Computer Modeling in Engineering & Sciences

Tech Science Press

https://doi.org/10.32604/cmes.2026.086105

ARTICLE

TrajFusionNet: Pedestrian Crossing Intention Prediction via Fusion of Sequential and Visual Trajectory Representations

François G. Landry* **and Moulay A. Akhloufi**

Perception, Robotics and Intelligent Machines Research Group (PRIME), Department of Computer Science, Université de Moncton, Moncton, NB, Canada

*Corresponding Author: François G. Landry. Email: efl7126@umoncton.ca

Received: 24 May 2026; Accepted: 20 August 2026; Published: 28 September 2026

ABSTRACT: With the introduction of vehicles with autonomous capabilities on public roads, predicting pedestrian crossing intention has emerged as an active area of research. The task of predicting pedestrian crossing intention involves determining whether pedestrians in the scene are likely to cross the road or not. In this work, we propose TrajFusionNet, a novel transformer-based model that leverages future pedestrian trajectory and vehicle speed predictions as priors for predicting crossing intention. TrajFusionNet comprises two branches: a Sequence Attention Module (SAM) and a Visual Attention Module (VAM). The SAM branch learns from a sequential representation of the observed and predicted pedestrian trajectory and vehicle speed. Complementarily, the VAM branch learns from a visual representation of the observed and predicted pedestrian trajectory by overlaying corresponding pedestrian bounding boxes onto scene images. In terms of performance, TrajFusionNet achieves state-of-the-art results on the Joint Attention in Autonomous Driving (JAAD) and Pedestrian Intention Estimation (PIE) datasets. By utilizing a small number of lightweight modalities, it also achieves the lowest total inference time (including model runtime and data preprocessing) among state-of-the-art approaches.

KEYWORDS: Pedestrian crossing intention; pedestrian trajectory; autonomous vehicle; transformer

1 Introduction

Substantial efforts have been devoted to developing methods for pedestrian detection and identifying pedestrians actively crossing [1,2]. The latter is typically framed as an action recognition problem, where the goal is to infer a class label (e.g., whether a pedestrian is *currently* crossing) based on observed actions. Action prediction, on the other hand, involves anticipating whether an action will occur in the future, making it inherently more challenging. Predicting pedestrian crossing intention is particularly challenging due to the fact that pedestrians are highly dynamic, can change direction quickly [3] and that many factors can change the trajectories of pedestrians [4]. From the ego-vehicle perspective, pedestrians can easily get occluded by other pedestrians or by objects such as other vehicles and structures along the road. Pedestrian movement is highly dependent on other traffic objects in the scene such as other agents (e.g., vehicles, other pedestrians) and traffic elements (e.g., sidewalks, zebra crossings, traffic lights). Despite the difficulty surrounding crossing intention prediction, the task has significant practical applications. Predicting whether a pedestrian is likely to cross the road can enable autonomous vehicles to compute and potentially execute an avoidance plan (e.g., braking or changing direction) in advance, reducing the risk of collisions.

Previous works in the field of pedestrian crossing intention prediction have reached improved predictive performance by leveraging multiple modalities and computing higher-level features such as semantic segmentation maps [5–8] and pose keypoints [6,8–11]. However, computing these features substantially increases inference time, as generating segmentation maps and pose keypoints is computationally expensive. When computed over multiple past frames or for multiple pedestrians in the scene, these features can incur an overhead of several hundred milliseconds on a commodity GPU [12], limiting the suitability of these approaches for real-time applications.

In this work, we introduce TrajFusionNet[1], a novel model for predicting pedestrian crossing intention (Fig. 1). We demonstrate that leveraging lightweight modalities, specifically pedestrian coordinates, vehicle speed, and raw scene images, is sufficient to achieve state-of-the-art performance while enabling low inference latency necessary for real-time applications. TrajFusionNet is composed of two branches: a Sequence Attention Module (SAM) and a Visual Attention Module (VAM). In the SAM branch, an encoder-decoder transformer is used to predict the future pedestrian trajectory and vehicle speed. This sequential representation, along with the past observed trajectory, is then used as input to a transformer encoder. Complementarily, the VAM branch enables learning from a visual representation of the pedestrian trajectory by overlaying observed and predicted pedestrian bounding boxes onto scene images. The VAM branch is composed of two VAN (Visual Attention Network [13]) blocks.

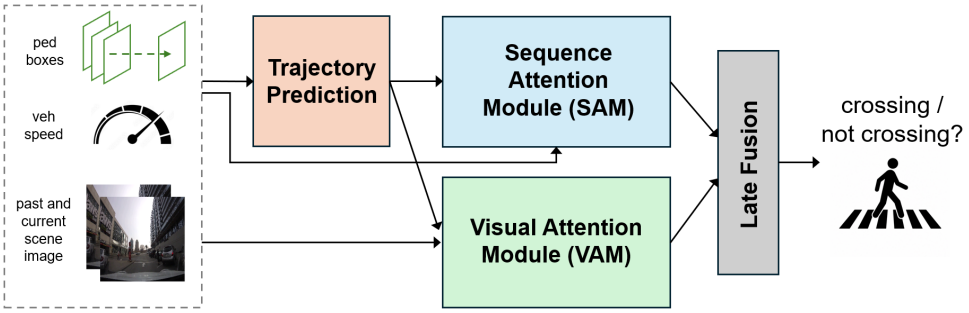

Figure 1: Overview of the proposed TrajFusionNet architecture.

TrajFusionNet therefore learns from a fusion of complementary trajectory representations: a sequential representation in the SAM branch (as pedestrian coordinates) and a visual representation in the VAM branch (as trajectory bounding boxes overlaid on scene images). Both trajectory representations are enriched with predicted pedestrian trajectories, which are used as priors by the network for predicting crossing intention.

TrajFusionNet achieves state-of-the-art performance across the most widely used datasets for pedestrian crossing intention prediction, the Joint Attention in Autonomous Driving (JAAD) dataset [14] and the Pedestrian Intention Estimation (PIE) dataset [15]. Additionally, by employing a small number of lightweight modalities, TrajFusionNet minimizes total inference time (including model runtime and data preprocessing), outperforming state-of-the-art approaches in terms of inference speed.

The main contributions of this work are summarized as follows:

- We propose TrajFusionNet, a novel framework for pedestrian crossing intention prediction based on a dual-branch architecture composed of a Sequence Attention Module (SAM) and a Visual Attention Module (VAM).

[1]The source code is publicly available at: https://github.com/fglandry/TrajFusionNet

- We show that lightweight modalities, namely pedestrian coordinates, vehicle speed, and raw scene images, are sufficient to achieve state-of-the-art performance and facilitate low inference times.
- We demonstrate performance gains by leveraging predicted pedestrian trajectories and vehicle speed as priors for predicting pedestrian crossing intention.
- We validate our approach on the JAAD and PIE datasets, where it achieves state-of-the-art performance with significantly reduced inference time.

2 Related Work

In this section, we provide a short review of the literature on pedestrian crossing intention prediction. For more extensive surveys, the reader is referred to the following reviews: [3,16,17]. The first approaches proposed in the literature for pedestrian crossing intention were based on probabilistic models [18,19] and on applying machine learning classifiers to hand-crafted features [20,21]. The early deep learning methods involved applying CNNs (convolutional neural networks) to parts of the scene image such as the area surrounding the pedestrian [14,22]. To integrate the time dimension across video frames, authors have proposed using 3D CNN-based models [23] or applying RNNs (recurrent neural networks) to sequential features [24,25].

A variety of features were proposed for being used as part of sequential models, including pedestrian pose keypoints, bounding box coordinates and the vehicle speed. Researchers were able to improve predictive results by leveraging more modalities as input into RNN-based models, usually at the cost of increased inference time [26,27]. State-of-the-art results were obtained using hierarchical RNN architectures, which are composed of multiple RNNs where the output of a lower-level RNN is used as the input of a higher-level RNN, such as SF-GRU [28] and SafeCrossNet [29].

The advent of transformers [30] led to an increase in predictive performance compared to other sequential models such as RNNs. Achaji et al. [31] obtained state-of-the-art results by using an encoder-decoder transformer model and using solely bounding box coordinates as input features. LIM [11] utilizes only skeleton sequences and passes features at different levels (joint, part, and global) through attention layers. Zhou et al. [32] propose PIT (Progressive Interaction Transformer), where a transformer layer is applied to each video frame in a temporal sequence, allowing early interactions between modalities. Vision transformers, which apply the self-attention mechanism to image patches, have been used for pedestrian intention prediction as part of approaches proposed in [33–35]. For instance, Elgazwy et al. [35] employ vision transformers as part of a multi-branch architecture to extract both local and global context features.

A few works have employed knowledge distillation to transfer knowledge from a teacher network to a student model, enabling the deployment of a lightweight student model with lower inference time [10] or facilitating the transfer of knowledge learned in the synthetic domain to the real domain [36]. Bai et al. [10] propose to use knowledge distillation with a student network composed of a basic transformer encoder using only bounding box coordinates and a teacher network leveraging 3D convolutions and attention layers applied to multiple modalities.

Another family of approaches proposed in the literature consists of generative approaches, which tackle the pedestrian intention problem by predicting future representations, which are typically future frames. Gujjar and Vaughan [37] and Chaabane et al. [38] propose to use an encoder-decoder network to predict future frames, using a 3D CNN as the encoder and a convolutional LSTM (long short-term memory) at the decoder level. An important weakness of generative approaches is their high inference time which is reported as over 100 ms on commodity GPUs [37,38]. TrajFusionNet draws inspiration from previously proposed generative approaches; however, instead of predicting future image frames as priors, we predict lightweight modalities (future bounding box locations and vehicle speed), allowing to limit inference time.

Graph-based approaches have been very successful in pedestrian crossing intention prediction [7,8, 39–41], as well as in the closely related task of trajectory prediction [42,43]. Graph-based approaches aim to model spatiotemporal dependencies between elements in the scene with graphs. The most common deep learning model that allows learning from graph structures is the graph convolutional network (GCN). A GCN applied to a scene graph is used by Song et al. [7], where each node represents an object in the scene (pedestrian or traffic light) and each edge represents the strength of the interaction between the two nodes. GCN models have also been applied to pedestrian pose keypoints in [8,39,40]. In PedAST-GCN, Ling et al. [41] use graph representations that allow to integrate the bounding box and vehicle speed modalities in addition to pose keypoints into GCN layers, which are then followed by attention layers.

Some authors have proposed approaches that leverage multi-task learning, where the model is trained to solve multiple tasks at the same time. As such, pedestrian crossing prediction has been combined with other tasks such as trajectory prediction [5,27,31,44,45], target location prediction [44], time to cross the street [46] and predicting additional pedestrian actions (e.g., walking, standing) [27,34,47,48]. Multi-task learning allows the sharing of commonalities across tasks through a shared representation, leading to an efficient use of model parameters. Multi-task models typically contain a backbone of common layers followed by task-specific heads at the end of the network. In other models, the output from one subtask is used as the input of another task [31,45]. TrajFusionNet follows this approach by leveraging pedestrian trajectory and vehicle speed predictions as priors for predicting crossing intention.

3 Method

3.1 Problem Formulation

The problem at hand consists of predicting whether a pedestrian is going to cross (or not cross) during a video sequence, given a set of observation frames, and for all pedestrians detected. The pedestrian crossing intention problem can be formulated as follows. It consists of a binary classification task at time t where crossing action $A_i \in \{0,1\}$ is predicted in the future for pedestrian i given previous observations $\mathbf{M}_i$:

$$\mathbf{M}_i = \{\mathbf{m}_i^{t-n}, \mathbf{m}_i^{t-n+1}, \ldots, \mathbf{m}_i^{t}\} \tag{1}$$

where $\mathbf{m}_i^t$ is an observation at time t for pedestrian i and n is the number of frames in the observation period.

We follow the benchmark parameters proposed by Kotseruba et al. [49] and predict whether the pedestrian will cross between 1 and 2 s after time t.

3.2 Input Modalities

Our model uses three input modalities: a sequence of pedestrian bounding boxes, a sequence of vehicle speeds, and a sequence of scene images. Accordingly, $\mathbf{M}_i$ is composed of three tensors: $\mathbf{B}_i = \{\mathbf{b}_i^{t-n}, \mathbf{b}_i^{t-n+1}, \ldots, \mathbf{b}_i^{t}\}$, the bounding box sequence, $\mathbf{V} = \{v^{t-n}, v^{t-n+1}, \ldots, v^{t}\}$, the vehicle speed sequence, and $\mathbf{I} = \{\mathbf{i}^{t-n}, \mathbf{i}^{t-n+1}, \ldots, \mathbf{i}^{t}\}$, the scene video sequence. Each bounding box $\mathbf{b}_i^t$ consists of four values: the x and y coordinates of its top-left and bottom-right corners. Vehicle speed is treated differently depending on the dataset used. For the PIE dataset [15], we use the raw speed values directly. For the JAAD dataset [14], which provides only categorical speed labels, we encode them ordinally as follows: {0: stopped, 1: decelerating, 2: moving slow, 3: moving fast, 4: accelerating}.

3.3 Architecture

The proposed architecture is shown in Fig. 2. The model is composed of two branches: a *Sequence Attention Module (SAM)* and a *Visual Attention Module (VAM)*. The two branches are merged into a late-fusion fashion with a dense layer.

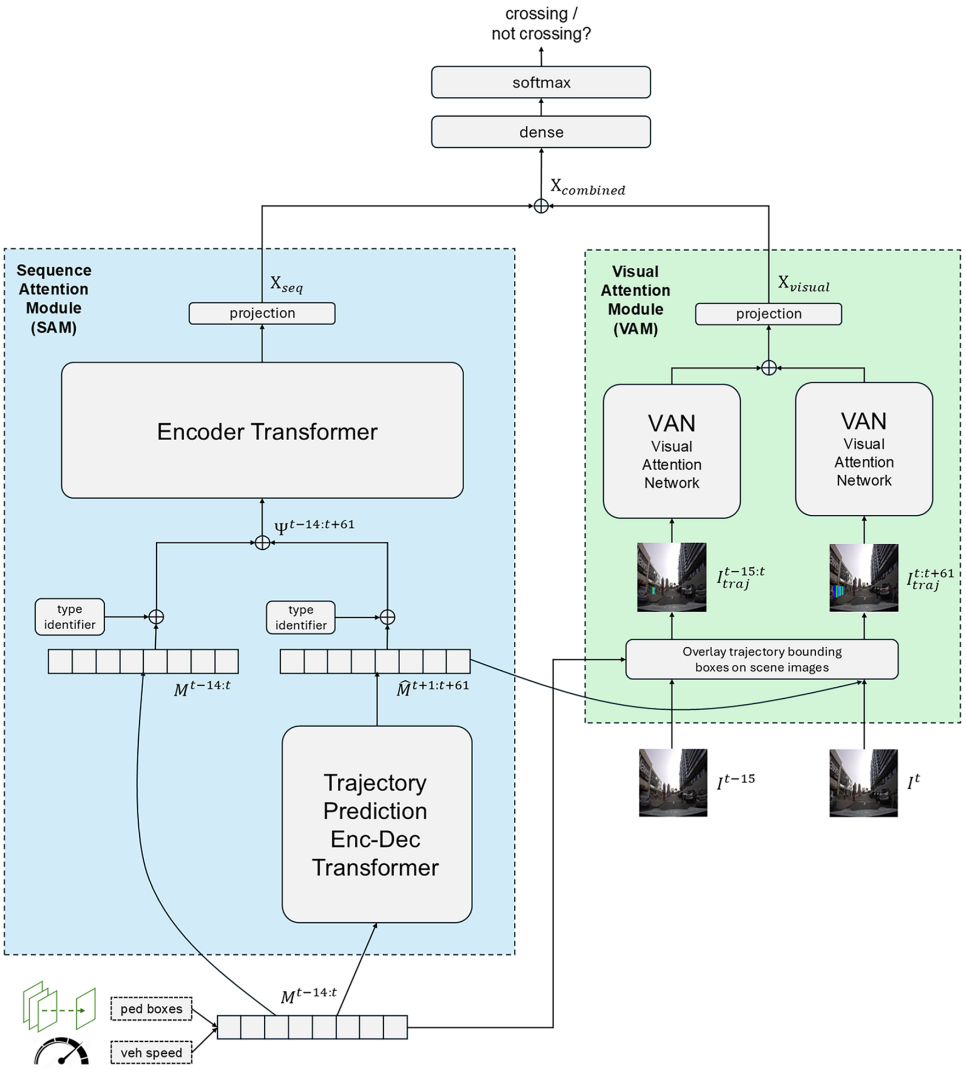

Figure 2: TrajFusionNet architecture. The SAM branch (**left**) predicts the future pedestrian trajectory and vehicle speed, which are appended to the past observed trajectory and vehicle speed to form a sequence that is then processed by a transformer encoder. In the VAM branch (**right**), observed and predicted pedestrian bounding boxes are overlaid on the first and last scene images of the observation period, respectively, with each image processed by a separate instance of the VAN network [13]. The outputs of the two branches are then combined through late fusion.

Sequence Attention Module (SAM): The role of the SAM branch is to extract insights by applying the attention mechanism to a sequential representation of past observed and predicted pedestrian coordinates and vehicle speed. The SAM branch is composed of two transformer blocks. The first transformer is an encoder-decoder transformer that is used to perform trajectory prediction to estimate future pedestrian bounding boxes and future vehicle speed. The trajectory is predicted over the next 60 frames (up until 2 s after time t). The trajectory prediction tensor, $\hat{\mathbf{M}}^{t+1:t+61}$, is then concatenated with the past trajectory tensor,

$\mathbf{M}^{t-14:t}$, to form a new tensor, $\psi^{t-14:t+61}$. $\psi^{t-14:t+61}$ is fed to the input of an encoder-only transformer, whose task is to output an encoding to be used for classification.

Trajectory Prediction Transformer: The trajectory prediction transformer consists of a non-autoregressive encoder-decoder transformer. The future trajectories are all predicted in one pass (i.e., the previously generated token is not required to generate the next token), which allows to reduce the inference time considerably. When predicting the trajectory, we predict both the future bounding box coordinates and the future vehicle speed.

We use the non-autoregressive encoder-decoder transformer implementation provided in the TSLib times series library [50]. The input to the encoder consists of the past observed trajectory sequence, $\mathbf{M}^{t-14:t} \in \mathbb{R}^{d_{\text{seq}} \times m}$, where d_{seq} is the length of the past trajectory (15 frames) and m is the input modalities dimension ($m = 5$). For the pedestrian bounding box modality, all coordinates in the sequence are offset by subtracting the coordinates of the first bounding box at time $t - 15$. The input to the decoder consists of the concatenation of the past trajectory sequence, $\mathbf{M}^{t-14:t} \in \mathbb{R}^{d_{\text{seq}} \times m}$, with an empty tensor, $\mathbf{0} \in \mathbb{R}^{d_{\text{pred}} \times m}$, where d_{pred} is the length of the predicted trajectory (60 frames).

At the end of the decoder layer stack, a projection layer transforms the output into a tensor of dimensions $(d_{\text{seq}} + d_{\text{pred}}) \times m$, where $d_{seq} + d_{pred}$ represents the total length of the past and predicted sequences (75 frames). At the output of the decoder, we only keep the predicted sequence (last 60 frames), leading to the trajectory prediction tensor $\hat{\mathbf{M}}^{t+1:t+61} \in \mathbb{R}^{d_{\text{pred}} \times m}$.

Sequence Type Identifiers: Before sending the past observed trajectory tensor, $\mathbf{M}^{t-14:t}$, and the predicted trajectory tensor, $\hat{\mathbf{M}}^{t+1:t+61}$, to the subsequent encoder transformer, we augment the two tensors with sequence type identifiers. Sequence type identifiers encode whether sequence tokens consist of past trajectory values or rather of trajectory values predicted in the future. This helps attention heads to attend specifically to past trajectory tokens and to ignore future trajectory tokens, or vice versa. The sequence type identifiers are inspired by the node/edge type identifiers proposed by Kim et al. [51] for applying transformers to graph structures. We simply append a fixed scalar of 0 to past trajectory tokens and a fixed scalar of 1 to future trajectory tokens, such that:

- $\mathbf{M}^t$ is augmented as $[\mathbf{M}^t, 0]$
- $\hat{\mathbf{M}}^t$ is augmented as $[\hat{\mathbf{M}}^t, 1]$

The resulting tensors are concatenated to form a new tensor, $\boldsymbol{\psi}^{t-14:t+61}$. $\boldsymbol{\psi}^{t-14:t+61}$ is then fed to the input of an encoder-only transformer whose task is to output an encoding to be used for classification. A final projection layer is added at the end of the SAM branch.

Visual Attention Module (VAM): The role of the VAM branch is to extract insights by applying the attention mechanism to a visual representation of the pedestrian trajectories within the surrounding contextual scene. The VAM branch is composed of Visual Attention Networks (VANs) [13]. The VAN network is based on large kernel attention (LKA), which is composed of three components: a spatial local convolution, a spatial long-range convolution, and a channel convolution. The LKA combines the advantages of convolution, such as the ability to capture local structures, with the advantages of self-attention, notably the capacity to model long-range dependencies. In pedestrian crossing intention prediction, local cues such as pedestrian appearance and orientation can be important for understanding behavior, while long-range dependencies may arise from interactions between pedestrians and vehicles that are spatially distant in the ego view.

The VAN network is applied to full scene images captured from the ego-vehicle. These images are augmented with colored rectangles that correspond to the pedestrian bounding boxes associated with the

past observed and predicted trajectories. This allows the VAN network to capture interactions between the pedestrian trajectory and the surrounding context. Two instances of VAN are used:

- The first VAN instance is applied to the first video frame available in the observation period, $\mathbf{I}^{t-15}$. Rectangles corresponding to the observed pedestrian bounding boxes ($\mathbf{M}_{1:4}^{t-14:t}$) are added to the image, forming $\mathbf{I}_{traj}^{t-15:t}$, which is used as input into the first VAN network.
- The second VAN instance is applied to the last frame available in the observation period, $\mathbf{I}^{t}$. Rectangles corresponding to the predicted pedestrian bounding boxes ($\hat{\mathbf{M}}_{1:4}^{t+1:t+61}$) are added to the image, forming $\mathbf{I}_{traj}^{t:t+61}$, which is used as input into the second VAN network.

Rectangles corresponding to the pedestrian trajectory bounding boxes are added to two channels only (blue and green channels in the RGB image). One channel is kept so that the pedestrian appearance can still be leveraged by the network. Examples of scene images augmented with trajectory bounding boxes are shown in Fig. 3. The output from the two VAN networks are concatenated and then fed into a projection layer whose output is to be used for classification.

Figure 3: Scene images augmented with trajectory boxes. (**left**) corresponds to the first frame available in the observation period augmented with observed pedestrian bounding boxes $\mathbf{I}_{traj}^{t-15:t}$. (**right**) corresponds to the last frame available in the observation period augmented with predicted pedestrian bounding boxes $\mathbf{I}_{traj}^{t:t+61}$.

Late Fusion: At the end of the network, the outputs from the Sequence Attention Module (SAM) and the Visual Attention Module (VAM) are merged in a late-fusion fashion using dense layers of sizes 80, 40 and 2.

3.4 Training and Model Settings

Modular Training: Given the modularity and relative size of our model, we train it in steps, where each module is pretrained separately. After pretraining the lower modules, we freeze their weights and proceed to fine-tune the subsequent modules. To train each module, we append a classification head composed of two dense layers (with 40 and 2 neurons, respectively), which then gets removed when the trained module is added to the rest of the model. Our approach resembles layer-wise pretraining from early deep learning models [52] and shares similarities with Progressive Neural Networks [53], where new modules are trained sequentially while previously learned components are frozen.

Training Procedure and Learning Objectives: The first module that we pretrain is the trajectory prediction encoder-decoder transformer. This transformer is trained to predict the pedestrian bounding

box and vehicle speed over the next 60 frames ($\hat{\mathbf{M}}^{t+1:t+61}$). To train the trajectory prediction transformer, we extract overlapping sequences from pedestrian tracks, with the overlap factor being equal to the one suggested in the benchmark proposed by Kotseruba et al. [49]. For the PIE dataset, this results in 49,527 sequences in the training set. We use a mean square error loss function as the training objective:

$$L = \frac{1}{NTC} \sum_{i=1}^{N} \sum_{j=1}^{T} \sum_{k=1}^{C} \left(y_{ijk} - \hat{y}_{ijk} \right)^2 \tag{2}$$

where N is the number of training samples, T is the length of the predicted trajectory (60 frames), and C is the number of values predicted as part of the trajectory. The latter includes five z-score normalized values: the x and y coordinates of the upper-left and bottom-right corners of the bounding box, as well as the vehicle speed. We provide an evaluation of trajectory prediction error in Appendix A.

Once the trajectory prediction transformer has been trained, we now train the encoder transformer. We freeze the weights of the trajectory prediction transformer and only train the new weights added with the encoder transformer, as well as additional dense layers for classification. For training, we use the same dataset splits and evaluation parameters (track split, observation length and prediction horizon) as specified in the benchmark proposed by Kotseruba et al. [49]. For the PIE dataset, this results in 4770 sequences in the training set. We use a weighted cross-entropy loss function as the training objective:

$$L = -\frac{1}{N} \sum_{i=1}^{N} \Big(\alpha y_i \log(\hat{y}_i) + (1-\alpha)(1-y_i) \log(1-\hat{y}_i) \Big) \tag{3}$$

where N is the number of training samples and α is a weighting factor to account for class imbalance.

The VAN networks are trained in a similar manner, with additional dense layers for classification. Each VAN is trained independently. The trajectory prediction transformer's weights are frozen, and its output is used to overlay trajectory bounding boxes on the scene images, which serve as input to the second VAN instance.

In the final training step, the complete model is trained with the weights of the SAM and VAM branches frozen. Only the weights of the projection layers at the ends of the branches and the final dense layers are updated.

Implementation and Training Parameters: When conducting experiments, we keep the implementation and training parameters consistent across datasets during training/inference. In the trajectory prediction encoder-decoder transformer, we set the number of encoder layers to 8, the number of decoder layers to 8, the number of attention heads to 4, d_{model} to 128, and the dimension of the fully-connected layers to 512. In the subsequent encoder transformer, we set the number of encoder layers to 6, the number of attention heads to 8, d_{model} to 128, and the dimension of the fully-connected layers to 1024. These architecture parameters, as well as the learning rate, were optimized on the validation set using grid search.

When it comes to the two VAN models, we keep the parameters suggested by the authors in the VAN-B2 version [13], which contains 26.6M parameters. The training parameters used for each module of the model are shown in Table 1. We provide the training and validation loss curves for the final training stage of TrajFusionNet in Appendix B.

Table 1: Parameters used for training each module of the model during modular training.

Module	Learning Rate	Epochs	Batch Size
Trajectory prediction enc-dec TF	5e−5	40	64
Encoder TF	5e−6	60	16
VAN	5e−5	15	16
Remaining layers	5e−6	60	16

We also train a lightweight version of the TrajFusionNet architecture, which we name TrajFusionNet-Small, and that contains significantly less parameters (5.20M parameters vs. 58.28M parameters for the full TrajFusionNet model). In TrajFusionNet-Small, the dimension of fully-connected layers in both transformers is reduced to 256; the encoder-decoder transformer is downsized to 2 encoder layers, 2 decoder layers, and 2 attention heads; and the encoder transformer is downsized to 2 encoder layers and 2 attention heads. A single VAN instance is applied to the last frame available in the observation period, overlaid with both the observed and predicted bounding boxes. The VAN-B0 version [13] of VAN is used, which contains 4.1M parameters.

4 Evaluation

4.1 Datasets

Two datasets are used to evaluate our model on predicting pedestrian crossing intention: JAAD (Joint Attention for Autonomous Driving) [14] and PIE (Pedestrian Intention Estimation) [15]. These two datasets are the most popular datasets used in the literature for the prediction of pedestrian crossing intention.

JAAD: The JAAD dataset [14] is a naturalistic dataset providing 346 clips of pedestrians prior to crossing events. It was filmed in North America and Europe and under varying weather conditions. The dataset provides ground truth bounding boxes for pedestrians as well as behavioral tags describing pedestrian actions. There are also behavioral tags assigned to the vehicle driver, although a numerical value for vehicle speed is not provided. The JAAD dataset has been divided into two subsets [49]: JAAD_{beh}, which is skewed towards pedestrians who are crossing or are about to cross, and JAAD_{all}, which consists of the complete dataset and contains an additional 2100 visible pedestrians who are away from the road and are not crossing.

PIE: The same authors who proposed the JAAD dataset compiled a second dataset called PIE (Pedestrian Intention Estimation) [15]. PIE is a larger dataset (with almost 10 times the number of frames) and contains longer pedestrian clips and better ego-vehicle information (including speed, GPS location, and heading angle). Compared to JAAD, pedestrians are more diverse in appearance, in the type of behavior they exhibit, and in their location with respect to the curb. However, the PIE dataset was filmed only in Toronto, Canada in clear weather, making it less diverse in terms of driving context.

4.2 Evaluation Procedure

We follow the same evaluation procedure as proposed in the benchmark by Kotseruba et al. [49]. We use 0.53 s for the observation length (16 frames) and predict pedestrian crossing 1 to 2 s (30 to 60 frames) after the observation period. We train and test our model on three datasets: PIE, JAAD_{all}, and JAAD_{beh}. The results reported for each dataset correspond to the model trained and tested on that specific dataset. We provide results using the following common classification metrics: accuracy (Acc), area under the curve (AUC), F1-score (F1), precision (P), and recall (R).

4.3 Results

Table 2 shows a comparison of state-of-the-art (SOTA) methods for the prediction of crossing intention. Values shown in bold correspond to the best value obtained across all models, while values that are underlined correspond to the second best value.

Table 2: Comparison of state-of-the-art methods for the prediction of crossing intention on the PIE and JAAD datasets. Values in bold indicate the best performance, while underlined values represent the second-best performance. Results for our models are reported as the mean ± standard deviation over five independent runs.

Model	Year	PIE					$JAAD_{all}$					$JAAD_{beh}$				
		Acc	AUC	F1	P	R	Acc	AUC	F1	P	R	Acc	AUC	F1	P	R
PCPA [49]	2020	0.86	0.91	0.78	0.69	0.89	0.83	0.77	0.57	0.50	0.66	0.58	0.47	0.73	0.61	0.92
TrouSPI-Net [9]	2021	0.88	0.88	0.80	0.73	0.89	0.85	0.73	0.56	0.57	0.55	0.64	0.56	0.76	0.66	0.91
Yang et al. [6]	2022	0.89	0.86	0.80	0.79	0.81	0.83	0.82	0.63	0.51	0.81	0.62	0.51	0.76	0.63	**0.95**
Pedestrian Graph + [8]	2022	0.89	0.90	0.81	0.83	0.79	0.86	0.88	0.65	0.58	0.75	0.70	0.70	0.76	0.77	0.75
Song et al. [7]	2022	**0.92**	0.91	**0.86**	0.82	**0.90**	0.87	0.81	0.65	0.60	0.71	0.68	0.60	0.78	0.68	0.91
Bai et al. [10]	2022	0.89	0.88	0.79	0.74	0.84	0.86	0.81	0.77	0.74	0.81	0.64	0.66	0.76	0.70	0.89
DPCIAN [39]	2023	0.91	0.88	0.83	0.83	0.84	**0.89**	0.77	0.59	0.61	0.58	0.71	0.66	0.78	0.73	0.85
PIT [32]	2023	0.91	0.90	0.82	0.85	0.79	0.87	0.87	0.66	0.54	**0.85**	0.70	0.65	0.81	0.71	0.93
PedAST-GCN [41]	2024	0.91	0.94	0.83	**0.88**	0.79	**0.89**	0.83	0.68	0.67	0.69	0.69	0.66	0.79	0.68	0.93
LIM [11]	2025	0.91	**0.96**	0.84	0.84	0.85	**0.89**	**0.91**	0.68	0.69	0.67	0.72	**0.77**	0.75	**0.85**	0.67
Gated-S2R-PCP [36]	2025	**0.92**	0.90	0.82	0.83	0.80	0.88	0.84	**0.79**	**0.77**	0.82	0.66	0.68	0.77	0.71	0.85
TrajFusionNet-Small (ours)	2026	**0.92** ± 0.01	0.91 ± 0.01	**0.86** ± 0.01	0.85 ± 0.03	0.87 ± 0.02	0.86 ± 0.01	0.82 ± 0.02	0.66 ± 0.04	0.59 ± 0.06	0.76 ± 0.04	0.69 ± 0.02	0.60 ± 0.02	0.79 ± 0.02	0.69 ± 0.02	0.89 ± 0.01
TrajFusionNet (ours)	2026	**0.92** ± 0.01	0.91 ± 0.01	**0.86** ± 0.01	0.85 ± 0.02	0.88 ± 0.03	**0.89** ± 0.01	0.85 ± 0.03	0.72 ± 0.03	0.67 ± 0.03	0.78 ± 0.05	**0.74** ± 0.03	0.67 ± 0.04	**0.82** ± 0.02	0.72 ± 0.03	**0.95** ± 0.02

Although the best metric scores are distributed across multiple models, TrajFusionNet achieves the highest number of top scores (when including Acc, AUC, and F1), establishing state-of-the-art overall performance. We note that one other method, LIM [11], attains a comparable, albeit slightly lower, number of top metric scores. In particular, TrajFusionNet consistently performs well in terms of accuracy and F1 score, achieving the highest values on all datasets except $JAAD_{all}$, where it ranks third in F1 score. TrajFusionNet also achieves consistently strong results across datasets, indicating that the model is less prone to overfitting on specific datasets.

4.4 Inference Time

Table 3 compares the inference times of TrajFusionNet with several SOTA approaches. The inference times were measured on a consumer-grade GPU (NVIDIA GeForce RTX 3060). Only approaches for which the authors have provided the source code are included. Two inference times are reported: the inference time of the model alone (M) and the inference time of the model combined with data preprocessing (M + D). Pedestrian detection and tracking are not included in D, as they are shared across all approaches compared.

Data preprocessing includes the time required to compute modalities required by the models, such as pose estimation and segmentation maps. For instance, PCPA [49] uses OpenPose [54] to compute pose estimation, which takes 13.99 ms to run on an RTX 3060 GPU. PCPA, as well as other approaches such as Yang et al. [6] and Pedestrian Graph+ [8], generate pose keypoints for each observation frame. The latter

two approaches also compute segmentation maps using DeepLabV3 [55], which requires 13.95 ms to execute on the RTX 3060 GPU. Yang et al. generate one map per observation frame, whereas Pedestrian Graph + generates a single map.

Table 3: Inference time obtained by state-of-the-art models and their respective number of parameters. M represents the inference time of the model alone, while M + D represents the inference time of the model combined with data preprocessing.

Model	Inference Time (ms)		Params (millions)
	M	M + D	
PCPA [49]	15.51	239.35	31.17
Yang et al. [6]	2.64	449.68	2.99
Pedestrian Graph+ [8]	2.67	240.46	0.07
TrajFusionNet-Small (ours)	4.63	4.63	5.20
TrajFusionNet (ours)	12.04	12.04	58.28

Table 3 shows that although TrajFusionNet has a relatively large number of parameters (58.28M) and an intermediate model-only inference time, its total inference time (M + D) is only 12.04 ms, which is by far the lowest among all compared approaches. TrajFusionNet only requires three lightweight modalities (pedestrian coordinates, vehicle speed, and raw scene images), meaning it does not spend time computing computationally expensive modalities such as pose keypoints and segmentation maps. The lightweight version of TrajFusionNet, TrajFusionNet-Small, has approximately 10 times fewer parameters (5.20M) and reduces the total inference time (M + D) to only 4.63 ms.

4.5 Ablation Study

An ablation study was performed to identify how different components of TrajFusionNet contribute to the model performance. Table 4 shows different scenarios in which one component of the network was removed or modified.

Table 4: Ablation study where the proposed TrajFusionNet architecture is compared with various architectural modifications.

	Scenario	PIE					$JAAD_{all}$					$JAAD_{beh}$				
		Acc	AUC	F1	P	R	Acc	AUC	F1	P	R	Acc	AUC	F1	P	R
Base	TrajFusionNet architecture	0.92	0.91	0.86	0.85	0.88	0.89	0.85	0.72	0.67	0.78	0.74	0.67	0.82	0.72	0.95
1	Combine branches with a modality self-attention layer	0.92	0.91	0.86	0.83	0.89	0.88	0.84	0.71	0.66	0.77	0.71	0.64	0.81	0.70	0.95
2	Use a single VAN network with combined overlays of past and predicted trajectories	0.92	0.91	0.86	0.83	0.89	0.88	0.82	0.68	0.66	0.70	0.72	0.66	0.81	0.73	0.91
3	Remove type IDs in SAM branch	0.92	0.90	0.85	0.85	0.86	0.88	0.82	0.69	0.65	0.72	0.72	0.66	0.80	0.72	0.88

(Continued)

Table 4 (continued)

	Scenario	PIE					$JAAD_{all}$					$JAAD_{beh}$				
		Acc	AUC	F1	P	R	Acc	AUC	F1	P	R	Acc	AUC	F1	P	R
4	Remove vehicle speed from input modalities	0.88	0.87	0.79	0.78	0.80	0.89	0.83	0.70	0.66	0.75	0.71	0.62	0.80	0.69	0.95
5	Remove trajectory prediction from SAM branch	0.89	0.88	0.82	0.79	0.84	0.89	0.81	0.70	0.69	0.70	0.70	0.63	0.79	0.70	0.91
6	Remove trajectory prediction from VAM branch	0.92	0.91	0.86	0.83	0.89	0.87	0.79	0.64	0.61	0.67	0.71	0.62	0.80	0.70	0.92
7	Remove SAM branch	0.85	0.81	0.73	0.75	0.72	0.87	0.83	0.68	0.60	0.77	0.71	0.64	0.80	0.70	0.93
8	Remove VAM branch	0.91	0.90	0.86	0.84	0.88	0.78	0.76	0.52	0.43	0.69	0.63	0.51	0.77	0.64	0.96

In Scenario 1, instead of merging the outputs from the SAM and VAM branches, using a dense layer, the outputs are merged with a modality self-attention layer[2]. Late fusion with an attention layer has been employed in previous works [6,9,33,49]. However, as shown in Scenario 1 of Table 4, no improvement is observed in TrajFusionNet when merging the two branches with a modality self-attention layer.

In Scenario 2, the number of VAN networks forming the VAM branch is reduced from two to one. The single VAN is applied to the last frame available in the observation period, $\mathbf{I}^t$. The pedestrian bounding box overlays for the observed trajectory ($\mathbf{M}_{1:4}^{t-14:t}$) and the predicted trajectory ($\hat{\mathbf{M}}_{1:4}^{t+1:t+61}$) are combined into the same image. Using a single VAN network instead of two does not affect the results on the PIE dataset but leads to a decrease in several metrics on the $JAAD_{all}$ and $JAAD_{beh}$ datasets.

In Scenario 3 of Table 4, sequence type identifiers are removed before sending the past trajectory tensor ($\mathbf{M}^{t-14:t}$) and the predicted trajectory tensor ($\hat{\mathbf{M}}^{t+1:t+61}$) to the encoder transformer. A slight decrease in performance across metrics and datasets is observed. In Scenario 4, vehicle speed is removed from the input modalities in the SAM branch (the trajectory prediction transformer only predicts future bounding boxes). Removing the vehicle speed leads to a noticeable decrease in performance across metrics on the PIE and $JAAD_{beh}$ datasets, but has a smaller impact on $JAAD_{all}$.

In Scenarios 5 and 6, trajectory prediction is excluded from the SAM and VAM branches, respectively. In Scenario 5, the predicted trajectory tensor ($\hat{\mathbf{M}}^{t+1:t+61}$), which includes predicted pedestrian coordinates and vehicle speed, is removed from the input to the encoder transformer. As shown in Table 4, this leads to a significant decrease in metrics on both the PIE and $JAAD_{beh}$ datasets, and a smaller decrease on the $JAAD_{all}$ dataset. In Scenario 6, predicted pedestrian boxes are not added as overlays on the scene images provided to the second VAN block (i.e., the predicted trajectory tensor $\hat{\mathbf{M}}_{1:4}^{t+1:t+61}$ is not used in the VAM branch). While the results on the PIE dataset remain unaffected, there is a significant decrease in performance on the $JAAD_{all}$ and $JAAD_{beh}$ datasets. Overall, the ablations in scenarios 5 and 6 highlight the importance of leveraging predicted pedestrian trajectories and vehicle speed as priors for predicting crossing intention in TrajFusionNet.

[2]The output from the SAM branch, χ_{traj}, is first concatenated with the output of the VAM branch, χ_{visual}, to form $\chi_{combined} \in \mathbb{R}^{d_{\text{mod}} \times d_{\text{enc}}}$, with $d_{mod} = 2$ and $d_{enc} = 40$. Attention is then computed along the d_{mod} dimension using scaled dot product attention [30]. The computed attention context is concatenated with $\chi_{combined}$, and the result is then passed to the final dense layer.

We conclude with Scenarios 7 and 8, which evaluate the contribution of each branch by removing one branch at a time. In Scenario 7, the SAM branch is removed, resulting in a substantial decline in performance across all datasets, with the most pronounced degradation observed on the PIE dataset, where the model performs particularly poorly. In Scenario 8, the VAM branch is removed, leading to only a modest decrease on the PIE dataset but a substantial reduction in performance on the $JAAD_{all}$ and $JAAD_{beh}$ datasets. Together, these ablations demonstrate the importance of TrajFusionNet's dual-branch architecture.

4.6 Qualitative Results

In this section, we analyze examples of correct and incorrect predictions by TrajFusionNet across the three datasets (Fig. 4). The left column of the figure presents cases where TrajFusionNet made correct predictions, while the right column shows examples of incorrect predictions. For the PIE dataset, the incorrect example (first row, second column) involves a pedestrian positioned near but not directly in front of a zebra crossing. This seems to mislead the model into inferring that the pedestrian does not intend to cross yet. Additionally, the pedestrian's trajectory during the observation period is relatively static. The pedestrian quickly decides to jaywalk (bypassing the zebra crossing) after the vehicle stops at the crossing to allow the other pedestrians to cross. The model incorrectly predicts "no cross".

Figure 4: Qualitative results showing examples of correct predictions by TrajFusionNet (**left column**) and incorrect predictions (**right column**).

For the $JAAD_{all}$ dataset, the incorrect example (second row, second column) depicts a vehicle turning at an intersection. Due to the turn, the target pedestrian's trajectory appears to move horizontally over time, resembling the trajectory of a crossing pedestrian. However, the pedestrian is merely walking along the roadside. TrajFusionNet incorrectly predicts "cross".

For the $JAAD_{beh}$ dataset, the incorrect example (third row, second column) shows a pedestrian exiting a store and turning perpendicularly to cross the road, which is an untypical trajectory. Low illumination around the pedestrian further complicates the VAM branch's ability to perceive the scene. TrajFusionNet predicts "no cross", but the pedestrian ultimately crosses the road.

5 Conclusion

This work introduced TrajFusionNet, a novel model for predicting pedestrian crossing intention. The model leverages future pedestrian trajectory and vehicle speed predictions as priors for predicting crossing intention. The Sequence Attention Module (SAM) processes a sequential representation of past and future trajectories, while the Visual Attention Module (VAM) utilizes a visual representation of the pedestrian trajectories by overlaying observed and predicted bounding boxes onto scene images. TrajFusionNet achieves state-of-the-art performance on the JAAD and PIE datasets. Moreover, by employing a small number of lightweight modalities, TrajFusionNet achieves the lowest total inference time (including model runtime and data preprocessing) among state-of-the-art approaches.

Although TrajFusionNet demonstrates strong performance on the JAAD and PIE datasets, further evaluation on larger and more diverse datasets containing additional geographic regions, weather conditions, traffic cultures, and sensor configurations would further validate its generalization capabilities. Future work could also investigate further reducing inference time through techniques such as knowledge distillation and adaptive computation. Additionally, incorporating explainability and uncertainty estimation into the model could be a promising direction for future research.

Acknowledgement: Not applicable.

Funding Statement: This research was enabled in part by support provided by the Natural Sciences and Engineering Research Council of Canada (NSERC), funding reference number RGPIN-2024-05287.

Author Contributions: Conceptualization, François G. Landry and Moulay A. Akhloufi; Methodology, François G. Landry; Formal analysis and investigation, François G. Landry; Writing—original draft preparation, François G. Landry; Writing—review and editing, François G. Landry and Moulay A. Akhloufi; Funding acquisition, Moulay A. Akhloufi; Resources: Moulay A. Akhloufi; Supervision: Moulay A. Akhloufi. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The datasets used in this study were collected and made publicly available by their respective original authors. The JAAD dataset is available at https://doi.org/10.1109/ICCVW.2017.33, and the PIE dataset is available at https://doi.org/10.1109/ICCV.2019.00636.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A Evaluation of Trajectory Prediction Error

Table A1 provides the trajectory prediction errors obtained by the trajectory prediction transformer used in TrajFusionNet's SAM branch. We report the average displacement error (ADE) and final displacement error (FDE), based on the center coordinates of pedestrian bounding boxes, as well as the mean absolute

error (MAE) of the predicted speed. It should be noted that speed is provided as continuous values in the PIE dataset, whereas the JAAD dataset provides categorical speed labels, which were ordinally encoded. Results are reported for prediction horizons of 1 and 2 s.

Table A1: Trajectory prediction errors of the trajectory prediction transformer used in TrajFusionNet's SAM branch on the PIE and JAAD test sets. Metrics include the average displacement error (ADE), final displacement error (FDE), and mean absolute error (MAE) of the predicted speed.

Model	Prediction Horizon	PIE			$JAAD_{all}$			$JAAD_{beh}$		
		Position Error		Speed Error	Position Error		Speed Error	Position Error		Speed Error
		ADE	FDE	MAE	ADE	FDE	MAE	ADE	FDE	MAE
TrajFusionNet-Small	1 s	15.37	27.99	0.64	27.21	49.32	0.49	43.05	78.55	0.46
	2 s	33.63	80.29	1.06	59.13	139.01	0.70	96.35	230.45	0.66
TrajFusionNet	1 s	14.76	27.27	0.67	24.59	45.82	0.48	40.24	72.86	0.47
	2 s	32.69	78.36	1.09	54.64	129.53	0.69	89.21	215.04	0.66

Appendix B Training/Validation Loss and Evaluation Metric Curves

Fig. A1 provides the training and validation loss curves for the final training stage of TrajFusionNet, during which the complete model is trained while the weights of the SAM and VAM branches remain frozen. The right side of Fig. A1 also shows the validation metrics (accuracy and AUC) for each epoch. For evaluation, we select the checkpoint with the highest validation AUC.

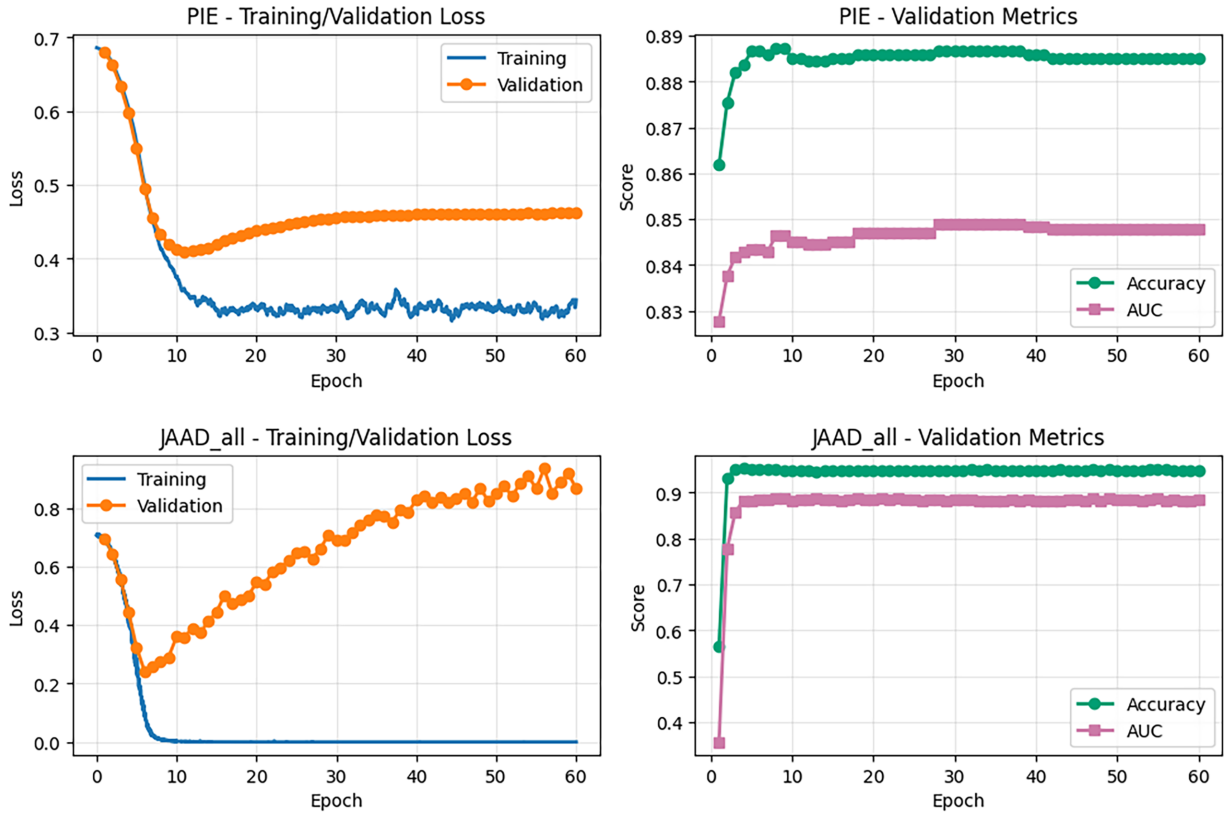

Figure A1: (Continued)

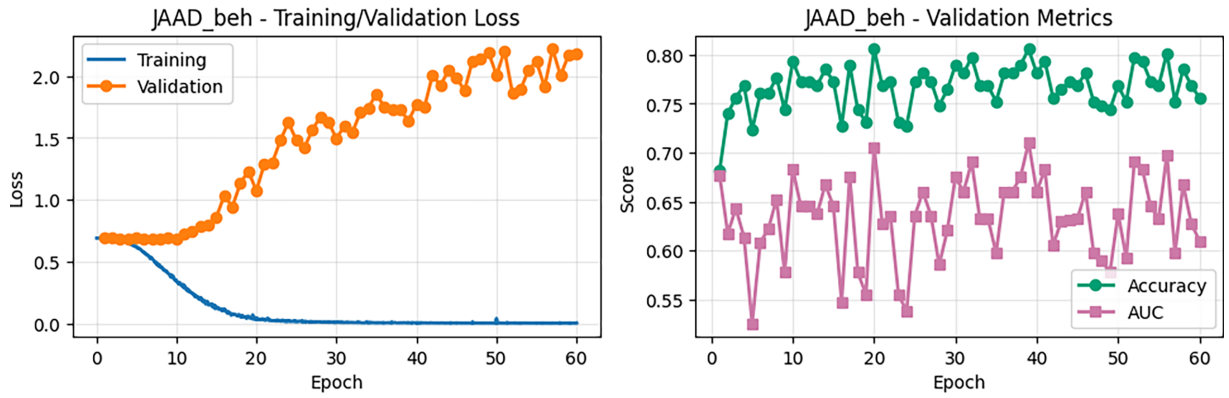

Figure A1: Training/validation loss curves and validation evaluation metrics during the final stage of modular training in TrajFusionNet.

References

1. Dollar P, Wojek C, Schiele B, Perona P. Pedestrian detection: an evaluation of the state of the art. IEEE Trans Pattern Anal Mach Intell. 2011;34(4):743–61.
2. Gandhi T, Trivedi MM. Pedestrian protection systems: issues, survey, and challenges. IEEE Trans Intell Transp Syst. 2007;8(3):413–30.
3. Galvão LG, Huda MN. Pedestrian and vehicle behaviour prediction in autonomous vehicle system—a review. Expert Syst Appl. 2023;238(3):121983. doi:10.1016/j.eswa.2023.121983.
4. Liu B, Adeli E, Cao Z, Lee KH, Shenoi A, Gaidon A, et al. Spatiotemporal relationship reasoning for pedestrian intent prediction. IEEE Robot Autom Lett. 2020;5(2):3485–92. doi:10.1109/lra.2020.2976305.
5. Sui Z, Zhou Y, Zhao X, Chen A, Ni Y. Joint intention and trajectory prediction based on transformer. In: Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2021 Sep 27–Oct 1; Prague, Czech Republic. New York, NY, USA: IEEE; 2021. p. 7082–8.
6. Yang D, Zhang H, Yurtsever E, Redmill KA, Özgüner Ü. Predicting pedestrian crossing intention with feature fusion and spatio-temporal attention. IEEE Trans Intell Veh. 2022;7(2):221–30. doi:10.1109/tiv.2022.3162719.
7. Song X, Kang M, Zhou S, Wang J, Mao Y, Zheng N. Pedestrian intention prediction based on traffic-aware scene graph model. In: Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2022 Oct 23–27; Kyoto, Japan. New York, NY, USA: IEEE; 2022. p. 9851–8.
8. Cadena PRG, Qian Y, Wang C, Yang M. Pedestrian graph+: a fast pedestrian crossing prediction model based on graph convolutional networks. IEEE Trans Intell Transp Syst. 2022;23(11):21050–61.
9. Gesnouin J, Pechberti S, Stanciulcscu B, Moutarde F. TrouSPI-Net: spatio-temporal attention on parallel atrous convolutions and U-GRUs for skeletal pedestrian crossing prediction. In: Proceedings of the 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021); 2021 Dec 15–18; Jodhpur, India. New York, NY, USA: IEEE; 2021. p. 1–7.
10. Bai J, Fang X, Fang J, Xue J, Yuan C. Deep virtual-to-real distillation for pedestrian crossing prediction. In: Proceedings of the 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC); 2022 Oct 8–12; Macau, China. New York, NY, USA: IEEE; 2022. p. 1586–92.
11. Chen H, Sun J, Liu Y, Peng Z, Hu D. Causal confusion in pedestrian crossing intention prediction for autonomous vehicles: the role of ego-vehicle speed. IEEE Trans Intell Transp Syst. 2025;26(8):12224–37. doi:10.1109/tits.2025.3557125.
12. Ling Y, Ma Z. Pedestrian crossing intention prediction in the wild: a survey. CHAIN. 2024;1(4):263–79. doi:10.23919/chain.2024.000008.
13. Guo MH, Lu CZ, Liu ZN, Cheng MM, Hu SM. Visual attention network. Comput Vis Media. 2023;9(4):733–52. doi:10.1007/s41095-023-0364-2.

14. Rasouli A, Kotseruba I, Tsotsos JK. Are they going to cross? A benchmark dataset and baseline for pedestrian crosswalk behavior. In: Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops (ICCVW); 2017 Oct 22–29; Venice, Italy. p. 206–13.
15. Rasouli A, Kotseruba I, Kunic T, Tsotsos JK. PIE: a large-scale dataset and models for pedestrian intention estimation and trajectory prediction. In: Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV); 2019 Oct 27–Nov 2; Seoul, Republic of Korea. p. 6261–70.
16. Landry FG, Akhloufi MA. Predicting pedestrian crossing intention in autonomous vehicles: a review. Neurocomputing. 2025;618(4):129105. doi:10.1016/j.neucom.2024.129105.
17. Zhang C, Berger C. Pedestrian behavior prediction using deep learning methods for urban scenarios: a review. IEEE Trans Intell Transp Syst. 2023;24(10):10279–301. doi:10.1109/tits.2023.3281393.
18. Bandyopadhyay T, Jie CZ, Hsu D, Ang MH, Rus D, Frazzoli E. Intention-aware pedestrian avoidance. In: Experimental Robotics: The 13th International Symposium on Experimental Robotics. Cham, Switzerland: Springer; 2013. p. 963–77.
19. Kooij JFP, Schneider N, Flohr F, Gavrila DM. Context-based pedestrian path prediction. In: Proceedings of the Computer Vision–ECCV 2014: 13th European Conference; 2014 Sep 6–12; Zurich, Switzerland. Cham, Switzerland: Springer; 2014. p. 618–33.
20. Köhler S, Schreiner B, Ronalter S, Doll K, Brunsmann U, Zindler K. Autonomous evasive maneuvers triggered by infrastructure-based detection of pedestrian intentions. In: Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IV); 2013 Jun 23–26; Gold Coast, Australia. New York, NY, USA: IEEE; 2013. p. 519–26.
21. Schneemann F, Heinemann P. Context-based detection of pedestrian crossing intention for autonomous driving in urban environments. In: Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2016 Oct 9–14; Daejeon, Republic of Korea. New York, NY, USA: IEEE; 2016. p. 2243–8.
22. Varytimidis D, Alonso-Fernandez F, Duran B, Englund C. Action and intention recognition of pedestrians in urban traffic. In: Proceedings of the 2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS); 2018 Nov 26–29; Las Palmas de Gran Canaria, Spain. New York, NY, USA: IEEE; 2018. p. 676–82.
23. Saleh K, Hossny M, Nahavandi S. Real-time intent prediction of pedestrians for autonomous ground vehicles via spatio-temporal densenet. In: Proceedings of the 2019 International Conference on Robotics and Automation (ICRA); 2019 May 20–24; Montreal, QC, Canada. New York, NY, USA: IEEE; 2019. p. 9704–10.
24. Li F, Fan S, Chen P, Li X. Pedestrian motion state estimation from 2D pose. In: Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV); 2020 Oct 19–23; Las Palmas de Gran Canaria, Spain. New York, NY, USA: IEEE; 2020. p. 1682–7.
25. Lorenzo J, Parra I, Wirth F, Stiller C, Llorca DF, Sotelo MA. RNN-based pedestrian crossing prediction using activity and pose-related features. In: Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV); 2020 Oct 19–23; Las Palmas de Gran Canaria, Spain. New York, NY, USA: IEEE; 2020. p. 1801–6.
26. Kotseruba I, Rasouli A, Tsotsos JK. Do they want to cross? Understanding pedestrian intention for behavior prediction. In: Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV); 2020 Oct 19–23; Las Palmas de Gran Canaria, Spain. New York, NY, USA: IEEE; 2020. p. 1688–93.
27. Ranga A, Giruzzi F, Bhanushali J, Wirbel E, Pérez P, Vu TH, et al. VRUNet: multi-task learning model for intent prediction of vulnerable road users. Electron Imaging. 2020;32(16):109-1–10
28. Rasouli A, Kotseruba I, Tsotsos JK. Pedestrian action anticipation using contextual feature fusion in stacked RNNs. arXiv:2005.06582. 2020.
29. Du Q, Xu L, Wu Q, Ning H, Wang X, Lin L, et al. SafeCrossNet: multi-modal fusion with social-aware for pedestrian crossing intention prediction. Inf Fusion. 2025:103609.
30. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Adv Neural Inf Process Syst. 2017;30:6000–10. doi:10.65215/ctdc8e75.
31. Achaji L, Moreau J, Fouqueray T, Aioun F, Charpillet F. Is attention to bounding boxes all you need for pedestrian action prediction?. In: Proceedings of the 2022 IEEE Intelligent Vehicles Symposium (IV); 2022 Jun 5–9; Aachen, Germany. New York, NY, USA: IEEE; 2022. p. 895–902.

32. Zhou Y, Tan G, Zhong R, Li Y, Gou C. PIT: progressive interaction transformer for pedestrian crossing intention prediction. IEEE Trans Intell Transp Syst. 2023;24(12):14213–25.
33. Lorenzo J, Alonso IP, Izquierdo R, Ballardini AL, Saz Á.H, Llorca DF, et al. CAPformer: pedestrian crossing action prediction using transformer. Sensors. 2021;21(17):5694.
34. Zhao S, Li H, Ke Q, Liu L, Zhang R. Action-ViT: pedestrian intent prediction in traffic scenes. IEEE Signal Process Lett. 2022;29:324–8. doi:10.1109/lsp.2021.3134194.
35. Elgazwy A, Elgazzar K, Khamis A. Predicting pedestrian crossing intentions in adverse weather with self-attention models. IEEE Trans Intell Transp Syst. 2025;26(3):3250–61. doi:10.1109/tits.2024.3524117.
36. Bai J, Fang J, Lv Y, Lv C, Xue J, Li Z. Gating syn-to-real knowledge for pedestrian crossing prediction in safe driving. IEEE Trans Intell Transp Syst. 2025;26(6):7509–22. doi:10.1109/tits.2025.3554767.
37. Gujjar P, Vaughan R. Classifying pedestrian actions in advance using predicted video of urban driving scenes. In: Proceedings of the 2019 International Conference on Robotics and Automation (ICRA); 2019 May 20–24; Montreal, QC, Canada. New York, NY, USA: IEEE; 2019. p. 2097–103.
38. Chaabane M, Trabelsi A, Blanchard N, Beveridge R. Looking ahead: anticipating pedestrians crossing with future frames prediction. In: Proceedings of the 2020 IEEE Winter Conference on Applications of Computer Vision (WACV); 2020 Mar 1–5; Snowmass Village, CO, USA. p. 2297–306.
39. Yang B, Wei Z, Hu H, Wang R, Yang C, Ni R. DPCIAN: a novel dual-channel pedestrian crossing intention anticipation network. IEEE Trans Intell Transp Syst. 2024;25(6):6023–34.
40. Ni R, Yang B, Wei Z, Hu H, Yang C. Pedestrians crossing intention anticipation based on dual-channel action recognition and hierarchical environmental context. IET Intell Transp Syst. 2023;17(2):255–69. doi:10.1049/itr2.12253.
41. Ling Y, Ma Z, Zhang Q, Xie B, Weng X. PedAST-GCN: fast pedestrian crossing intention prediction using spatial–temporal attention graph convolution networks. IEEE Trans Intell Transp Syst. 2024;25(10):13277–90. doi:10.1109/tits.2024.3398252.
42. Sang H, Chen W, Zhao Z. NaCGCN: node-augmented complementary graph convolution networks for pedestrian trajectory prediction. Expert Syst Appl. 2026;323(1):132473. doi:10.1016/j.eswa.2026.132473.
43. Chen W, Sang H, Wang J, Zhao Z. IGGCN: individual-guided graph convolution network for pedestrian trajectory prediction. Digit Signal Process. 2025;156:104862.
44. Schörkhuber D, Pröll M, Gelautz M. Feature selection and multi-task learning for pedestrian crossing prediction. In: Proceedings of the 2022 16th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS); 2022 Oct 19–21; Dijon, France. New York, NY, USA: IEEE; 2022. p. 439–44.
45. Cao D, Fu Y. Using graph convolutional networks skeleton-based pedestrian intention estimation models for trajectory prediction. J Phys Conf Ser. 2020;1621(1):012047. doi:10.1088/1742-6596/1621/1/012047.
46. Pop DO, Rogozan A, Chatelain C, Nashashibi F, Bensrhair A. Multi-task deep learning for pedestrian detection, action recognition and time to cross prediction. IEEE Access. 2019;7:149318–27. doi:10.1109/access.2019.2944792.
47. Yao Y, Atkins E, Roberson MJ, Vasudevan R, Du X. Coupling intent and action for pedestrian crossing behavior prediction. arXiv:2105.04133. 2021.
48. Zhai X, Hu Z, Yang D, Zhou L, Liu J. Social aware multi-modal pedestrian crossing behavior prediction. In: Proceedings of the Asian Conference on Computer Vision; 2022 Dec 4–8; Macau, China. p. 4428–43.
49. Kotseruba I, Rasouli A, Tsotsos JK. Benchmark for evaluating pedestrian action prediction. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision; 2021 Jan 5–9; Waikoloa, HI, USA. p. 1258–68.
50. Wang Y, Wu H, Dong J, Liu Y, Long M, Wang J. Deep time series models: a comprehensive survey and benchmark. arXiv:2407.13278. 2024.
51. Kim J, Nguyen D, Min S, Cho S, Lee M, Lee H, et al. Pure transformers are powerful graph learners. Adv Neural Inf Process Syst. 2022;35:14582–95. doi:10.52202/068431-1060.
52. Bengio Y, Lamblin P, Popovici D, Larochelle H. Greedy layer-wise training of deep networks. Adv Neural Inf Process Syst. 2006;19:1–17.
53. Rusu AA, Rabinowitz NC, Desjardins G, Soyer H, Kirkpatrick J, Kavukcuoglu K, et al. Progressive neural networks. arXiv:1606.04671. 2016.

54. Cao Z, Simon T, Wei SE, Sheikh Y. Realtime multi-person 2D pose estimation using part affinity fields. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017 Jul 21–26; Honolulu, HI, USA. p. 7291–9.
55. Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL. Rethinking atrous convolution for semantic image segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017 Jul 21–26; Honolulu, HI, USA. p. 7268–77.