PedAST-GCN: Fast Pedestrian Crossing Intention Prediction Using Spatial–Temporal Attention Graph Convolution Networks

Yancheng Ling, Zhenliang Ma, Qi Zhang, Bangquan Xie, and Xiaoxiong Weng

***Abstract*— Accurately and timely predicting pedestrian crossing intentions in real-time is critical for operating intelligent vehicles on roads. Although existing models achieve promising accuracy using complex models and video image data, they are constrained for real-time practical use given the high model complexity, time-consuming data preprocessing, and low-quality image data in the wild. To address these, the paper proposes a Spatial-Temporal Attention Graph Convolution Network model for fast pedestrian crossing intention prediction (PedAST-GCN). It uses a lightweight GCN model as the backbone network with simple but robust graph representations of pedestrian crossing intention modality features, including pedestrian pose, bounding box, and vehicle speeds. The model is validated by comparing it with state-of-the-art models on two large-scale public datasets (JAAD and PIE). The results highlight the better performance of the PedAST-GCN model for pedestrian crossing intention prediction in terms of accuracy and computation times. The ablation analysis confirms the value of the backbone layer and graph design, the designed modality features, the effectiveness of attention mechanisms in capturing long-term dependencies (spatial-temporal attention) and fusing heterogeneous features (modality attention), and the robust performance across various observation lengths and in the presence of noisy data.**

***Index Terms*— Pedestrian crossing intention prediction, graph convolution networks, modality features, video image data.**

I. Introduction

IMPROVING road safety with autonomous vehicles (AVs) is crucial for pedestrian safety. Studies have reported that the majority of autonomous vehicle accidents occur due to a failure to accurately predict pedestrian behavior [1]. Understanding pedestrian behavior [2], [3] at crossings is indispensable for developing AVs, in which predicting pedestrian crossing intention is particularly important for safe AV driving and avoiding collisions.

The State-Of-The-Art (SOTA) models for pedestrian crossing intention prediction can be categorized as (1) Convolutional Neural Network (CNN) based methods [4], [5], [6], [7], [8], [9], [10], (2) Recurrent Neural Networks (RNN) based methods [11], [12], [13], [14], [15], [16], and (3) Graph Convolution Networks (GCN) based methods [17], [18], [19]. The CNN-based models have powerful feature extraction capabilities for images. They can use as model inputs either a single image [5], [10] or a sequence of images [20], [21]. However, the CNN-based models are sensitive to image quality since they can only take a single type of data which is highly influenced by environmental factors in the wild. Compared to CNN-based models, RNN-based models are capable of capturing long-term dependencies of images and taking inputs of various types of data, such as bounding box sequences, pedestrian crop image sequences, skeleton sequences, ego-vehicle velocity sequences, and semantic segmentation maps. However, both RNN and CNN-based models have significantly high computational complexity with a large number of model parameters, which hinders their applicability for practical use for real-time prediction of pedestrian crossing intentions, particularly considering limited onboard computing resources on AVs. Recently, GCN-based models [19] are reported to achieve promising performance for predicting pedestrian crossing intentions, in terms of accuracy and speed. However, the current model architecture, where the pedestrian pose is treated as the main branch and other features are added as the second branch, may limit its performance in diverse scenarios by not fully leveraging the potential of various features.

The data representation has an important influence on prediction accuracy and speed. Cadena et al. [17], Zhang et al. [18] and Ling et al. [22] introduced the spatial-temporal graph convolution networks (ST-GCN) as the backbone network and used pose keypoints as model input, which achieved a high prediction speed. However, their performance is limited. For example, the pure pose keypoints can only capture the action and state information of the pedestrian, but lack interaction information with vehicles. Besides, the pose keypoints detection depends on the image quality, whose precision decreases significantly when the pedestrian crop images are blurry. To improve the prediction accuracy, Cadena et al. [19] designed a Pedestrian Graph + model

Manuscript received 10 April 2023; revised 24 October 2023, 4 January 2024, and 27 February 2024; accepted 5 May 2024. This work was supported by the China Scholarship Council under Grant 202206150027. The Associate Editor for this article was H. Han. *(Corresponding author: Zhenliang Ma.)*

Yancheng Ling, Bangquan Xie, and Xiaoxiong Weng are with the School of Civil Engineering Transportation, South China University of Technology, Guangzhou 510000, China (e-mail: lingyancheng@126.com; ctbqx51@mail.scut.edu.cn; ctxxweng@scut.edu.cn).

Zhenliang Ma and Qi Zhang are with the Department of Civil and Architectural Engineering, KTH Royal Institute of Technology, 114 28 Stockholm, Sweden (e-mail: zhema@kth.se; qzhan@kth.se).

This article has supplementary downloadable material available at https://doi.org/10.1109/TITS.2024.3398252, provided by the authors.

Digital Object Identifier 10.1109/TITS.2024.3398252

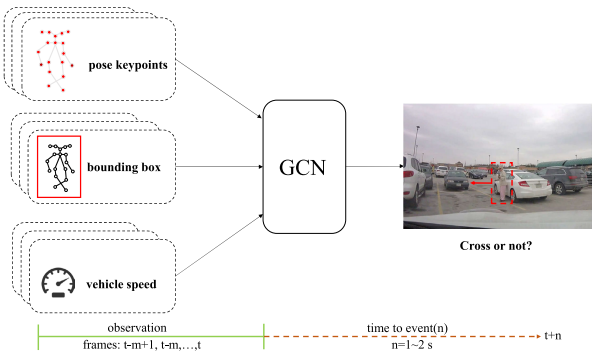


Fig. 1. Illustration of the PedAST-GCN model for pedestrian crossing intention prediction. The variable 'm' represents the observation length; for instance, when 'm' is equal to 32, it indicates an observation length of 32 frames. It uses the lightweight GCN model as the backbone network with simple but robust input representations (pose skeleton, bounding box, and vehicle speed).

for pedestrian crossing intention prediction. It employs an ST-GCN as the main branch to extract information from the skeleton data. Other data, including cropped images, segmentation maps, and ego-vehicle velocity data, are compressed into a weight vector using Global Average Pooling (GAP) and a sigmoid function and then added to the main branch by multiplication channel weighting. However, the generation of segmentation maps from the original images requires high computing resources, hindering its execution speed in practice.

To address these issues, we propose a GCN-based model with spatial-temporal attention for pedestrian crossing intention prediction (PedAST-GCN). As is shown in Fig. 1, we employ human keypoints to extract pedestrian attributes, including actions and eye contact, while utilizing ego-vehicle velocity to derive vehicle characteristics, such as speed and vehicle motion behavior. We also introduce bounding boxes to obtain information about pedestrian size and motion. These multimodal inputs also facilitate the capture of environment-related information by integrating various types of data (e.g., capturing the dynamic traffic situation around pedestrians by fusing their motion and action information over time). In contrast to methods directly employing segmentation maps for environmental information extraction, this approach indirectly captures environment-related information providing a comprehensive understanding of the context. The model inputs can be obtained from images with less preprocessing time, leading to improved execution speed for crossing intention prediction in practical applications.

For the model structure, We introduce an attention module (temporal attention) to capture the long dependencies in the sequence, and an attention module (modality attention) to fuse multiple channels of information. Different from closely relevant studies [13], [16], we propose a bounding box graph to extract pedestrian motion and size features from bounding box sequences which captures more information by preserving the spatial structure. Generally, the GCN model is used for the human poses keypoints data (coordinate data), but few directly for the ego-vehicle data (discrete sequence data). We design an ego-vehicle graph to represent the vehicle data instead of embedding the ego-vehicle velocity as a branch of the GCN model. The main contributions of the paper are:

1) Propose a GCN-based model (with attention modules for feature extraction and feature fusion) that is comparable or outperforms other models with a lower inference time.
2) Propose a graph representation for the bounding boxes and ego-vehicle speed that increases the model's accuracy.
3) Validate the model performance on two large-scale public data sets (JAAD and PIE) for pedestrian crossing intention prediction by comparing with state-of-art models and conducting ablation studies.

The remaining paper is structured as follows. Section II reviews the related studies. Section III formulates the problem and proposes the methodology. Section IV validates the model performance and conducts ablation studies. The final Section v summarizes the main conclusions and future studies.

II. Related Work

The review focuses on CNN, RNN, and GCN models for predicting pedestrian crossing intentions, as well as the spatial-temporal GCN models for sequential learning tasks.

A. Pedestrian Crossing Intention Prediction

The early CNN-based models used the 2D ConvNets and the last frame in the observation sequence to predict pedestrian crossing intention [5], [6]. Their performance is limited due to a lack of temporal information. To address this gap, Saleh et al. [7] developed the SORT [23] with the unscented Kalman filter (UKF) [24] to track pedestrians and proposed the Spatial-Temporal DenseNet(ST-DenseNet) to predict pedestrian crossing from image sequences. Singh and Suddamalla [9] used the Convolutional 3D (C3D) [21] to extract features from the skeleton, local context, and global context image sequences, and then concatenated features extracted from the last convolutional layer of Resnet 3D with the bounding box coordinates to predict pedestrian crossing intention.

Compared with prediction using a single image, the sequence of images captures more temporal features and thus improves prediction accuracy. However, the CNN models are not good at processing discrete sparse data, such as speed sequences and bounding box sequences. Rasouli et al. [5] developed a stacked RNN architecture to extract different information from various data types and they are used as inputs of pedestrian crops, surrounding context, poses, bounding box, and speed. They used the C3D to extract the information from the pedestrian crops and surrounding context and used the RNN model to process the poses, bounding box, and speed data. Different from the CNN-based model, the C3D was used as a module for image sequence processing in this proposed method. Kotseruba et al. [16] developed a pedestrian crossing intention prediction with an attention model (PCPA). They used the RNN model to extract features from the bounding box sequence, pose sequence, and vehicle speed sequence and used the C3D to extract the information from pedestrian crops. The attention mechanism was also used to capture long spatial dependencies and fuse different data types [25].

Besides, Kotseruba et al. [16] developed the publicly available benchmark for pedestrian crossing intention prediction, which provided standard public models for future work. Yang et al. [13] developed a hierarchical RNN-based model and incorporated the global context (semantic map) in the PCPA model to capture the scene information, and compared different combinations of data type streams. Zhou et al. [26] introduced a transformer-based model for pedestrian crossing intention prediction, which incorporates a temporal fusion block and a self-attention mechanism to capture richer information. Although these models are reported to achieve a high prediction accuracy, they use complex models for preparing different types of model input data which tends to be computationally intensive and limited for real-time applications.

The GCN model is widely used for the skeleton data for its prominent performance on non-euclidean data. For example, Cadena et al. [17] proposed a two-layer GCN model and used 14 keypoints to recognize pedestrian crossing intention. Zhang et al. [18] introduced the spatial-temporal graph convolution networks [27] for pedestrian crossing intention prediction, which learns high-level features of spatial and temporal information. These models have a high inference speed given the graph representation inputs, however, they are limited to using a single data type. To address that, Cadena et al. [19] proposed a Pedestrian Graph + model fusing pedestrian pose keypoints, image crop, segmentation maps, and ego-vehicle velocity to predict pedestrian crossing intention. They input the skeleton information into the main GCN backbone network and design a branch to embed the speed, image crop, and segmentation maps into the backbone. To improve accuracy, the Pedestrian Graph + model also uses a human pose forecasting model [28] to predict the following 30 coordinates (one second). However, the segmentation maps and human pose forecasting take significant time for computing which may limit their real-time applications in practice. Also, the human pose forecasting may increase the model's instability by adding uncertainty.

B. Spatial-Temporal GCN for Sequential Learning Tasks

The spatial-temporal GCN models are widely used for sequence skeleton data based action recognition tasks [27], [29], [30], [31], [32], [33], [34], [35], [36]. Yan et al. [27] firstly proposed the spatial-temporal GCN model (ST-GCN) for the action recognition. The following works focus on developing models to capture more abundant spatial information and longer time-dependent information. For example, Liu et al. [37] proposed the disentangling and unifying graph convolution to capture the long-range skeleton joints dependencies and complex spatial-temporal dependencies for action recognition. Shi et al. [29] proposed a directed acyclic graph to represent the skeleton data, which effectively incorporates the skeleton joint and bone data.

The ST-GCN model is also used for other recognition tasks. For example, Liu et al. [33] developed a novel Symmetry-Driven Hyper Feature Graph Convolutional Network (SDHF-GCN) for Gait recognition. Compared to previous models, the SDHF-GCN model automatically learns multiple dynamic patterns and hierarchical semantic features. It contains natural connection, temporal correlation, and symmetric interaction, which highly enriches the description of dynamic patterns by exploiting symmetry perceptual principles. Zhang et al. [35] developed a two-stream Graph Convolutional Network with spatial-temporal attention(STA-GCN) for hand gesture recognition. They proposed a data-driven updated skeleton graph for spatial information aggregation and fused the pose and motion streams to improve the recognition accuracy.

III. METHODOLOGY

A. Problem Definition

We defined the pedestrian crossing intention prediction as a binary classification problem [16]. Mathematically, the problem predicts the crossing intention $I \in \{0, 1\}$ of a pedestrian i in future time $t + n, n \in \{30, 60\}$(about 1 to 2 seconds), using the pose keypoints sequence $P_i^t = \left\{p_i^{t-m+1}, p_i^{t-m+2}, \ldots, p_i^t\right\}$, bounding box sequence $B_i^t = \left\{b_i^{t-m+1}, b_i^{t-m+2}, \ldots, b_i^t\right\}$, and vehicle speed sequence $V_i^t = \left\{v_i^{t-m+1}, v_i^{t-m+2}, \ldots, v_i^t\right\}$ in m consecutive frames. For example, we predict whether a pedestrian will cross the street or not in the next 1 to 2 seconds by observing 32 frames of video images.

The pedestrian crossing intention prediction is challenging in the road environment. The model structure and data representation have important influences on accuracy and prediction speed in practice. The paper proposes the spatial-temporal attention GCN model to predict pedestrian crossing intention that has a low computational cost but high accuracy and robustness. Fig. 2 shows the overall architecture of PedAST-GCN, which consists of a Modality layer, a Backbone layer, a Fusion layer, and a Prediction layer.

- The Modality layer has three different types of modality data, including: 1) pose keypoints, obtained from pedestrian crop images via the pose detector [38]; 2) bounding box, obtained from the image through object detection [39]; and 3) vehicle speed, obtained through the On-Board Diagnostics system. The modality feature data are complementary in capturing the dynamic information importance in predicting the pedestrian crossing intention.
- The Backbone layer comprises three distinct streams to extract hidden features of the pose keypoints, bounding box, and vehicle velocity, correspondingly. It includes three STA-GCN units, with each unit consisting of the GCN layer, attention layer, and TCN layer (Fig. 4).
- The Fusion layer includes several crucial components. The GAP operation calculates the average of hidden features across both temporal and spatial dimensions. The Temporal Average Pooling (TAP) operation calculates the average of hidden features across the temporal dimension. The concatenate operation combines the hidden features from different streams. The modality attention mechanism fuses the modality information by selectively weighting the importance of different modality information.

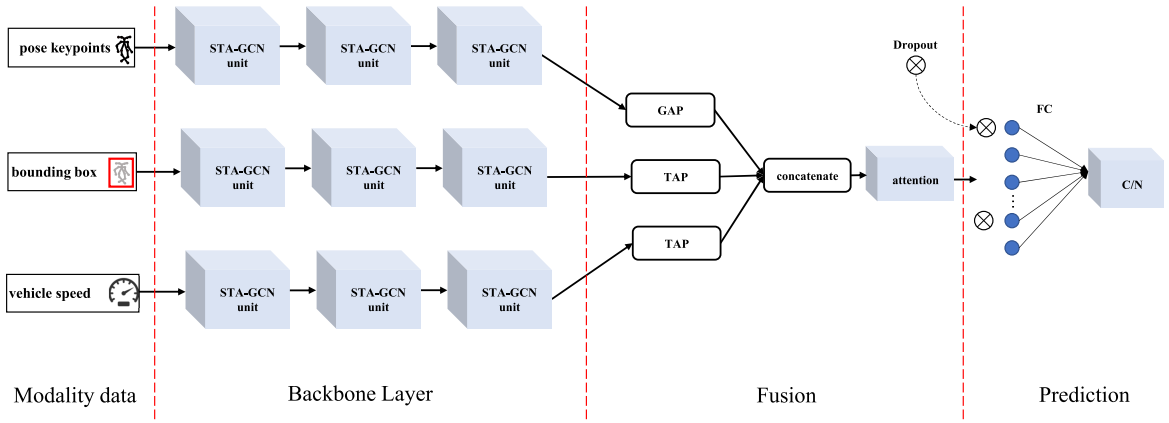
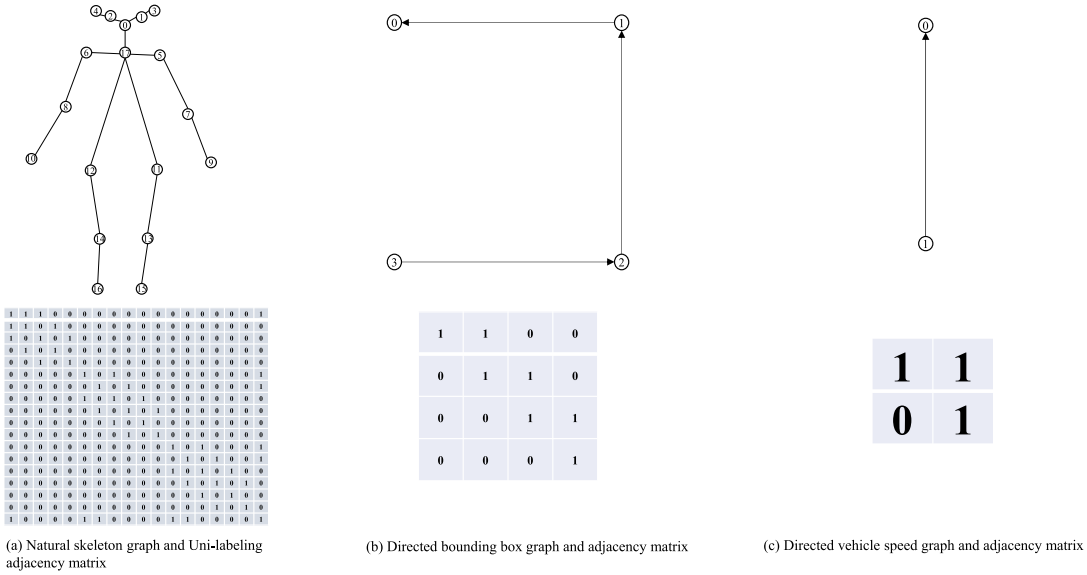


Fig. 2. The PedAST-GCN model for pedestrian crossing intention prediction. The input data types are pose skeleton, bounding box, and vehicle speed. Three ST-GCN modules are used to extract features from corresponding input data. The average pooling is used to compress the extracted features. The GAP is the global average pooling which performs the averaging operation on both the spatial and temporal dimensions. The TAP is the temporal average pooling which performs the averaging operation on the temporal dimension. The Fusion attention layer hybrids the features from different information streams. The fully connected (FC) layer outputs the prediction of pedestrian crossing intentions.



(a) Natural skeleton graph and Uni-labeling adjacency matrix

(b) Directed bounding box graph and adjacency matrix

(c) Directed vehicle speed graph and adjacency matrix

Fig. 3. The graph representations of pose keypoints, bounding box, and vehicle speed. The first row shows (a) the natural skeleton connection graph (pose keypoints), (b) the directed bounding connection graph (bounding box), and (c) the directed speed graph (vehicle speed). The bottom row shows the corresponding uni-labeling adjacency matrices.

- The prediction layer comprises dropout and full connection layers. It outputs the predicted probability of the pedestrian crossing intentions.

To summarize, the PedAST-GCN model takes inputs of heterogeneous modality information (pedestrian pose keypoints, bounding box, and vehicle speed) and outputs the hidden representation in the Backbone layer. Then, the Fusion layer compresses the hidden representation of modality information and extracts the final fusion representation. Finally, The Prediction layer predicts the pedestrian crossing probability. The core modules of the proposed model include modality graph design, spatial-temporal attention GCN unit, and modality attention.

B. Modality Graph Design

Fig. 3 shows the graph representations of pose keypoints, bounding box, and vehicle speed. Fig. 3 (a) is the natural skeleton connection graph and the corresponding uni-labeling adjacency matrix for the pose keypoints (The efficacy of various partition strategies is detailed in Appendix A). It is generated by 18 natural physical connections of the human keypoints. Fig. 3 (b) shows the directed bounding connection graph (only the features of the 0-th node will be utilized in the subsequent step) and its adjacency matrix for the bounding box. It has four vertexes and the shape of the bounding box contains the size and location information of the pedestrian. The natural connection between vertexes is used to construct the bounding box graph in order to preserve its spatial characteristics. Besides, we add the direction in the bounding box graph to speed up the information aggregation in the training process.

Fig. 3 (c) shows the directed vehicle speed graph (only the features of the 1-th node will be utilized in the subsequent step) and its adjacency matrix for the vehicle speed. Different from the skeleton and bounding box that contain the coordinate

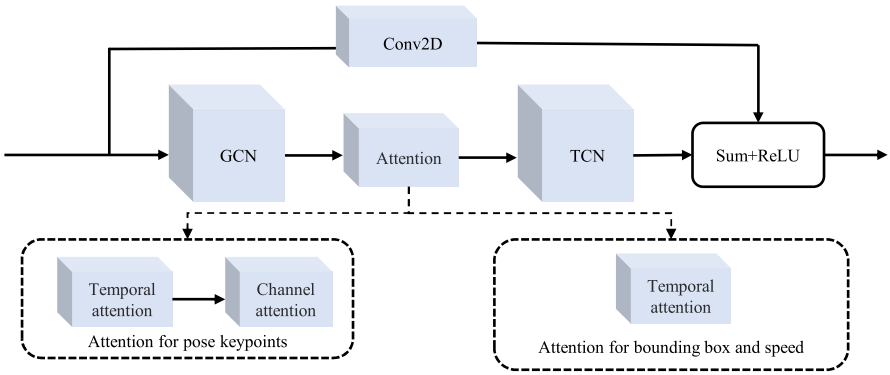


Fig. 4. The STA-GCN unit consists of the GCN layer, Attention layer, and TCN layer. The Attention layer contains temporal attention and channel attention for pose skeleton data while containing only temporal attention for bounding box and vehicle speed data.

data and have a visual physical structure, the vehicle speed is discrete sparse data and does not have coordinates. In the real world, the vehicle speed belongs to the feature of the vehicle. We can treat the vehicle as a particle (joints-1 in Fig. 3(c)) in the image and the vehicle speed is the feature of the particle. We further assume a virtual point (joints-0 in Fig. 3(c)) also with a speed feature in the image. Practically, we use the pedestrian bounding box center as the virtual point (joints-0 in Fig. 3(c)). The virtual point speed is calculated as a ratio of the center coordinate difference to the time between frames. Basically, the virtual point can represent the speed of the pedestrian.

The key advantages of the modality graph designs are: (1) We use the uni-labeling adjacency matrix as the adjacency matrix instead of the spatial configuration partitioning adjacency matrix (verified for action recognition). The action of crossing the street is a binary classification problem and the pattern is more regular than the action recognition. The uni-labeling adjacency matrix is relatively simple and fit for the problem which eventually facilitates a fast learning of patterns. (2) We design a directed bounding box graph based on the natural connection between vertexes and generate the bounding box adjacency matrix. Compared to existing methods, the bounding box graph can preserve the spatial information as well as learn more implicit spatial information (e.g., the size of a pedestrian and the relative distance between the vehicle and pedestrian) and the movement information between frames. In addition, the directed graph can help the model effectively aggregate and update the joints features. (3) We treat the vehicle as a particle and the vehicle speed as the feature of the vehicle. We also add a virtual point with a speed considering human attributes. We can generate a directed vehicle speed graph and the corroding adjacency matrix based on the two points. The vehicle speed graph can make speed data adapt to the GCN model and the directed graph can prevent the feature of the vehicle speed aggregation and update from the virtual point.

C. Spatial-Temporal Attention GCN Unit

The STA-GCN unit contains the GCN layer, Attention layer, and TCN layer as shown in Fig. 4. It takes inputs of modality data sequences and outputs their hidden representations. We add the Conv2D layer as the ResNet [40] connection to stabilize the training and use the Sum and ReLu operations to obtain hidden feature representations.

1) The GCN Layer: The GCN layer is used to aggregate and update the features of joints, which captures the spatial graph information. Given the original vertex state matrix (skeleton, box, and speed) $\mathbf{X} \rightarrow \mathbb{R}^{C\times T\times N}$, where the C denotes the number of channels, T denotes the temporal length, and the N denotes the number of vertexes. The core layer to the updated vertexes hidden states matrix $\mathbf{X^{gcn}} \rightarrow \mathbb{R}^{C^{gcn}\times T\times N}$ is calculated as:

$$\mathbf{X^{gcn}} = \mathbf{W_g X W_e} \odot \mathbf{A} \tag{1}$$

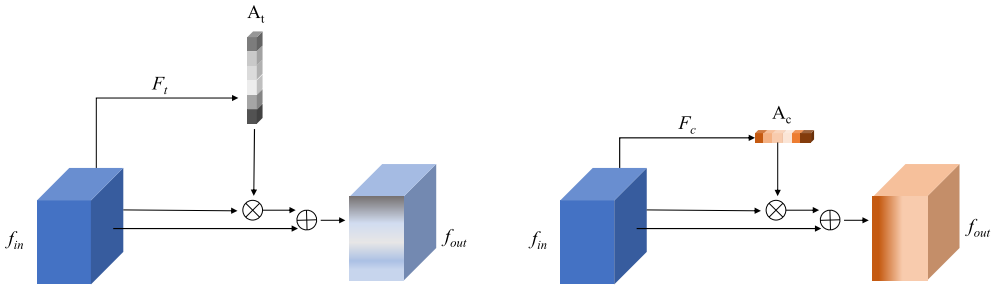
where $\mathbf{W_g} \rightarrow \mathbb{R}^{C^{gcn}\times C\times 1\ \times\ 1}$ is the weight vector of the $1\ \times\ 1$ convolution operation, $\mathbf{W_e} \rightarrow \mathbb{R}^{N\times N}$ is the learnable edge weights, $\odot$ is the element-wise product, and $\mathbf{A} \rightarrow \mathbb{R}^{N\times N}$ is the adjacency matrix.

2) The Attention Layer: The attention mechanism [25], [41], [42] is widely used for spatial-temporal information learning tasks and varies in formulations [31]. As is shown in Fig. 4, we propose two different attention layers for different data types. The attention model for pose keypoints contains two sub-modules: the temporal attention module(as shown in Fig. 5(a)) and the channel attention module(as shown in Fig. 5(b)), while the attention model for bounding box and vehicle speed only contains the temporal attention module.

The temporal attention module assigns different attention to frames and lends the model ability to capture longer temporal dependencies [31]. To assist the model in capturing information from sequence data more effectively, we introduce this module to allocate varying levels of attention to each frame and fuse features in a weighted manner. Given the output vertex hidden states matrix of the GCN layer $\mathbf{X^{gcn}} \rightarrow \mathbb{R}^{C^{gcn}\times T\times N}$. The core attention score $\mathbf{A_t} \rightarrow \mathbb{R}^{1\times T\times 1}$ is calculated as:

$$\mathbf{A}_t = \sigma(F_t(AvgPool(\mathbf{X^{gcn}}))) \tag{2}$$

where $Avgpool$ is the operation to average the features of all joints. F_t is a 1-D convolutional operation, $\mathbf{W_{F_t}} \rightarrow \mathbb{R}^{1\times C^{gcn}\times K_s}$ where K_s is the kernel size, and σ is the Sigmoid activation function.



(a) The temporal attention module (b) The channel attention module

Fig. 5. The temporal attention and the channel attention. $\otimes$ denotes the element-wise multiplication. $\oplus$ denotes the element-wise addition.

Then, the output vertex hidden states matrix of the temporal attention module $\mathbf{X^{ta}} \rightarrow \mathbb{R}^{C^{ta} \times T \times N}$ is calculated as:

$$\mathbf{X^{ta}} = \mathbf{X^{gcn}}\mathbf{A}_t + \mathbf{X^{gcn}} \tag{3}$$

The channel attention module is used to strengthen the channel features [31]. We introduce it to reinforce the discriminative features of different input samples. Given the output vertex hidden states matrix of the temporal attention module $\mathbf{X^{ta}} \rightarrow \mathbb{R}^{C^{ta} \times T \times N}$. The core attention score $\mathbf{A_c} \rightarrow \mathbb{R}^{C^{ta} \times 1 \times 1}$ is calculated as:

$$\mathbf{A}_c = \sigma(\mathbf{W_{c2}}(\delta(\mathbf{W_{c1}}(AvgPool(\mathbf{X^{ta}})))) \tag{4}$$

where $Avgpool$ is the operation to average the features of all joints in all frames. $\mathbf{W_{c1}} \rightarrow \mathbb{R}^{C^{ta} \times \frac{C^{ta}}{r}}$ and $\mathbf{W_{c2}} \rightarrow \mathbb{R}^{\frac{C^{ta}}{r} \times C^{ta}}$ are the learnable weights, $\frac{C^{ta}}{r}$ is the number of channels after scaling, r is the scale factor. δ is the ReLu activation function and σ is the Sigmoid activation function.

Then, the output vertex hidden states matrix of the channel attention module $\mathbf{X^{ca}} \rightarrow \mathbb{R}^{C^{ca} \times T \times N}$ is calculated as:

$$\mathbf{X^{ca}} = \mathbf{X^{ta}}\mathbf{A}_c + \mathbf{X^{ta}} \tag{5}$$

3) The TCN Layer: The TCN layer is used to extract and compress the temporal features between the frame sequences. The TCN layer contains a BatchNorm2d layer, a ReLu activation function, a Conv2d layer, and a BatchNorm2d layer in sequence. Given the output vertex hidden states matrix of the Attention layer (temporal attention or channel attention) $\mathbf{X^{a}} \rightarrow \mathbb{R}^{C^{a} \times T \times N}$. The output vertex hidden states matrix of the TCN layer $\mathbf{X^{tcn}} \rightarrow \mathbb{R}^{C^{tcn} \times T \times N}$ is calculated as:

$$\mathbf{X^{tcn}} = \mathrm{BN}(\mathbf{W_t}(\delta(\mathrm{BN}(\mathbf{X^a})))) \tag{6}$$

where BN represents batch normalization operation, $\mathbf{W_t} \rightarrow \mathbb{R}^{C^{tcn} \times C^{a} \times K_t \times 1}$ is the weight vector of the $K_t \times 1$ convolution operation, and δ is the ReLu activation function.

D. Modality Attention

The attention mechanism selectively focuses on significant features, aiding the model in capturing crucial information by combining different branches of data in a weighted manner [13], [16]. To improve the model's robustness, we further introduce the modality attention model to fuse the hidden representation from different modality data (pose keypoints, bounding box, and speed). The core operation is attention weights computing. Given the sequence of the modality information features $S = \{\mathbf{s_0}, \mathbf{s_1}, \ldots, \mathbf{s_E}\}$, the attention weight α_{fe} between the hidden representations $\mathbf{s_f}$ and $\mathbf{s_e}$ is computed as:

$$\alpha_{fe} = \frac{\exp(score(\mathbf{s_f}, \mathbf{s_e}))}{\sum_{e=1}^{E} \exp(score(\mathbf{s_f}, \mathbf{s_e}))} \tag{7}$$

where the $score(\mathbf{s_f}, \mathbf{s_e}) = \mathbf{s_f}(\mathbf{s_e}\mathbf{W_a})^T$. $\mathbf{s_f}$ and $\mathbf{s_e}$ are hidden features from different data types, respectively. $\mathbf{W_a}$ is a learned weight matrix and E is the number of data types.

The attention module is used to better memorize sequential information by selectively focusing on parts of features relevant to the task. We calculated the attention weight sequence between the bound box and modality data $\alpha_0 = \{\alpha_{00}, \alpha_{01}, \ldots, \alpha_{0E}\}$. The attention weights trade off the $\mathbf{s_0}$ with other modality data features $\mathbf{s_e}$, which can gather useful information to improve the prediction model robustness.

E. Data Processing and Preparation

We utilize bounding box sequence $B = \{b^0, b^1, \ldots, b^{m-1}\}$, vehicle speed sequence $V = \{v^0, v^1, \ldots, v^{m-1}\}$, and skeleton sequence $P = \{p^0, p^1, \ldots, p^{m-1}\}$ in m consecutive frames as our input(e.g., m = 32). For the b^i in the i-th frame, it consists of 4 bounding box vertex pixel coordinates, which are sourced directly from the dataset. Each box vertex coordinate is denoted as (x_{box}, y_{box}). To enhance the model's generalization capabilities, we normalize each box vertex coordinate using the following procedure:

$$\begin{cases} x'_{box} = \dfrac{x_{box}}{w_{image}}, \\ y'_{box} = \dfrac{y_{box}}{h_{image}}, \end{cases} \tag{8}$$

where (x'_{box}, y'_{box}) the normalized coordinates. w_{image} and h_{image} are the width and height of the image, respectively.

In the i-th frame, v^i comprises both the pedestrian movement speed $v_{pedestrian}$, and the vehicle speed $v_{vehicle}$. The vehicle speed $v_{vehicle}$ is directly obtained from the dataset, and we apply the following normalization procedure to each $v_{vehicle}$:

$$v'_{vehicle} = \frac{v_{vehicle}}{v^{max}_{vehicle}}, \tag{9}$$

where $v'_{vehicle}$ the normalized vehicle speed. $v^{max}_{vehicle}$ is the maximum vehicle speed in the dataset.

We can obtain the bounding box centre coordinate sequence $B_{centre} = \left\{b^0_{centre}, b^1_{centre}, \ldots, b^{m-1}_{centre}\right\}$, from the bounding box data. Each centre coordinate is represented as (x_{centre}, y_{centre}). The $v_{pedestrian}$ in the i-th frame(i < m-1) is calculate as:

$$v_{pedestrain} = fps * \sqrt{(x^{i+1}_{centre} - x^{i}_{centre})^2 + (y^{i+1}_{centre} - y^{i}_{centre})^2} \quad (10)$$

where fps is the frame rate of the camera. When i equals $m-1$ (the last frame), $v_{pedestrian}$ is equal to the speed of the previous frame.

For processing the skeleton data, we employ the HRNet [38] to extract 18 keypoints from the pedestrian image. HRNet provides keypoint pixel coordinates $(x_{skeleton}, y_{skeleton})$ along with associated confidence scores, denoted as $s_{skeleton}$. To increase the model generalization, we normalize each keypoint as follows:

$$\begin{cases} x'_{skeleton} = \dfrac{x_{skeleton} - x_{left}}{w_{box}}, \\ y'_{skeleton} = \dfrac{y_{skeleton} - y_{left}}{h_{box}}, \\ s'_{skeleton} = s_{skeleton}, \end{cases} \quad (11)$$

where $(x'_{skeleton}, y'_{skeleton})$ is the normalized coordinates. x_{left} and y_{left} are the coordinates of the top left corner pedestrian box. w_{box} and h_{box} are the width and height of the pedestrian box, respectively. $s'_{skeleton}$ is the transformed confidence score.

IV. EXPERIMENTS AND EVALUATIONS

A. Data Set

1) JAAD: The JAAD [5] is a specialized autonomous driving data set and contains 346 video clips, and each clip lasts 5-10 seconds. The JAAD behavioral data ($JAAD_{beh}$) contains 495 crossing pedestrians and 191 pedestrians intending to cross. The complete JAAD dataset ($JAAD_{all}$) adds 2100 other visible pedestrians who are far away from the road and do not intend to cross. To have a fair comparison, We use the same split as in [16] for the training and testing, which contains 177 videos for training, 29 videos for validation, and 117 for testing.

2) PIE: The PIE [43] is also a real data set for pedestrian crossing intention prediction. The data set contains 1322 non-crossing and 512 crossings. The PIE dataset contains all pedestrians close to the road who are hesitant to cross or not. For the training and testing split, as suggested in [16], we use set01, set02, and set04 for training, set05, set06 for validation, and set03 for the testing set.

B. Compared Models and Evaluation Metrics

We compared our method with state-of-the-art methods on two standard benchmark datasets: JAAD [5], and PIE [43]. The compared approaches include:

- **ATGC** [5]. It only uses the last video frame for pedestrian crossing intention prediction based on a fully connected layer. The backbone networks for feature extraction are VGG16 [44] or ResNet50 [40].
- **ConvLSTM** [12]. It uses a pre-trained CNN model to extract features from the image sequences. Then, it uses the LSTM model to extract the hidden representation from the feature sequences. Finally, it uses a fully connected layer for pedestrian crossing intention prediction based on the extracted hidden representation.
- **SingleRNN** [45]. It uses the 2D bounding box, pedestrians and their surrounding images, vehicle speed, and intention information as inputs. It uses the LSTM or GRU as the backbone network and uses a fully connected layer based on the last hidden state for pedestrian crossing intention prediction.
- **Stacked RNN** [46]. It uses the structure of a stack of RNN layers and each RNN layer takes as inputs the hidden state of the RNN layer below.
- **MultiRNN** [47]. It uses different RNN streams to extract features from different types of data and feds the final hidden states into a fully connected layer for pedestrian crossing intention prediction.
- **HierarchicalRNN** [15]. It uses different RNN branches to extract features from different types of data and the concatenated hidden states are fed to the other RNN model and a fully connected layer is finally used for pedestrian crossing intention prediction.
- **SFRNN** [48]. It uses pedestrian crops, surrounding context, poses, bounding box, and ego-vehicle speed as inputs. It uses the structure of a stack of RNN layers as the backbone network in which the complex features are fed at the bottom layers and simpler features at the top.
- **C3D** [21]. It uses pedestrian crop sequences as inputs and the 3D convolutional networks as the backbone network. The fully connected layer is used for pedestrian crossing intention prediction.
- **I3D** [20]. It uses pedestrian crop sequences as inputs and the two-streams Inflated 3D convolutional networks as the backbone network. The fully connected layer is used for pedestrian crossing intention prediction.
- **TwoStream** [49]. It uses pedestrian crop sequences and the optical flow as inputs and the two CNN branches as the backbone network. The average of the predictions of the two branches is used for pedestrian crossing intention prediction.
- **Fussi-Net** [50]. It uses the skeleton and bounding box as inputs and the DenseNet model as the backbone network.
- **PCPA** [16]. It uses the skeleton, box, speed, and local context as inputs. It uses three RNN models and a 3DConv model as four streams to extract features from different types of data and the attention model for information fusion. The fully connected layer is used for pedestrian crossing intention prediction.
- **Global PCPA** [13]. It uses the skeleton, box, speed, local context, and global context as inputs. It uses five GRU models as five streams to extract features from different types of data and the attention model for information fusion. The fully connected layer is used for pedestrian crossing intention prediction.

- **TrouSPI-Net** [51]. It uses the skeleton data, bounding box data, speed information, and the relative pairwise distances of skeletal joints as inputs. It utilizes the GRU as the backbone for pedestrian crossing intention prediction.
- **Pedestrian Graph** [17]. It uses the pose keypoints as the input and a 2 layers GCN model for pedestrian crossing intention prediction.
- **Pedestrian Graph +** [19]. It uses the pose keypoints, image crop, vehicle speed, and segmentation maps as inputs. It uses a 2 layers GCN model as the backbone network and the Conv branches to embed the speed information. The fully connected layer is used for pedestrian crossing intention prediction.
- **ST CrossingPose** [18]. It uses the pose keypoints as the input and the spatial-temporal GCN model for pedestrian crossing intention prediction.
- **PIT** [26]. It uses the pose keypoints, image crop, vehicle speed, bounding box, and global image as the input and the transformer-based model for pedestrian crossing intention prediction.

To make a fair and comprehensive model comparison, we use five evaluation metrics (Accuracy, AUC, F1 score, Precision, Recall) to evaluate the performance of models, as proposed in [16]. The definitions of these evaluation metrics are as follows:

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP} \tag{12}$$

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

$$F1\,score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{15}$$

where TP represents the true positive, TN the true negative, FP the false positive, and FN the false negative. AUC is the area under the ROC curve.

C. Training and Model Settings

We used three labels ('Irrelevant', 'Cross', and 'No cross') to train our model on the JAAD_{all} dataset to obtain a better performance as suggested in [19]. In the test phase, we used the Class Mapping [19] to map the 'Irrelevant' into 'No cross'. As for the JAAD_{beh} and PIE dataset, we used two labels ('Cross', and 'No cross') to train and test. For the input of the model, both JAAD and PIE data provide the bounding box and vehicle speed for each sample and we used the HRNet [38] to obtain the pose keypoints.

We implemented the proposed method using PyTorch. During training, we used the Adam optimizer with weight-decay 5e-4. We used the CrossEntropyLoss function and the learning rate of 0.01 with 300 epochs to train the PedAST-GCN model on JAAD_{all} dataset and used the Binary Cross Entropy (BCE) Loss function and the learning rate of 0.001 with 300 epochs to train the PedAST-GCN model and PIE dataset. As for JAAD_{beh}, we used the BCE Loss function and the learning rate of 0.01 with 300 epochs to train. The model settings are provided in Table I.

TABLE I
THE DETAILS OF MODE SETTING

Data	Data split type	default
	Data set	JAAD_{all}, JAAD_{beh}, PIE
	$v_{vehicle}^{max}$	5, 5, 56
	fps	30
Model	Input	keypoints, bounding box, speed
	Model	PedAST-GCN
Net	Backbone	STA-GCN unit
	Layers of backbone	3
	Fusion layer	modality attention
Implementation	$C^{gcn} \vee C^{ta} \vee C^{ca} \vee C^{tcn}$	64, 128, 256
	stride(W_g), padding(W_g)	(1,1), (0,0)
	K_s, stride(W_{F_t}), padding(W_{F_t})	9, 1, 4
	r	2
	K_t, stride(W_t), padding(W_t)	9, {(1, 1), (2, 1), (2, 1)}, (4,0)
Train	Batch size	32, 64, 256
	Epochs	300, 300, 300
	Learning rate	0.01, 0.01, 0.001
	Optimizer	Adam
	Loss function	CrossEntropyLoss, BCE Loss, BCE Loss
Test	Batch size	32, 64, 256
	Test label transition	Class Mapping [19], No, No

D. Results

Table II shows the model comparison results for the PIE and JAAD dataset. The PedAST-GCN model achieves comparable or better results compared to the state-of-the-art Pedestrian Graph + model [19] and PIT [26]. The improvement is attributed to the use of multi-modality data, such as pose keypoints data to capture pedestrian pose and action information, bounding box data to capture pedestrian motion, distance, size information, and vehicle speed data to capture vehicle movement information. The fusion of these sources enables higher-level reasoning, contributing to the maintenance of prediction accuracy. Furthermore, our model maintains accuracy across various observation lengths due to its robust temporal feature extraction capabilities. The early research based on a 2D convolutional model used the last pedestrian frame image which lacks capturing time dependencies in predicting crossing intentions [5]. Compared with the 2D convolutional models, the RNN-based models capture temporal dependence information and allow taking heterogeneous types of data as inputs [13], [16]. The 3D convolutional models also perform better than the 2D ones since they capture the time dependencies (e.g. the C3D [21] and I3D [20]). The GCN-based models reported in [17] and [18] have a faster inference speed but use limited modality data (i.e., only pose keypoints). The state-of-the-art Pedestrian Graph + model [19] performs well in the benchmark models which take inputs of pose keypoints, image crop, vehicle speed, and segmentation map (built environment information). However, the preprocessing time for the segmentation map is significantly high, thus limiting its real-time use in practice.

E. Ablation Study

The model accuracy, execution speed, and robustness are important requirements for pedestrian crossing intention prediction. This section includes an ablation study on observation length and various ablation studies using 32 frames as observation length to examine the impact of variations in backbone

TABLE II

MODEL PERFORMANCE COMPARISON RESULTS FOR PIE AND JAAD DATASET. $JAAD_{beh}$ IS A SUBSET OF THE JAAD DATASET WITH BEHAVIORAL LABELS (ONLY PEDESTRIANS THAT HAVE INTERACTION WITH THE EGO-VEHICLE), AND $JAAD_{all}$ INCLUDES ALL DETECTED PEDESTRIANS. ACC MEANS ACCURACY, AUC AREA UNDER THE CURVE, F1 IS F1 SCORE, P PRECISION AND R RECALL

Model name	Year	Model Variant	Use frames	Input data	PIE					$JAAD_{beh}$					$JAAD_{all}$				
					Acc	AUC	F1	P	R	Acc	AUC	F1	P	R	Acc	AUC	F1	P	R
ATGC [5]	2017	VGG16	1	G,L,W	71	60	41	49	36	49	52	71	63	82	82	75	55	49	63
		ResNet50	1	G,L,W	70	59	38	47	32	46	45	54	58	51	81	72	52	47	56
ConvLSTM [12]	2015	VGG19+LSTM	16	I	58	55	39	32	49	53	49	64	64	64	63	57	32	24	48
		ResNet50+LSTM	16	I	54	46	26	23	29	59	55	69	68	70	63	58	33	25	49
TwoStream [49]	2014	VGG16	16	I,OF	64	54	32	33	31	56	52	66	66	66	60	69	43	29	83
SingleRNN [45]	2020	GRU	16	I,B,S,Int	83	77	67	70	64	58	54	67	67	68	65	59	34	26	49
		LSTM	16	I,B,S,Int	81	75	64	67	61	51	48	61	63	59	78	75	54	44	70
MultiRNN [47]	2018	GRU	16	-	83	80	71	69	73	61	50	74	64	86	79	79	58	45	79
StakedRNN [46]	2015	GRU	16	I,OF	82	78	67	67	68	60	60	66	73	61	79	79	58	46	79
HerarchicalRNN [15]	2015	GRU	16	K	82	77	67	68	66	53	50	63	64	61	80	79	59	47	79
SFRNN [48]	2020	GRU	16	I,G,K,B,S	82	79	69	67	70	51	45	63	61	64	84	84	65	54	84
C3D [21]	2015	3DConv	16	I	77	67	52	63	44	61	51	75	63	91	84	81	65	57	75
I3D [20]	2017	3DConv	16	I	80	73	62	67	58	62	56	73	68	79	81	74	63	66	61
		Opticflow+3DConv	16	I,OF	81	83	72	60	**90**	62	51	75	65	88	84	80	63	55	73
FUSSI-Net [50]	2020	DenseNet	16	B,K	-	-	-	-	-	59	58	69	66	73	60	72	40	27	73
PCPA [16]	2021	3DConv	16	I,B,K,S	86	91	78	69	89	50	47	59	61	58	70	85	51	36	**87**
Global PCPA [13]	2021	VGG+GRU	16	I,B,K,S,SGM	-	-	-	-	-	62	55	73	65	85	83	86	63	51	82
TrouSPI-Net [51]	2021	GRU	16	B,K,S,ED	88	88	80	73	89	64	56	76	66	91	85	73	56	57	55
Pedestrian Graph [17]	2019	GCN	16	K	76	69	48	62	39	62	69	70	71	68	80	84	55	46	68
Pedestrian Graph + [19]	2022	Conv+GCN	32	I,K,S,SGM	89	90	81	83	79	**70**	**70**	76	**77**	75	86	**88**	65	58	75
ST CrossingPose [18]	2022	ST-GCN	16	K	-	-	-	-	-	63	56	74	66	83	-	-	-	-	-
PIT [26]	2023	Transformer	16	I,K,S,B,G	**91**	90	82	85	79	**70**	65	**81**	71	93	87	87	66	54	85
PedAST-GCN	2023	STA-GCN	16	K,S,B	**91**	**94**	**83**	**88**	79	69	66	79	68	93	**89**	83	**68**	**67**	69
PedAST-GCN	2023	STA-GCN	32	K,S,B	90	86	81	83	79	69	59	80	68	**96**	88	81	**68**	66	70

Notes: G represents the global context, L represents looking or not, W represents walking or not, I represents the image crop, OF represents the optical flow, B represents the bounding box, S represents the speed, int represents the intention, K represents pose keypoints, SGM represents the segmentation maps, ED represents the relative pairwise distances of skeletal joints.

TABLE III

ABLATION STUDY OF GCN BACKBONE AND GRAPH DESIGN. $JAAD_{beh}$ IS A SUBSET OF THE JAAD DATASET WITH BEHAVIORAL LABELS (ONLY PEDESTRIANS THAT HAVE INTERACTION WITH THE EGO-VEHICLE), AND $JAAD_{all}$ INCLUDES ALL DETECTED PEDESTRIANS. ACC MEANS ACCURACY, AUC AREA UNDER THE CURVE, F1 IS F1 SCORE, P PRECISION AND R RECALL

	Backbone	PIE					$JAAD_{beh}$					$JAAD_{all}$				
		Acc	AUC	F1	P	R	Acc	AUC	F1	P	R	Acc	AUC	F1	P	R
B	GRU*	75.00	78.00	66.00	54.00	84.00	46.00	44.00	55.00	59.00	51.00	75.00	79.00	55.00	41.00	85.00
	LSTM*	82.00	82.00	72.00	64.00	83.00	49.00	48.00	56.00	62.00	50.00	79.00	81.00	59.00	45.00	84.00
	ST-GCN-UD	83.22	78.24	69.13	71.55	68.87	64.51	52.57	77.47	65.21	95.39	85.35	77.03	61.24	58.70	64.01
	ST-GCN-D	83.72	79.65	69.84	72.85	67.06	64.85	51.94	78.14	64.87	98.23	87.44	78.03	64.56	65.87	63.30
S	GRU*	84.00	85.00	75.00	66.00	88.00	41.00	52.00	23.00	72.00	14.00	50.00	58.00	33.00	22.00	70.00
	LSTM*	82.00	83.00	73.00	64.00	84.00	42.00	53.00	22.00	77.00	13.00	50.00	58.00	34.00	22.00	72.00
	ST-GCN-UD	85.95	86.61	77.89	69.81	88.10	63.27	50.09	77.22	63.99	97.34	80.00	62.92	39.54	43.59	36.17
	ST-GCN-D	86.85	83.12	76.11	77.69	74.60	63.95	50.00	78.01	63.95	99.98	81.96	50.09	0.35	99.98	0.18

1 "B" signifies the bounding box, while "S" denotes speed. "ST-GCN-UD" corresponds to the ST-GCN model utilizing an undirected graph, and "ST-GCN-D" pertains to the ST-GCN model employing a directed graph. None of these ST-GCN models incorporate temporal attention.
2 The '*' are the results tested by the authors using the open source code at https://github.com/OSU-Haolin/Pedestrian_Crossing_Intention_Prediction.

layer and graph design, modality data type, attention modules, and noisy data.

1) Impact of the Backbone Layer and Graph Design: To verify the performance of the backbone layer and graph design, we conduct a comparison between the proposed ST-GCN based backbone (without temporal attention) and the RNN-based counterpart (GRU, LSTM), while also evaluating different graph designs of ST-GCN (ST-GCN-UD corresponds to the ST-GCN model utilizing an undirected graph, and ST-GCN-D pertains to the ST-GCN model employing a directed graph). Table III shows the comparison results. Compared to the RNN-based backbone, the ST-GCN based model demonstrates a significant improvement across datasets in handling box and speed data. This enhancement is attributed to the fact that while the RNN-based model can capture temporal information from the sequential data, it treats the box data as the discrete information in each frame, thereby losing valuable spatial structure information. In contrast, the proposed ST-GCN based model excels at capturing both spatial and temporal information associated with the bounding boxes. Consequently, it effectively captures details regarding pedestrian movements (speed/direction) and sizes (distance) which are important features driving the pedestrian cross intention prediction. Regarding the speed data, we constructed a graph that incorporates both vehicle speed and pedestrian movement speed. This approach empowers the ST-GCN backbone to effectively process speed-related spatial and temporal information, allowing it to capture vital details regarding the relative

TABLE IV

ABLATION STUDY OF DIFFERENT TYPES OF MODALITY DATA. $JAAD_{beh}$ IS A SUBSET OF THE JAAD DATASET WITH BEHAVIORAL LABELS (ONLY PEDESTRIANS THAT HAVE INTERACTION WITH THE EGO-VEHICLE), AND $JAAD_{all}$ INCLUDES ALL DETECTED PEDESTRIANS. ACC MEANS ACCURACY, AUC AREA UNDER THE CURVE, F1 IS F1 SCORE, P PRECISION AND R RECALL

	PedAST-GCN	PIE					$JAAD_{beh}$					$JAAD_{all}$					Inference Time (ms)
		Acc	AUC	F1	P	R	Acc	AUC	F1	P	R	Acc	AUC	F1	P	R	
Single	K	72.63	55.76	26.17	54.04	17.26	67.35	58.69	77.85	68.75	89.72	86.76	69.05	53.02	73.97	41.31	5
	B	83.78	78.14	69.34	73.93	65.28	66.33	56.80	77.55	67.59	90.96	88.17	77.93	65.42	69.38	61.88	4
	S	89.19	87.40	81.24	79.25	83.33	65.08	53.49	77.68	65.69	95.04	81.96	50.16	0.71	66.67	0.36	4
Double	K+B	83.06	74.43	64.49	78.41	54.76	67.77	56.97	80.06	69.00	95.33	88.49	78.33	66.23	70.54	62.41	8
	K+S	89.58	87.86	81.90	79.96	83.93	66.78	56.95	78.02	67.62	92.20	86.35	70.80	55.16	67.88	46.45	8
	B+S	90.64	87.38	82.75	85.75	79.96	66.67	55.97	78.35	67.00	94.33	87.12	75.42	61.57	66.81	57.09	7
Triple	K+B+S	89.69	86.48	81.18	83.30	79.17	69.05	58.58	79.88	68.75	96.10	88.01	80.52	67.50	66.21	69.00	10

Notes: K represents the pose keypoints, B represents the bounding box, and S represents the speed. All the backbones incorporate the attention module for feature extraction and the modality attention module for fusing multiple data types. The modality attention module is not used for the single input data model.

movement dynamics between pedestrians and vehicles in the studied task.

Compared to the ST-GCN-UD model utilizing a graph without direction, the ST-GCN-D model employing a directed graph exhibits superior performance. The box graph differs from the skeleton graph used in action recognition tasks, as the latter employs a complex graph structure and strategy to capture a wide range of relationships between joints to capture more actions. We structured the graph for the bounding boxes to align with their inherent structure, aiming to capture pedestrian movement and size information effectively. In contrast to an undirected graph, the use of a directed graph addresses the over-smoothing problem through directed information transfer during model training. This design also expedites information aggregation to the terminal node (represented as the 0-th node in Fig. 3(b)). In the case of the vehicle speed graph, as depicted in Fig. 3(c), when comparing the graph without direction to the one with direction, it's noteworthy that there's no information transfer from the 0-th node to the 1-th node. Superior performance was observed when using only the 1-th node in the directed graph as the output hidden feature than using the average of the 0-th and 1-th nodes in the undirected graph. This can be attributed to the unique nature of the data sources. Specifically, pedestrian speed is estimated from pixel points, whereas vehicle speed is based on actual data. As a result, combining these two sources of information does not yield improved results.

2) Impact of the Modality Data Type: To verify the performance of data types, we use different modality features as inputs, including pose skeleton(K), bounding box(B), and vehicle speed(S). In all the sub-experiments, the backbones include the attention module for feature extraction and the modality attention module for fusing multiple data types (the modality attention module is not applied when dealing with single input data). Table IV shows the model prediction results with different combinations of modality data. Among the models utilizing different input data types, the model incorporating vehicle speed data exhibited the highest performance on the PIE dataset, surpassing those using pose keypoints or bounding box information. However, the model that utilizes only pure vehicle speed data resulted in poor performance for the JAAD dataset ($JAAD_{beh}$ and $JAAD_{all}$), due to the lack of precise raw velocity observations in the dataset [19].

The model utilizing pose keypoints data demonstrates strong performance in predicting pedestrian crossing intention in the JAAD dataset ($JAAD_{beh}$ and $JAAD_{all}$) but shows poor performance on the PIE dataset. The reason can be that the JAAD data($JAAD_{beh}$ and $JAAD_{all}$) contain many pedestrians directed to intend to cross or not, while the PIE data set contains many pedestrians close to the road who are irresolute to cross [16]. The irresolute pedestrian would add ambiguity for intention prediction based on the pose keypoints. The model with the bounding box information performs well for both PIE and JAAD data sets($JAAD_{beh}$ and $JAAD_{all}$) since the bounding box captures the motion and size information of pedestrians.

Compared with the model using a single modality feature, the combined features generally improve the model performance in almost all cases. For example, the performance of the model with both bounding box and vehicle speed features shows a significant improvement on the PIE dataset, compared to the model that utilizes only a single feature (either bounding box or vehicle speed). The model incorporating pose keypoints, bounding box, and vehicle speed achieves the best performance on all three datasets. This is because these modality features are complementary in capturing the dynamic importance of different information and are thus effective in predicting pedestrian crossing intention. For example, in situations where pedestrians are crossing the road, the pose keypoints and bounding box are more important in capturing the pedestrian's pose and motion information. However, when the pedestrian is hesitant to cross, the vehicle speed becomes more important in predicting their crossing intention.

3) Impact of the Attention Module: We also conducted an experiment to verify the effectiveness of the attention modules. Table V shows the model prediction results with different attention modules. The 'No-attention' model uses the 'sum' operation for the fusion with no temporal attention or modality attention in the PedAST-GCN model. The 'Temporal attention' only uses the temporal attention module in the PedAST-GCN model. The 'Modality attention' only uses the modality attention module in the PedAST-GCN model. The 'Temporal and Modality attention' uses both the temporal attention module and the modality attention module in the PedAST-GCN model.

TABLE V

ABLATION STUDY OF ATTENTION MODULES. $JAAD_{beh}$ IS A SUBSET OF THE JAAD DATASET WITH BEHAVIORAL LABELS (ONLY PEDESTRIANS THAT HAVE INTERACTION WITH THE EGO-VEHICLE), AND $JAAD_{all}$ INCLUDES ALL DETECTED PEDESTRIANS. ACC MEANS ACCURACY, AUC AREA UNDER THE CURVE, F1 IS F1 SCORE, P PRECISION AND R RECALL

PedAST-GCN	PIE					$JAAD_{beh}$					$JAAD_{all}$				
	Acc	AUC	F1	P	R	Acc	AUC	F1	P	R	Acc	AUC	F1	P	R
No-attention	87.51	84.00	77.37	78.81	75.99	65.42	56.02	76.84	67.20	89.72	87.21	78.03	64.28	64.92	63.65
Temporal attention	88.80	84.96	79.26	82.58	76.19	65.19	53.92	77.61	65.92	94.33	87.79	75.14	62.09	70.75	55.32
Modality attention	88.57	87.70	80.82	76.46	85.71	66.33	53.30	79.16	65.51	99.00	87.95	81.24	67.97	65.41	70.75
Temporal and Modality attention	89.69	86.48	81.18	83.30	79.17	69.05	58.58	79.88	68.75	96.10	88.01	80.52	67.50	66.21	69.00

TABLE VI

ABLATION STUDY OF OBSERVATION LENGTH. $JAAD_{beh}$ IS A SUBSET OF THE JAAD DATASET WITH BEHAVIORAL LABELS (ONLY PEDESTRIANS THAT HAVE INTERACTION WITH THE EGO-VEHICLE), AND $JAAD_{all}$ INCLUDES ALL DETECTED PEDESTRIANS. ACC MEANS ACCURACY, AUC AREA UNDER THE CURVE, F1 IS F1 SCORE, P PRECISION AND R RECALL

Observation frames	PIE					$JAAD_{beh}$					$JAAD_{all}$				
	Acc	AUC	F1	P	R	Acc	AUC	F1	P	R	Acc	AUC	F1	P	R
8	90.44	92.39	82.49	84.56	80.53	67.09	61.11	76.38	69.56	84.68	87.43	84.70	64.59	63.50	65.72
16	91.14	94.26	83.43	88.10	79.24	68.53	65.66	78.72	68.22	93.03	88.56	83.11	67.73	66.83	68.65
24	90.01	92.74	83.00	81.58	84.46	66.09	60.73	76.84	67.54	89.11	87.52	88.40	66.35	64.27	68.56
32	89.69	86.48	81.18	83.30	79.17	69.05	58.58	79.88	68.75	96.10	88.01	80.52	67.50	66.21	69.00
40	89.28	88.64	80.13	84.28	76.36	67.34	53.95	79.53	66.42	99.10	88.32	89.07	67.42	67.88	67.00

Generally, the model with temporal and modality attention significantly improves the prediction performance in most cases. This is because the temporal attention mechanism is able to selectively attend to different frames within a long sequence, allowing for the capture of more crucial information [31]. The Modality attention mechanism can selectively attend to different data types and fuse them in a weighted manner [13]. Compared with the 'No-attention', the model with 'Temporal attention and Modality attention' improves the model performance by 2.18%, 3.63%, and 0.8% on PIE, $JAAD_{beh}$, and $JAAD_{all}$ data set.

4) Impact of Observation Length: We employ diverse observation frames (8, 16, 24, 32, etc.) as inputs to evaluate the performance of our proposed model. Table VI shows that our model demonstrates the ability to adapt to varying input lengths while consistently delivering good performance. This is because the choice of observation frames exerts an influence on both the number of samples and the spatial-temporal information of each sample available. When the shorter observation length is chosen, more samples can be generated for model training, thereby enhancing the overall performance of the model [18]. Meanwhile, when longer observation frames are chosen, they have the capacity to capture a greater amount of spatial-temporal information for each sample, leading to potential improvements in performance compared to shorter observation frames.

5) Impact of Noisy Data: In real-world scenarios, encountering incomplete data sequences is inevitable, for example, tracking algorithm failures(resulting in the loss of skeleton and box data) and sensor data transfer issues(resulting in the loss of speed data). To evaluate our model's performance in the presence of noisy data, we introduce input data variability by randomly dropping frames or signals with different probabilities, thereby generating noisy data from the original dataset. Additionally, we employ two strategies for addressing lost frames or signals. 'Zero padding' (ZP) involves filling the missing data with zeros, while 'Median filling' (MF) entails using the average of the nearest data points on both sides to fill the missing data (employing the nearest available data from one side to fill in the gap when the first or last data is missing).

Table VII displays the test results obtained with various noisy data sets, employing different strategies. In the case of frame dropping, the prediction results experience a sharp decline as the loss rate increases across all datasets when the ZP strategy is employed. Conversely, the model manages to maintain a high level of performance when utilizing the MF strategy, sustaining this performance until the loss rate reaches 0.9, which corresponds to approximately 3 frames remaining. This is attributed to the skeleton data and box data are continuously changed over time. The MF strategy aids in preserving this temporal continuity and movement trend to a certain extent, which conveys information about the motion, movement, and size of the pedestrian.

When it comes to sensor data dropping, the decline in performance trend in PIE data mirrors that of frame dropping when applying the ZP and MF strategy. Nonetheless, the model consistently maintains a high level of performance in both the $JAAD_{all}$ and $JAAD_{beh}$ datasets, even when the ZP strategy is employed. This resilience can be attributed to the nature of the speed data in these datasets, which is manually estimated and represented in discrete states (such as stopped, moving slowly, moving fast, decelerating, and accelerating). Consequently, our model places a greater emphasis on the box and skeleton data, allowing it to maintain strong performance even in situations where the speed signal is lost within the sequence in these two datasets.

F. Model Profile and Prediction Examples

Table VIII shows the profiles of different SOTA models and PedAST-GCN(test on a GTX 1080). The preprocessing steps

TABLE VII
ABLATION STUDY OF NOISY DATA

	Dataset	Strategy	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Frame Dropping	$JAAD_{all}$	ZP	77.31	75.29	75.29	74.55	74.55	73.37	73.17	73.15	73.12	0
		MF	88.01	88.01	87.98	87.98	87.98	87.53	87.40	87.16	86.80	0
	$JAAD_{beh}$	ZP	67.35	62.25	53.29	46.83	41.61	40.51	40.32	40.32	40.32	0
		MF	68.82	68.82	68.71	67.68	67.57	67.46	67.26	67.01	66.33	0
	PIE	ZP	86.23	80.27	74.75	68.40	58.58	52.95	46.54	40.36	34.84	29.60
		MF	89.58	89.53	89.41	89.24	89.19	89.19	88.91	88.86	88.58	29.60
Sensor Dropping	$JAAD_{all}$	ZP	87.47	87.34	87.24	87.13	87.04	87.01	86.99	86.94	86.88	86.80
		MF	88.01	88.01	88.01	88.01	87.98	87.98	87.98	87.98	87.98	86.80
	$JAAD_{beh}$	ZP	69.05	68.93	68.93	68.93	68.93	68.93	68.93	68.93	68.05	67.82
		MF	69.05	69.05	69.05	69.05	69.05	69.05	69.05	69.05	69.05	67.82
	PIE	ZP	82.16	73.52	69.29	64.60	60.98	57.69	56.47	54.24	52.68	34.23
		MF	89.74	89.63	89.63	89.63	89.58	89.52	89.52	89.47	89.35	34.23

Notes:Various strategies are employed for data recovery in the face of missing values. "ZP" denotes zero padding, while "MF" stands for median filling.

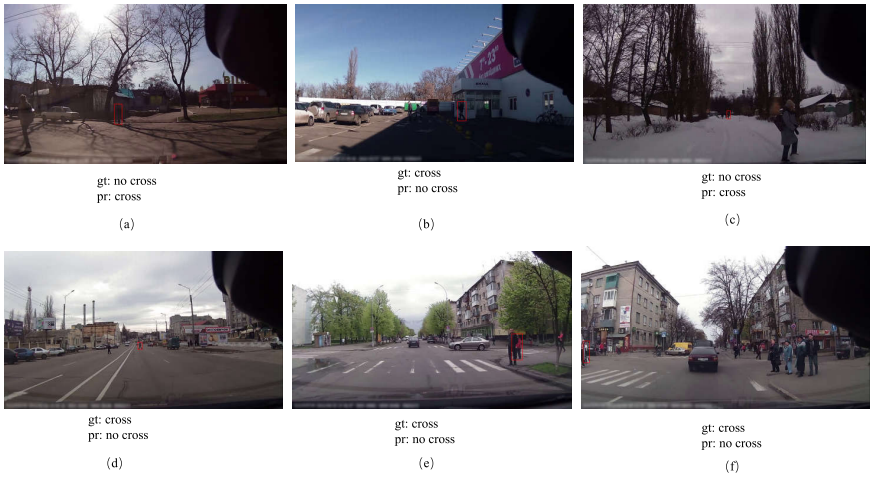


Fig. 6. Typical examples of detection failures. 'gt' is the ground truth classification value, and 'pr' is the predicted value. The illumination, remote distance, and occlusion of obstacles are the main reasons for failure detections.

TABLE VIII
COMPARISON OF PROFILES OF DIFFERENT SOTA MODELS AND PEDAST-GCN. M MEANS THE INFERENCE TIME OF THE MODEL BY ITSELF, AND M+D MEANS THE INFERENCE TIME OF THE MODEL AND THE INPUT DATA

Model	$JAAD_{all}$					
	Acc	GFlops	Inference time (ms)		Size (MB)	Params (Millions)
			M	M+D		
PCPA [16]	70.93	77.2	38.6	194.6	118.8	31.165
Global PCPA [13]	83.49	154.3	70.83	416.83	374.2	60.919
FUSSI-net [50]	60.96	4.34	34.92	190.92	8.4	0.996
Pedestrian Graph [17]	80.08	1.08	29.01	185.01	**0.22**	**0.06**
Pedestrian Graph + [19]	86.97	0.52	**5.47**	351.47	0.27	0.0703
PedAST-GCN	**88.01**	**0.229**	10	**166**	10.4	2.68

Notes: The process to obtain pose skeleton data using HRNet [38] takes 156 ms, while the method for acquiring segmentation maps using DeepLab v3 [52] requires approximately 190 ms.

involve acquiring pose skeleton data using keypoint detection with HRNet [38] and obtaining a segmentation map using DeepLab v3 [52]. GFLOPS stands for Giga Floating-Point Operations per Second. We compared the performance profiles of the SOTA models with that of PedAST-GCN. The PedAST-GCN model has the best accuracy performance while requiring minimum GFlops. The model prediction time (M) and the execution time it takes to produce the input data and perform the inference (M+D) were compared. The results indicate that our model has a consistently lower execution time (M+D) compared to the other models. We also demonstrated the effectiveness of using YOLOv5s+ DeepSORT to obtain bounding boxes in real scenarios (See Appendix B).

Fig. 6 shows typical examples of prediction failures. Generally, the reasons can be categorized into three classes, including illumination, remote distance, and occlusion of obstacles. For example, figure (a) and (b) has illumination issues which makes it difficult to recognize the pedestrian pose. The remote distance would also make the detection difficult due to the pedestrian being too small to recognize, as shown in (c) and (d). The failure detection in (e) and (f) is

the occlusion of obstacles, in which the model could not get the full features of pedestrians.

We also visually examined detection results using different models across various scenarios to further elucidate the advantages of our proposed method (See Appendix C).

V. Conclusion

Understanding and predicting the crossing behaviors of pedestrians is critical for vehicle intelligence. The paper proposes a Spatial-Temporal Attention Graph Convolution Network model for pedestrian crossing intention prediction. It uses a lightweight GCN model as the backbone network with simple but robust graph representations of pedestrian crossing intention modality features, including pedestrian pose keypoints, bounding box, and vehicle speeds. Attention mechanisms are developed to capture long-term temporal dependencies of dynamic graphs and fuse different modality features.

The model was validated by comparing benchmark models on two public datasets, including JAAD [5] and PIE [43]. The results highlight the accuracy and high computing speed performance of the proposed PedAST-GCN model in pedestrian crossing intention prediction. The ablation analysis verifies the importance of the backbone layer and graph design, different modality features for pedestrian crossing intention prediction, the prominent role of the attention mechanism in feature extraction and fusion, and the robust performance across various observation lengths and in the presence of noisy data. For example, The model with 'Temporal attention and Fusion attention' improves the prediction performance by 2.18%, 3.63%, and 0.8% on PIE, $JAAD_{beh}$, and $JAAD_{all}$ data set compared to the model without attention. The model prediction failures are mainly caused by the video quality, such as illumination, remote distances, and occlusion of obstacles. The proposed model currently does not utilize the global context as input to directly capture visual features encompassing multi-interactions, future studies will explore incorporating this information efficiently to enhance the model's robustness. Additionally, deploying our model in vehicles and enhancing its performance by observing more real-world data will be key areas of focus.

References

[1] C DMV. (2017). *Autonomous Vehicle Disengagement Reports 2017*. Accessed: Jan. 24, 2019. [Online]. Available: https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/disengagement_report_2017

[2] T. Mordan, M. Cord, P. Pérez, and A. Alahi, "Detecting 32 pedestrian attributes for autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 11823–11835, Aug. 2022.

[3] Y. Ling, Z. Ma, B. Xie, Q. Zhang, and X. Weng, "SA-BiGCN: Bi-stream graph convolution networks with spatial attentions for the eye contact detection in the wild," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 2, pp. 2089–2100, Feb. 2024.

[4] J. Gesnouin, S. Pechberti, G. Bresson, B. Stanciulescu, and F. Moutarde, "Predicting intentions of pedestrians from 2D skeletal pose sequences with a representation-focused multi-branch deep learning network," *Algorithms*, vol. 13, no. 12, p. 331, Dec. 2020.

[5] A. Rasouli, I. Kotseruba, and J. K. Tsotsos, "Are they going to cross? A benchmark dataset and baseline for pedestrian crosswalk behavior," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 206–213.

[6] D. Varytimidis, F. Alonso-Fernandez, B. Duran, and C. Englund, "Action and intention recognition of pedestrians in urban traffic," in *Proc. 14th Int. Conf. Signal-Image Technol. Internet-Based Syst. (SITIS)*, Nov. 2018, pp. 676–682.

[7] K. Saleh, M. Hossny, and S. Nahavandi, "Real-time intent prediction of pedestrians for autonomous ground vehicles via spatio-temporal DenseNet," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 9704–9710.

[8] B. Yang, W. Zhan, P. Wang, C. Chan, Y. Cai, and N. Wang, "Crossing or not? Context-based recognition of pedestrian crossing intention in the urban environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5338–5349, Jun. 2021.

[9] A. Singh and U. Suddamalla, "Multi-input fusion for practical pedestrian intention prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 2304–2311.

[10] H. Razali, T. Mordan, and A. Alahi, "Pedestrian intention prediction: A convolutional bottom-up multi-task approach," *Transp. Res. C, Emerg. Technol.*, vol. 130, Sep. 2021, Art. no. 103259.

[11] F. Li, S. Fan, P. Chen, and X. Li, "Pedestrian motion state estimation from 2D pose," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 1682–1687.

[12] X. Shi et al., "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.

[13] D. Yang, H. Zhang, E. Yurtsever, K. A. Redmill, and Ü. Özgüner, "Predicting pedestrian crossing intention with feature fusion and spatio-temporal attention," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 2, pp. 221–230, Jun. 2022.

[14] A. Ranga et al., "VRUNet: Multi-task learning model for intent prediction of vulnerable road users," 2020, *arXiv:2007.05397*.

[15] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1110–1118.

[16] I. Kotseruba, A. Rasouli, and J. K. Tsotsos, "Benchmark for evaluating pedestrian action prediction," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 1257–1267.

[17] P. R. G. Cadena, M. Yang, Y. Qian, and C. Wang, "Pedestrian graph: Pedestrian crossing prediction based on 2D pose estimation and graph convolutional networks," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 2000–2005.

[18] X. Zhang, P. Angeloudis, and Y. Demiris, "ST CrossingPose: A spatial–temporal graph convolutional network for skeleton-based pedestrian crossing intention prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 20773–20782, Nov. 2022.

[19] P. R. G. Cadena, Y. Qian, C. Wang, and M. Yang, "Pedestrian graph+: A fast pedestrian crossing prediction model based on graph convolutional networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 21050–21061, Nov. 2022.

[20] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6299–6308.

[21] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4489–4497.

[22] Y. Ling, Q. Zhang, X. Weng, and Z. Ma, "STMA-GCN_PedCross: Skeleton based spatial–temporal graph convolution networks with multiple attentions for fast pedestrian crossing intention prediction," in *Proc. IEEE 26th Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2023, pp. 500–506.

[23] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2016, pp. 3464–3468.

[24] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proc. IEEE Adapt. Syst. Signal Process., Commun., Control Symp.*, Oct. 2000, pp. 153–158.

[25] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," 2015, *arXiv:1508.04025*.

[26] Y. Zhou, G. Tan, R. Zhong, Y. Li, and C. Gou, "PIT: Progressive interaction transformer for pedestrian crossing intention prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 14213–14225, Dec. 2023.

[27] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 7444–7452.

[28] T. Sofianos, A. Sampieri, L. Franco, and F. Galasso, "Space-time-separable graph convolutional network for pose forecasting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 11209–11218.

[29] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with directed graph neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7912–7921.

[30] X. Zhang, C. Xu, and D. Tao, "Context aware graph convolution for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14321–14330.

[31] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 12026–12035.

[32] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3595–3603.

[33] X. Liu, Z. You, Y. He, S. Bi, and J. Wang, "Symmetry-driven hyper feature GCN for skeleton-based gait recognition," *Pattern Recognit.*, vol. 125, May 2022, Art. no. 108520.

[34] F. Zhou, X. Tu, Q. Wang, and G. Jiang, "Improved GCN framework for human motion recognition," *Scientific Program.*, vol. 2022, pp. 1–10, May 2022.

[35] W. Zhang, Z. Lin, J. Cheng, C. Ma, X. Deng, and H. Wang, "STA-GCN: Two-stream graph convolutional network with spatial–temporal attention for hand gesture recognition," *Vis. Comput.*, vol. 36, nos. 10–12, pp. 2433–2444, Oct. 2020.

[36] M. Shopon, A. S. M. H. Bari, and M. L. Gavrilova, "Residual connection-based graph convolutional neural networks for gait recognition," *Vis. Comput.*, vol. 37, nos. 9–11, pp. 2713–2724, Sep. 2021.

[37] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang, "Disentangling and unifying graph convolutions for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 143–152.

[38] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 5693–5703.

[39] C. Li et al., "YOLOv6: A single-stage object detection framework for industrial applications," 2022, *arXiv:2209.02976*.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[41] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lió, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

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

[43] A. Rasouli, I. Kotseruba, T. Kunic, and J. K. Tsotsos, "PIE: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 6262–6271.

[44] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[45] I. Kotseruba, A. Rasouli, and J. K. Tsotsos, "Do they want to cross? Understanding pedestrian intention for behavior prediction," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 1688–1693.

[46] J. Yue-Hei Ng et al., "Beyond short snippets: Deep networks for video classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4694–4702.

[47] A. Bhattacharyya, M. Fritz, and B. Schiele, "Long-term on-board prediction of people in traffic scenes under uncertainty," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4194–4202.

[48] A. Rasouli, I. Kotseruba, and J. K. Tsotsos, "Pedestrian action anticipation using contextual feature fusion in stacked RNNs," 2020, *arXiv:2005.06582*.

[49] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–9.

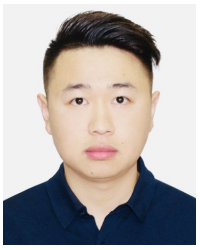
[50] F. Piccoli et al., "FuSSI-Net: Fusion of spatio-temporal skeletons for intention prediction network," in *Proc. 54th Asilomar Conf. Signals, Syst., Comput.*, Nov. 2020, pp. 68–72.

[51] J. Gesnouin, S. Pechberti, B. Stanciulcscu, and F. Moutarde, "TrouSPI-net: Spatio-temporal attention on parallel atrous convolutions and U-GRUs for skeletal pedestrian crossing prediction," in *Proc. 16th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, Dec. 2021, pp. 1–7.

[52] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.

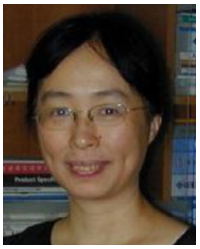
Yancheng Ling received the B.E. degree in traffic engineering and the Internet of Things engineering from East China Jiaotong University, Nanchang, China, in 2017, and the M.S. degree in traffic engineering from South China University of Technology, Guangzhou, Guangdong, China, in 2019, where he is currently pursuing the Ph.D. degree in traffic information engineering and control. He has been a Visiting Ph.D. Student with the Department of Civil and Architectural Engineering, KTH Royal Institute of Technology, since 2022. His research interests include mobile phone information collection and processing, intelligent transportation systems (ITSs), computer vision, video analysis, NLP, and traffic accident analysis.

Zhenliang Ma is currently an Associate Professor of road traffic engineering with the KTH Royal Institute of Technology. His research interests include statistics, machine learning, computer science-based modeling, simulation, optimization, and control within the framework of selected mobility-related complex systems, which are intelligent transport systems (traffic/public transport/rails) and personal information systems (transport/energy).



Qi Zhang received the B.E. and M.S. degrees in traffic information engineering and control from Wuhan University of Technology. He is currently pursuing the Ph.D. degree with the Department of Civil and Architectural Engineering, KTH Royal Institute of Technology, Stockholm. His research interests include deep learning, urban public transport, individual mobility, and knowledge graph.

Bangquan Xie received the B.S. degree from Changsha University of Science and Technology, China, in 2007, the M.S. degree from South China University of Technology, China, in 2013, where he is currently pursuing the Ph.D. degree. He has been a joint training Ph.D. Student with Clemson University International Center for Automotive Research (CU-ICAR), USA, since 2019. His research interests include sensor fusion, 3D detection, tracking and segmentation, multi-task learning, autoML, unsupervised and self-supervised, robot perception, autonomous driving, and automotive technology.



Xiaoxiong Weng received the bachelor's degree in industrial automation from Dalian University of Technology, China, the master's degree in automatic control theory and application from Shanghai Jiaotong University, China, and the Ph.D. degree in control theory and control engineering from South China University of Technology (SCUT), China. She is currently a Professor with the School of Civil Engineering and Transportation, SCUT. Her research interests include intelligent transportation systems, computer vision, dynamic modeling of urban traffic flow, data mining on transit systems, traffic signal control systems, transit commuter behavior analysis, and traffic accident analyzing.