, T.-K., & Shum, H. P. H., "BP-SGCN: Behavioral Pseudo-Label Informed Sparse Graph Convolution Network for Pedestrian and Heterogeneous Trajectory Prediction." *IEEE Transactions on Neural Networks and Learning Systems* (TNNLS), 2025.

Trajectory prediction allows better decision-making in applications of autonomous vehicles or surveillance by predicting the short-term future movement of traffic agents. It is classified into pedestrian or heterogeneous trajectory prediction. The former exploits the relatively consistent behavior of pedestrians, but is limited in real-world scenarios with heterogeneous traffic agents such as cyclists and vehicles. The latter typically relies on extra class label information to distinguish the heterogeneous agents, but such labels are costly to annotate and cannot be generalized to represent different behaviors within

the same class of agents. In this chapter, we introduce the behavioral pseudo-labels that effectively capture the behavior distributions of pedestrians and heterogeneous agents solely based on their motion features, significantly improving the accuracy of trajectory prediction. To implement the framework, we propose the Behavioral Pseudo-Label Informed Sparse Graph Convolution Network (BP-SGCN) that learns pseudo-labels and informs to a trajectory predictor. For optimization, we propose a cascaded training scheme, in which we first learn the pseudo-labels in an unsupervised manner, and then perform end-to-end fine-tuning on the labels in the direction of increasing the trajectory prediction accuracy. Experiments show that our pseudo-labels effectively model different behavior clusters and improve trajectory prediction. Our proposed BP-SGCN outperforms existing methods using both pedestrian (ETH/UCY, homogeneous pedestrian SDD) and heterogeneous agent datasets (SDD, Argoverse 1).

5.1 Introduction

Predicting the future movement of traffic agents, known as trajectory prediction, is crucial for safe and efficient decision-making in applications such as autonomous vehicles [10]. Thanks to reliable data-driven [167] object tracking methods [168], accurate geometric trajectories can be extracted from videos, serving as a more representative feature set for modeling. Graph Convolutional Networks (GCNs) [131] have shown exceptional performance across diverse fields due to their adeptness at capturing spatial relationships [169–173]. This enables them to excel in applications ranging from trajectory agent interaction modeling [27, 29, 38, 154] to human skeleton-based behavior modeling [174–178], highlighting the superior capabilities in handling graph-based data structures. Similarly, recognizing distinct movement behavior patterns among agents is pivotal to model the temporal dependency [123]. These patterns, when integrated with GCN, further enhance the precision of predictions by accounting for the inherent behavioral tendencies.

Existing trajectory prediction methods can be broadly classified into two categories. The first focuses on predicting *pedestrian trajectories* in datasets that are exclusively composed of pedestrians [1, 73] or deliberately omit non-pedestrian traffic agents [4, 5, 179].

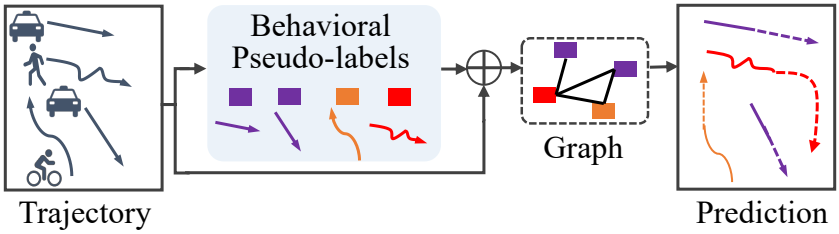


Figure 5.1: We propose the behavioral pseudo-labels learned from observed trajectories, effectively representing inter- and intra-type behavioral differences to improve pedestrian and heterogeneous trajectory prediction accuracy.

These methods primarily employ neural networks to account for pedestrian social interactions, such as the pooling window mechanism [28] and social interaction graphs [27,29,38]. The second category encompasses *heterogeneous trajectory* prediction, considering a diverse range of traffic agents (e.g. cars, cyclists, pedestrians, etc.). Recent methods [6,7,34] exploit the annotated class labels of traffic agents to better model agent interactions in intricate urban scenarios. These labels facilitate the system's understanding on multifaceted interactions among various agent types [7].

A notable research gap can be observed between homogeneous and heterogeneous trajectory prediction. Methods tailored solely for pedestrian behavior excel due to its predictable patterns but lack applicability in real-world scenarios like autonomous driving, since pedestrians behave very differently from heterogeneous agents [6,7]. The fundamental differences in modeling the motion patterns of different types of agents stem from their distinct dynamics, speed ranges, spatial needs, interaction behaviors, decision-making processes, and ways of perceiving the environment, necessitating varied modeling approaches to accurately predict their trajectories. For heterogeneous trajectory prediction, ground-truth (GT) labels for agent types have traditionally been used to guide discriminative learning [6,7,34,42]. However, these labels often fail to capture diverse within-class behaviors: for example, 'vans' and 'compact cars' are both labeled simply as 'cars,' while 'pedestrians' can range from ordinary walkers to skateboarders [74]. This granularity issue can lead to mislabeling, especially when visually similar categories are grouped together. Moreover, obtaining such detailed GT labels is time-consuming and expensive. We argue that purely relying on manual labels is both insufficient and cost-ineffective for representing the nuanced motion patterns seen in real-world traffic

scenarios.

In this paper, we present a unified framework utilizing machine-learned behavioral pseudo-labels applicable to both heterogeneous and exclusively pedestrian domains. Our insight is that behavioral pseudo-labels can capture both inter-class and intra-class behavioral variations among agents, thereby improving the accuracy of our model. For heterogeneous scenarios, the use of behavioral pseudo-labels eliminates the need for manual label annotations, streamlining the process and reducing the reliance on extensive labeled datasets. In homogeneous pedestrian scenarios, these pseudo-labels facilitate the differentiation and learning of intrinsic motion patterns among pedestrians, offering a more nuanced understanding of pedestrian behavior. A shared advantage across both contexts is the significant improvement in overall prediction performance, demonstrating the versatility and efficacy of behavioral pseudo-labels in diverse trajectory prediction tasks (Figure 5.1).

We propose the Behavioral Pseudo-Label informed Sparse Graph Convolution Network (BP-SGCN) for pedestrian and heterogeneous trajectory prediction. The network includes two modules. First, we introduce a deep unsupervised behavior clustering module that assigns pseudo-labels to agents based on their observed trajectories. This module marks a novel application of deep embedded clustering [125], utilizing high-level temporal latent features. It is supported by a Variational Recurrent Neural Network (VRNN) [180] that processes a set of customized geometric features, crucial for capturing motion dynamics such as speed, angle, and acceleration. Additionally, a soft dynamic time warping loss addresses temporal variances in trajectories, uniquely tailoring our approach for trajectory modeling. The generated behavioral pseudo-labels are specifically designed to enhance trajectory forecasting, highlighting our model's focus on the nuanced demands of trajectory prediction in complex environments. Second, we propose a goal-guided pseudo-label informed trajectory prediction module, which adapts SGCN [27], a powerful GCN backbone for trajectory prediction that utilizes a sparse spatial-temporal attention mechanism to effectively model spatial interactions and temporal dependencies of agents. We then employ a Gumbel-Softmax straight-through estimator to link up the clustering module, allowing the prediction module and clustering module to be fine-tuned in an end-to-end manner. Finally, we design a cascaded training

scheme [181] that first trains pseudo-label clustering in an unsupervised manner, and then fine-tunes both clustering and trajectory prediction together with the prediction loss to maximize their compatibility.

BP-SGCN surpasses SOTAs in both heterogeneous prediction on the SDD [74] and Argoverse 1 [92] datasets, and in pedestrian prediction on the ETH/UCY [1, 73] dataset and the homogeneous pedestrian setup of SDD [182]. Our source code is available at https://github.com/Carrotsniper/BP-SGCN to facilitate further research. Our contributions are:

- We propose the novel concept of behavioral pseudo-labels to represent clusters of traffic agents with different movement behaviors, improving trajectory prediction without the need for any extra annotation.
- To implement the idea, we propose BP-SGCN, which introduces a cascaded training scheme to optimize the compatibility of its two core modules: the pseudo-label clustering module and the trajectory prediction module.
- We propose a deep unsupervised behavior clustering module to obtain behavioral pseudo-labels, tailoring the geometric feature representation and the loss to best learn the agents' behaviors.
- We propose a pseudo-label informed goal-guided trajectory prediction module, which facilitates end-to-end fine-tuning with its prediction loss for better clustering and prediction, outperforming existing pedestrian and heterogeneous prediction methods.

5.2 Behavior Pseudo-Label Informed Sparse Graph Convolution Network

5.2.1 The High-Level Network Architecture

We observe a research gap in pedestrian and heterogeneous trajectory prediction. Existing pedestrian prediction approaches have limited applicability to heterogeneous traffic agents due to the diverse behaviors of agents. For instance, in Figure 5.2, (a) and (c) depict

intricate heterogeneous scenarios with bikers and cars exhibiting longer, non-linear paths, while homogeneous pedestrian scenarios (b) and (d) overlook interactions among pedestrians, bikers and cars.

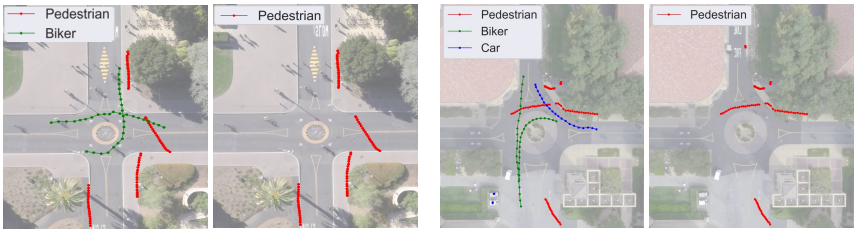


Figure 5.2: Trajectory visualization on heterogeneous SSD dataset, where red, green and blue dots represent pedestrians, bikers and cars, respectively. (a) and (c) represent heterogeneous scenarios with all agent types, (b) and (d) represent the homogeneous pedestrian scenarios commonly used by pedestrian trajectory predictions [4, 5] by simply removing all non-pedestrian agents.

Although introducing annotated class labels for heterogeneous agents leads to better prediction performance [6, 7, 34], such labels are only a proxy of movement behaviors, which cannot represent intra-class behavioral differences and inter-class behavioral similarity.

To this end, we present the concept of *behavioral pseudo-labels*, which capture movement behaviors to enhance trajectory prediction. Our pseudo-labels do not require annotations, mitigating the risk of mislabeling and reducing labor costs. It can be applied to both homogeneous pedestrian and heterogeneous datasets, resulting in superior prediction performance.

To realize pseudo-label informed trajectory prediction, we propose the *Behavioral Pseudo-Label Informed Sparse Graph Convolution Network* (BP-SGCN). As shown in Figure 5.3, BP-SGCN includes two modules: deep unsupervised clustering and pseudo-label informed trajectory prediction. The former learns the pseudo-labels in an unsupervised manner, while the latter performs end-to-end optimization to improve pseudo-label clustering while predicting trajectories with such labels.

We propose a *cascaded training scheme* to obtain the pseudo-labels and thus high-quality trajectory prediction. First, highlighted with the orange dotted block in Figure 5.3, the unsupervised behavior representation learning module derives behavior latent representations from observed trajectories through a Variational Recurrent Neural Network

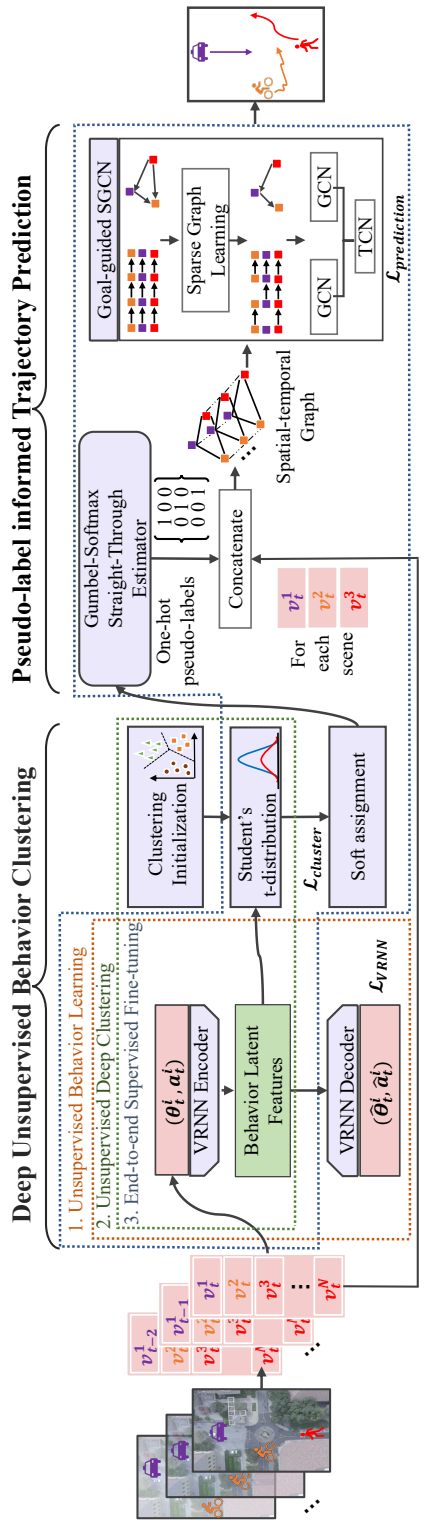


Figure 5.3: The overview of BP-SGCN to learn the pseudo-labels for trajectory prediction, consisting of the deep unsupervised clustering module and the pseudo-label informed trajectory prediction module. We propose a cascaded optimization scheme to first learn pseudo-labels in an unsupervised manner, and then fine-tune them in an end-to-end manner with trajectory prediction supervision.

(VRNN) [180] module. Then, in the green dotted block, the behavior latent representations are fed into simple clustering modules (e.g., K-means, GMM, etc.) for cluster center initialization. We then perform unsupervised deep clustering to learn the distribution of pseudo-labels by feeding the VRNN latent representations to the Student's t-distribution kernel [183]. This allows fine-tuning the VRNN encoder to create a better latent space and refine the cluster centers. Finally, indicated by the blue dotted block, we utilize a Gumbel-Softmax straight-through estimator to sample one-hot pseudo-labels, which are concatenated to the trajectory features as the input of goal-guided SGCN [27] for trajectory prediction. The whole network is optimized end-to-end, fine-tuning the pseudo-label clustering module to maximize its compatibility for trajectory prediction.

5.2.2 Deep Unsupervised Behavior Clustering

Here, we explain how we obtain behavior clusters, which serve as powerful features for effective trajectory prediction.

Geometric Representation of Trajectories

Given a series of observed video frames of N agents over time $t \in [1, T_{obs}]$, and the corresponding 2-D trajectory coordinates (x_t^i, y_t^i), $i \in [1, N]$, our objective is to predict the future trajectory coordinates $p_t^i = (x_t^i, y_t^i)$ of each traffic agent i within a time horizon $t \in [T_{obs+1}, T_{pred}]$.

We introduce relative angle and acceleration magnitude to learn behavior latents. While global velocity is an effective feature for trajectory prediction [7, 27], it is less representative of behaviors, as it depends on global movement directions, and is less sensitive to velocity changes. Relative angles provide a representation that is invariant to the initial facing direction, which is complemented with the magnitude of acceleration that has been shown to be effective for modeling behaviors [26].

For each traffic agent *i*, we calculate its velocity vector at time t. For simplicity, we remove the notation *i* in the following equation:

$$\boldsymbol{v}_t = \left(\frac{x_t - x_{t-1}}{t - (t-1)}, \frac{y_t - y_{t-1}}{t - (t-1)} \right), \tag{5.1}$$

where $\forall t \in [1, T_{obs}]$, we compute the cosine of the angle, $\cos(\theta_t)$ between velocity vectors, $\boldsymbol{v}_t$ and $\boldsymbol{v}_{t-1}$:

$$\cos(\theta_t) = \frac{\boldsymbol{v}_t \cdot \boldsymbol{v}_{t-1}}{|\boldsymbol{v}_t| \cdot |\boldsymbol{v}_{t-1}|}, \tag{5.2}$$

and the magnitude of corresponding acceleration at time t:

$$|\boldsymbol{a}_t| = \left| \frac{\boldsymbol{v}_t - \boldsymbol{v}_{t-1}}{t - (t-1)} \right| \tag{5.3}$$

The geometric feature is constructed as $g_t = (\cos(\theta_t), |\boldsymbol{a}_t|)$. These motion primitives offer informative inductive cues that are difficult to reliably disentangle from noise through latent learning alone, while still leaving higher-order temporal representations.

Behavior Representation Learning

We adapt VRNN to learn latent representations for behavior clustering [125, 184]. VRNN learns the temporal dependencies of a sequence by modeling the distribution over its hidden states with an encoder-decoder architecture. Compared to LSTM-based autoencoders [184], it effectively models the highly nonlinear dynamics and captures the uncertainties of latent space. Its probabilistic nature of variational inference improves the learning of implicit sequential data distributions.

In particular, the encoder network $\varphi_{enc}(\cdot, \cdot)$ receives the embedded geometric data $\varphi^g(g_t)$ and recurrent hidden state h_{t-1} to approximate the posterior distribution $q_\phi(\cdot)$:

$$\begin{aligned} q_\phi(z_t | g_{\leq t}, z_{<t}) &= \mathcal{N}(z_t | (\mu_{z,t}, \sigma^2_{z,t})), \\ [\mu_{z,t}, \sigma_{z,t}] &= \varphi_{enc}(\varphi^g(g_t), h_{t-1}), \end{aligned} \tag{5.4}$$

where z_t is sampled using a reparameterization trick [185]. The decoder network $\varphi_{dec}(\cdot, \cdot)$ takes the embedded latent $\varphi^z(z_t)$ and h_{t-1} to approximate the reconstruction distribution $p_\delta(\cdot)$:

$$\begin{aligned} p_\delta(g_t | z_{\leq t}, g_{<t}) &= \mathcal{N}(g_t | (\mu_{g,t}, \sigma^2_{g,t})), \\ [\mu_{g,t}, \sigma_{g,t}] &= \varphi_{dec}(\varphi^z(z_t), h_{t-1}). \end{aligned} \tag{5.5}$$

To enhance the temporal dependencies in sequences, the prior distribution in VRNN

relies on h_{t-1} with $\varphi_{prior}(\cdot)$:

$$\begin{aligned} p_\delta(z_t|z_{<t}, g_{<t}) &= \mathcal{N}(z_t|(\mu_{0,t}, \sigma_{0,t}^2)), \\ [\mu_{0,t}, \sigma_{0,t}] &= \varphi_{prior}(h_{t-1}). \end{aligned} \tag{5.6}$$

We employ the Gated Recurrent Unit (GRU) [110] to update the RNN hidden state, which outperforms LSTM [83] when the sequence length is relatively short:

$$h_t = GRU(\varphi^g(g_t), \varphi^z(z_t), h_{t-1}). \tag{5.7}$$

The VRNN is optimized with a customized loss:

$$\mathcal{L}_{VRNN} = \mathcal{L}_{Soft\text{-}DTW} + \mathcal{L}_{ELBO}, \tag{5.8}$$

where $\mathcal{L}_{Soft\text{-}DTW}$ is a differentiable soft Dynamic Time Warping (DTW) loss [186]:

$$\mathcal{L}_{Soft\text{-}DTW} = \min_{\mu_{g,t}} \sum_{i=1}^{N} \frac{1}{T_{obs}} DTW_\gamma(\mu_{g,t}, g_t), \tag{5.9}$$

DTW_γ refers to the original DTW [187] discrepancy that measures and aligns the similarity between two time series, γ is a parameter indicating the acceptable distortion for aligning two sequences, $\mu_{g,t}$ is the decoded mean of the VRNN decoder. The loss allows capturing non-linear temporal alignment [188], which cannot be achieved with MSE. $\mathcal{L}_{\text{ELBO}}$ is the variational evidence lower-bound with the Kullback–Leibler (KL) divergence [180, 185]:

$$\begin{aligned} \mathcal{L}_{ELBO} = \mathbb{E}_{q_\phi(z_{\leq T_{obs}}|g_{\leq T_{obs}})} \Bigg[\sum_{t=1}^{T_{obs}} (\log p_\delta(g_t|z_{\leq t}, g_{<t}) \\ - KL(q_\phi(z_t|g_{\leq t}, z_{<t}) \,||\, p_\delta(z_t|z_{<t}, g_{<t})) \Bigg]. \end{aligned} \tag{5.10}$$

By optimizing $\mathcal{L}_{VRNN}$, the model aligns predicted and observed sequences while maintaining a theoretically grounded variational framework. This alignment enhances flexibility in handling non-linear temporal dynamics, and the KL regularization constrains the latent structure, thus ensuring stable training. Consequently, the VRNN encoder provides

richer latent representations for subsequent unsupervised deep clustering, effectively leveraging spatial-temporal structures to capture nuanced agent behaviors.

Deep Embedded Clustering

We present a new application of Deep Embedded Clustering (DEC) [125] to cluster the agent behaviors latents from the VRNN encoder, thereby generating a distribution of pseudo-labels. DEC allows jointly optimizing the cluster centers and the VRNN encoder, enhancing the latent representation via back-propagation. This significantly outperforms traditional methods like k-means [189] and Gaussian mixture models [190], which lack the capability to refine input feature representations.

The initial phase of DEC involves setting cluster centers using VRNN behavior latents. We input all training data into the VRNN encoder to obtain the set of behavior latent features $\mathbb{Z}$, and then apply k-means to determine initial centers, $c_j \in [1, k]$. Given the variance in agent behaviors across datasets, k is an empirically tuned hyperparameter.

We then apply Student's T-Distribution [183], that is, Q distribution to compute the soft assignment between each initialized cluster center and latent vector [125]. Its kernel measures the probability of each encoded vector $z_i \in \mathbb{Z}$ belonging to the cluster j:

$$q_{ij} = \frac{\left(1 + \frac{d(z_i, c_j)}{\alpha}\right)^{-\frac{\alpha+1}{2}}}{\sum_{j'} \left(1 + \frac{d(z_i, c_{j'})}{\alpha}\right)^{-\frac{\alpha+1}{2}}}, \tag{5.11}$$

where d is a similarity metric that refers to the distance between the encoded vector z_i and center c_j, and α is the number of degrees of freedom of the Q distribution. We denote d as the Euclidean distance and set α to 1.

Meanwhile, we optimize the clustering network with a KL divergence loss to minimize the discrepancy between the two distributions:

$$\mathcal{L}_{\text{cluster}} = KL(P||Q) = \sum_i \sum_j \left(p_{ij} \log \frac{p_{ij}}{q_{ij}}\right), \tag{5.12}$$

where P is the auxiliary distribution:

$$p_{ij} = \frac{q_{ij}^2 / f_j}{\sum_{j'} q_{ij'}^2 / f_{j'}}, \tag{5.13}$$

and $f_j = \sum_i q_{ij}$ are soft cluster frequencies. Here, P is re-weighted from Q distribution in a way that sharpens high-confidence assignments and de-emphasizes low-confidence ones [125], thereby systematically increasing the separation between clusters in the latent space. Finally, we derive soft assignments from the Student's t-distribution, reflecting the probability of the latent z_t^i in each cluster c_j. This approach not only offers greater flexibility in representing complex behaviors but also sharpens cluster boundaries by reinforcing high-confidence assignments and reducing ambiguity in low-confidence ones. Consequently, it yields more coherent clusters and better captures the inherent diversity in agent dynamics, ultimately enhancing the overall clustering quality.

5.2.3 Pseudo-label Informed Trajectory Prediction

Here, we introduce the concept of behavioral pseudo-labels for more accurate trajectory prediction.

Gumbel-Softmax Straight-Through Estimator

While the soft assignment represents good behavior clusters, such clusters are unsupervised and trained only on feature representations, meaning that they are still sub-optimal for any given task. This explains the sub-optimal prediction accuracy in existing methods [123, 124]. Here, we present a framework to improve the compatibility between the clusters and the task via fine-tuning the behavior latent.

To enable end-to-end fine-tuning of the behavior latent with a task objective, an operator is needed to connect the clustering and the prediction modules. We employ the Gumbel-Softmax straight-through estimator [191], which facilitates the gradient propagation and computes one-hot vectors representing the pseudo-labels. The estimator uses a differentiable Softmax, as opposed to the non-differentiable Argmax, allowing end-to-end optimization. An agent's class label is l_j, where $j \in 1, \dots k$ is the cluster center.

Apart from performance gains, as one-hot labels fit the human understanding of a class concept, they allow better interpretability via visualization tools. They are also immediately compatible with existing network architectures trained with ground-truth labels [6, 7], allowing effective adaptations.

Behavioral Pseudo-Label Informed SGCN

We adopt a Sparse Graph Convolution Network (SGCN) [27] as our backbone and introduce the pseudo-labels and a new loss function. SGCN has shown outstanding performance and is computationally efficient on pedestrian trajectory prediction [27]. It introduces sparsified spatial-temporal attention mechanism [38,67,192], which effectively models spatial interactions and temporal dependencies among agents. The sparse graph learning component removes spatial superfluous interactions and temporal motion tendencies, improving both computational speed and accuracy. In reality, our pseudo-label framework is compatible with a wide range of trajectory prediction networks.

We introduce the usage of semantic-goal features into SGCN, which enhances the prediction accuracy [5, 179]. To this end, we integrate the goal-retrieval operation [179] into the SGCN, we first subtract each observation step $\boldsymbol{v}_t$ in $t \in [1, T_{obs}]$ by the corresponding trajectory endpoint $\boldsymbol{v}_{T_{pred}}$ as $\boldsymbol{v}_t = \boldsymbol{v}_t - \boldsymbol{v}_{T_{pred}}$. We then construct the spatial graph $\mathcal{G}_s = \left\{(\mathcal{V}_s, \mathcal{A}_s) | \mathcal{V}_s \in \mathbb{R}^{T_{obs} \times N \times D_s}, \mathcal{A}_s \in \mathbb{R}^{T_{obs} \times N \times N}\right\}$, where $\mathcal{V}_s$ represents the spatial interactions among all agents at time step t, $\mathcal{A}_s$ is the spatial adjacency matrix and D_s refers to the spatial feature dimension.

To add heterogeneity to the graph, we concatenate the pseudo-labels l to the trajectory feature vector for each agent at each time step as $\mathcal{V}_t^i = \mathit{concat}(v_t^i, l^i)$, $\forall t \in [1, T_{obs}]$ and $\forall i \in [1, N]$. Similarly, we establish the
temporal graph $\mathcal{G}_t = \left\{(\mathcal{V}_t, \mathcal{A}_t) | \mathcal{V}_t \in \mathbb{R}^{N \times T_{obs} \times D_t}, \mathcal{A}_t \in \mathbb{R}^{N \times T_{obs} \times T_{obs}}\right\}$ to represent the temporal correlations of each individual agent during T_{obs} steps, where $\mathcal{A}_t$ is the temporal adjacency matrix and D_t is the temporal feature dimension. Finally, these spatial and temporal goal-guided heterogeneous graphs are passed into SGCN for final trajectory prediction.

We propose a joint training strategy with a novel loss function to jointly optimize trajectory prediction and pseudo-label clustering. Thanks to our Gumbel-Softmax estimator, back-propagation is performed from the prediction all the way back to the VRNN encoder, resulting in better compatibility between the clustering and prediction modules. We present a combined loss:

$$\mathcal{L}_{final} = \mathcal{L}_{cluster} + \mathcal{L}_{prediction}, \tag{5.14}$$

where $\mathcal{L}_{cluster}$ is defined in Eq. 5.12, and $\mathcal{L}_{prediction}$ as:

$$\mathcal{L}_{prediction} = -\sum_{t=T_{obs}+1}^{T_{pred}} \log P(p_t|\hat{\mu}, \hat{\sigma}, \hat{\rho}), \tag{5.15}$$

where $\hat{\mu}$ and $\hat{\sigma}$ are the mean and variance of the bi-variate Gaussian distribution of trajectory prediction, and $\hat{\rho}$ represents the correlation coefficient.

5.3 Experiments

5.3.1 Datasets

We evaluate BP-SGCN on multiple benchmark datasets, including the Stanford Drone Dataset (SDD) [74], Argoverse 1 [92], ETH [1] and UCY [73], and the homogeneous pedestrian version of SDD [182]. For pedestrian trajectory prediction, ETH/UCY consists of five homogeneous pedestrian datasets (ETH, HOTEL, UNIV, ZARA1, ZARA2) with 1,536 pedestrians. homogeneous pedestrian SDD is the simplified version where non-pedestrian agents are removed. For heterogeneous trajectory prediction, we follow [7, 193, 194] that consider all trajectories, consisting of 8 scenes, 60 videos and 6 categories of traffic agents (i.e., pedestrians, bicyclists, skateboarders, carts, cars, and buses). Argoverse 1 consists of over 30K urban traffic scenarios that include 3 types of agents (i.e. AVs, agents, and others).

5.3.2 Experimental Setup

By default, we follow the experimental setup of [27, 154], using 3.2 seconds (8 frames) of observation trajectories to predict the next 4.8 seconds (12 frames). For homogeneous pedestrian prediction, we employ the data augmentation approach introduced in [154] and the official leave-one-out strategy [21] during the training and validation. For heterogeneous trajectory prediction on Argoverse 1 dataset, our experimental setup and dataset split strategy follow [34, 35]. Specifically, we utilize 2 seconds (20 frames) of observation trajectories to predict the trajectories of all tracked objects over the subsequent 3 seconds (30 frames) within each scene. In particular, our experimental setup on the Argoverse 1 dataset for heterogeneous trajectory prediction predicts trajectories

for all agents [34, 195], unlike methods focusing on a single agent [196] or two specific agents [197], our approach captures multi-agent interactions, reflecting real-world traffic complexity and improving predictive robustness, situational awareness, and adaptability to diverse urban environments.

During testing, we adhere to the standard protocol by generating 20 predictions for both heterogeneous [6, 7, 35] and homogeneous pedestrian trajectory predictions [21, 27, 198]. This approach ensures our results are comparable to those established in the field. The sample with the lowest error is then used to compute the evaluation metrics. We employ the Average Displacement Error (ADE) and Final Displacement Error (FDE) [21, 27–29] as our evaluation metrics:

$$\begin{aligned} ADE &= \frac{1}{(T_{pred} - T_{obs}) \times N} \sum_{i=1}^{N} \sum_{t=T_{obs}+1}^{T_{pred}} ||\hat{p}_t^i - p_t^i||_2, \\ FDE &= \frac{1}{N} \sum_{i=1}^{N} ||\hat{p}_t^i - p_t^i||_2, t = T_{pred}, \end{aligned} \tag{5.16}$$

where $\hat{p}_t^i$ represents the ground-truth trajectory coordinates. Table 5.1 summarizes the primary notations and their definitions used throughout the BP-SGCN framework.

5.3.3 Quantitative Evaluation

Heterogeneous Prediction

Table 5.2 compares our BP-SGCN with previous state-of-the-art methods on heterogeneous SDD. These methods can be categorized into three groups based on the input features, including trajectory-only [25, 28, 38, 41, 42], trajectory with ground-truth labels [6, 7, 33, 34, 42, 91], and trajectory with extra scene features such as scene semantics [40, 193, 194, 199–201]. BP-SGCN outperforms all the methods that utilize ground-truth agent class labels [6, 7, 33, 34, 42, 91]. Compared to the best method VNAGT [91], BP-SGCN demonstrates the superiority by reducing ADE/FDE by 28.23%/44.43%. Crucially, for SOTA approaches that incorporate scene semantic features such as V^2-Net [200] and TDOR [193], our BP-SGCN improves the performance by reducing ADE/FDE by 2.5%/15.9% compared to V^2-Net and 19.3%/31.2% compared to TDOR. The results indicate that without the need for additional inputs, our BP-SGCN can still achieve SOTA

Table 5.1: A summary of main symbols and definitions

Symbols	Definition
$p_t^i = (x_t^i, y_t^i)$	2D coordinates of agent i at time t.
N	Number of pedestrians
T_{obs}	Observed time steps
T_{pred}	Prediction time steps
$\boldsymbol{v}_t$	Velocity vector
$\cos(\theta_t)$	Cosine of the angle
$\lvert\boldsymbol{a}_t\rvert$	Magnitude of acceleration
g_t	Agent geometric feature
φ_{enc}	Encoder network of VRNN
h	Recurrent hidden state
q_ϕ	Posterior distribution
φ_{dec}	Decoder network of VRNN
p_δ	Reconstruction distribution
φ_{prior}	Prior distribution
$\mathcal{L}_{VRNN}$	Loss of VRNN
$\mathcal{L}_{Soft\text{-}DTW}$	Loss of soft-DTW
$\mathcal{L}_{ELBO}$	Loss of ELBO
q	Q Distribution for soft assignment
p	Auxiliary distribution P
$\mathcal{L}_{cluster}$	Loss of deep clustering
f	Soft cluster frequency
l	Agent class label
$\mathcal{G}_s$	Spatial graph
$\mathcal{V}_s$	Node of $\mathcal{G}_s$
$\mathcal{A}_s$	Adjacency matrix of $\mathcal{G}_s$
$\mathcal{G}_t$	Temporal graph
$\mathcal{V}_t$	Node of $\mathcal{G}_t$
$\mathcal{A}_t$	Adjacency matrix of $\mathcal{G}_t$
$\mathcal{L}_{prediction}$	Loss of bi-variate Gaussian distribution
$\mathcal{L}_{final}$	Combined loss

performance in heterogeneous trajectory prediction.

Table 5.3 compares the BP-SGCN with those state-of-the-art methods in heterogeneous trajectory prediction on Argoverse 1, following the setup in [34, 35, 195]. Results show that our BP-SGCN outperforms all the methods by a significant margin, especially in the ADE metric. BP-SGCN surpasses NLNI [34], which integrates ground-truth labels, by reducing 12.7% in ADE and 8.7% in FDE, further showcasing the effectiveness of our proposed pseudo-label module. Notably, although NLNI utilizes label-based category features, its performance is limited by the simplistic nature of the “GT Labels" in the Argoverse 1 dataset, which are broadly classified as “1 AV" (1 Autonomous Vehicle), “1 Focal" (the primary vehicle whose trajectory is predicted), and “N other" (other tracked objects,

Table 5.2: Results on SDD for heterogeneous prediction.

Methods	Venue	Year	GT Labels	SDD	
				ADE(↓)	FDE(↓)
Social-LSTM [28]	CVPR	2016	No	31.19	56.97
DESIRE [40]	CVPR	2017	No	19.25	34.05
MATF [81]	CVPR	2019	No	22.59	33.53
STGAT [38]	ICCV	2019	No	18.80	31.30
Multiverse [194]	CVPR	2020	No	14.78	27.09
SimAug [199]	ECCV	2020	No	10.27	19.71
NLNI [34]	ICCV	2021	Yes	15.90	26.30
STSF-Net [41]	TMM	2021	No	14.81	28.03
Semantic-STGCNN [6]	SMC	2021	Yes	18.12	29.70
V^2-Net [200]	ECCV	2022	No	<u>7.12</u>	<u>11.39</u>
Multiclass-SGCN [7]	ICIP	2022	Yes	14.36	25.99
TDOR [193]	CVPR	2022	No	8.60	13.90
CAPHA [201]	TVT	2023	No	9.13	14.34
VNAGT [91]	TVT	2023	Yes	9.67	17.22
SFEM-GCN [42]	TIV	2024	Yes	15.31	25.72
SMGCN [33]	IJCAI	2024	Yes	20.89	36.84
BP-SGCN (Ours)			No	**6.94**	**9.57**

Table 5.3: Results on Argoverse 1 for heterogeneous prediction.

Methods	Venue	Year	GT Labels	Argoverse 1	
				ADE(↓)	FDE(↓)
Social-LSTM [28]	CVPR	2016	No	1.39	2.57
DESIRE [40]	CVPR	2017	No	0.90	1.45
R2P2-MA [202]	ECCV	2018	No	1.11	1.77
MATFG [81]	CVPR	2019	No	1.26	2.31
CAM [85]	ECCV	2020	No	1.13	2.50
MFP [203]	NeurIPs	2020	No	1.40	2.68
Social-STGCNN [29]	CVPR	2020	No	1.31	2.34
NLNI [34]	ICCV	2021	Yes	0.79	1.26
DD [204]	Inf. Sci.	2022	No	<u>0.74</u>	1.28
HRG+HSG [35]	TITS	2023	No	0.85	**1.12**
BIP-Tree [195]	TITS	2023	No	0.78	1.35
BP-SGCN (Ours)			No	**0.69**	<u>1.15</u>

which can include vehicles, pedestrians, or bicycles). This coarse categorization restricts the algorithm's ability to accurately capture and analyze the nuanced interactions among diverse traffic agents. In contrast, BP-SGCN effectively overcomes these constraints by

Table 5.4: Results on ETH/UCY on homogeneous pedestrian prediction;
- denotes missing result due to unavailability from original authors.

Method	Venue	Year	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
			ADE(↓)/FDE(↓)	ADE(↓)/FDE(↓)	ADE(↓)/FDE(↓)	ADE(↓)/FDE(↓)	ADE(↓)/FDE(↓)	ADE(↓)/FDE(↓)
Social LSTM [28]	CVPR	2016	1.09/2.35	0.79/1.76	0.67/1.40	0.47/1.00	0.56/1.17	0.72/1.54
Social GAN [21]	CVPR	2018	0.87/1.62	0.67/1.37	0.76/1.52	0.35/0.68	0.42/0.84	0.61/1.21
Social-STGCNN [29]	CVPR	2020	0.64/1.11	0.49/0.85	0.44/0.79	0.34/0.53	0.30/0.48	0.44/0.75
PECNet [4]	ECCV	2020	0.54/0.87	0.18/0.24	0.35/0.60	0.22/0.39	0.17/0.30	0.29/0.48
SGCN [27]	CVPR	2021	0.63/1.03	0.32/0.55	0.37/0.70	0.29/0.53	0.25/0.45	0.37/0.65
AgentFormer [68]	ICCV	2021	0.45/0.75	0.14/0.22	0.25/0.45	0.18/0.30	0.14/0.24	0.23/0.39
PCCSNet [123]	ICCV	2021	0.28/0.54	0.11/0.19	0.29/0.60	0.21/0.44	0.15/0.34	0.21/0.42
ExpertTraj [1][179]	ICCV	2021	0.37/0.65	0.11/0.15	0.20/0.44	0.15/0.31	0.12/0.25	0.19/0.36
STSF-Net [41]	TMM	2021	0.63/1.13	0.24/0.43	0.28/0.52	0.23/0.45	0.21/0.41	0.32/0.59
Social-Implicit [154]	ECCV	2022	0.66/1.44	0.20/0.36	0.31/0.60	0.25/0.50	0.22/0.43	0.33/0.67
GP-Graph [25]	ECCV	2022	0.43/0.63	0.18/0.30	0.24/0.42	0.17/0.31	0.15/0.29	0.23/0.39
Social-VAE [22]	ECCV	2022	0.41/0.58	0.13/0.19	0.21/0.36	0.17/0.29	0.13/0.22	0.21/0.33
MemoNet [155]	CVPR	2022	0.40/0.61	0.11/0.17	0.24/0.43	0.18/0.32	0.14/0.24	0.21/0.35
GroupNet [63]	CVPR	2022	0.46/0.73	0.15/0.25	0.26/0.49	0.21/0.39	0.17/0.33	0.25/0.44
MID [71]	CVPR	2022	0.39/0.66	0.13/0.22	0.22/0.45	0.17/0.30	0.13/0.27	0.21/0.38
GTPPO [98]	TNNLS	2022	0.63/0.98	0.19/0.30	0.35/0.60	0.20/0.32	0.18/0.31	0.31/0.50
Graph-TERN [39]	AAAI	2023	0.42/0.58	0.14/0.23	0.26/0.45	0.21/0.37	0.17/0.29	0.24/0.88
MSRL [151]	AAAI	2023	0.28/0.47	0.14/0.22	0.24/0.43	0.17/0.30	0.14/0.23	0.19/0.33
LED [72]	CVPR	2023	0.39/0.58	0.11/0.17	0.26/0.43	0.18/0.26	0.13/0.22	0.21/0.33
EqMotion [26]	CVPR	2023	0.40/0.61	0.12/0.18	0.23/0.43	0.18/0.32	0.13/0.23	0.21/0.35
FEND [124]	CVPR	2023	-	-	-	-	-	0.17/0.32
EigenTrajectory [30]	ICCV	2023	0.36/0.53	0.12/0.19	0.24/0.43	0.19/0.33	0.14/0.24	0.21/0.34
TUTR [69]	ICCV	2023	0.40/0.61	0.11/0.18	0.23/0.42	0.18/0.34	0.13/0.25	0.21/0.36
SICNet [205]	ICCV	2023	**0.27**/**0.45**	0.11/0.16	0.26/0.46	0.19/0.33	0.13/0.26	0.19/0.33
TP-EGT [206]	TITS	2023	0.41/0.68	0.13/0.21	0.29/0.50	0.18/0.30	0.16/0.27	0.23/0.39
DynGroupNet [104]	NN	2023	0.42/0.66	0.13/0.20	0.24/0.44	0.19/0.34	0.15/0.28	0.23/0.38
SMEMO [156]	TPAMI	2024	0.39/0.59	0.14/0.20	0.23/0.41	0.19/0.32	0.15/0.26	0.22/0.35
STGlow [23]	TNNLS	2024	0.31/0.49	**0.09**/**0.14**	**0.16**/0.33	**0.12**/0.24	**0.09**/0.19	**0.15**/0.28
MRGTraj [158]	TCSVT	2024	0.28/0.47	0.21/0.39	0.33/0.60	0.24/0.44	0.22/0.41	0.26/0.46
HighGraph [31]	CVPR	2024	0.40/0.55	0.13/0.17	0.20/0.33	0.17/0.27	0.11/0.21	0.20/0.30
PPT [207]	ECCV	2024	0.36/0.51	0.11/0.15	0.22/0.40	0.17/0.30	0.12/0.21	0.20/0.31
BP-SGCN (Ours)			0.33/0.47	0.10/**0.14**	0.17/**0.26**	0.13/**0.19**	0.10/**0.16**	0.17/**0.24**

[1] For ExpertTraj [179], the discrepancy from the original paper arises due to an error highlighted by the authors: https://github.com/JoeHEZHAO/expert_traj

Table 5.5: Results on the homogeneous pedestrian version of SDD.

Methods	Venue	Year	SDD-human	
			ADE(↓)	FDE(↓)
STGAT [38]	ICCV	2019	0.58	1.11
Social-Ways [208]	CVPRW	2019	0.62	1.16
DAG-Net [198]	ICPR	2020	0.53	1.04
Social-implicit [154]	ECCV	2022	0.47	0.89
WTGCN [209]	IJMLC	2024	0.43	0.72
IGGCN [210]	DSP	2024	0.44	0.71
BP-SGCN (Ours)			**0.28**	**0.41**

conducting a comprehensive analysis of the behavior dynamics of all agents within the scene. By employing our advanced pseudo-label module, we significantly enhance the representational capabilities of our system, leading to markedly improved prediction

accuracy across diverse traffic scenarios. This improvement is achieved without the need for direct matching with ground-truth labels, demonstrating the robustness and adaptability of our approach in interpreting complex interactive behaviors. Importantly, DD [204] and HRG+HSG [35] achieve comparable performance on ADE and FDE mainly due to the use of scene images that better capture the interactions between traffic agents and environments, our BP-SGCN still shows the best ADE performance compared to these methods.

Pedestrian Prediction

For ETH/UCY, we conduct quantitative comparisons with a wide range of methods with various techniques, as shown in Table 5.4. Following [26, 72], we compare with methods utilizing trajectory data only.

For distribution-based methods, Social-LSTM [28] introduces bi-variate Gaussian distribution to sample predictions from the trained mean and variance, which is widely used in recently published methods [25, 27, 29, 66, 104, 179]. Following this, our BP-SGCN also uses the bi-variate Gaussian distribution to represent the distribution parameters of the predicted trajectories. It outperforms almost all methods under this setting by a significant margin. In addition, both ExpertTraj [179] and our BP-SGCN utilize the goal-retrieve mechanism but we have a significant improvement of 10.5% in ADE and 33.3% in FDE.

For generative-based methods, Social-GAN [21] is the pioneer method that introduces GANs [60] to generate trajectories with special pooling modules. PECNet [4] utilizes the CVAEs [211] to generate trajectories conditioned on the pre-sampled goal points, which add an extra constraint to the predicted trajectories for better accuracy. Methods like [22, 30, 31, 63, 68, 151] follow the CVAEs basis to train the encoder with ground-truth trajectories for better latent representations. MID [71] and LED [72] further introduce the diffusion models [212] to enhance training and reduce mode collapses. Results reveal that our BP-SGCN outperforms generative-based methods.

For transformer-based methods, TUTR [69] proposes a novel global prediction system incorporated with a motion-level transformer encoder and a social-level transformer decoder for accurate trajectory representation. MRGTraj [158] introduces a

non-autoregressive enhanced transformer decoder for trajectory prediction. PPT [207] proposes multi-stage transformer progressively modeling trajectories. STGlow [23] further introduces the flow-based generative framework with dual-graphormer to precisely model motion distributions. Compared to STGlow, our BP-SGCN achieves comparable ADE with STGlow, while reducing FDE by 14%.

Besides these categories, LSTM decoder-based methods [41, 98, 123, 206] and [124] directly predict trajectories using LSTM decoder, which also show comparable results to the transformer-based methods. Social-implicit [154] introduces the concept of implicit maximum likelihood estimation mechanism. Memonet [155], SICNet [205] and SMEMO [156] incorporate memory bank/module concepts into the system, demonstrating considerable performance. Notably, SICNet presents the best results on ETH subset in both ADE and FDE metrics compared with all other methods. Graph-TERN [39] shows a novel trajectory refinement module that first samples the endpoint and then linearly interpolates the predictions. EqMotion [26] further introduces the concepts of invariance and equivariance into trajectory prediction to learn motion patterns. Nevertheless, results in Table 5.4 illustrate that our BP-SGCN outperforms all of these methods.

For homogeneous pedestrian SDD, Table 5.5 highlights the comparative performance of our BP-SGCN, which secures substantial improvements over all listed models, including the latest STOA models, WTGCN [209] and IGGCN [210]. Specifically, BP-SGCN achieves a 35% reduction in ADE compared to WTGCN and a 42% reduction in FDE compared to IGGCN. The results in both heterogeneous SDD and homogeneous pedestrian SDD show the superiority of our BP-SGCN in multiple scenarios.

5.3.4 Qualitative Evaluation

Figure 5.4 presents a t-SNE [183] visualization of the latent representations and their corresponding pseudo-classes during unsupervised deep clustering on SDD ($k = 6$). These clusters do not correspond to ground-truth semantic labels, but instead reflect behavior-driven groupings learned from motion patterns. At epoch 0, cluster centers are initialized using k-means, and the latent representations are not yet structured, resulting in overlapping and ambiguous clusters. As training progresses, the VRNN encoder learns more discriminative behavioral representations, leading to increasingly compact and

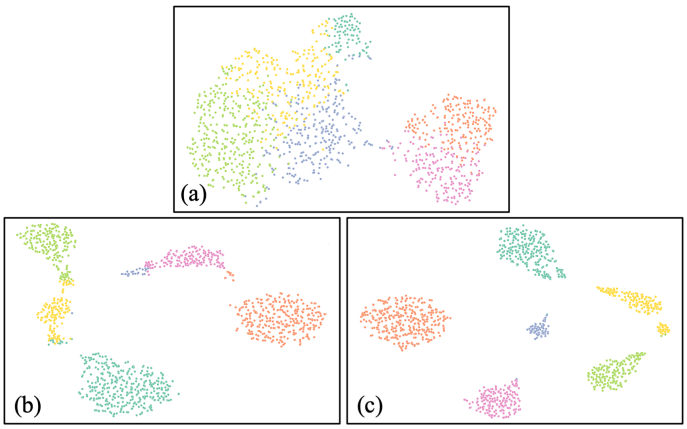


Figure 5.4: The t-SNE visualization of pseudo-class clustering on SDD (k=6) during unsupervised deep clustering. (a) 0 epochs (initialized by k-means), (b) 200 epochs, (c) 800 epochs.

well-separated clusters. This visualization primarily serves to illustrate the improvement in clustering quality and representation consistency over training, rather than to indicate predefined or interpretable semantic categories.

Figure 5.5 and Figure 5.6 visualize the trajectory predictions for the SDD and ETH/UCY datasets, respectively. Blue and red dots represent observed and ground-truth future trajectories, respectively. For the SDD dataset, we visualize the predictions in Figure 5.5, where light blue indicates the predicted distributions and yellow dots represent the predicted single trajectory. The visualizations demonstrate that our BP-SGCN exhibits superior performance compared to methods integrating ground-truth labels [6,7] in three challenging scenarios characterized by complex social interactions among agents.

In Figure 5.6, we visualize the predicted distribution in the ETH/UCY datasets across various scenarios, encompassing both simple and complex interactions, and compare our method with SGCN [27] and GP-Graph [25]. We visualize the parameterized distribution of future trajectories, as they are the learning objective of these methods. Qualitative comparisons reveal that our predicted distributions closely align with the ground truth and adeptly capture the non-linear trajectories. Specifically, scenario (a) illustrates a scene with numerous pedestrians on the street engaging in complex interactions, such as meeting, colliding, and standing still. While all the predicted distributions can accurately represent linear trajectories, both SGCN and GP-Graph falter in predicting the movements of pedestrians exhibiting non-linear behaviors. In contrast, BP-SGCN consistently

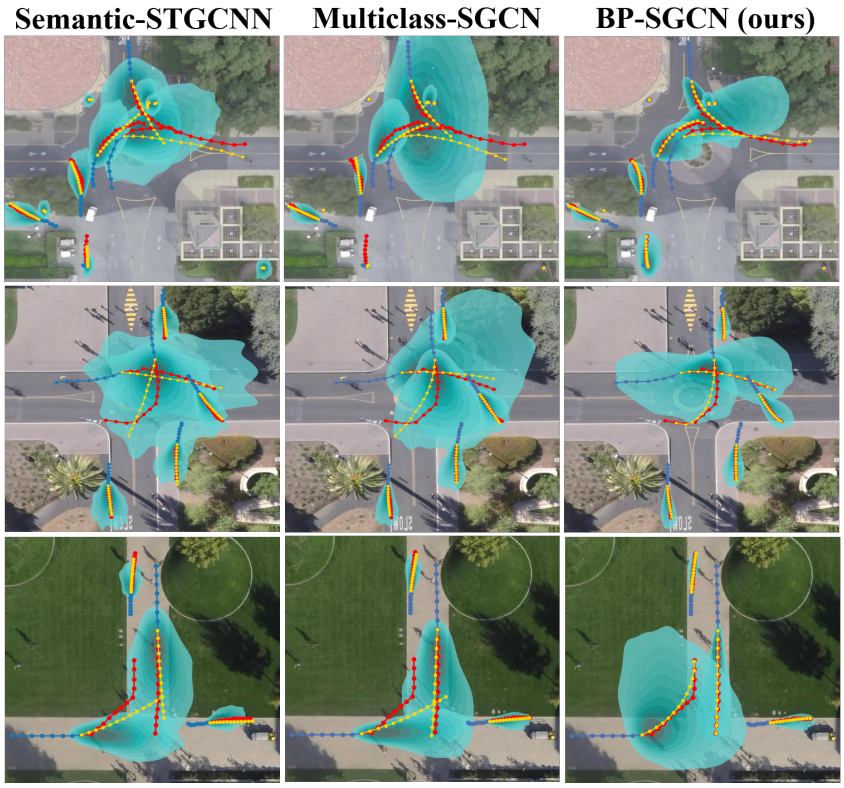


Figure 5.5: Visualization of trajectory prediction on SDD of Semantic-STGCNN [6], Multiclass-SGCN [7], and BP-SGCN (ours). Blue and red represent observed and ground-truth trajectories respectively, yellow represents the predicted trajectory and light-blue shade represents the predicted distribution.

generates plausible predictions. Scenario (b) displays four stationary pedestrians; however, both SGCN and GP-Graph yield wrong predictions, whereas BP-SGCN accurately captures the static behaviors. In scenario (c), the predicted distributions from both SGCN and GP-Graph demonstrate significant overlaps, leading to a heightened risk of predicted collisions. On the other hand, BP-SGCN's predictions show reduced overlaps. In scenario (d), while GP-Graph continues to display overlap issues, SGCN exhibits overconfidence in its predictions, resulting in a lack of diversity and a propensity to deviate from the ground truth. BP-SGCN effectively addresses both of these challenges, striking a balance between prediction accuracy and diversity.

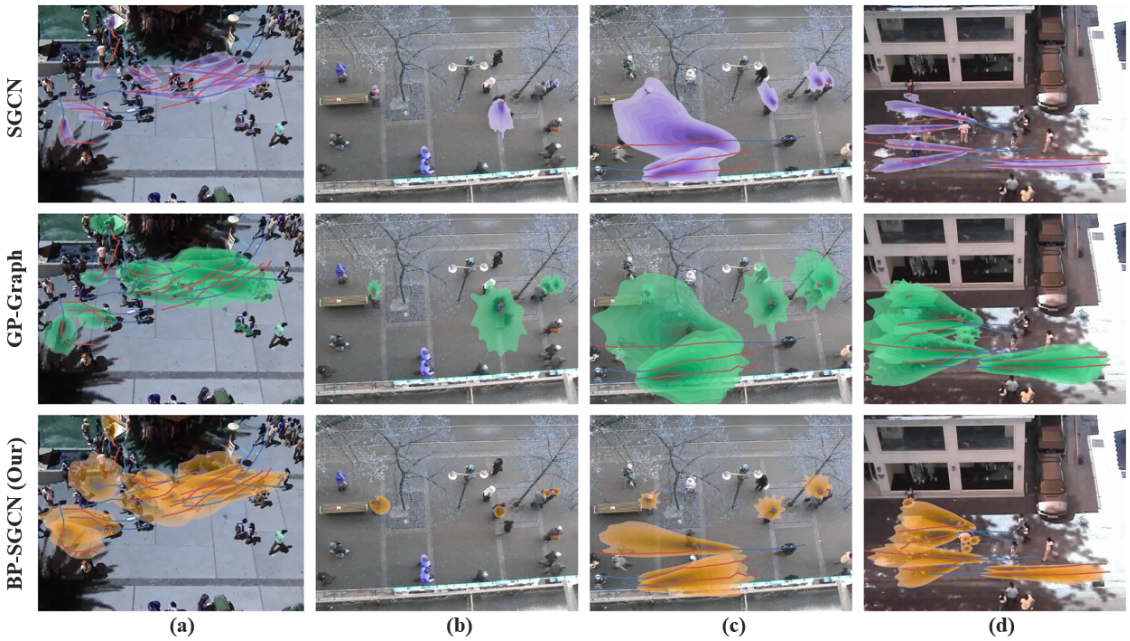


Figure 5.6: Visualization of the trajectory prediction on ETH/UCY in the scenario of pedestrian walking behaviors. Past trajectories are shown in blue, and ground-truth trajectories are in red. (a) shows the pedestrians in a crowded scenario with complex interactions. (b) shows the scene where four pedestrians are almost static. (c) and (d) show scenes including multiple pedestrian behaviors, such as walking, meeting, and standing.

5.3.5 Ablation Study and Parameter Analysis

Cluster Number Analysis

The effects of cluster number on heterogeneous datasets are shown in Table 5.6 (Heterogeneous SDD) and Table 5.7 (Argoverse 1). The results on homogeneous pedestrian datasets are shown in Table 5.8 (ETH/UCY) and Table 5.9 (homogeneous pedestrian SDD). In general, the cluster number depends on the diversity of behaviors, which is strongly correlated with the location. For instance, choosing six clusters for SDD is reasonable given the presence of six types of agents, and this choice yields good performance. Tuning the cluster number for a scene provides extra improvements, and this only has to be done once. These results further reflect that the heterogeneous dataset is more sensitive to the cluster numbers and the pedestrian dataset results exhibit diminished sensitivity, attributable to the inherent behavioral homogeneity and comparatively lower variance observed in human actions. Note that due to its large data size, for Argoverse 1, we run

Table 5.6: Cluster number analysis on heterogeneous SDD.

Clusters	ADE(↓)	FDE(↓)
1	7.26	10.03
3	7.11	9.81
6	**6.94**	**9.57**
9	7.03	9.74
12	7.58	10.92

Table 5.7: Cluster number analysis on Argoverse 1.

Clusters	ADE(↓)	FDE(↓)
1	0.86	1.63
3	0.80	1.45
6	**0.69**	**1.15**
9	0.79	1.47

ablation studies and parameter analysis using a partial dataset in a simplified setup.

Notably, in our experiments on cluster numbers, a cluster's number equal to 1 denotes that there is no pseudo-label applied on each agent because all the agents are considered to belong to the same class, and consequently the model performance relies solely on the trajectories themselves. In particular, results in Table 5.8 and Table 5.9 demonstrate that, within datasets exclusively comprising pedestrian agents, our BP-SGCN model is adept at discerning the nuanced variances in their movement patterns. Despite the apparent homogeneity of the agents as pedestrians, our analysis reveals intrinsic behavioral differentiations that our model capitalizes on to significantly improve prediction accuracy. This not only underscores the importance of individualized learning even among seemingly similar entities, but also showcases the efficacy of our model in enhancing predictive outcomes by leveraging these subtle distinctions.

Network Components Analysis

Table 5.10 shows ablation studies to evidence the effectiveness of network components used in BP-SGCN on heterogeneous and homogeneous pedestrian SDD. The "No Deep Clustering" setup uses k-means cluster centers directly for trajectory prediction, and therefore does not implement unsupervised deep learning and end-to-end fine-tuning.

Table 5.8: Cluster number analysis on ETH/UCY.

Clusters	ADE(↓) / FDE(↓)				
	ETH	HOTEL	UNIV	ZARA1	ZARA2
1	0.37/0.51	0.14/0.19	0.27/0.37	0.15/0.21	0.24/0.34
2	0.37/0.52	0.15/0.21	0.18/0.27	0.20/0.37	0.25/0.35
3	0.45/0.61	**0.10**/**0.14**	0.27/0.36	0.24/0.33	0.17/0.34
4	**0.33**/**0.47**	0.12/0.16	0.27/0.37	0.14/0.20	**0.10**/**0.16**
5	0.36/0.50	0.17/0.22	0.18/0.27	**0.13**/**0.19**	0.12/0.18
6	0.39/0.53	0.15/0.21	0.18/0.27	0.15/0.21	0.11/0.17
7	0.37/0.51	0.11/0.15	**0.17**/**0.26**	0.26/0.37	0.13/0.19

Table 5.9: Cluster number analysis on homogeneous pedestrian SDD.

Clusters	ADE(↓)	FDE(↓)
1	0.33	0.49
3	**0.28**	**0.41**
6	0.47	0.72
9	0.31	0.47

The "No Gumbel-Softmax" setup directly concatenates the soft assignment to the trajectory features for trajectory prediction. The "No End-to-End Training" setup uses only $L_{prediction}$ to optimize the trajectory prediction module but not the deep clustering module; here, the Gumbel-Softmax estimator is substituted with the non-differentiable Argmax function. Results from both the heterogeneous and pedestrian datasets emphasize the significance of all the proposed components in BP-SGCN.

In addition, our proposed Goal-Guided SGCN module utilizes the spatial attention and temporal attention mechanism to enhance the final prediction accuracy. We conduct experiments on ETH/UCY datasets to validate the effectiveness of these two modules. The results shown in Table 5.11 indicate that both spatial attention and temporal attention modules are important for the best performance.

Table 5.10: Network components analysis on heterogeneous SDD (upper) and homogeneous pedestrian SDD (lower).

Method	ADE(↓)	FDE(↓)
BP-SGCN (No Deep Clustering)	7.52	10.50
BP-SGCN (No Gumbel-Softmax)	7.65	10.85
BP-SGCN (No End-to-End Training)	10.82	15.32
BP-SGCN (Ours)	**6.94**	**9.57**

Method	ADE(↓)	FDE(↓)
BP-SGCN (No Deep Clustering)	0.30	0.44
BP-SGCN (No Gumbel-Softmax)	0.40	0.60
BP-SGCN (No End-to-End Training)	0.30	0.46
BP-SGCN (Ours)	**0.28**	**0.41**

Table 5.11: Prediction module analysis on ETH/UCY datasets.

Method	ADE(↓)	FDE(↓)
BP-SGCN (No Spatial Attention)	0.25	0.30
BP-SGCN (No Temporal Attention)	0.28	0.35
BP-SGCN (Ours)	**0.17**	**0.24**

Trajectory Prediction Loss Analysis

As discussed above, we propose a cascaded training strategy with a novel loss function to jointly optimize trajectory prediction and pseudo-label clustering, defined as:

$$\mathcal{L}_{final} = \mathcal{L}_{prediction} + \mathcal{L}_{cluster}. \tag{5.17}$$

In the proposed loss function, $\mathcal{L}_{prediction}$ and $\mathcal{L}_{cluster}$ contribute equally to the final loss $\mathcal{L}_{final}$. We conduct an ablation study by introducing a weighted sum of losses with a new hyperparameter λ to explore the effect and contribution of the two losses on trajectory prediction on both heterogeneous and homogeneous pedestrian SDD datasets:

$$\mathcal{L}_{final} = \lambda\mathcal{L}_{prediction} + (1-\lambda)\mathcal{L}_{cluster}. \tag{5.18}$$

Here, we analyze the effect of λ. For the proposed BP-SGCN, the default value of λ

Table 5.12: Loss weight analysis between $\mathcal{L}_{prediction}$ and $\mathcal{L}_{cluster}$ on heterogeneous SDD (left) and homogeneous pedestrian SDD (right).

Method	ADE(↓)	FDE(↓)
BP-SGCN ($\lambda = 0.25$)	19.33	24.26
BP-SGCN ($\lambda = 0.75$)	7.08	9.84
BP-SGCN (Ours)	**6.94**	**9.57**

Method	ADE(↓)	FDE(↓)
BP-SGCN ($\lambda = 0.25$)	0.46	0.70
BP-SGCN ($\lambda = 0.75$)	0.31	0.46
BP-SGCN (Ours)	**0.28**	**0.41**

can be considered as 0.5, as both losses contribute equally to the final loss. We further adjust the value of λ as 0.25, and 0.75, respectively. The experimental results presented in Table 5.12 show that the performance of BP-SGCN reaches its peak when the ratio of $\mathcal{L}_{prediction}$ and $\mathcal{L}_{cluster}$ is equal, as presented in the main paper, which further indicates that the trajectory prediction and pseudo-label clustering modules are equally important for the overall trajectory prediction performance.

Clustering Features Analysis

Finally, Table 5.13 shows ablation studies on heterogeneous and homogeneous pedestrian SDD datasets with regard to the geometric features used for behavior clustering. These features play a pivotal role, enabling our unsupervised deep clustering module to differentiate agent behaviors effectively. The outcomes highlight the outstanding performance of our proposed features, which integrate relative angle and acceleration magnitude.

5.3.6 Model Complexity and Inference Time Analysis

To verify the efficiency of our proposed method, we conduct experiments on inference time and model parameters with existing mainstream trajectory prediction frameworks. As demonstrated in Table 5.14, our method is inferior to EigenTrajectory [30] and better than all other methods in terms of inference time and model parameters. We leave it as future work to improve the efficiency of our BP-SGCN with more advanced sequential modeling methods such as Transformers [67] and State Space Models (SSMs) [213, 214].

Table 5.13: Clustering features analysis on heterogeneous SDD (upper) and homogeneous pedestrian SDD (lower).

Method	ADE(↓)	FDE(↓)
BP-SGCN (Relative Angle)	19.52	34.05
BP-SGCN (Acceleration Magnitude)	9.07	13.02
BP-SGCN (Ours)	**6.94**	**9.57**

Method	ADE(↓)	FDE(↓)
BP-SGCN (Relative Angle)	0.45	0.68
BP-SGCN (Acceleration Magnitude)	0.42	0.63
BP-SGCN (Ours)	**0.28**	**0.41**

Table 5.14: COMPARISON OF THE PROPOSED APPROACHES IN TERMS OF NUMBER OF PARAMETER AND INFERENCE TIME.

Methods	Venue	Year	Param $\times 10^6$	Infer. Time/Iter.
ExpertTraj [179]	ICCV	2021	0.32	130 *ms*
Social-VAE [22]	ECCV	2022	5.69	1110 *ms*
GroupNet [63]	CVPR	2022	3.14	-
MSRL [151]	AAAI	2023	11.32	970 *ms*
EqMotion [26]	CVPR	2023	2.08	800 *ms*
TUTR [69]	ICCV	2023	0.44	360 *ms*
EigenTrajectory [30]	ICCV	2023	**0.02**	**72** *ms*
BP-SGCN (Ours)			0.13	110 *ms*

Moreover, we validate the stability and reliability of our BP-SGCN on heterogeneous trajectory prediction by 10 experiments. Results shown in Table 5.15 showcase the stability of our method.

Table 5.15: STABILITY TESTS ON ARGOVERSE 1 AND HETEROGENEOUS VERSION OF SDD

Methods	Argoverse 1		SDD	
	ADE(↓)	FDE(↓)	ADE(↓)	FDE(↓)
BP-SGCN (ours)	0.68 ± 0.031	1.16 ± 0.034	6.97 ± 0.069	9.59 ± 0.043

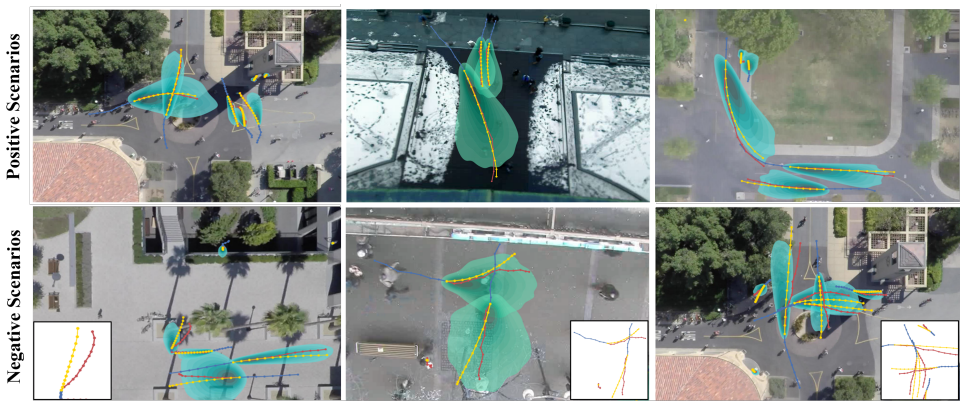


Figure 5.7: Visualization of the trajectory prediction of BP-SGCN in different social scenarios including positive predictions and negative predictions (we highlight erroneous predictions inside the white boxes). Past trajectories are shown in blue, ground-truth trajectories are in red, predicted trajectories are shown in yellow, and distributions are shown in light blue.

.

5.3.7 Discussion

In our experiments, we observed that methods [22, 30, 69, 151] tailored exclusively for pedestrians exhibit a sensitivity to the threshold settings that dictate the count of nearby agents. These methods, while ensuring state-of-the-art performance in homogeneous pedestrian trajectory prediction, perform sub-optimally in heterogeneous scenarios due to the challenge of predefining neighbors. The result is shown in Table 5.16. Unlike these pedestrian-specific approaches, which require manual neighbor selection based on metrics like relative distances, our BP-SGCN model automatically considers all proximate agents as initial neighbors, adaptively filtering out the less relevant ones. Thus, our proposed BP-SGCN is better than these methods in heterogeneous trajectory prediction.

Next, we showcase inaccurate predictions made by our BP-SGCN and delve into the method's limitations. As depicted in Figure 5.7, the first row illustrates the BP-SGCN's proficiency in accurately predicting trajectories across various social contexts. Nonetheless, the second row highlights instances where our BP-SGCN falls short, particularly in scenarios where: 1) trajectories undergo abrupt changes; 2) paths are highly erratic and frequently alter; and 3) social dynamics become exceedingly intricate with numerous agents involved. Looking ahead, our objective is to rectify these inaccuracies by en-

Table 5.16: RESULTS BY homogeneous pedestrian METHODS ON THE HETEROGENEOUS VERSION OF SDD.

Methods	Venue	Year	SDD	
			ADE(↓)	FDE(↓)
Social-VAE + FPC [22]	ECCV	2022	9.41	<u>13.49</u>
MSRL [151]	AAAI	2023	10.72	16.15
EigenTrajectory [30]	ICCV	2023	<u>8.85</u>	15.15
TUTR [69]	ECCV	2023	8.93	15.66
BP-SGCN (Ours)			**6.94**	**9.57**

hancing BP-SGCN's capabilities through the incorporation of cutting-edge deep learning methodologies, including Transformers [67] and Diffusion models [72], among others.

The quantity of behavior clusters is an adjustable hyperparameter. We manually select the number of clusters for the unsupervised deep clustering module. This approach brings several challenges, including subjectivity and potential bias, scalability issues, and potential impacts on model performance due to overfitting or underfitting. Moreover, the optimal number of clusters is sensitive to the datasets, which further complicates the selection process. Especially in heterogeneous scenarios, the high variance between different types of agents' motions makes it challenging to identify the best number of clusters to represent behavior features accurately than homogeneous pedestrian scenarios. In the future, we aim to scrutinize the behavior distributions of traffic agents more closely and dynamically estimate the optimal number of clusters [215, 216].

Despite BP-SGCN's effectiveness in both heterogeneous and homogeneous pedestrian trajectory prediction, another notable limitation of our model is its current omission of scene semantic features. Although only using trajectories as inputs brings the benefit of computation efficiency and emphasizes the importance of behavior motions, the integration of agent interactions with their surrounding environment can benefit in developing effective trajectory prediction models for use in real-world scenarios [5, 99, 193, 200]. Recognizing this, a significant direction for our future is to explore how to effectively combine trajectory data with scene semantic features to capture the interactions between static barriers and dynamic agents. We hypothesize that this will not only enhance the model's prediction accuracy, but also improve the refinement of pseudo-label identification by leveraging the rich context provided by environmental cues.

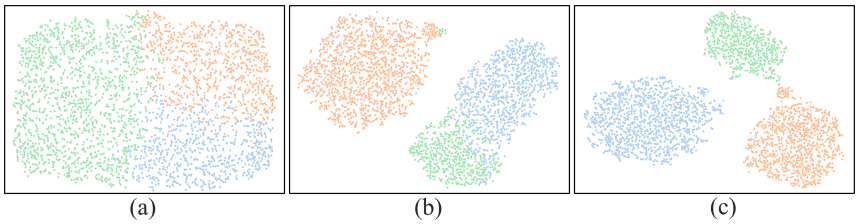


Figure 5.8: The t-SNE Visualization of clustering distribution with different features on homogeneous pedestrian SDD (k=3), using (a) acceleration, (b) angle, and (c) acceleration + angle (ours).

5.3.8 More Qualitative Visualizations

In this section, we present additional qualitative experiments to further demonstrate the prediction performance of BP-SGCN in various scenarios.

In Figure 5.8, we assess the quality of the clustering using various geometric features. Both (a) and (b) indicate that when solely relying on acceleration or angle as input feature vectors, our unsupervised deep clustering module struggles to differentiate between the hidden representations of the three pedestrian behavioral groups. However, when combining acceleration and angle (as introduced in BP-SGCN), the distinction between these three behavioral groups becomes evident and thus leads to better trajectory prediction accuracy.

Figure 5.9, Figure 5.10 and Figure 5.11 illustrate additional qualitative results on various scenes of the heterogeneous SDD dataset, heterogeneous Argoverse 1 dataset and the homogeneous pedestrian ETH/UCY datasets, respectively. Since our model relies on sampling from a bi-variate Gaussian distribution to compute the predicted trajectory, we plot the predicted distributions instead of a single trajectory to present a comprehensive view of the prediction quality in this supplementary document.

Specifically, for the SDD dataset, we visualize the predicted trajectory distributions in real-world scenarios by overlaying them on the original background images. Figure 5.9 depicts that the proposed BP-SGCN is able to predict realistic trajectory distributions that fall within valid movement areas in both simple and complex scenarios.

For Argoverse 1 dataset, Figure 5.10 showcases the predictions generated by our model adhering to the map. In straightforward scenarios, BP-SGCN effectively forecasts trajectories with varied speed profiles. When faced with intersections, the model offers

multimodal predictions, capturing the potential intentions of the agents.

For the ETH/UCY datasets, we visualize the trajectory distribution across scenarios, ranging from simple to complex scenarios (from top row to bottom row). Figure 5.11 demonstrates that BP-SGCN capably produces realistic pedestrian trajectory predictions across varied social contexts.

Notably, there are some sub-optimal results shown in the visualizations if the number of agents is large, mainly due to the randomness of agent movements. However, the proposed BP-SGCN can still provide plausible trajectory distribution predictions in these cases, as the predicted trajectory distributions can almost cover the ground-truth trajectories. Overall, the provided trajectory prediction visualizations demonstrate the effectiveness of the proposed BP-SGCN for heterogeneous and homogeneous pedestrian trajectory prediction in diverse traffic scenarios.

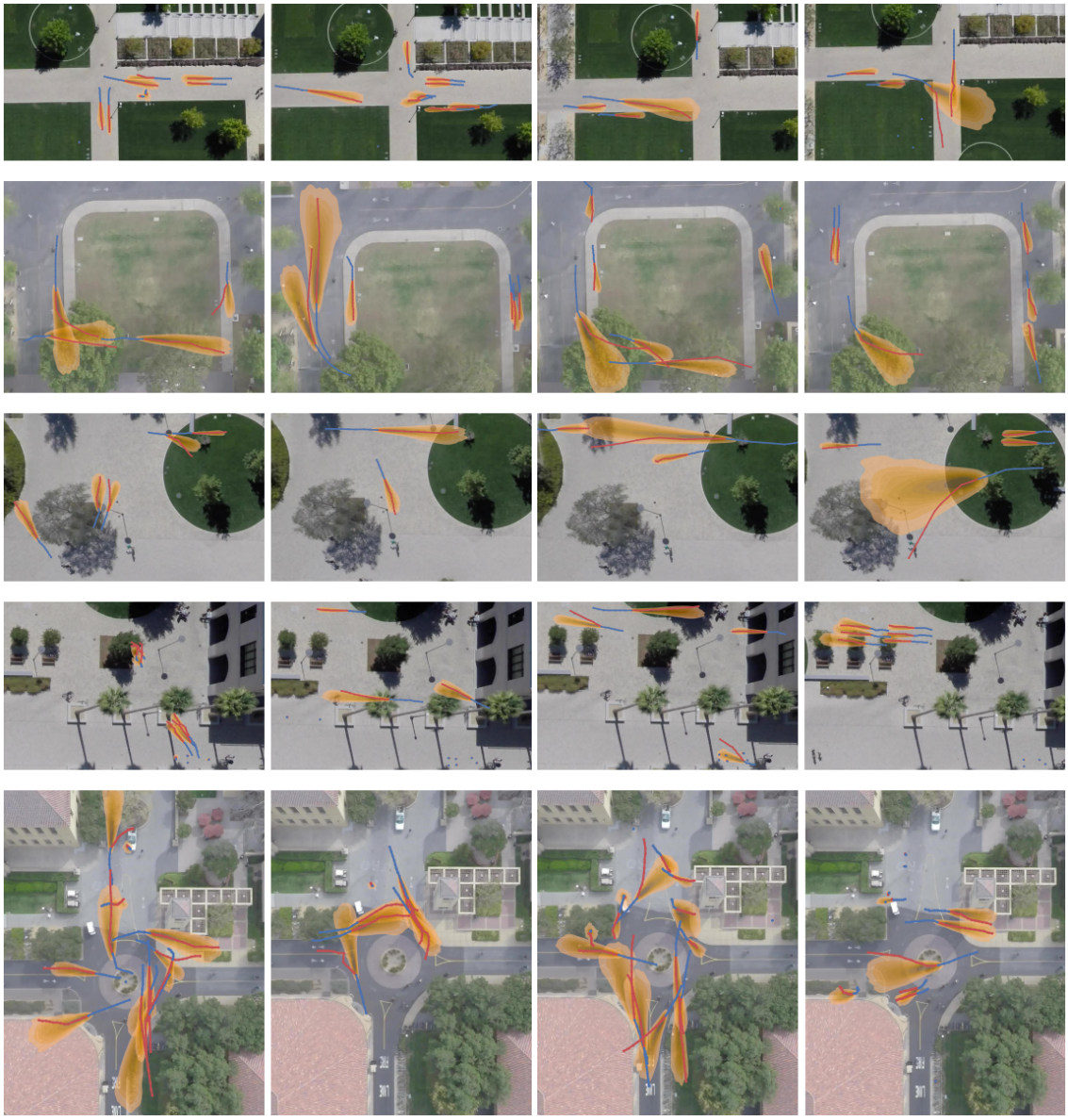


Figure 5.9: Predicted trajectory distributions using the proposed BP-SGCN on the SDD dataset. Past trajectories are shown in blue, ground-truth trajectories in red, and predicted trajectory distributions in orange.

Figure 5.10: Predicted trajectory distributions using the proposed BP-SGCN on the Argoverse 1 dataset. Past trajectories are shown in blue, ground-truth trajectories in red, and predicted trajectory distributions in orange.

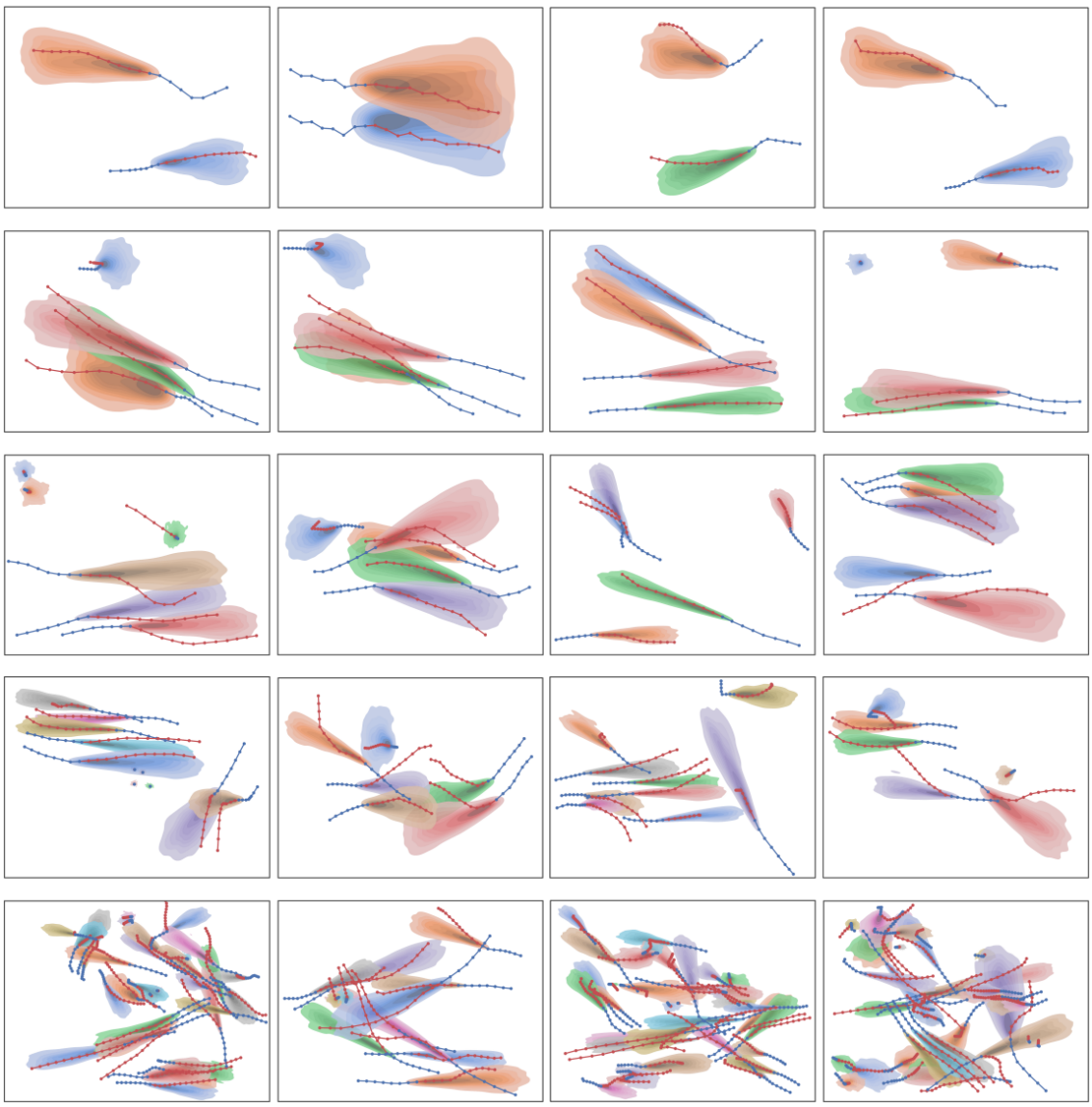


Figure 5.11: Predicted trajectory distributions using the proposed BP-SGCN on the ETH/UCY datasets. The complexity level of social interactions among pedestrians increases from the top row to the bottom row. Past trajectories are shown in blue and ground-truth trajectories are shown in red. Due to the relatively high pedestrian density, we use different colors to represent the predicted trajectory distributions of different pedestrians

5.4 Summary

In this chapter, we introduce BP-SGCN for heterogeneous and pedestrian trajectory prediction, showcasing its superior performance compared to existing models. In particular, we introduce the concept of behavioral pseudo-labels, which effectively represent the different behavior clusters of agents and do not require extra ground-truth information. BP-SGCN includes a deep unsupervised clustering module that learns the pseudo-label, as well as a pseudo-label informed sparse graph convolution network for trajectory prediction. It implements a cascaded training scheme that first learns the pseudo-labels in an unsupervised manner, and then fine-tunes the labels by optimizing the network end-to-end for better compatibility.

Beyond pedestrian scenarios, BP-SGCN also shows promising potential in broader domains. In robotic path planning [10, 23], BP-SGCN can enhance collision avoidance systems through behavioral pattern analysis [98, 111] of surrounding agents, facilitating more effective navigation in intricate settings. Additionally, in video monitoring and surveillance systems as suggested in [23, 27], BP-SGCN can enhance anomaly detection through behavioral pattern analysis of system dynamics, enabling early detection of potential operational irregularities. These applications demonstrate the applicability of BP-SGCN in modeling interactive behaviors across different domains, highlighting its potential for various real-world trajectory prediction tasks.

CHAPTER 6

Conclusion

In the domain of multi-agent trajectory prediction, the integration of advanced spatial–temporal reasoning mechanisms, semantic interaction modeling, and innovative graph neural network architectures is essential for achieving robust performance in diverse traffic environments. The contributions made during this doctoral research include three novel graph-based frameworks, each targeting a distinct yet complementary aspect of multi-agent trajectory prediction. Multiclass-SGCN (Chapter 3) is designed for heterogeneous traffic prediction, integrating agent-class semantics with an adaptive sparse graph architecture to efficiently capture asymmetric cross-type interactions. UniEdge (Chapter 4) addresses homogeneous pedestrian trajectory forecasting by introducing a unified spatial–temporal edge-enhanced graph structure that models high-order cross-time dependencies and implicit edge-to-edge influences within dense crowds. BP-SGCN (Chapter 5) builds on insights from both settings, leveraging unsupervised behavioral pseudo-labels and a cascaded clustering–prediction scheme to enhance spatial–temporal interaction modeling and improve generalization across diverse scenarios. These advancements collectively enhance the understanding and modeling of complex agent interactions, addressing challenges such as asymmetric inter-class dynamics, high-order cross-time dependencies, and cross-scenario generalization in real-world environments.

6.1 Review of Contributions

This doctoral research set out to develop robust and accurate trajectory prediction frameworks capable of operating effectively across both heterogeneous and homogeneous traffic environments. The contributions made in Chapter 3 to Chapter 5 address distinct aspects of this aim, progressively expanding from specialized solutions for individual contexts to a unified framework applicable to diverse real-world scenarios.

First, in Chapter 3, we addressed the challenges of heterogeneous traffic environments with multiple interacting agent types. Multiclass-SGCN integrates agent-class semantics with motion features via a velocity–label graph and employs an adaptive interaction mask to sparsify the spatial–temporal graph, improving efficiency without loss of accuracy. Experiments show it effectively models asymmetric cross-type dynamics and outperforms state-of-the-art baselines, fulfilling the first objective of understanding interactions in heterogeneous settings.

Second, in Chapter 4, we addressed homogeneous pedestrian scenarios where dense social interactions demand unified modeling of individual and collective behaviors. UniEdge employs a patch-based spatial–temporal graph to convert high-order cross-time dependencies into simplified first-order relationships, improving message propagation and reducing under-reaching. A dual-graph GCN and Transformer-based decoder jointly capture spatial influences and long-range temporal dependencies. Experiments show UniEdge delivers accurate, socially-aware forecasts, fulfilling the second objective of modeling dependencies in homogeneous crowds.

Finally, in Chapter 5, we proposed BP-SGCN, a unified framework for both heterogeneous and homogeneous settings. It uses unsupervised deep clustering to generate behavioral pseudo-labels, guiding the construction of sparse, semantically informed interaction graphs that capture both inter- and intra-class variations. A cascaded training scheme jointly optimizes clustering and prediction, enhancing representation learning and generalization. Experiments confirm BP-SGCN's state-of-the-art performance, fulfilling the third objective of unifying trajectory prediction frameworks for broad applicability.

6.2 Future Research Directions

Despite our achievements in advancing trajectory prediction across heterogeneous and homogeneous settings, several open challenges remain that present promising avenues for further exploration. This section outlines potential future research directions for our proposed frameworks.

6.2.1 Integration of Multimodal and Contextual Information

While the frameworks proposed in this thesis primarily leverage motion trajectories to model spatial–temporal dependencies, they do not explicitly incorporate other rich multimodal cues that are readily available in real-world traffic environments, which have been shown to significantly improve prediction performance and robustness [5, 35, 159, 196]. Such cues include scene semantics (e.g., road layout, crosswalks, sidewalks), high-definition (HD) maps, dynamic traffic signals, social norms, and even environmental factors like weather or lighting conditions. The absence of these contextual elements limits the model's ability to resolve ambiguous motion patterns, particularly in complex or unfamiliar environments. Future work will therefore focus on integrating these multimodal signals into the graph-based prediction pipeline, enriching both node and edge representations with scene- and context-aware features. This integration is expected to enhance the interpretability, safety-awareness, and generalization capacity of trajectory prediction models across a wider range of traffic scenarios.

6.2.2 Adaptive and Continual Learning

In this research, the proposed models are trained in an offline setting using fixed benchmark datasets. However, this limits their ability to adapt to evolving real-world traffic conditions. In practice, the spatial–temporal dynamics of both heterogeneous and homogeneous environments can change significantly over time due to seasonal variations, construction works, changes in traffic regulations, or the emergence of novel interaction patterns. A promising future direction is to equip trajectory prediction frameworks with adaptive and continual learning capabilities [217–219], allowing them to incrementally update their knowledge without the need for full retraining. Approaches such as on-

line graph neural networks [220] and domain adaptation [159, 221] could be explored to mitigate catastrophic forgetting while preserving previously learned behaviors. By enabling models to adapt in real time to changing environments, this line of research would enhance both the robustness and long-term deployability of trajectory prediction systems in safety-critical applications.

6.2.3 Closed-Loop Evaluation in High-Fidelity Simulation

In this research, all proposed models are evaluated in an open-loop setting, generating predictions from fixed benchmark datasets without interacting with the environment. While such protocols are common in trajectory prediction research, they may not fully reflect a model's real-world performance when deployed in dynamic, safety-critical applications. In practice, prediction errors can accumulate and propagate over time, influencing downstream modules such as planning and control. A promising future direction is to adopt closed-loop evaluation within high-fidelity simulation environments, where the trajectory predictor interacts continuously with simulated agents and their surroundings. Recent advances in sensor simulation, behavior modeling, and interactive traffic simulators (e.g., CARLA [222]) enable realistic, controllable, and reproducible testing scenarios, bridging the gap between offline evaluation and deployment. Such closed-loop testing can expose failure modes hidden by open-loop metrics, accelerate model iteration, and ensure more reliable performance in safety-critical autonomous systems.

Bibliography

[1] 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/CVF Int. Conf. Comput. Vis.*, pp. 261–268, IEEE, 2009. x, 1, 15, 45, 55, 73, 76, 85

[2] B. Stoler, "T2fpv: Dataset and method for correcting first-person view errors in pedestrian trajectory prediction." https://www.cs.cmu.edu/~csd-phd-blog/2024/t2fpv/, 2024. Accessed May 11, 2025. x, 1

[3] B. A. Rainbow, Q. Men, and H. P. Shum, "Semantics-stgcnn: A semantics-guided spatial-temporal graph convolutional network for multi-class trajectory prediction," in *IEEE Int. Conf. Syst. Man Cybern.*, pp. 2959–2966, IEEE, 2021. x, 3, 29, 31, 35, 36, 37, 38

[4] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik, and A. Gaidon, "It is not the journey but the destination: Endpoint conditioned trajectory prediction," in *Proc. Eur. Conf. Comput. Vis.*, pp. 759–776, 2020. xii, 73, 77, 89, 90

[5] K. Mangalam, Y. An, H. Girase, and J. Malik, "From goals, waypoints & paths to long term human trajectory forecasting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pp. 15233–15242, 2021. xii, 13, 73, 77, 84, 101, 110

[6] B. A. Rainbow, Q. Men, and H. P. Shum, "Semantics-stgcnn: A semantics-guided spatial-temporal graph convolutional network for multi-class trajectory prediction," in *IEEE Int. Conf. Syst. Man Cybern.*, pp. 2959–2966, IEEE, 2021. xiii, 4, 17, 74, 77, 83, 86, 88, 92, 93

[7] R. Li, S. Katsigiannis, and H. P. Shum, "Multiclass-sgcn: Sparse graph-based trajectory prediction with agent class embedding," in *IEEE Int. Conf. Image Process.*, pp. 2346–2350, IEEE, 2022. xiii, 42, 54, 74, 77, 79, 83, 85, 86, 88, 92, 93

[8] H. Bai, S. Cai, N. Ye, D. Hsu, and W. S. Lee, "Intention-aware online pomdp planning for autonomous driving in a crowd," in *IEEE Int. Conf. Robot. Autom.*, pp. 454–460, IEEE, 2015. 1, 42

[9] J. Liu, X. Mao, Y. Fang, D. Zhu, and M. Q.-H. Meng, "A survey on deep-learning approaches for vehicle trajectory prediction in autonomous driving," in *IEEE Int. Conf. Robot. Biomimetics*, pp. 978–985, IEEE, 2021. 1, 10, 29

[10] Y. Luo, P. Cai, A. Bera, D. Hsu, W. S. Lee, and D. Manocha, "Porca: Modeling and planning for autonomous driving among many pedestrians," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3418–3425, 2018. 1, 10, 73, 107

[11] P. Raksincharoensak, T. Hasegawa, and M. Nagai, "Motion planning and control of autonomous driving intelligence system based on risk potential optimization framework," *Int. J. Automot. Eng.*, vol. 7, no. AVEC14, pp. 53–60, 2016. 1, 10

[12] M. Luber, J. A. Stork, G. D. Tipaldi, and K. O. Arras, "People tracking with human motion predictions from social forces," in *IEEE Int. Conf. Robot. Autom.*, pp. 464–469, IEEE, 2010. 1, 10

[13] M. Yasuno, N. Yasuda, and M. Aoki, "Pedestrian detection and tracking in far infrared images," in *Conf. Comput. Vis. Pattern Recognit. Workshop*, pp. 125–125, IEEE, 2004. 1, 10

[14] A. M. Kanu-Asiegbu, R. Vasudevan, and X. Du, "Leveraging trajectory prediction for pedestrian video anomaly detection," in *IEEE Symp. Ser. Comput. Intell.*, pp. 11–08, IEEE, 2021. 1, 10

[15] B. Musleh, F. García, J. Otamendi, J. M. Armingol, and A. De la Escalera, "Identifying and tracking pedestrians based on sensor fusion and motion stability predictions," *Sensors*, vol. 10, no. 9, pp. 8028–8053, 2010. 1, 10

[16] S. Lin, W. Lin, X. Hu, W. Wu, R. Mo, and H. Zhong, "Cyclenet: enhancing time series forecasting through modeling periodic patterns," *Adv. Neural Inf. Process. Syst.*, vol. 37, pp. 106315–106345, 2024. 2

[17] K. Yi, J. Fei, Q. Zhang, H. He, S. Hao, D. Lian, and W. Fan, "Filternet: Harnessing frequency filters for time series forecasting," *Adv. Neural Inf. Process. Syst.*, vol. 37, pp. 55115–55140, 2024. 2

[18] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 22419–22430, 2021. 2

[19] R. Li, S. Katsigiannis, T.-K. Kim, and H. P. Shum, "Bp-sgcn: Behavioral pseudo-label informed sparse graph convolution network for pedestrian and heterogeneous trajectory prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, 2025. 2, 3, 16, 72

[20] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *IEEE/CVF CVPR*, pp. 2255–2264, 2018. 2, 4, 35, 36

[21] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 2255–2264, 2018. 2, 3, 13, 19, 27, 52, 55, 56, 58, 85, 86, 89, 90

[22] P. Xu, J.-B. Hayet, and I. Karamouzas, "Socialvae: Human trajectory prediction using timewise latents," in *Proc. Eur. Conf. Comput. Vis.*, pp. 511–528, 2022. 2, 13, 56, 68, 89, 90, 99, 100, 101

[23] R. Liang, Y. Li, J. Zhou, and X. Li, "Stglow: A flow-based generative framework with dual-graphormer for pedestrian trajectory prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 11, pp. 16504–16517, 2024. 2, 89, 91, 107

[24] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995. 2

[25] I. Bae, J.-H. Park, and H.-G. Jeon, "Learning pedestrian group representations for multi-modal trajectory prediction," in *Proc. Eur. Conf. Comput. Vis.*, pp. 270–289, 2022. 2, 25, 42, 43, 46, 50, 55, 56, 58, 59, 60, 61, 86, 89, 90, 92

[26] C. Xu, R. T. Tan, Y. Tan, S. Chen, Y. G. Wang, X. Wang, and Y. Wang, "Eqmotion: Equivariant multi-agent motion prediction with invariant interaction reasoning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 1410–1420, 2023. 2, 56, 57, 68, 79, 89, 90, 91, 99

[27] L. Shi, L. Wang, C. Long, S. Zhou, M. Zhou, Z. Niu, and G. Hua, "Sgcn: Sparse graph convolution network for pedestrian trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 8994–9003, 2021. 3, 4, 20, 23, 27, 29, 31, 33, 34, 35, 36, 42, 43, 46, 48, 50, 52, 54, 55, 56, 65, 68, 73, 74, 75, 79, 84, 85, 86, 89, 90, 92, 107

[28] 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/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 961–971, 2016. 3, 4, 13, 19, 24, 27, 29, 35, 36, 48, 52, 55, 56, 74, 86, 88, 89, 90

[29] 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.*, pp. 14424–14432, 2020. 3, 4, 13, 19, 23, 24, 29, 31, 34, 35, 36, 42, 43, 46, 48, 50, 52, 54, 55, 56, 58, 65, 73, 74, 86, 88, 89, 90

[30] I. Bae, J. Oh, and H.-G. Jeon, "Eigentrajectory: Low-rank descriptors for multi-modal trajectory forecasting," *arXiv preprint arXiv:2307.09306*, 2023. 3, 4, 42, 46, 50, 55, 56, 58, 59, 60, 61, 65, 66, 67, 68, 89, 90, 98, 99, 100, 101

[31] S. Kim, H.-g. Chi, H. Lim, K. Ramani, J. Kim, and S. Kim, "Higher-order relational reasoning for pedestrian trajectory prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 15251–15260, 2024. 3, 4, 20, 56, 58, 68, 89, 90

[32] R. Li, S. Katsigiannis, and H. P. Shum, "Multiclass-sgcn: Sparse graph-based trajectory prediction with agent class embedding," in *IEEE Int. Conf. Image Process.*, pp. 2346–2350, IEEE, 2022. 3, 28

[33] J. Zhang, J. Yao, L. Yan, Y. Xu, and Z. Wang, "Sparse multi-relational graph convolutional network for multi-type object trajectory prediction," in *Int. Joint Conf. Artif. Intell.*, pp. 1697–1705, 2024. 3, 17, 20, 22, 86, 88

[34] F. Zheng, L. Wang, S. Zhou, W. Tang, Z. Niu, N. Zheng, and G. Hua, "Unlimited neighborhood interaction for heterogeneous trajectory prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pp. 13168–13177, 2021. 3, 4, 16, 21, 24, 74, 77, 85, 86, 87, 88

[35] J. Fang, C. Zhu, P. Zhang, H. Yu, and J. Xue, "Heterogeneous trajectory forecasting via risk and scene graph learning," *IEEE Trans. Intell. Transp. Syst.*, 2023. 3, 17, 24, 85, 86, 87, 88, 90, 110

[36] Q. Du, X. Wang, S. Yin, L. Li, and H. Ning, "Social force embedded mixed graph convolutional network for multi-class trajectory prediction," *IEEE Trans. Intell. Veh.*, 2024. 3

[37] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha, "Trafficpredict: Trajectory prediction for heterogeneous traffic-agents," in *Proc. AAAI Conf. Art. Intel.*, vol. 33, pp. 6120–6127, 2019. 3

[38] 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.*, pp. 6272–6281, 2019. 4, 13, 19, 24, 42, 43, 49, 65, 73, 74, 84, 86, 88, 89

[39] I. Bae and H.-G. Jeon, "A set of control points conditioned pedestrian trajectory prediction," in *Proc. AAAI Conf. Art. Intel.*, vol. 37, pp. 6155–6165, 2023. 4, 42, 55, 56, 58, 59, 60, 61, 68, 89, 91

[40] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "Desire: Distant future prediction in dynamic scenes with interacting agents," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 336–345, 2017. 4, 13, 35, 36, 86, 88

[41] Y. Wang and S. Chen, "Multi-agent trajectory prediction with spatio-temporal sequence fusion," *IEEE Trans. Multimedia*, 2021. 4, 86, 88, 89, 91

[42] Q. Du, X. Wang, S. Yin, L. Li, and H. Ning, "Social force embedded mixed graph convolutional network for multi-class trajectory prediction," *IEEE Trans. Intell. Veh.*, pp. 1–11, 2024. 4, 16, 74, 86, 88

[43] R. Tamaru, P. Li, and B. Ran, "Enhancing pedestrian trajectory prediction with crowd trip information," *arXiv preprint arXiv:2409.15224*, 2024. 10

[44] M. Zong, Y. Chang, Y. Dang, and K. Wang, "Pedestrian trajectory prediction in crowded environments using social attention graph neural networks," *Applied Sciences*, vol. 14, no. 20, p. 9349, 2024. 10

[45] J. Jiang, K. Yan, X. Xia, and B. Yang, "A survey of deep learning-based pedestrian trajectory prediction: Challenges and solutions," *Sensors*, vol. 25, no. 3, p. 957, 2025. 11

[46] R. Korbmacher and A. Tordeux, "Review of pedestrian trajectory prediction methods: Comparing deep learning and knowledge-based approaches," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 24126–24144, 2022. 11, 18

[47] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, pp. 4282–4286, may 1995. 11, 16

[48] N. Rinke, C. Schiermeyer, F. Pascucci, V. Berkhahn, and B. Friedrich, "A multi-layer social force approach to model interactions in shared spaces using collision prediction," *Transp. Res. Procedia*, vol. 25, pp. 1249–1267, 2017. 12

[49] M. Chraibi, A. Seyfried, and A. Schadschneider, "Generalized centrifugal-force model for pedestrian dynamics," *Phys. Rev. E*, vol. 82, no. 4, p. 046111, 2010. 12

[50] B. Anvari, M. G. Bell, A. Sivakumar, and W. Y. Ochieng, "Modelling shared space users via rule-based social force model," *Transp. Res. Part C Emerg. Technol.*, vol. 51, pp. 83–103, 2015. 12

[51] F. T. Johora and J. P. Müller, "Modeling interactions of multimodal road users in shared spaces," in *IEEE Int. Conf. Intell. Transp. Syst.*, pp. 3568–3574, 2018. 12

[52] F. T. Johora and J. P. Müller, "On transferability and calibration of pedestrian and car motion models in shared spaces," *Transp. Lett.*, vol. 13, no. 3, pp. 172–182, 2021.

[53] J. Felcman and P. Kubera, "A cellular automaton model for a pedestrian flow problem," *Math. Model. Nat. Phenom.*, vol. 16, p. 11, 2021. 12

[54] M. Zong, Y. Chang, Y. Dang, and K. Wang, "Pedestrian trajectory prediction in crowded environments using social attention graph neural networks," *Applied Sciences*, vol. 14, no. 20, p. 9349, 2024. 12

[55] C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittartz, "Simulation of pedestrian dynamics using a two-dimensional cellular automaton," *Physica A*, vol. 295, no. 3-4, pp. 507–525, 2001. 12

[56] X. Li and J.-Q. Sun, "Studies of vehicle lane-changing to avoid pedestrians with cellular automata," *Phys. A, Stat. Mech. Appl.*, vol. 438, pp. 251–271, 2015. 12

[57] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *Int. J. Robot. Res.*, vol. 17, no. 7, pp. 760–772, 1998. 12

[58] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *IEEE Int. Conf. Robot. Autom.*, pp. 1928–1935, 2008. 12

[59] H. Xue, D. Q. Huynh, and M. Reynolds, "Ss-lstm: A hierarchical lstm model for pedestrian trajectory prediction," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1186–1194, IEEE, 2018. 13, 19

[60] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Proc. Adv. Neu. Inf. Process. Syst.*, vol. 27, 2014. 13, 90

[61] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *Proc. Eur. Conf. Comput. Vis.*, pp. 683–700, 2020. 13, 24

[62] V. Kosaraju, A. Sadeghian, R. Martín-Martín, I. Reid, H. Rezatofighi, and S. Savarese, "Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks," *Proc. Adv. Neu. Inf. Process. Syst.*, vol. 32, 2019. 13, 42, 49

[63] C. Xu, M. Li, Z. Ni, Y. Zhang, and S. Chen, "Groupnet: Multiscale hypergraph neural networks for trajectory prediction with relational reasoning," *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 6488–6497, 2022. 14, 20, 43, 50, 56, 57, 58, 89, 90, 99

[64] Y. Xu, A. Bazarjani, H.-g. Chi, C. Choi, and Y. Fu, "Uncovering the missing pattern: Unified framework towards trajectory imputation and prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 9632–9643, 2023. 14, 24

[65] X. Mo, Z. Huang, Y. Xing, and C. Lv, "Multi-agent trajectory prediction with heterogeneous edge-enhanced graph attention network," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 9554–9567, 2022. 14, 21, 24, 43, 50

[66] J. Li, L. Yang, Y. Chen, and Y. Jin, "Mfan: Mixing feature attention network for trajectory prediction," *Pattern Recognition*, vol. 146, p. 109997, 2024. 14, 24, 56, 58, 68, 90

[67] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Proc. Adv. Neu. Inf. Process. Syst.*, vol. 30, 2017. 14, 19, 29, 33, 45, 52, 53, 64, 84, 98, 101

[68] Y. Yuan, X. Weng, Y. Ou, and K. M. Kitani, "Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pp. 9813–9823, 2021. 14, 23, 24, 89, 90

[69] L. Shi, L. Wang, S. Zhou, and G. Hua, "Trajectory unified transformer for pedestrian trajectory prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pp. 9675–9684, 2023. 14, 23, 52, 55, 56, 57, 58, 68, 89, 90, 99, 100, 101

[70] Z. Yin, R. Liu, Z. Xiong, and Z. Yuan, "Multimodal transformer networks for pedestrian trajectory prediction.," in *Int. Joint Conf. Artif. Intell.*, pp. 1259–1265, 2021. 14

[71] T. Gu, G. Chen, J. Li, C. Lin, Y. Rao, J. Zhou, and J. Lu, "Stochastic trajectory prediction via motion indeterminacy diffusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 17113–17122, 2022. 14, 89, 90

[72] W. Mao, C. Xu, Q. Zhu, S. Chen, and Y. Wang, "Leapfrog diffusion model for stochastic trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 5517–5526, 2023. 14, 56, 57, 58, 89, 90, 101

[73] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," in *Computer graphics forum*, vol. 26, pp. 655–664, Wiley Online Library, 2007. 15, 45, 55, 73, 76, 85

[74] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *Proc. Eur. Conf. Comput. Vis.*, pp. 549–565, 2016. 15, 17, 30, 35, 45, 55, 74, 76, 85

[75] D. Yang, Ü. Özgüner, and K. Redmill, "Social force based microscopic modeling of vehicle-crowd interaction," in *IEEE Intell. Vehicles Symp.*, pp. 1537–1542, 2018. 15

[76] D. Yang, Ü. Özgüner, and K. Redmill, "A social force based pedestrian motion model considering multi-pedestrian interaction with a vehicle," *ACM Trans. Spatial Algorithms Syst.*, vol. 6, no. 2, pp. 1–27, 2020. 15

[77] F. T. Johora and J. P. Müller, "Modeling interactions of multimodal road users in shared spaces," in *Int. Conf. Intell. Transp. Syst.*, pp. 3568–3574, 2018. 15

[78] F. T. Johora and J. P. Müller, "On transferability and calibration of pedestrian and car motion models in shared spaces," *Transp. Lett.*, vol. 13, no. 3, pp. 172–182, 2021. 15

[79] B. Anvari, M. G. Bell, A. Sivakumar, and W. Y. Ochieng, "Modelling shared space users via rule-based social force model," *Transp. Res. C, Emerg. Technol.*, vol. 51, pp. 83–103, 2015. 15

[80] J. Wei, N. Vödisch, A. Rehr, C. Feist, and A. Valada, "Parkdiffusion: Heterogeneous multi-agent multi-modal trajectory prediction for automated parking using diffusion models," *arXiv preprint arXiv:2505.00586*, 2025. 16

[81] T. Zhao, Y. Xu, M. Monfort, W. Choi, C. Baker, Y. Zhao, Y. Wang, and Y. N. Wu, "Multi-agent tensor fusion for contextual trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 12126–12134, 2019. 16, 88

[82] H. Bi, Z. Fang, T. Mao, Z. Wang, and Z. Deng, "Joint prediction for kinematic trajectories in vehicle-pedestrian-mixed scenes," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pp. 10383–10392, 2019. 16

[83] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997. 16, 23, 43, 64, 81

[84] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff, "Covernet: Multimodal behavior prediction using trajectory sets," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 14074–14083, 2020. 16

[85] S. H. Park, G. Lee, J. Seo, M. Bhat, M. Kang, J. Francis, A. Jadhav, P. P. Liang, and L.-P. Morency, "Diverse and admissible trajectory forecasting through multimodal context understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 282–298, 2020. 16, 88

[86] A. Jafari and Y.-C. Liu, "A heterogeneous social force model for personal mobility vehicles on futuristic sidewalks," *Simul. Model. Pract. Theory*, vol. 131, p. 102879, 2024. 16

[87] X. Xu, W. Liu, and L. Yu, "Trajectory prediction for heterogeneous traffic-agents using knowledge correction data-driven model," *Information Sciences*, vol. 608, pp. 375–391, 2022. 16

[88] D. Grimm, M. Zipfl, F. Hertlein, A. Naumann, J. Luettin, S. Thoma, S. Schmid, L. Halilaj, A. Rettinger, and J. M. Zöllner, "Heterogeneous graph-based trajectory prediction using local map context and social interactions," in *Int. Conf. Intell. Transp. Syst.*, pp. 2901–2907, 2023. 16

[89] J. Fan, Z. Liu, Y. Fang, Z. Huang, Y. Liu, and S. Lin, "Multi-class agent trajectory prediction with selective state spaces for autonomous driving," *Eng. Appl. Artif. Intell.*, vol. 156, p. 111027, 2025. 17

[90] D. Xu, X. Shang, H. Peng, and H. Li, "Mvhgn: Multi-view adaptive hierarchical spatial graph convolution network based trajectory prediction for heterogeneous traffic-agents," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 6, pp. 6217–6226, 2023. 17, 22

[91] X. Chen, H. Zhang, Y. Hu, J. Liang, and H. Wang, "Vnagt: Variational non-autoregressive graph transformer network for multi-agent trajectory prediction," *IEEE Trans. Veh. Technol*, vol. 72, no. 10, pp. 12540–12552, 2023. 17, 21, 86, 88

[92] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 8748–8757, 2019. 17, 76, 85

[93] C. Finet, S. D. S. Martins, J.-B. Hayet, I. Karamouzas, J. Amirian, S. L. Hégarat-Mascle, J. Pettré, and E. Aldea, "Recent advances in multi-agent human trajectory prediction: A comprehensive review," *arXiv preprint arXiv:2506.14831*, 2025. 18

[94] M. Golchoubian, M. Ghafurian, K. Dautenhahn, and N. L. Azad, "Pedestrian trajectory prediction in pedestrian-vehicle mixed environments: A systematic review," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 11, pp. 11544–11567, 2023. 18

[95] Z. Fu, K. Jiang, C. Xie, Y. Xu, J. Huang, and D. Yang, "Summary and reflections on pedestrian trajectory prediction in the field of autonomous driving," *IEEE Trans. Intell. Veh.*, 2024. 18

[96] H. Zhou, X. Yang, M. Fan, H. Huang, D. Ren, and H. Xia, "Static-dynamic global graph representation for pedestrian trajectory prediction," *Knowl.-Based Syst.*, vol. 277, p. 110775, 2023. 19

[97] 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.*, pp. 507–523, 2020. 19

[98] B. Yang, G. Yan, P. Wang, C.-Y. Chan, X. Song, and Y. Chen, "A novel graph-based trajectory predictor with pseudo-oracle," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 7064–7078, 2021. 19, 89, 91, 107

[99] P. Lv, W. Wang, Y. Wang, Y. Zhang, M. Xu, and C. Xu, "Ssagcn: Social soft attention graph convolution network for pedestrian trajectory prediction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 9, pp. 11989–12003, 2024. 19, 101

[100] Y. Liu, H. Guo, Q. Meng, and J. Li, "Spatial-temporal graph attention network for pedestrian trajectory prediction," in *CAA Int. Conf. Veh. Control Intell.*, pp. 1–6, 2022. 19

[101] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, *et al.*, "Graph attention networks," *stat*, vol. 1050, no. 20, pp. 10–48550, 2017. 19, 43

[102] M. Black, Z. Wan, A. Nayyeri, and Y. Wang, "Understanding oversquashing in gnns through the lens of effective resistance," in *Proc. Int. Conf. Mach. Learn.*, pp. 2528–2547, PMLR, 2023. 20, 43, 44, 48

[103] C. Sun, B. Wang, J. Leng, X. Zhang, and B. Wang, "Sdagcn: Sparse directed attention graph convolutional network for spatial interaction in pedestrian trajectory prediction," *IEEE Internet Things J.*, vol. 11, no. 24, pp. 39225–39235, 2024. 20

[104] C. Xu, Y. Wei, B. Tang, S. Yin, Y. Zhang, S. Chen, and Y. Wang, "Dynamic-group-aware networks for multi-agent trajectory prediction with relational reasoning," *Neural Networks*, vol. 170, pp. 564–577, 2024. 20, 89, 90

[105] S. Lee, J. Lee, Y. Yu, T. Kim, and K. Lee, "Mart: Multiscale relational transformer networks for multi-agent trajectory prediction," in *Proc. Eur. Conf. Comput. Vis.*, pp. 89–107, 2024. 20

[106] G. Li, G. Luo, Q. Yuan, and J. Li, "Trajectory prediction with heterogeneous graph neural network," in *Pac. Rim Int. Conf. Artif. Intell.*, pp. 375–387, 2022. 21

[107] J. Li, H. Shi, Y. Guo, G. Han, R. Yu, and X. Wang, "Tragcan: Trajectory prediction of heterogeneous traffic agents in iov systems," *IEEE Internet Things J.*, vol. 10, pp. 7100–7113, 2023. 22

[108] X. Jia, P. Wu, L. Chen, H. Li, Y. Liu, and J. Yan, "Hdgt: Heterogeneous driving graph transformer for multi-agent trajectory prediction via scene encoding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, pp. 13860–13875, 2022. 22

[109] V. Capone, A. Casolaro, and F. Camastra, "Spatio-temporal prediction using graph neural networks: A survey," *Neurocomputing*, p. 130400, 2025. 23

[110] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014. 23, 81

[111] H. Xue, D. Q. Huynh, and M. Reynolds, "Poppl: Pedestrian trajectory prediction by lstm with automatic route class clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 77–90, 2020. 23, 25, 107

[112] J. Wang, K. Liu, H. Li, Q. Gao, and X. Wang, "Vehicle trajectory prediction using hierarchical lstm and graph attention network," *IEEE Internet Things J.*, vol. 12, no. 6, pp. 7010–7025, 2025. 23

[113] Y. Zhou, H. Wu, H. Cheng, K. Qi, K. Hu, C. Kang, and J. Zheng, "Social graph convolutional lstm for pedestrian trajectory prediction," *IET Intell. Transp. Syst.*, vol. 15, no. 3, pp. 396–405, 2021. 23

[114] P. Zhang, W. Ouyang, P. Zhang, J. Xue, and N. Zheng, "Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 12085–12094, 2019. 23

[115] J. Mi, X. Zhang, H. Zeng, and L. Wang, "Dergcn: Dynamic-evolving graph convolutional networks for human trajectory prediction," *Neurocomputing*, vol. 569, p. 127117, 2024. 23

[116] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *CoRR*, vol. abs/1803.01271, 2018. 23, 29, 30, 34, 43, 64

[117] K. Yi, Q. Zhang, W. Fan, H. He, L. Hu, P. Wang, N. An, L. Cao, and Z. Niu, "FourierGNN: Rethinking multivariate time series forecasting from a pure graph perspective," in *Proc. Adv. Neu. Inf. Process. Syst.*, 2023. 24, 43

[118] Y. Wang, Y. Xu, J. Yang, M. Wu, X. Li, L. Xie, and Z. Chen, "Fully-connected spatial-temporal graph for multivariate time-series data," in *Proc. AAAI Conf. Artif. Intell.*, vol. 38, pp. 15715–15724, 2024. 24, 43

[119] W. Lu, Z. Guan, W. Zhao, Y. Yang, and L. Jin, "Nodemixup: Tackling under-reaching for graph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 38, pp. 14175–14183, 2024. 24, 43, 44, 48

[120] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Soft+ hardwired attention: An lstm framework for human trajectory prediction and abnormal event detection," *Neural networks*, vol. 108, pp. 466–478, 2018. 25

[121] I. A. Lawal, F. Poiesi, D. Anguita, and A. Cavallaro, "Support vector motion clustering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 11, pp. 2395–2408, 2016. 25

[122] A. K.-F. Lui, Y.-H. Chan, and M.-F. Leung, "Modelling of destinations for data-driven pedestrian trajectory prediction in public buildings," in *IEEE Int. Conf. Big Data*, pp. 1709–1717, IEEE, 2021. 25

[123] J. Sun, Y. Li, H.-S. Fang, and C. Lu, "Three steps to multimodal trajectory prediction: Modality clustering, classification and synthesis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pp. 13250–13259, 2021. 26, 73, 83, 89, 91

[124] Y. Wang, P. Zhang, L. Bai, and J. Xue, "Fend: A future enhanced distribution-aware contrastive learning framework for long-tail trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 1400–1409, 2023. 26, 83, 89, 91

[125] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. Int. Conf. Mach. Learn.*, pp. 478–487, 2016. 26, 75, 80, 82, 83

[126] B. Zhou, X. Tang, and X. Wang, "Learning collective crowd behaviors with dynamic pedestrian-agents," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 50–68, 2015. 29

[127] C. Yu, X. Ma, J. Ren, H. Zhao, and S. Yi, "Spatio-temporal graph transformer networks for pedestrian trajectory prediction," in *ECCV*, August 2020. 29

[128] I. Giuliari, F.and Hasan, M. Cristani, and F. Galasso, "Transformer networks for trajectory forecasting," in *ICPR*, pp. 10335–10342, 2021. 29

[129] Q. Men and H. P. H. Shum, "Pytorch-based implementation of label-aware graph representation for multi-class trajectory prediction," *Software Impacts*, vol. 11, p. 100201, 2021. 29

[130] P. Zhang, C. Lan, W. Zeng, J. Xing, J. Xue, and N. Zheng, "Semantics-guided neural networks for efficient skeleton-based human action recognition," in *IEEE/CVF CVPR*, pp. 1109–1118, 2020. 29

[131] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent.*, 2017. 30, 34, 43, 62, 73

[132] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *IEEE CVPR*, pp. 2818–2826, 2016. 33

[133] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015. 35

[134] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg, "Who are you with and where are you going?," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1345–1352, IEEE, 2011. 35, 36

[135] A. Sadeghian, F. Legros, M. Voisin, R. Vesel, A. Alahi, and S. Savarese, "Car-net: Clairvoyant attentive recurrent network," in *ECCV*, pp. 162–180, 2018. 35, 36

[136] R. Li, T. Qiao, S. Katsigiannis, Z. Zhu, and H. P. Shum, "Unified spatial-temporal edge-enhanced graph networks for pedestrian trajectory prediction," *IEEE Trans. Circuits Syst. Video Technol.*, 2025. 41

[137] W. Chen, Z. Yang, L. Xue, J. Duan, H. Sun, and N. Zheng, "Multimodal pedestrian trajectory prediction using probabilistic proposal network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 6, pp. 2877–2891, 2023. 42, 52

[138] H. Sun, Z. Zhao, Z. Yin, and Z. He, "Reciprocal twin networks for pedestrian motion learning and future path prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 1483–1497, 2021. 42, 70

[139] X. Zhou, H. Ren, T. Zhang, X. Mou, Y. He, and C.-Y. Chan, "Prediction of pedestrian crossing behavior based on surveillance video," *Sensors*, 2022. 42

[140] X. Liu, J. Yin, J. Liu, P. Ding, J. Liu, and H. Liu, "Trajectorycnn: a new spatio-temporal feature learning network for human motion prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 6, pp. 2133–2146, 2020. 42

[141] N. Wang, G. Zhu, H. Li, M. Feng, X. Zhao, L. Ni, P. Shen, L. Mei, and L. Zhang, "Exploring spatio–temporal graph convolution for video-based human–object interaction recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 10, pp. 5814–5827, 2023. 42

[142] Y. Xia, Y. Liang, H. Wen, X. Liu, K. Wang, Z. Zhou, and R. Zimmermann, "Deciphering spatio-temporal graph forecasting: A causal lens and treatment," in *Proc. Adv. Neu. Inf. Process. Syst.*, 2023. 44, 51, 52

[143] J. Huang, M. K. Chung, and A. Qiu, "Heterogeneous graph convolutional neural network via hodge-laplacian for brain functional data," in *Int. Conf. Inf. Process. Med. Imaging*, pp. 278–290, Springer, 2023. 44, 45, 51

[144] X. Wu, W. Lu, Y. Quan, Q. Miao, and P. G. Sun, "Deep dual graph attention auto-encoder for community detection," *Expert Syst. Appl.*, vol. 238, p. 122182, 2024. 45

[145] O. Post, "First-order operators and boundary triples," *Russian Journal of Mathematical Physics*, vol. 14, no. 4, pp. 482–492, 2007. 45

[146] H. Wang, W. Zhi, G. Batista, and R. Chandra, "Pedestrian trajectory prediction using dynamics-based deep learning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15068–15075, IEEE, 2024. 46, 56, 57

[147] A. Ghosh, S. Boyd, and A. Saberi, "Minimizing effective resistance of a graph," *SIAM review*, vol. 50, no. 1, pp. 37–66, 2008. 48

[148] E. Bozzo, "The moore–penrose inverse of the normalized graph laplacian," *Linear Algebra Appl.*, vol. 439, no. 10, pp. 3038–3043, 2013. 49

[149] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 25, 2012. 49

[150] S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?," *arXiv preprint arXiv:2105.14491*, 2021. 49

[151] Y. Wu, L. Wang, S. Zhou, J. Duan, G. Hua, and W. Tang, "Multi-stream representation learning for pedestrian trajectory prediction," in *Proc. AAAI Conf. Art. Intel.*, vol. 37, pp. 2875–2882, 2023. 50, 56, 57, 58, 89, 90, 99, 100, 101

[152] T. Li, Y. Tian, H. Li, M. Deng, and K. He, "Autoregressive image generation without vector quantization," *arXiv preprint arXiv:2406.11838*, 2024. 53

[153] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "itransformer: Inverted transformers are effective for time series forecasting," *arXiv preprint arXiv:2310.06625*, 2023. 53, 64

[154] A. Mohamed, D. Zhu, W. Vu, M. Elhoseiny, and C. Claudel, "Social-implicit: Rethinking trajectory prediction evaluation and the effectiveness of implicit maximum likelihood estimation," in *Proc. Eur. Conf. Comput. Vis.*, pp. 463–479, 2022. 55, 68, 73, 85, 89, 91

[155] 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.*, pp. 6488–6497, 2022. 56, 57, 58, 89, 91

[156] F. Marchetti, F. Becattini, L. Seidenari, and A. Del Bimbo, "Smemo: social memory for trajectory forecasting," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2024. 56, 57, 58, 89, 91

[157] Z. Pei, J. Zhang, W. Zhang, M. Wang, J. Wang, and Y.-H. Yang, "Autofocusing for synthetic aperture imaging based on pedestrian trajectory prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 5, pp. 3551–3562, 2024. 56, 57

[158] Y. Peng, G. Zhang, J. Shi, X. Li, and L. Zheng, "Mrgtraj: A novel non-autoregressive approach for human trajectory prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 4, pp. 2318–2331, 2024. 56, 57, 68, 89, 90

[159] I. Bae, Y.-J. Park, and H.-G. Jeon, "Singulartrajectory: Universal trajectory predictor using diffusion model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 17890–17901, 2024. 56, 57, 110, 111

[160] Y. Xu, L. Wang, Y. Wang, and Y. Fu, "Adaptive trajectory prediction via transferable gnn," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 6520–6531, 2022. 62

[161] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017. 62

[162] T. Ahmad, L. Jin, L. Lin, and G. Tang, "Skeleton-based action recognition using sparse spatio-temporal gcn with edge effective resistance," *Neurocomputing*, vol. 423, pp. 389–398, 2021. 69

[163] Q. Huang, L. Shen, R. Zhang, J. Cheng, S. Ding, Z. Zhou, and Y. Wang, "Hdmixer: Hierarchical dependency with extendable patch for multivariate time series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 38, pp. 12608–12616, 2024. 70

[164] A. Díaz Berenguer, M. Alioscha-Perez, M. C. Oveneke, and H. Sahli, "Context-aware human trajectories prediction via latent variational model," *IEEE Trans. Circuits Syst. Video Technol*, vol. 31, no. 5, pp. 1876–1889, 2021. 70

[165] T. Dao, D. Fu, S. Ermon, A. Rudra, and C. Ré, "Flashattention: Fast and memory-efficient exact attention with io-awareness," *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 16344–16359, 2022. 70

[166] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient transformers: A survey," *ACM Comput. Surv.*, vol. 55, no. 6, 2022. 70

[167] W. Yang, S. Li, and X. Luo, "Data driven vibration control: A review," *IEEE/CAA J. Autom. Sin.*, vol. 11, no. 9, pp. 1898–1917, 2024. 73

[168] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 779–788, 2016. 73

[169] M. Li, A. Micheli, Y. G. Wang, S. Pan, P. Lió, G. S. Gnecco, and M. Sanguineti, "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, 2024. 73

[170] Y. Zheng, M. Jin, S. Pan, Y.-F. Li, H. Peng, M. Li, and Z. Li, "Toward graph self-supervised learning with contrastive adjusted zooming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 7, pp. 8882–8896, 2024. 73

[171] J. Li, R. Zheng, H. Feng, M. Li, and X. Zhuang, "Permutation equivariant graph framelets for heterophilous graph learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 9, pp. 11634–11648, 2024. 73

[172] D. Wu, Y. He, and X. Luo, "A graph-incorporated latent factor analysis model for high-dimensional and sparse data," *IEEE Trans. Emerg. Top. Comput.*, vol. 11, no. 4, pp. 907–917, 2023. 73

[173] X. Luo, H. Wu, Z. Wang, J. Wang, and D. Meng, "A novel approach to large-scale dynamically weighted directed network representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 9756–9773, 2022. 73

[174] B. Xu, X. Shu, J. Zhang, G. Dai, and Y. Song, "Spatiotemporal decouple-and-squeeze contrastive learning for semisupervised skeleton-based action recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, 2023. 73

[175] X. Shu, B. Xu, L. Zhang, and J. Tang, "Multi-granularity anchor-contrastive representation learning for semi-supervised skeleton-based action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022. 73

[176] B. Xu, X. Shu, and Y. Song, "X-invariant contrastive augmentation and representation learning for semi-supervised skeleton-based action recognition," *IEEE Trans. Image Process.*, vol. 31, pp. 3852–3867, 2022. 73

[177] B. Xu and X. Shu, "Pyramid self-attention polymerization learning for semi-supervised skeleton-based action recognition," *arXiv preprint arXiv:2302.02327*, 2023. 73

[178] T. Qiao, Q. Men, F. W. B. Li, Y. Kubotani, S. Morishima, and H. P. H. Shum, "Geometric features informed multi-person human-object interaction recognition in videos," in *Proc. Eur. Conf. Comput. Vis.*, 2022. 73

[179] H. Zhao and R. P. Wildes, "Where are you heading? dynamic trajectory prediction with expert goal examples," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pp. 7629–7638, 2021. 73, 84, 89, 90, 99

[180] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," *Proc. Adv. Neu. Inf. Process. Syst.*, vol. 28, 2015. 75, 79, 81

[181] W. Yang, S. Li, Z. Li, and X. Luo, "Highly accurate manipulator calibration via extended kalman filter-incorporated residual neural network," *IEEE Trans. Ind. Inform.*, vol. 19, no. 11, pp. 10831–10841, 2023. 76

[182] S. Becker, R. Hug, W. Hübner, and M. Arens, "An evaluation of trajectory prediction approaches and notes on the trajnet benchmark," *arXiv preprint arXiv:1805.07663*, 2018. 76, 85

[183] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *J. Mach. Learn. Res.*, vol. 9, no. 11, 2008. 79, 82, 91

[184] N. Sai Madiraju, S. M. Sadat, D. Fisher, and H. Karimabadi, "Deep temporal clustering: Fully unsupervised learning of time-domain features," *arXiv e-prints*, pp. arXiv–1802, 2018. 80

[185] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013. 80, 81

[186] M. Cuturi and M. Blondel, "Soft-dtw: a differentiable loss function for time-series," in *Proc. Int. Conf. Mach. Learn.*, pp. 894–903, 2017. 81

[187] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE transactions on acoustics, speech, and signal processing*, vol. 26, no. 1, pp. 43–49, 1978. 81

[188] J. Zhao and L. Itti, "shapedtw: Shape dynamic time warping," *Pattern Recognition*, vol. 74, pp. 171–184, 2018. 81

[189] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proc. Fifth Berkeley Symp. Math. Statist. Probab.*, vol. 1, pp. 281–297, 1967. 82

[190] D. A. Reynolds *et al.*, "Gaussian mixture models.," *Encyclopedia of biometrics*, vol. 741, no. 659-663, 2009. 82

[191] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016. 83

[192] Y. Li, R. Liang, W. Wei, W. Wang, J. Zhou, and X. Li, "Temporal pyramid network with spatial-temporal attention for pedestrian trajectory prediction," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 3, pp. 1006–1019, 2021. 84

[193] K. Guo, W. Liu, and J. Pan, "End-to-end trajectory distribution prediction based on occupancy grid maps," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 2242–2251, 2022. 85, 86, 88, 101

[194] J. Liang, L. Jiang, K. Murphy, T. Yu, and A. Hauptmann, "The garden of forking paths: Towards multi-future trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 10508–10518, 2020. 85, 86, 88

[195] Y. Zhang, W. Guo, J. Su, P. Lv, and M. Xu, "Bip-tree: Tree variant with behavioral intention perception for heterogeneous trajectory prediction," *IEEE Trans. Intell. Transp. Syst.*, 2023. 86, 87, 88

[196] Z. Zhou, L. Ye, J. Wang, K. Wu, and K. Lu, "Hivt: Hierarchical vector transformer for multi-agent motion prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 8813–8823, 2022. 86, 110

[197] D. Li, Q. Zhang, S. Lu, Y. Pan, and D. Zhao, "Conditional goal-oriented trajectory prediction for interacting vehicles," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 12, pp. 18758–18770, 2024. 86

[198] A. Monti, A. Bertugli, S. Calderara, and R. Cucchiara, "Dag-net: Double attentive graph neural network for trajectory forecasting," in *Int. Conf. Pattern Recognit.*, pp. 2551–2558, IEEE, 2021. 86, 89

[199] J. Liang, L. Jiang, and A. Hauptmann, "Simaug: Learning robust representations from simulation for trajectory prediction," in *Proc. Eur. Conf. Comput. Vis.*, pp. 275–292, 2020. 86, 88

[200] C. Wong, B. Xia, Z. Hong, Q. Peng, W. Yuan, Q. Cao, Y. Yang, and X. You, "View vertically: A hierarchical network for trajectory prediction via fourier spectrums," in *Proc. Eur. Conf. Comput. Vis.*, pp. 682–700, 2022. 86, 88, 101

[201] M. N. Azadani and A. Boukerche, "Capha: A novel context-aware behavior prediction system of heterogeneous agents for autonomous vehicles," *IEEE Trans. Veh. Technol*, 2023. 86, 88

[202] N. Rhinehart, K. M. Kitani, and P. Vernaza, "R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting," in *Proc. Eur. Conf. Comput. Vis.*, pp. 772–788, 2018. 88

[203] C. Tang and R. R. Salakhutdinov, "Multiple futures prediction," *Proc. Adv. Neu. Inf. Process. Syst.*, vol. 32, 2019. 88

[204] X. Xu, W. Liu, and L. Yu, "Trajectory prediction for heterogeneous traffic-agents using knowledge correction data-driven model," *Information Sciences*, vol. 608, pp. 375–391, 2022. 88, 90

[205] Y. Dong, L. Wang, S. Zhou, and G. Hua, "Sparse instance conditioned multimodal trajectory prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pp. 9763–9772, 2023. 89, 91

[206] B. Yang, F. Fan, R. Ni, H. Wang, A. Jafaripournimchahi, and H. Hu, "A multi-task learning network with a collision-aware graph transformer for traffic-agents trajectory prediction," *IEEE Trans. Intell. Transp. Syst.*, 2024. 89, 91

[207] X. Lin, T. Liang, J. Lai, and J.-F. Hu, "Progressive pretext task learning for human trajectory prediction," in *Proc. Eur. Conf. Comput. Vis.*, pp. 197–214, 2024. 89, 91

[208] 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*, pp. 0–0, 2019. 89

[209] W. Chen, H. Sang, J. Wang, and Z. Zhao, "Wtgcn: wavelet transform graph convolution network for pedestrian trajectory prediction," *Int. J. Mach. Learn. Cybern.*, pp. 1–18, 2024. 89, 91

[210] W. Chen, H. Sang, J. Wang, and Z. Zhao, "Iggcn: Individual-guided graph convolution network for pedestrian trajectory prediction," *Digital Signal Processing*, vol. 156, p. 104862, 2025. 89, 91

[211] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Proc. Adv. Neu. Inf. Process. Syst.*, vol. 28, 2015. 90

[212] Z. Chang, G. A. Koulieris, and H. P. H. Shum, "On the design fundamentals of diffusion models: A survey," *arXiv*, 2023. 90

[213] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," *arXiv preprint arXiv:2111.00396*, 2021. 98

[214] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv preprint arXiv:2312.00752*, 2023. 98

[215] M. Ronen, S. E. Finder, and O. Freifeld, "Deepdpm: Deep clustering with an unknown number of clusters," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 9861–9870, 2022. 101

[216] A. Xiao, H. Chen, T. Guo, Q. Zhang, and Y. Wang, "Deep plug-and-play clustering with unknown number of clusters," *Trans. Mach. Learn. Res.*, 2022. 101

[217] D. Kang, D. Kum, and S. Kim, "Continual learning for motion prediction model via meta-representation learning and optimal memory buffer retention strategy," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 15438–15448, 2024. 110

[218] B. Yang, F. Fan, R. Ni, J. Li, L. Kiong, and X. Liu, "Continual learning-based trajectory prediction with memory augmented networks," *Knowl.-Based Syst.*, vol. 258, p. 110022, 2022. 110

[219] N. Song, B. Zhang, X. Zhu, and L. Zhang, "Motion forecasting in continuous driving," *Proc. Adv. Neu. Inf. Process. Syst.*, vol. 37, pp. 78147–78168, 2024. 110

[220] X. Zhang, D. Song, and D. Tao, "Hierarchical prototype networks for continual graph representation learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4622–4636, 2023. 111

[221] Y. Xu, L. Wang, Y. Wang, and Y. Fu, "Adaptive trajectory prediction via transferable gnn," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 6520–6531, 2022. 111

[222] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. 1st Annu. Conf. Robot Learn.*, pp. 1–16, 2017. 111

APPENDIX A

Hardware Acknowledgements

In addition to the individuals acknowledged for their contributions to this thesis, we also recognize the essential hardware support that made this research possible.

We extend our sincere gratitude to Durham University's NVIDIA CUDA Center (NCC) GPU system (https://nccadmin.webspace.durham.ac.uk), whose computational resources were instrumental in conducting the experiments presented in this work. The NCC cluster, established through Durham University's strategic investment funds and managed by the Department of Computer Science, provided a high-performance computing environment that enabled the efficient processing of large-scale datasets and the execution of complex deep learning models. This infrastructure was vital for the rigorous testing and refinement of the methodologies developed in this thesis, and we are grateful for the access to such advanced resources, which have been critical to the success of this research.