Unified Spatial-Temporal Edge-Enhanced Graph Networks for Pedestrian Trajectory Prediction

Ruochen Li, Tanqiu Qiao, Stamos Katsigiannis, *Member, IEEE*, Zhanxing Zhu, Hubert P. H. Shum†, *Senior Member, IEEE*

Pedestrian trajectory prediction aims to forecast future movements based on historical paths. Spatial-temporal (ST) 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 ST inter-dependencies and hinders prediction performance. To address these limitations, we propose UniEdge with three major designs. Firstly, we introduce a unified ST graph data structure that simplifies high-order cross-time interactions into first-order relationships, enabling the learning of ST 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.

***Index Terms*—Pedestrian trajectory prediction, Spatial-temporal graph, Edge graph, Transformer**

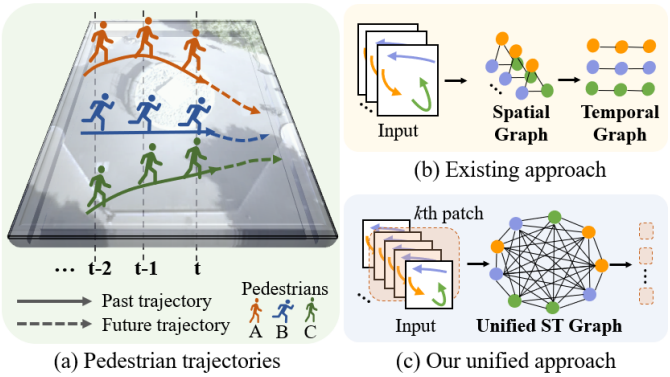
I. Introduction

THE aim of pedestrian trajectory prediction is to forecast future paths based on observed movements (Fig. 1(a)). High-precision prediction systems are crucial for applications like self-driving vehicles [1], [2] and video surveillance [3]. 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 [4].

R. Li, T. Qiao, S. Katasigiannis and H. P. H. Shum are with Durham University, UK. (e-mail: {ruochen.li, tanqiu.qiao, stamos.katsigiannis, hubert.shum}@durham.ac.uk).

Z. Zhu is with the University of Southampton, UK. (e-mail: z.zhu@soton.ac.uk)

†Corresponding author: H. P. H. Shum



(a) Pedestrian trajectories (b) Existing approach (c) Our unified approach

Fig. 1. Motivation Illustration. **(a) Real-world pedestrian trajectories** over multiple time frames. **(b) Existing ST approaches** separately model the spatial interactions among pedestrians and temporal dependencies of individuals. **(c) Our unified ST graph** integrates ST inter-dependencies and simplifies high-order cross-time interactions into first-order relationships.

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 (ST) graph architectures (Fig. 1(b)) are widely used to analyze human motions [5], [6] and pedestrian trajectories [7]–[14], 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 ST graph architectures require multiple steps to capture these interactions, where each node first aggregates spatial information at individual time steps and then addresses temporal dependencies through temporal networks. STGAT [10] combines graph attention [15] with Long Short-Term Memory (LSTM) [16] for sequential temporal modeling, while Social-STGCNN [11] and SGCN [7] advance to integrating Graph Convolutional Network (GCN) [17] with Temporal Convolutional Network (TCN) [18] 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 [19] due to increased effective resistance [20], 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 ST inter-dependencies observed in real-

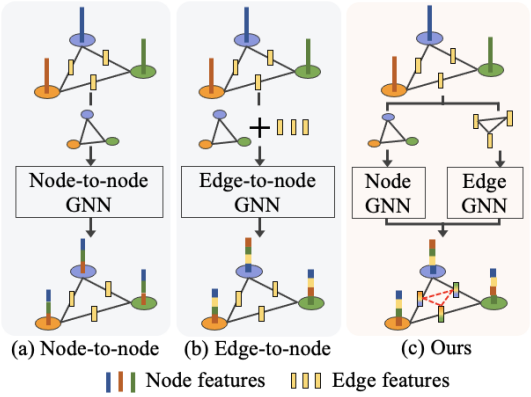


Fig. 2. Illustration of graph learning procedures. (a) Node-to-Node (N2N), (b) Edge-to-Node (E2N), and (c) Our novel dual-graph introduces the combination of N2N and Edge-to-Edge (E2E) paradigm.

world scenarios [21], [22], 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 [10]–[12], existing approaches primarily focus on **Node-to-Node** (N2N) interactions (Fig. 2(a)) through GNNs, e.g., using inverse distance [11] or attention-based [7], [10] weighting. Recent works like GroupNet [23] and HEAT [24] advance to **Edge-to-Node** (E2N) interactions (Fig. 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 [25], [26]. This fundamental limitation restricts GNNs' ability to capture the full spectrum of interaction dynamics in pedestrian behaviors, particularly in complex ST scenarios where one pedestrian's behavior can implicitly influence others through cascade effects [25].

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 ST graph segments input trajectories into patch-based structures (Fig. 1 (c)), simplifying high-order cross-time interactions into first-order relationships. This approach reduces effective resistance [20] and mitigates the under-reaching problem [19], preventing information dilution during propagation. By processing ST information jointly in a single step, each unified patch maintains natural ST 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 Fig. 2 (c). Dual-graph design provides a deeper understanding of graph topology in various domains [26], [27]. 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 [28] 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 [29] 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 [30], UCY [31] and Stanford Drone Dataset (SDD) [32]. 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 ST graph data structure that simplifies high-order cross-time interactions into first-order relationships. This enables direct learning of ST 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.

II. Related Work

A. Spatial-Temporal Modeling for Trajectory Prediction

Spatial-temporal architecture is widely used in trajectory prediction which considers both spatial interactions and temporal dependencies. Pioneering methods such as Social-LSTM [33] and Social-GAN [34] propose pooling window mechanisms to compute pedestrian spatial interactions and Long Short-Term Memory (LSTM) [16] for temporal aggregation. Due to the outstanding performance of graphs in representation learning, they are widely used to represent pedestrian interactions. STGAT [10] and Social-BiGAT [9] employ Graph Attention Network (GAT) [15] to measure interactions strength and LSTM to capture temporal dependencies. Social-STGCNN [11] proposes to use a Graph Convolutional Network (GCN) [17] combined with the TCN [18] to model pedestrian trajectories. To simplify the complexity of the graph, sparse GCN-based approaches [7], [8], [13] further propose directed graphs to dynamically update graph topology during message passing, and TCN is used to learn temporal correlations. In recent years,

group-wise methods [12], [23] have garnered attention due to their superior capability in analyzing group behaviors.

However, these methods characterize spatial interactions and temporal dependencies separately, leading to diluted information and delayed responses in complex scenarios. To this end, we introduce unified ST graphs that transform high-order interactions into simplified first-order relationships, efficiently capture ST inter-dependencies.

B. Graph Neural Networks

Graph Neural Networks (GNNs) have gained considerable traction in computer vision tasks due to their ability to model complex relationships and interactions between entities. Harnessing their representational power, GNNs have been successfully applied across various domains, including human skeleton analysis [35]–[37], drug design [38], and recommendation systems [39]. In the trajectory prediction domain, the evolution of GNN architectures reflects increasingly sophisticated approaches to modeling social interactions. Early works [11], [40] primarily relied on the representation capabilities of GCN to model social interactions. Following the success of the self-attention mechanism [29], subsequent studies [7]–[10], [41] enhanced this N2N paradigm by incorporating attention-based GNNs, enabling more adaptive and context-aware relationship modeling. Recent works have begun exploring E2N interactions to capture richer relational information between edge and node. GroupNet [23] pioneered this direction by introducing interaction strength and category features to enhance edge significance beyond simple connections. Following this trend, GC-VRNN [42], HEAT [24], and MFAN [43] further advanced E2N modeling by integrating edge features into node embeddings, enhancing relational reasoning capabilities.

However, existing trajectory prediction methods primarily focus on updating node representations. In this paper, we introduce Edge-to-Edge-Node-to-Node Graph Convolution (E2E-N2N-GCN), a dual-graph architecture that jointly captures both explicit N2N social interactions and implicit E2E influence propagation, providing a more comprehensive modeling of social interactions.

C. Trajectory Predictor

Trajectory prediction has seen various architectural developments. Early RNN-based approaches [3], [33], [34], [44]–[46] process trajectories sequentially through hidden states. Among these methods, Social-LSTM [33] processes trajectories where hidden states are updated recursively to capture temporal patterns. Recent works like Social-VAE [44] and ATP-VAE [45] combine RNN with variational autoencoders to model the uncertainty in trajectory predictions, achieving promising results. Subsequently, TCN-based predictor [7], [8], [11], [43] emerged as an alternative approach. Social-STGCNN [11] combines graph convolutions with TCN to achieve efficient parallel processing through increased receptive fields. SGCN [7] further advances this design by introducing sparse attention mechanisms to adaptively aggregate temporal features. Recently, transformer-based methods [2], [41], [47] have gained prominence in trajectory prediction, where self-attention mechanisms compute pairwise interactions between all time steps, enabling global temporal modeling without the constraints of sequential processing or fixed receptive fields.

However, RNNs suffer from long-term dependencies due to their auto-regressive nature, and TCNs are limited by fixed receptive fields due to their convolutional structure, while full transformer models have high computational costs. To balance modeling capability and efficiency, we propose a Transformer encoder-based predictor that learns global dependencies within the sequence without high computational costs.

III. Methodology

A. 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 $i \in [1, \dots, N]$ can be represented as $X_i = (x_t^i, y_t^i),\ t \in [1, \dots, 0]$ and its ground-truth future trajectories can be defined as $Y_i = (x_t^i, y_t^i),\ t \in [1, \dots, T_{pred}]$. For N pedestrians, the observed and ground-truth future trajectories are $\mathbf{X} = [X_1, X_2, \dots, X_N] \in \mathbb{R}^{N \times T_{obs} \times 2}$ and $\mathbf{Y} = [Y_1, Y_2, \dots, Y_N] \in \mathbb{R}^{N \times T_{pred} \times 2}$ respectively, where 2 denotes the 2D coordinates. Our proposed UniEdge aims to learn a prediction function $F_{pred}(\cdot)$ that minimizes the differences between the predicted trajectories $\hat{\mathbf{Y}} = F_{pred}(X)$ and the ground-truth future trajectories $\mathbf{Y}$. Instead of directly predicting absolute coordinates, we follow [7], [11]–[13] 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 $= \|\mathbf{v}\|_2$, and pedestrian movement angles $= \text{angle}(\mathbf{v})$, where $\|\cdot\|_2$ denotes the vector 2-norm and angle$(\cdot)$ is the function that computes the angle of the velocity vectors. We follow [48] 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:

$$X = \text{CONCAT}(f(\mathbf{v}; W_v); f(\ ; W_{norm}); f(\ ; W_{angle}));$$

where $X \in \mathbb{R}^{N \times T_{obs} \times D}$, N and T_{obs} represent the total number of pedestrians and time steps, respectively, and D denotes the embedded feature dimension. Here, $f(\ ;\ )$ represents Multi-Layer Perceptron (MLP) for feature embedding, and W represents the corresponding weights.

B. Unified ST Graph

Previous trajectory prediction methods often adopt a two-step approach, separately modeling pedestrian spatial interactions and individual temporal dependencies [7], [11], [33]. This approach is limited in capturing high-order cross-time

Fig. 3. Overview of the proposed UniEdge. (a) Construction of patch-based uni ed ST graphs that simplify cross-time interactions into rst-order relationships, (b) Edge-to-Edge-Node-to-Node Graph Convolution (E2E-N2N-GCN) that jointly processes N2N interactions and E2E in uence propagation, and (c) Transformer Encoder-based trajectory predictor.

Fig. 4. Comparison of effective resistance (R_{ij}) between traditional ST approach (left, $R_{ij} = 1.50$) and our uni ed ST graph (right, $R_{ij} = 0.27$). Lower R_{ij} indicates better message propagation ef ciency.

interactions, which require multi-step aggregation. Such multi-step processing increases the effective resistance - a measurement of graph connectivity that quanti es the ef ciency of information ow between nodes [20], [49]. High effective resistance impedes graph message-passing, leading to under-reaching problem [19], where message ows from distant nodes are diluted and compressed.

To address these challenges, we propose a uni ed ST graph to simplify high-order cross-time interactions among pedestrians into rst-order relationships, enabling direct learning of ST inter-dependencies, and preserving high-order interactions without information dilution. This design signi cantly reduces the effective resistance during message passing, improving information ow ef ciency [20], [49] and alleviating the risk of under-reaching [19]. Fig. 4 illustrates the difference in effective R between the message-passing paradigms of traditional ST approach and our uni ed approach:

$$R_{ij} = (e_i - e_j)^T L^+ (e_i - e_j) \tag{1}$$

where L^+ denotes the Moore-Penrose pseudoinverse of the graph Laplacian matrix representing the graph connectivity [50], and e_i, e_j are standard basis vectors corresponding to nodes i and j. Lower R_{ij} values indicate better message propagation ef ciency between nodes.

To reduce computational overhead in processing entire sequences and to better capture ne-grained pedestrian dynamics, we adopt a patch-based strategy akin to the local receptive elds used in convolution kernel for image processing. [51]. Speci cally, to construct the uni ed ST graph depicted in Fig. 3 (a), the input features are segmented into K overlapping patches across the temporal dimension T_{obs}. These patches are de ned by a length L and a stride S, yielding $K = \frac{T_{obs} - L}{S} + 1$. For each patch k, ranging from 1 to K, a graph $G^k_{node} = (Z^k; A^k_{node})$ is constructed. Here $Z^k \in R^{NL \times D}$ represents the node features, and $A^k_{node} \in R^{NL \times NL}$ denotes the node adjacency matrix, which encapsulates the node connections. This con guration further bene ts 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 [9], [10], [52] to initialize interactions strength for the kth graph G^k as:

$$H^k_{node} = GAT(Z^k; A^k_{node}); \tag{2}$$

where each node $H^k_{node;i}$ is embedded as:

$$H^k_{node;i} = \Big(\sum_{j \in N(i) \cup \{i\}} A^k_{i;j} Z^k_j \Big); \tag{3}$$

Fig. 5. Illustration of edge graph construction from a uni ed ST graph using the rst-order boundary operator B_1. Nodes are represented by numbers, and edges connecting these nodes are labeled with letters. Applying the rst-order boundary operator transforms each edge into a node in the edge graph, with connections formed based on shared nodes in the original graph.

$$\alpha^k_{i,j} = \frac{\exp\left(a^\top [Z^k_i \| Z^k_j]\right)}{\sum_{j \in \mathcal{N}(i) \cup \{i\}} \exp\left(a^\top [Z^k_i \| Z^k_j]\right)}, \quad (4)$$

where $\sigma(\cdot)$ is transformation function, $\phi(\cdot)$ and $\psi(\cdot)$ denote activation functions, $\mathcal{N}(\cdot)$ is the neighbor set of node i, and $a^\top$ represents learnable parameters. Attention coef cient $\alpha^k_{i,j}$ represents the weights between two nodes. During training, these weight coef cients are dynamically updated to re ect the importance of each node's contribution to its neighbors.

C. E2E-N2N Graph Convolution (E2E-N2N-GCN)

Previous pedestrian trajectory models typically adopt node-centric approaches, such as N2N [7], [11]–[13], [53] and E2N [23], [24] 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 in uence each other.

To address this limitation, we propose a novel Edge-to-Edge-Node-to-Node Graph Convolution (E2E-N2N-GCN) module (Fig. 3 (b)), a dual-graph architecture that leverages the rst-order boundary operator to construct edge graphs. By jointly modeling both explicit N2N social interactions among pedestrians and implicit E2E in uence propagation across interaction patterns, our approach enables more comprehensive modeling of complex pedestrian behaviors. This dual-graph design allows each uni ed ST graph to capture how interaction patterns evolve and in uence each other through connected edges, leading to more accurate trajectory predictions.

To construct the edge graph, we apply the rst-order boundary operator B_1 to transform it into its corresponding undirected edge graph $G^k_{edge} = (E^k; A^k_{edge})$, where E^k represents the node features in the edge graph, and 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. Fig. 5 illustrates this process, effectively showing how relationships are rede ned to highlight deeper interaction dynamics.

To analyze and update the feature propagation of each edge graph, we employ the rst-order Hodge Laplacian [25], [26] to analyze and learn the dynamics within these edge graphs:

$$L_1 = B_1^\top B_1 + B_2^\top B_2; \quad (5)$$

where L_1 represents rst-order Hodge Laplacian operator, and $B_1^\top$ captures and enhances edge relationships, focusing on direct interactions. 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) [25], [26] to obtain the high-level edge embedding H^k_{edge} for each edge graph k:

$$\begin{aligned} H^k_{edge} &= \mathrm{HLLConv}(E^k; A^k_{edge}) \\ &= \tilde{\sigma}_1(E^k) \\ &= \sum_{j=0}^{J-1} \theta_j \psi_j(L_1) E^k; \end{aligned} \quad (6)$$

where $\tilde{\sigma}_1$ is a spectral lter based on L_1 applied to update edge features E^k, with θ_j representing learnable parameters, and $\psi_j(\cdot)$ indicates the Laguerre polynomial functions. Detailed explanations of spectral lter $\tilde{\sigma}_1$ are shown in Algorithm 1. Finally, after obtaining the embedded node features H^k_{node} and edge features H^k_{edge} for the kth uni ed ST graph, we leverage a fusion GCN to integrate node and edge embeddings, enhancing the understanding of graph dynamics. Speci cally, we incorporate normalized edge embedding as weights into the aggregation process of GCN:

$$H^k = \mathrm{GCN}(H^k_{node}; H^k_{edge}; A^k_{node}); \quad (7)$$

and each node i in the graph is embedded as:

$$H^k_i = \left(H^k_{node;i} \right) + \sum_{j \in \mathcal{N}(i)} \left(H^k_{edge;ij} \right) \left(H^k_{node;j} \right), \quad (8)$$

where $\phi(\cdot)$ and $\psi(\cdot)$ are linear transformations for node and edge features [25], with $\sigma(\cdot)$ as the activation function.

Algorithm 1 Hodge-Laplacian Laguerre Convolution

Input: rst-order Hodge Laplacian $L_1 = B_1^\top B_1 + B_2^\top B_2$
Output: Spectral lter $\tilde{\sigma}_1$
Step 1: Perform eigen-decomposition on L_1:

$$L_1 \phi^i_1 = \lambda^i_1 \phi^i_1$$

to obtain the orthonormal bases ϕ^i_1 for $i \in [0, 1, 2, \ldots, \lambda_1]$. The spectral lter $\tilde{\sigma}_1$ of the 1-st order HL can be represented as $\tilde{\sigma}_1(\lambda, \lambda) = \sum_{i=0}^{\lambda_1 - 1} \tilde{\sigma}_1(\lambda^i_1) \phi^i_1(\lambda) \phi^i_1(\lambda)$.
Step 2: Approximate the spectral lter $\tilde{\sigma}_1(\lambda_1)$ by Laguerre polynomial functions:

$$\tilde{\sigma}_1(\lambda_1) = \sum_{j=0}^{J-1} \theta_j \psi_j(\lambda_1)$$

where θ_j is the jth expansion coef cient with jth Laguerre polynomial, and $\psi_j(\lambda)$ is written in a recurrence format as:

$$\psi_{j+1}(\lambda_1) = \frac{(2j + 1 - \lambda_1)\psi_j(\lambda_1) - j\psi_{j-1}(\lambda_1)}{j + 1}$$

where $\psi_0(\lambda_1) = 1$ and $\psi_1(\lambda_1) = 1 - \lambda_1$.

Fig. 6. Illustration of the Transformer encoder-based predictor.

D. Transformer Encoder Predictor

Temporal dependency modeling in trajectory prediction has evolved through various architectures. RNNs [33], [34] and TCNs [7], [11] have been widely adopted, they suffer from limited receptive fields and struggle to capture long-range dependencies. Although Transformer encoder-decoder architectures [2], [29], [41] 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 Fig. 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 [54] for more accurate predictions. We simply stack the graph embeddings H^k output by E2E-N2N-GCN across all patches to obtain the integrated feature representations H:

$$H = \mathrm{STACK}(H^1; H^2; \ldots; H^K) \in \mathbb{R}^{K \times (NL) \times D}. \quad (9)$$

We perform temporal average pooling across the L channel, and the output $H \in \mathbb{R}^{N \times K \times D}$ is served as the historical input tokens. We then initialize a learnable placeholder to form the padded future tokens as $F \in \mathbb{R}^{N \times T_{pred} \times D}$. 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 [29], [55], 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 H and padded future tokens F, resulting in the concatenated feature representation $\hat{H}_{in} \in \mathbb{R}^{N \times (K+T_{pred}) \times D}$. We further enhance these tokens with a learnable additive positional embedding $P \in \mathbb{R}^{N \times (K+T_{pred}) \times D}$ [29] 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 $\hat{Y} \in \mathbb{R}^{N \times (K+T_{pred}) \times D}$:

$$\hat{Y} = \mathrm{Encoder}(\hat{H}_{in} + P);$$
$$\hat{H}_{in} = [H \,\|\, F]; \quad (10)$$

where $[\,\|\,]$ denotes the concatenation operation along the temporal dimension. Note that $\hat{Y}$ 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 F. The architecture of the Transformer encoder and the learning process are shown in Fig. 6. Similarly to [7], [8], [11], we employ the bi-variate Gaussian loss function $L_{prediction}$ to optimize the trajectory prediction:

$$L_{prediction} = \sum_{t=1}^{T_{pred}} \log P((x_t, y_t) \mid \hat{\mu}_t, \hat{\sigma}_t, \hat{\rho}_t); \quad (11)$$

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

E. 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.

IV. Experiments

A. Experimental Setup

We evaluate the proposed UniEdge on multiple benchmark datasets, including ETH [30], UCY [31], and Stanford Drone Dataset (SDD) [32]. 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 [7], [33], [56], 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 [7], [11]–[13], [34], [41] and use the leave-one-out strategy for training and evaluation. For SDD, we follow the existing train-test split [12]–[14] to train and test our proposed method. During training, we employ data augmentation following [56] 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 [33], [34] and sampling strategy [12] that generates 20 predictions from the predicted distributions; the best sample is used to compute

Table I
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 [34]	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 [11]	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 [7]	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 [12]	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 [44]	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 [57]	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 [23]	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 [14]	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 [53]	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 [58]	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 [59]	CVPR'23	0.40/0.61	0.10/0.18	0.23/0.43	0.18/0.32	0.10/0.23	0.21/0.35
EigenTrajectory [13]	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 [41]	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 [60]	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 [43]	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 [48]	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 [45]	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 [47]	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 [61]	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 [62]	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

the evaluation metrics. Average Displacement Error (ADE) and Final Displacement Error (FDE) [7], [11], [33], [34] are used as evaluation metrics:

$$\text{ADE} = \frac{1}{N\,T_{pred}} \sum_{i=1}^{N} \sum_{t=1}^{T_{pred}} \sqrt{(x_t^i - \hat{x}_t^i)^2 + (y_t^i - \hat{y}_t^i)^2};$$

$$\text{FDE} = \frac{1}{N} \sum_{i=1}^{N} \sqrt{(x_{T_{pred}}^i - \hat{x}_{T_{pred}}^i)^2 + (y_{T_{pred}}^i - \hat{y}_{T_{pred}}^i)^2}; \tag{12}$$

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.

B. Baseline Methods

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

Graph-based methods: Social-STGCNN [11]: an approach that models ST pedestrian interactions through graphs; SGCN [33]: an approach that models ST interactions through sparse directed spatial graph and sparse directed temporal graph; GP-Graph [12]: an approach that considers group-based pedestrian behaviors; Graph-TERN [14]: an approach that integrates multi-relational graph and control endpoint for trajectory prediction; EigenTrajectory(+SGCN) [13]: a model that learns trajectories in eigenspaces and graph representations. MFAN [43]: an approach that models ST interactions for both edges and nodes. HighGraph [62]: 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 [34]: a method that uses pooling window module with Generative Adversarial Network (GAN) to generate diverse trajectories; Social-VAE [44]: a method that employs timewise variational autoencoder(VAE) and attention mechanism to generate trajectories; GroupNet [23]: a method that introduces multiscale hypergraph with edge strength, utilizing conditional-VAE (CVAE) to generate trajectories; MSRL [53]: a method that models multi-stream interactions for trajectory prediction based on CVAE; MRGTraj [47]: a method based on CVAE and non-auto-regressive transformer encoder to generate diverse trajectories; ATP-VAE [45]: an attention-based VAE architecture for trajectory prediction; LED [58]: a multi-modal framework based on diffusion for prediction; SingularTrajectory [61]: a diffusion framework based on singular projection and adaptive anchor to generate trajectories.

Other methods: MemoNet [57]: an approach based on the retrospective-memory bank for trajectory representations; EqMotion [59]: an approach that models trajectories via equivariant dynamics and invariant interaction; TUTR [41]: a transformer-based framework; SMEMO [60]: an approach that models trajectories through social memory modules; DDL [48]: goal-based transformer for trajectory prediction.

C. Quantitative Comparison

1) ETH and UCY Datasets: Table I 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

Table II
RESULTS ON THE STANFORD DRONE DATASET (SDD) FOR PEDESTRIAN TRAJECTORY PREDICTION

Method	Venue/Year	ADE(#) / FDE(#) SDD
Social GAN [34]	CVPR'18	27.23/41.44
Social-STGCNN [11]	CVPR'20	26.46/42.71
GroupNet [23]	CVPR'22	9.31/16.11
MemoNet [57]	CVPR'22	8.56/12.66
GP-Graph [12]	ECCV'22	9.10/13.80
MSRL [53]	AAAI'23	8.22/13.39
Graph-TERN [14]	AAAI'23	8.43/14.26
LED [58]	CVPR'23	8.48/11.66
EigenTrajectory [13]	ICCV'23	8.05/13.25
TUTR [41]	ICCV'23	7.76/12.69
SMEMO [60]	TPAMI'24	8.11/13.06
MFAN [43]	PR'24	9.69/14.51
HighGraph [62]	CVPR'24	7.98/11.42
UniEdge (Ours)	-	7.51/10.89

signi cant 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 ow 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, 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.

2) SDD Dataset:Table II 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.

D. Qualitative Comparison

1) Trajectory Visualization Comparison:In this section, we compare the most likely predictions between our UniEdge and previous graph-based methods, GP-Graph [12], Graph-TERN [14] and EigenTrajectory [13] on the ETH and UCY datasets.

As shown in Fig. 7, our prediction results are signi cantly 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 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 signi cantly suffer pedestrian collision issues. Our UniEdge demonstrates superior capability in capturing non-linear 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.

2) Distribution Visualization Comparisons:In this section, we further compare the predicted distributions of UniEdge with GP-Graph [12], Graph-TERN [14] and EigenTrajectory [13] on the ETH and UCY datasets. As shown in Fig. 8, 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.

E. Ablation Study and Model Analysis

1) Model Component Analysis:To verify the in uence of each module incorporated in our UniEdge, we conduct ablation studies on the ETH and UCY datasets, which contain ve different social scenarios. The results of these studies are detailed in Table III. 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). Speci cally, 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 in uence propagation, the performance dropped 16.7% in ADE and 20.0% in FDE. Variant

Fig. 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.

(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, the UNIV subset, which contains the most pedestrians and the most complex interactions [63], 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.

Table III
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)		X	X	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)	X		X	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)	X	X		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	X	X	X	0.36/0.46	0.11/0.17	0.19/0.28	0.14/0.20	0.11/0.16	0.18/0.25

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 IV. The baseline GCN [17] exhibits limited performance due to its uniform neighborhood aggregation strategy. GraphSage [64] 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.

2) 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 V, we examine three edge feature types: a Gaussian kernel $E_{i,j} = \exp\frac{d_{i,j}}{2\ ^2}$, which captures spatial relationships through the distance $d_{i,j}$ between nodes i and j, and the standard deviation ; a reciprocal distance kernel $E_{i,j} = \frac{1}{d_{i,j} +}$, highlighting inverse distance to represent pedestrian interactions; and a

Fig. 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.

Table IV
FEATURE EMBEDDING ANALYSIS ON THE ETH AND UCY DATASETS

Method	ADE(#) / FDE(#)					
	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
w/ GCN [17]	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 [64]	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

Table V
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

Euclidean distance kernel $E_{i,j} = d_{i,j}$, quantifying node relationships based on direct distance. Results in Table V 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.

3) Trajectory Predictor Analysis:To evaluate the effectiveness of the core modules in our Transformer encoder-based predictor and the corresponding padding approach, we conduct extensive experiments on the predictor design. The

Table VI
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

results are presented in Table VI. 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.

Table VII
TRAJECTORY PREDICTOR COMPARISON ANALYSIS ON THE ETH AND UCY DATASETS

Trajectory Predictor	ADE(#) / FDE(#)					
	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
RNN-based [16]	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 [18]	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

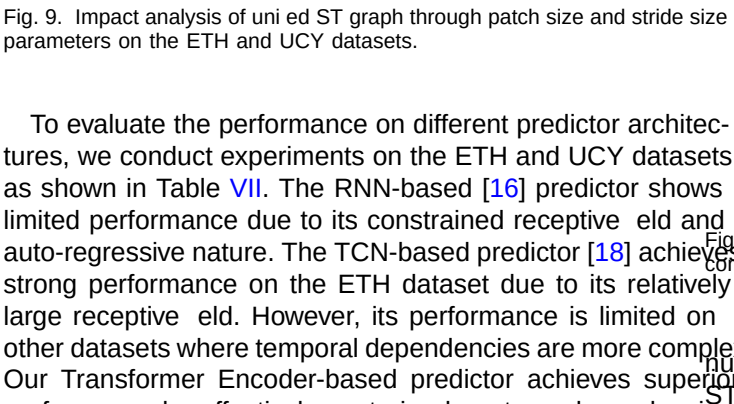


Fig. 9. Impact analysis of uni ed ST graph through patch size and stride size parameters on the ETH and UCY datasets.

To evaluate the performance on different predictor architectures, we conduct experiments on the ETH and UCY datasets, as shown in Table VII. The RNN-based [16] predictor shows limited performance due to its constrained receptive eld and auto-regressive nature. The TCN-based predictor [18] achieves strong performance on the ETH dataset due to its relatively large receptive eld. 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 [29], [55].

4) Uni ed ST Graph Analysis:In this section, we analyze the effectiveness and impact of our proposed uni ed ST graph data structure while keeping other components xed. The construction of this data structure is controlled by two key parameters: patch size and stride size. We conduct experiments on the ETH and UCY datasets to thoroughly analyze how these parameters affect the model's ability to capture ST inter-dependencies.

As shown in Fig. 9 (left), we evaluate how patch size affects uni ed ST graph construction. A patch size of 1 reduces our model to traditional two-stage ST approaches [7], [10], [11], [13], where cross-time interactions are not explicitly modeled. The model achieves optimal performance with a patch size of 3, effectively capturing local ST 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 Fig. 9 (right). The stride size determines the number of uni ed ST 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 uni ed ST graphs. A stride size of 1 yields the best performance in both ADE and FDE metrics, as it enables the capture of more ne-grained cross-time interactions through increased

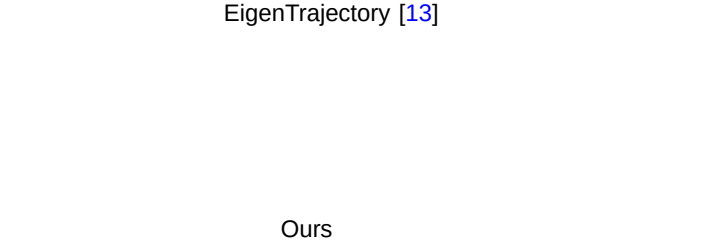


Fig. 10. Edge weight visualization of traditional two-stage ST approach EigenTrajectory and our UniEdge. Historical trajectories are in blue and ground-truth trajectories are in red.

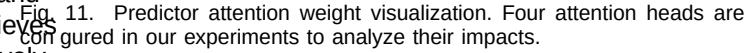


Fig. 11. Predictor attention weight visualization. Four attention heads are con gured in our experiments to analyze their impacts.

number of uni ed ST graphs. The increased number of uni ed ST graphs enables the transformer encoder-based predictor to leverage more ST contexts for enhanced performance.

5) Edge Weight Visualization:To provide qualitative insights into the differences between our UniEdge model and conventional ST architecture, we visualize the edge weights of our uni ed ST graph and EigenTrajectory [13]. Fig. 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 uni ed ST 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 ST frameworks may overlook.

6) Predictor Attention Weight Visualization:This section visualizes the attention weights of our Transformer encoder-based predictor to examine interactions between learnable placeholder padding and historical contexts. As shown in Fig. 11, the attention heads demonstrate distinct specialization patterns: heads 1 and 2 focus on temporal dependencies within historical trajectories, while heads 3 and 4 establish connections between learnable padding and relevant historical tokens. This specialized distribution reveals how the model decomposes trajectory prediction tasks and provides interpretable insights into its temporal information processing.

7) Complexity and Ef ciency Analysis: To evaluate the ef - ciency and computational complexity of UniEdge, Table VIII presents a comprehensive analysis of model complexity and computational ef ciency 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 [13], although UniEdge contains more parameters, it maintains competitive inference time while achieving signi cant improvements in prediction accuracy (18.2% in ADE and 30.6% in FDE). For common real-world trajectory prediction scenarios such as traf c collision avoidance and anomaly detection, we believe this trade-off is justi ed as prediction accuracy takes precedence over computational complexity, especially since higher accuracy in these applications can signi cantly reduce the risk of severe outcomes. Compared to transformer-based temporal modeling methods like TUTR [41] and MRGTraj [47], UniEdge demonstrates superior ef ciency with signi cantly 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 ef ciency and accuracy.

Table VIII
COMPLEXITY AND INFERENCETIME ANALYSIS. ALL MODELSARE EVALUATED ON NVIDIA RTX3080 GPU

Methods	Param 10^6	FLOPs (M)	Infer. Time (ms)	ADE/FDE(#)
Non-Transformer Temporal Modeling				
Social-VAE [44]	2.15	292.95	40.27	0.21/0.33
Graph-TERN [14]	0.05	22.59	40.15	0.24/0.38
EqMotion [59]	3.02	7.75	35.92	0.21/0.35
EigenTrajectory [13]	0.02	1.36	22.26	0.22/0.36
Transformer-based Temporal Modeling				
TUTR [41]	0.44	64.54	20.21	0.21/0.36
MRGTraj [47]	4.35	580.38	26.51	0.26/0.46
UniEdge (Ours)	0.34	26.49	27.02	0.18/0.25

F. Discussion

In this section, we discuss potential reasons for the relatively lower performance of graph-based trajectory prediction approaches [13], [14], [43], [62] on the ETH subset, as compared to other scenarios. As indicated in Table IX, the test set for the ETH subset averages only 2.59 pedestrians per sample, signi cantly less than other subsets, particularly the UNIV subset, which averages 25.70 pedestrians per sample. This stark variation in pedestrian density impacts the ef cacy of graph-based methods, which rely on graph structures to model social interactions [7], [56]. The relatively sparse graph connectivity in the ETH scenario may impair message passing, potentially limiting the model's ability to effectively propagate and re ne 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 Fig. 12. The example

Table IX
DATASET STATISTICS ONTHE 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

Fig. 12. Sample scenario in ETH dataset. Historical trajectories are in blue, ground-truth trajectories are in red.

shows how UniEdge constructs a uni ed ST 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 [65] 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 re ne 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 [66], addressing the limitations of our current xed patch size strategy and potentially improving prediction accuracy in varying crowd scenarios. Second, we plan to incorporate multimodal data sources, particularly environmental contextual images [3], [46], 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 [67], [68] to improve deployment ef ciency in real-time applications, reducing computation latency while maintaining prediction accuracy.

V. CONCLUSION

In this paper, we introduce a novel UniEdge framework for trajectory prediction. Firstly, to capture high-order cross-time social interactions, we propose a patch-based uni ed ST graph architecture that simpli es high-order cross-time interactions to rst-order relationships. Our approach reduces the steps required to aggregate spatial-temporal dependencies and effectively addresses the under-reaching problem by directly linking high-order nodes, offering a consistent improvement over traditional methods. Secondly, we propose the E2E-

N2N Graph Convolution (E2E-N2N-GCN), a dual-graph architecture that jointly models explicit N2N social interactions and implicit E2E influence propagation through first-order boundary operators. This design enables comprehensive modeling of both individual behaviors and collective interaction dynamics. Finally, we propose a Transformer encoder-based trajectory predictor with placeholder-based techniques, providing a global view of trajectory embeddings, and improving the prediction performance. Experiments on datasets demonstrate that UniEdge consistently outperforms state-of-the-art methods in both quantitative and qualitative evaluations.

ACKNOWLEDGMENT

This project is supported in part by the EPSRC NortHFutures project (ref: EP/X031012/1).

REFERENCES

[1] 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.* IEEE, 2015, pp. 454–460.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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 Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8994–9003.

[8] 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.* IEEE, 2022, pp. 2346–2350.

[9] 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.

[10] 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.*, 2019, pp. 6272–6281.

[11] 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 Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 14 424–14 432.

[12] I. Bae, J.-H. Park, and H.-G. Jeon, "Learning pedestrian group representations for multi-modal trajectory prediction," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2022, pp. 270–289.

[13] I. Bae, J. Oh, and H.-G. Jeon, "Eigentrajectory: Low-rank descriptors for multi-modal trajectory forecasting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023.

[14] I. Bae and H.-G. Jeon, "A set of control points conditioned pedestrian trajectory prediction," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 5, 2023, pp. 6155–6165.

[15] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," *Proc. Int. Conf. Learn. Represent.*, 2018.

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

[17] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent.*, 2017.

[18] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[19] 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, no. 13, 2024, pp. 14 175–14 183.

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

[21] 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, no. 14, 2024, pp. 15 715–15 724.

[22] 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.

[23] 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.

[24] X. Mo, Y. Xing, and C. Lv, "Heterogeneous edge-enhanced graph attention network for multi-agent trajectory prediction," *arXiv preprint arXiv:2106.07161*, 2021.

[25] 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.

[26] 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*. Springer, 2023, pp. 278–290.

[27] 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.

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

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[30] 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.*, 2009, pp. 261–268.

[31] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," in *Computer graphics forum*, vol. 26, no. 3. Wiley Online Library, 2007, pp. 655–664.

[32] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 549–565.

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

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

[35] 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.

[36] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[37] Y. Liu, H. Zhang, Y. Li, K. He, and D. Xu, "Skeleton-based human action recognition via large-kernel attention graph convolutional network," *IEEE Trans. Vis. Comput. Graph.*, vol. 29, no. 5, pp. 2575–2585, 2023.

[38] X.-S. Li, X. Liu, L. Lu, X.-S. Hua, Y. Chi, and K. Xia, "Multiphysical graph neural network (mp-gnn) for covid-19 drug design," *Briefings in bioinformatics*, vol. 23, no. 4, p. bbac231, 2022.

[39] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proc.Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval*, 2019, pp. 165–174.

[40] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Int. Joint Conf. Artif. Intell.*, 2018.

[41] L. Shi, L. Wang, S. Zhou, and G. Hua, "Trajectory unified transformer for pedestrian trajectory prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 9675–9684.