Computing and Informatics, Vol. 44, 2025, 202–222, doi: 10.31577/cai_2025_1_202

MINET: A PEDESTRIAN TRAJECTORY FORECASTING METHOD WITH MULTI-INFORMATION FEATURE FUSION

Tao Yuan, Xiaohong Han*

Taiyuan University of Technology
College of Computer Science and Technology (College of Data Science)
Taiyuan 030000, China
e-mail: runnableip@foxmail.com, hanxiaohong@tyut.edu.cn

Abstract. Pedestrian trajectory prediction plays an exceptionally vital role in autonomous driving, enabling advanced analysis and decision-making in certain scenarios to ensure driving safety. Predicting pedestrian trajectories is a highly complex task, encompassing static scenes, dynamic scenes, and subjective intent. To enhance the accuracy of pedestrian trajectory prediction, it is crucial to model these scenarios, extract relevant features, and fuse them effectively. However, existing methods only consider some of the scenarios mentioned above and extract static scene features through manual annotation of road key points, which fails to meet the demands of autonomous driving in complex traffic scenarios. To overcome these limitations, this paper introduces MINet – a network that employs multi-information feature fusion. Unlike previous approaches, MINet adopts a more automated approach to extract static scenes, including sidewalks and lawns. Moreover, the network incorporates pedestrian destination modeling to improve prediction accuracy. Furthermore, to tackle the challenge of collision avoidance in crowded spaces, this paper incorporates the extraction of dynamic scene changes through relative velocity modeling of objects. The proposed network achieved an improvement of 47.7 % in the ADE metric and 62.6 % in the FDE metric on the ETH/UCY dataset. In the SDD dataset, there was an improvement of 18.4 % in the ADE metric and 35.2 % in the FDE metric.

Keywords: Trajectory forecasting, scene feature encoding, interaction feature encoding, destination encoding, walkable area

1 INTRODUCTION

Pedestrian trajectory prediction plays a crucial role in diverse fields, including autonomous driving, intelligent transportation, video surveillance, and human-computer interaction [1, 2, 3, 4]. It enables autonomous vehicles to analyze current and historical scene information, proactively react to road conditions, and ensure the safety of passengers and other road users [5, 6, 7, 8]. Moreover, by accurately predicting pedestrian trajectories, monitoring devices can effectively detect abnormal behavior, leading to timely identification and handling of security incidents. Therefore, conducting thorough research in the area of pedestrian trajectory prediction is highly necessary to advance these fields and enhance overall safety measures.

However, pedestrian trajectory prediction is a complex process influenced by many factors, mainly including the following:

1. Destination: Pedestrians usually choose an optimal path to reach their destination. Although they may change their trajectory based on the state of other objects in the scene, such as evading obstacles, the overall path generally extends towards the destination, as shown in Figure 1 a) [9, 10].
2. Static Scene: A static scene typically refers to sidewalks, intersections, lawns, rivers, and other static obstacles [8, 11]. When pedestrians encounter static obstacles, they usually "detour" around them instead of crossing them (such as traffic barriers, lawns, and rivers), as shown in Figure 1 b).
3. Dynamic Scene: A dynamic scene includes other objects such as pedestrians, bicycles, and vehicles. In crowded spaces, people are easily influenced by the movements of surrounding pedestrians. To avoid collisions with others, pedestrians will change their movement direction [5, 6, 12]. As shown in Figure 1 c), when a target pedestrian is walking towards other objects, they will adjust their speed, trajectory, and direction to avoid colliding with surrounding pedestrians.
4. Subjective Awareness: Pedestrians' subjective awareness and emotional state can affect their movement trajectory. For example, when two pedestrians are performing evasive actions, the decision of who takes the evasive action is influenced by subjective awareness and emotional state. Another example is when a pedestrian crosses the street, they may choose to cross at the current intersection or wait until the next intersection, which is strongly influenced by their subjective awareness, as shown in Figure 1 d).

Therefore, to accurately predict pedestrian trajectories, it is essential to consider and comprehensively analyze multiple factors. Currently, pedestrian trajectory prediction algorithms commonly rely on rule-based traditional machine learning [13, 14, 15], or deep learning methods [5, 6, 11, 16], but they only take into account a subset of the aforementioned factors. As a result, their accuracy and generalization capability are inadequate.

This paper proposes a new pedestrian trajectory prediction method based on a multi-information feature fusion network (MINet). The main contribution of this

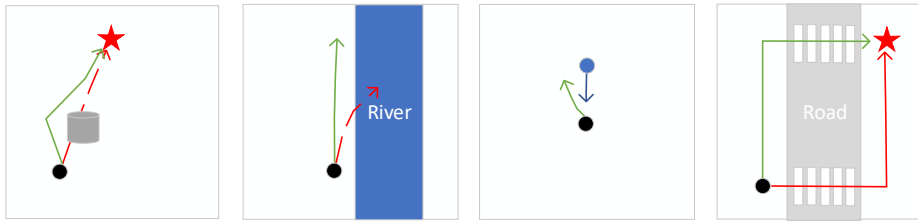

Figure 1. Some examples of pedestrian walking

study includes:

1. To address the limitation of relying on manual labeling of road key points to describe static scenes, this paper utilizes optical flow to track the rough trajectories of pedestrians and combines them with real trajectories to extract features of static scenes.
2. The proposed method employs pedestrian destinations to constrain the model to generate trajectories that are more in line with reality, in which Variational Autoencoder (VAE) is used to predict destinations of pedestrian in short-term.
3. The proposed method employs an interactive influence factor graph to extract interaction information among people. This can effectively solve the problem of avoiding pedestrians when they approach each other.

By incorporating these enhancements, the proposed MINet method significantly improves the accuracy and generalization capability of pedestrian trajectory prediction.

2 RELATED WORK

Pedestrian trajectory prediction plays a crucial role in the fields of autonomous driving and intelligent robotics [17, 18, 19], offering a wide range of applications. Currently, two primary methods are employed for pedestrian trajectory prediction: traditional knowledge-based modeling approaches [13, 14, 15, 20] and data-driven deep learning methods [5, 6, 7, 21, 22, 23].

Traditional knowledge-based modeling methods involve the manual design of rules and functions to capture the mutual interactions between pedestrians. These methods typically consider various data such as velocity, acceleration, direction, and distance, often utilizing models like attraction-repulsion models and velocity-obstacle models. While these methods demonstrate effectiveness in predicting pedestrian trajectories in simple scenarios, their performance tends to be unsatisfactory in complex traffic environments. On the other hand, data-driven deep learning methods leverage abundant positional information datasets to analyze the factors influencing

pedestrian walking trajectories. By modeling these factors, these methods achieve more accurate predictions of pedestrian trajectories.

Data-driven deep learning methods encompass a range of models, including:

Pedestrian Interaction Model. In crowded environments, when two pedestrians come close to each other, it is common for one of them to adjust their trajectory and avoid the other by changing direction. This observed phenomenon has inspired researchers to incorporate it into pedestrian trajectory prediction tasks. For instance, Social LSTM [5] introduces a social pooling module that gathers information about the neighbouring states of the target pedestrian, enabling the capture of subtle movements between individuals. In [24], a novel pedestrian encoding method called APG is introduced, which utilizes one-dimensional grids in polar space to effectively capture the interaction between pedestrians. SS-LSTM [21], on the other hand, proposes a circular shape neighborhood as an alternative to the traditional rectangular neighborhood used in social scale models. Additionally, Social Attention [25] incorporates an attention mechanism to accurately assess the relative importance of each neighbouring individual around the target pedestrian. While the aforementioned methods focus on simple interactions between pedestrians, real-life scenarios often involve more complex interactions that encompass not only pedestrians but also objects such as cars and bicycles.

Person-Scene Model. Scenes play a crucial role in shaping pedestrian walking trajectories, as individuals dynamically adjust their paths based on the surrounding environment. In the context of trajectory prediction, Scene-LSTM [8] partitions static scenes into Manhattan grids and utilizes LSTM to forecast pedestrian locations. CAR-Net [26] proposes an attention network that leverages scene semantic CNN to predict human trajectories. Study [24] introduces a binary two-dimensional occupancy grid, where static obstacles are represented by 1 and walkable areas by 0, effectively describing the scene's layout. On the other hand, MI-LSTM [27] is a network tailored for predicting cyclists' trajectories, employing manually labeled road key points to delineate road scene boundaries. By incorporating road boundaries, the predicted trajectory for cyclists can be confined to the correct road. Notably, existing methods rely on manual labeling to capture scene information, which may lack robustness. In this study, road features are automatically extracted, enabling a more comprehensive depiction of scene influence on pedestrian movement.

Pedestrian Intention Modeling. Pedestrians typically determine their walking trajectory based on their intended destination and the current scene conditions. In the field of pedestrian trajectory prediction, researchers have incorporated pedestrian intention into their models. For instance, in the study [28] the researchers aim to infer pedestrian intentions by predicting the direction of the pedestrian's head. Another approach, known as the Next model [11], treats the prediction of pedestrian's future behavior as an auxiliary task, aiding the model in inferring more realistic trajectories. However, given the countless variations

in human behaviors, accurately classifying all of them presents a significant challenge. In the study [29] a discrete choice model is introduced to calculate the pedestrian's next decision, further enhancing intention prediction. Another approach, [30] assesses the pedestrian's intention based on their current posture, leveraging stereo-based 3D deep pose estimation. Furthermore, [9] predicts the short-term endpoint of the pedestrian using a latent encoder and utilizes this predicted endpoint as a constraint for generating trajectories. These diverse approaches highlight the efforts to incorporate pedestrian intention into trajectory prediction models, aiming to enhance the accuracy and realism of the predicted trajectories.

However, Existing methods for pedestrian trajectory prediction only consider a subset of the factors that influence pedestrian trajectories, leading to incomplete models and unsatisfactory prediction outcomes. To address this limitation, this study aims to enhance the prediction accuracy by incorporating a comprehensive set of factors. Specifically, the study will integrate both static and dynamic scene information, along with destination features, into the pedestrian trajectory prediction framework. By considering a wider range of factors, this research endeavors to improve the overall prediction performance and provide more reliable trajectory estimations.

3 PROPOSED APPROACH

The current approaches in predicting pedestrian trajectories are limited in considering only a few factors, resulting in unsatisfactory prediction results. Therefore, there is a need for improved methods to enhance the accuracy of pedestrian trajectory predictions. This paper introduces a novel approach that integrates multiple influential factors for accurate prediction of pedestrian walking paths. The overall framework of the proposed method is depicted in Figure 2. It focuses on extracting relevant features from four dimensions: road scene S, pedestrian interaction I, historical trajectory T, and destination D. By combining these features, a multi-channel tensor is constructed. The road scene dimension encompasses comprehensive three-dimensional information, encompassing static obstacles and road boundaries. The pedestrian interaction dimension captures the mutual influence among pedestrians. The historical trajectory dimension incorporates pertinent characteristics from past pedestrian walking paths. Lastly, the destination dimension reflects the intentions of the pedestrians.

The proposed network architecture comprises four main components: the scene feature encoding module, the interaction feature encoding module, the destination encoding module, and the historical trajectory encoding module.

The scene feature encoding module employs optical flow [31] to track pedestrians and derive a preliminary trajectory T_R. It combines this trajectory with the original pedestrian trajectory data $T_{GroundTruth}$ from the dataset to generate a "walkable area" map, from which it extracts rich road information features S. This approach

surpasses previous manual methods [24, 27] in terms of robustness and flexibility, allowing for rapid adaptation to diverse scenes.

The primary objective of the interaction feature encoding module is to extract the interaction features I among pedestrians. This module employs a grid-based approach to represent the neighbouring individuals around the target pedestrian, effectively modeling them through directed pooling [32].

The goal of the destination encoding module is to extract the intended destinations D of pedestrians. To achieve this, the module utilizes a VAE to sample potential destinations [9]. The VAE parameters are trained using actual destination data, enabling the module to generate accurate destinations during prediction to guide trajectory generation. It is worth noting that the destinations generated by the destination encoding module pertain to the short-term goals of pedestrians rather than their long-term destinations.

The historical trajectory encoding module utilizes a multi-layer perceptron to extract informative features T from pedestrian historical trajectories.

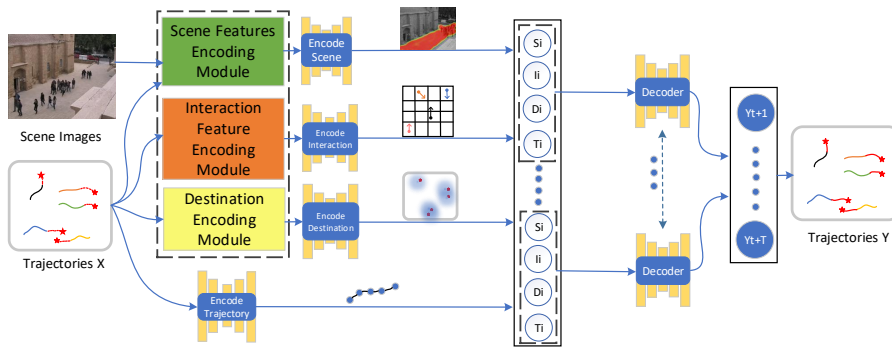

Figure 2. Overall model architecture

3.1 Scene Encoder

Figure 3 illustrates the scene feature encoding module developed in this paper, which is responsible for extracting a feature S representing the "walkable area". It is well known that different static scenes, such as sidewalks, lawns, motor lanes, and static obstacles, exert varying influences on pedestrian trajectories. For instance, pedestrians typically prefer walking on sidewalks rather than on lawns, motor lanes, or static obstacles. In this context, the sidewalk represents the "walkable area", while the lawn, motor lane, or static obstacle represents "non-walkable areas". By incorporating the feature S of the "walkable area", this module significantly improves the model's prediction performance by constraining the predicted region.

The initial step involves acquiring the corner points $C_{pedestrian}$ of pedestrians and employing optical flow to track them, resulting in a trajectory optical flow map F.

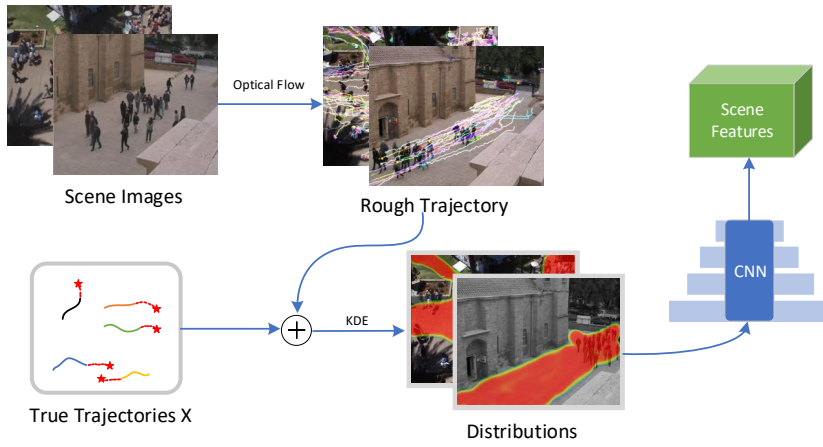

Figure 3. Scene feature encoding module

The process begins by gathering scene videos captured over a specific duration, assuming the existence of N frames in total. For each frame N_i, where $i \in [1, N]$, corner detection [33, 34] is performed, yielding a set of detected corners denoted as $C^i = C_1^i, C_2^i, \ldots, C_K^i$. Here, C_j^i represents the j^{th} corner detected in the i^{th} frame image.

The majority of points in C^i correspond to pedestrians' heads or feet, although there are also corner points unrelated to pedestrians. In Figure 4 a), the red dots represent the detected corner points. These points are not only found on pedestrians' heads, hands, and feet but also on buildings and fences, referred to as C_{Static}^i. The presence of C_{Static}^i can impact the determination of the final walkable area. As shown in Figure 4 c), the three elliptical areas in the upper left corner are non-walkable areas, necessitating the removal of C_{Static}^i. To distinguish between pedestrian corners $C_{Pedestrian}^i$ and static corners C_{Static}^i, we utilize the Euclidean distance d as a criterion. If the distance d between two corners falls below a specific threshold, we classify the corner point as static and include it in the set of C_{Static}^i. Within frame i, pedestrian corners are represented as Equation (1):

$$C_{Pedestrian}^i = C^i - C_{Static}^i. \tag{1}$$

Subsequently, we employ the optical flow algorithm to track $C_{Pedestrian}^i$ and compute the displacement of pedestrian corners between frame i and frame $i+1$, resulting in the corner displacement matrix $V^i = V_1^i, V_2^i, \ldots, V_K^i$, where V_j^i denotes the trajectory displacement length of the j^{th} corner point from the i^{th} frame image. Utilizing V^i, we obtain a rough trajectory optical flow map denoted as F_R^i Finally, we combine all the F_R^i to generate the final rough trajectory optical flow map, F_R.

$$F_R = F_R^1 + F_R^2 + \cdots + F_R^N. \tag{2}$$

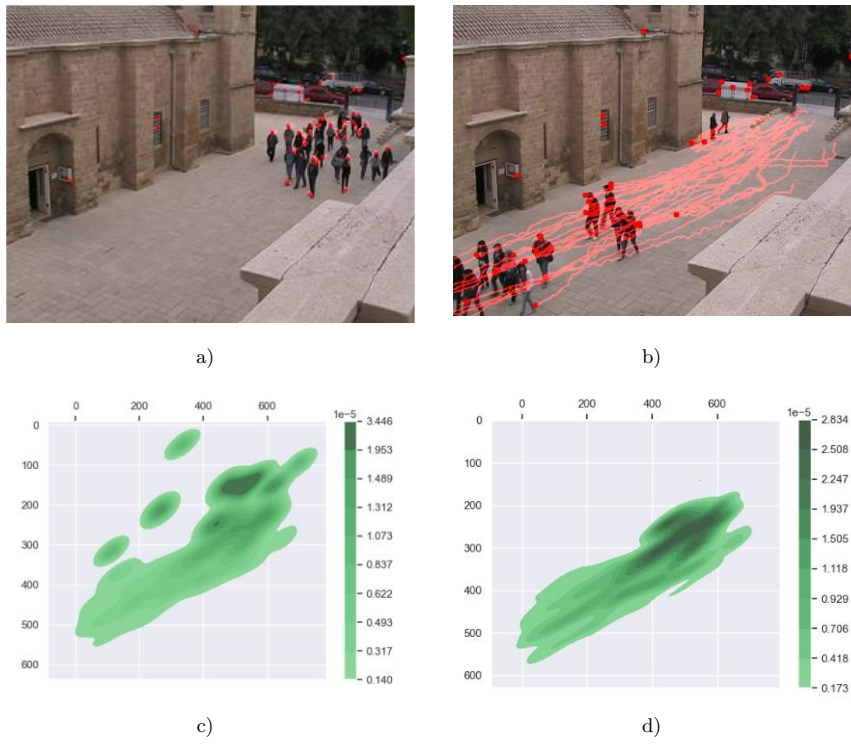

Figure 4. Visual analysis of Arxiepiskopi_flock.avi: corner detection, trajectory flow, and noise reduction in KDE

F_R illustrates the road outline, as depicted in Figure 4 b). To obtain an accurate trajectory optical flow map, $F_GroundTruth$, we rely on the true pedestrian trajectories annotated in the dataset. By merging F_R with $F_{GroundTruth}$, we generate a comprehensive trajectory optical flow map denoted as F.

$$F = F_R + F_{GroundTruth}. \tag{3}$$

The second step is to perform Kernel Density Estimation (KDE) [35] on the trajectories in F, in order to obtain a "pixel-level" probability distribution of the trajectories.

$$p(x,y) = \frac{1}{Rh^2}\sum_{i=1}^{R} max\left\{\frac{1}{h} - \frac{1}{h^2}\sqrt{\left(\frac{x-p_x^i}{h}\right)^2 + \left(\frac{y-p_y^i}{h}\right)^2}, 0\right\}. \tag{4}$$

Here, R is the number of trajectories in F, and h is the bandwidth. KDE provides a "pixel-level" probability distribution of the "walkable area" in the predicted

scene. This probability distribution can be used to generate the walkable area map, as shown in Figure 4 d), thereby constraining the region for pedestrian trajectory prediction. Finally, a CNN is used to obtain the scene representation S:

$$S = CNN(KDE(F)). \tag{5}$$

3.2 Interaction Encoder

The interaction feature encoding module focuses on capturing nuanced interactions among individuals through the use of an interaction influence factor map. This module employs directed pooling [32], utilizing relative velocity as a metric, to aggregate the hidden states of nearby crowds. The hidden state mimics the effects of factors such as the positions of neighbors, distances, relative velocities, and motion directions on pedestrian trajectories. The interaction influence factor map, which showcases these interactions, is depicted in Figure 5.

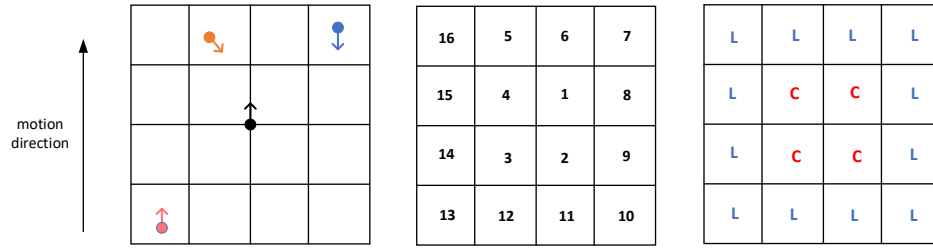

Figure 5. Graphical representation of close-Range and long-Range impact

The interaction influence factor map consists of a square grid with a side length of L, which determines the range of influence on the target pedestrian. It comprises 16 grids divided into two layers: inside and outside. L represents the influence over long distances, while C represents the influence over close ranges. At each time step t, all neighbouring individuals O_m possess a position vector P. This position vector, represented using one-hot encoding, has a size of 16×1, as illustrated in Figure 6. It indicates the position of O_m in the interaction influence factor map, reflecting their relative position to the target pedestrian. For instance, if neighbor O_2 is present in grid 2, then element 1 of the P vector is set to 1, while all other elements are set to 0.

The position vector P solely encompasses the relative positional details of the neighbors. However, this module requires a more comprehensive neighbor state vector, incorporating semantic information from both the inner and outer layers of the interaction influence factor map, along with velocity, direction, and distance. Previous studies have demonstrated that factors such as the direction, speed, and proximity of neighbors significantly influence the motion trajectory of the target pedestrian. Hence, it is essential to fuse these factors to generate a richer neighbor state vector.

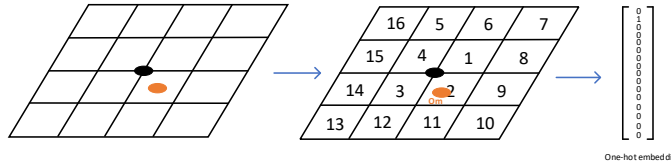

Figure 6. Target pedestrians and neighbors in grid:position encoding with one-Hot vectors

Neighbors can be broadly classified into two categories: "same-direction walking" and "opposite-direction walking". When the target pedestrian moves in the same direction as their neighbors, the influence of these neighbors on their trajectory diminishes, warranting lower weights to be assigned. Conversely, when the target pedestrian walks in the opposite direction to their neighbors, varying weights are set based on Θ, which represents the angle between the relative velocities of the neighbor and the target pedestrian, as well as the distance between them.

If Θ is small and falls within grid C, it implies a higher likelihood of the target pedestrian evading at that moment, thus necessitating a larger weight assignment. On the other hand, neighbors in grid L or those whose directional angles deviate from that of the target pedestrian exhibit reduced influence factors and are consequently assigned lower weights.

By incorporating the position vector P in this manner, we derive a hidden state vector I for each neighbor at time t, encompassing comprehensive information about their positions, directions, and more.

3.3 Destination Encoder

The destination encoding module plays a critical role in extracting destination features and utilizing them to enhance the precision of trajectory predictions. As depicted in Figure 7, the destination encoding module consists of two parts: the red segment represents the training stage. In this stage, the module is trained using the historical trajectory $T_i = T_i^1, T_i^2, \ldots, T_i^{obs}$ and the corresponding destination D_i of the pedestrian. Here, T_i represents the trajectory of pedestrian i from time $t = 0$ to $t = obs$. Subsequently, the trained VAE is employed to generate destinations for testing purposes.

During the training stage, we initiate the process by extracting the pedestrian's historical trajectory T_i and the associated destination D_i. Two MLP, denoted as M_t and M_d, resulting in the encoded representations $M_t(T_i)$ and $M_d(D_i)$. These encodings are then fused and input into the latent encoder, M_{latent}, producing $M_{latent}(M_t(T_i) + M_d(D_i))$. This fused representation is utilized to train the mean μ and variance σ of the VAE, generating the distribution $z = N(\mu, \sigma^2)$. The module samples potential destinations from the distribution $N(\mu, \sigma^2)$, which are subsequently concatenated with $M_t(T_i)$ and decoded using the decoder, D_{latent}, to obtain

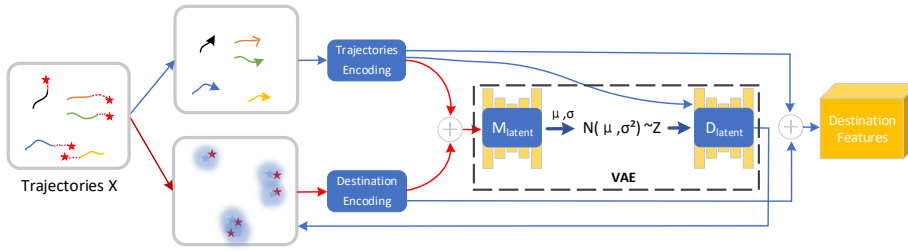

Figure 7. Destination encoding module

the predicted pedestrian destination, $\hat{D}_i$.

$$\hat{D}_i = D_{latent}(M_{latent}(M_t(T_i) + M_d(D_i)) + M_t(T_i)). \tag{6}$$

We utilize M_d to encode $\hat{D}_i$, the predicted destination, and connect it with $M_t(T_i)$ to generate the destination feature F_d.

$$F_d = M_t(T_i) + M_d(\hat{D}_i). \tag{7}$$

During the testing phase, as the model does not have access to the actual destinations of pedestrian trajectories, it directly samples destination samples from a normal distribution. Subsequently, similar to the training phase, the sampled destinations are concatenated with $M_t(T)$, and the trained decoder D_{latent} is employed to predict $\hat{D}_i$. Finally, the features $M_t(T_i)$ and $M_d(\hat{D}_i)$ are fused to form destination features, which are then fed into the main network.

4 EXPERIMENTS

4.1 Datasets

In this section, we conducted training and testing of our model using publicly available datasets, namely ETH/UCY [36, 37] and SDD [38]. These datasets encompass various challenging social behaviors, such as group walking, crowd crossing, and following, among others.

The ETH/UCY dataset is a collection of datasets from ETH [36] and UCY [37], consisting of five distinct scenarios: UNIV, Zara1, Zara2, ETH, and HOTEL. In total, this dataset comprises 1 536 human trajectories. Locations of pedestrians are marked in the real world, with meters as the unit of measurement. Video resolutions differ among the datasets: 720×576 for UCY and HOTEL, and 640×480 for ETH. For training and testing data separation, we employed a leave-one-out strategy [5, 6, 7, 39], utilizing data from four scenes for training and testing on the remaining one.

The SDD [38] serves as a recognized benchmark for evaluating the performance of human trajectory prediction. This dataset includes trajectory videos from 20 di-

verse scenarios, captured from a bird's-eye view using a drone. Besides pedestrians, it features cyclists, cars, skateboarders, buses, and other moving objects, totaling 11 200 pedestrians and 64 000 bicycles, with over 20 000 targets in total. This dataset enables effective observation and analysis of interactions between moving objects. To maintain consistency with prior works [6, 7, 9, 23, 40], the training and testing sets were divided accordingly.

4.2 Implementation Details

The network in this study was implemented using PyTorch and trained on a Linux server equipped with a Tesla K80 graphics card. For both training and testing, we set the batch size to 100, and each training run consisted of 650 epochs. The learning rate was set at 0.0003, and we used the ADAM optimizer.

In the Scene Feature Encoding module, we adopted the culling method described in Section 3.1 to remove static corner points. Table 1 provides the settings for the Euclidean distance threshold d used in different scenes within the ETH/UCY.

	ETH	HOTEL	UNIV	ZARA
d (meters)	0.36	0.15	0.32	0.4

Table 1. d for static corner point elimination in the ETH/UCY dataset

In our study, we first removed noise from the trajectory optical flow map. Subsequently, we applied KDE to obtain the "Walkable Area" map for each scenario, as illustrated by the green area in the right part of Figure 8. For the purpose of illustration, we utilized the ETH/UCY dataset, and the process is depicted in Figure 8.

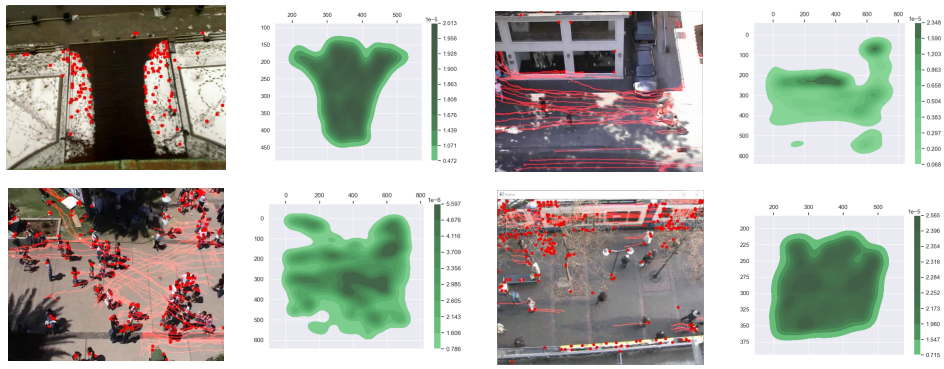

Figure 8. Scene visualization and walkable area maps

The process of extracting the walkable area from the raw video involves three primary steps: corner detection, optical flow tracking, and kernel density estimation.

To obtain the walkable area maps for the eight scenes in the SDD, we adopt the same approach. Furthermore, in Section 3.2, we set the side length L of the interaction influence factor map to 40 px.

For pedestrian trajectory prediction tasks, we aim to predict the next 12 time steps (4.8 s) of the trajectory based on observations from the previous 8 time steps (3.2 s), following the setup outlined in reference [5], which we also adhere to in this paper. In terms of the coefficients for the loss function, we set $\beta_1 = 1$ and $\beta_2 = 1$.

The network employs MLP with non-linear ReLU functions for both the encoders and decoders. Additionally, the scene feature encoding module utilizes a CNN to handle the walkable area. The structures of these components are detailed in Table 2.

Encoder/Decoder	Architecture
Scene CNN Encoder	$(1, 64, 3 \times 3) \rightarrow (64, 30, 3 \times 3) \rightarrow (30, 20, 3 \times 3) \rightarrow (20, 10, 3 \times 3)$
Scene MLP Encoder	$1024 \rightarrow 512 \rightarrow 256 \rightarrow 128 \rightarrow 64 \rightarrow 16$
Trajectory MLP Encoder	$16 \rightarrow 512 \rightarrow 256 \rightarrow 16$
Destination MLP Encoder	$2 \rightarrow 8 \rightarrow 16 \rightarrow 16$
Latent MLP Encoder	$32 \rightarrow 8 \rightarrow 50 \rightarrow 32$
Latent MLP Decoder	$32 \rightarrow 1\,024 \rightarrow 512 \rightarrow 1\,024 \rightarrow 2$
Interaction MLP Encoder	$8 \rightarrow 256 \rightarrow 128 \rightarrow 64 \rightarrow 16$
Predict MLP Encoder	$64 \rightarrow 1\,024 \rightarrow 512 \rightarrow 256 \rightarrow 22$

Table 2. Structure of the encoders and decoders mentioned in this article

Metrics: In our evaluation of the model's performance, we utilize two widely used metrics in pedestrian trajectory prediction tasks: the Average Displacement Error (ADE) and the Final Displacement Error (FDE) [5]. The ADE represents the average L_2 distance between the predicted trajectory and the actual trajectory, while the FDE is the L_2 distance between the final predicted point and the actual point. The formal expressions for these metrics are as follows:

$$ADE = \frac{\sum_i \sum_t \|(x_t^i, y_t^i) - (\hat{x_t^l}, \hat{y_t^l})\|_2}{N * T_{pred}}, \tag{8}$$

$$FDE = \frac{\sum_i \|(x_t^i, y_t^i) - (\hat{x_t^l}, \hat{y_t^l})\|_2}{N}. \tag{9}$$

We also employed the Best of N approach, as mentioned in [6], which involves selecting the minimum ADE and FDE from K randomly sampled trajectories. This method has become widely adopted in pedestrian trajectory prediction tasks.

Loss Function: For the end-to-end training of the entire network, we employ the following loss function:

$$Loss = \|\hat{T} - T\|^2 + \beta_1 \|\hat{D} - D\|_2^2 + \beta_2 D_{KL}(N(\mu, \sigma) \| N(0, I)). \tag{10}$$

In this equation, the first term represents the mean trajectory loss, which is used for training the entire network. The second term corresponds to the mean destination loss, utilized for training the destination encoder's Multilayer Perceptron (MLP) section. The third term, the KL divergence, is applied for training the variational autoencoder.

Baseline: We have selected 8 previously published baseline models for comparison with our MINet, and they are detailed below:

- *Social LSTM* [5]: Alahi et al. proposed an LSTM model that includes a social pooling layer, enabling the model to automatically learn interactions among pedestrians.
- *Conv2D* [41]: Zamboni et al. introduced a novel two-dimensional convolution model. This recurrent model showcases both high prediction accuracy and quick computation.
- *Social GAN* [6]: Gupta et al. combined sequence prediction with Generative Adversarial Networks, proposing a multimodal human trajectory prediction GAN. They trained a variety of losses to encourage diversity and utilized a new pooling mechanism to aggregate information between individuals.
- *SR-LSTM* [16]: Zhang et al. employed a data-driven state-refinement LSTM network that utilizes a message-passing mechanism to leverage the current intentions of neighbors.
- *NEXT* [11]: Liang et al. proposed an end-to-end multi-task learning system, which capitalizes on rich visual features about human behavior and interaction with the surrounding environment.
- *SoPhie* [7]: Sadeghian et al. combined scene context information with social interactions among agents to obey the environment's physical constraints.
- *SimAug* [42]: Liang et al. introduced a novel method that learns robust representations by augmenting simulated training data, enabling these representations to better generalize to unseen real-world test data.
- *DESIRE* [43]: Lee et al. proposed a trajectory planning method based on inverse optimal control. This approach utilizes a refinement structure to predict trajectories.

4.3 Comparison with Related Methods

In this section, we conduct a comprehensive comparison and discussion of our model with the aforementioned baseline networks using the ADE and FDE metrics.

In Table 3, we present the results of each model across various scenarios on the ETH/UCY dataset, following the leave-one-out evaluation methodology as described

Model	Eth	Hotel	Univ	Zara1	Zara2	AVG
Social LSTM	1.09/2.35	0.79/1.76	0.67/1.40	0.47/1.00	0.56/1.17	0.72/1.54
Conv2D	0.56/1.11	0.24/0.46	0.58/1.23	0.46/0.99	0.35/0.75	0.44/0.91
Social GAN	0.87/1.62	0.67/1.37	0.76/1.52	0.35/0.68	0.42/0.84	0.61/1.21
SR-LSTM	0.63/1.25	0.37/0.74	0.51/1.10	0.41/0.90	0.32/0.70	0.45/0.94
NEXT	0.73/1.65	0.30/0.95	0.60/1.27	0.38/0.81	0.31/0.68	0.46/1.00
SoPhie	0.70/1.43	0.76/1.67	0.54/1.24	0.30/0.63	0.38/0.78	0.54/1.15
Ours-D	0.58/0.96	0.19/0.34	0.39/0.67	0.23/0.39	0.24/0.35	0.32/0.54
Ours-D-S	0.49/0.72	0.15/0.20	0.18/0.28	0.21/0.32	**0.16/0.26**	0.24/0.36
Ours-D-S-I	**0.46/0.65**	**0.14/0.18**	**0.18/0.27**	**0.21/0.31**	**0.16**/0.28	**0.23/0.34**

Table 3. Comparison of baseline model and MINet on the ETH/UCY

in [5, 6, 7, 39]. The experiments were conducted with a K value of 20, meaning the trajectory with the smallest ADE and FDE values was selected from 20 generated trajectories. Our observations demonstrate that MINet exhibited strong competitiveness, significantly outperforming the state-of-the-art Conv2D and SR-LSTM models. Specifically, Conv2D achieved an average error of 0.44 on ADE and 0.91 on FDE, whereas MINet achieved a remarkable 47.7 % improvement on ADE and 62.6 % improvement on FDE.

In addition to comparing with baseline models, we conducted ablation experiments on our model by progressively including different modules. Starting with only the destination encoding module (Ours-D), we sequentially added the scene feature encoding module (Ours-D-S) and the interaction feature encoding module (Ours-D-S-I). The experimental results show a steady increase in performance, affirming the effectiveness of each module. Particularly, Ours-D-S achieved an average improvement of 23.5 % on the ADE metric and 33.3 % on the FDE metric compared to Ours-D.

Interestingly, Ours-D-S-I exhibited little improvement over the Ours-D-S model across all scenarios. For the Univ, Zara1, and Zara2 scenes, the ADE metric for Ours-D-S remained consistent. To investigate the possible reasons, we examined the original videos of the dataset. We observed that there were fewer pedestrian interactions in these three scenes, resulting in insufficient training data for the model, which explains the slight performance improvement.

	SoPhie	SimAug	Ours-D	Ours-D-S	Ours-D-S-I	DESIRE	Ours-D-S-I
K	20	20	20	20	20	5	1
ADE	16.27	10.27	10.25	**8.38**	**8.38**	19.25	**16.36**
FDE	29.38	19.71	16.72	**12.76**	12.77	34.05	**30.15**

Table 4. Comparison of the baseline model and MINet on SDD when setting different K

SDD: In Table 4, we present a comprehensive comparison of our approach with previous baselines on the SDD dataset. Our approach showcases notable improvements over prior state-of-the-art methods [6, 7, 42], as evident from both the ADE

and FDE metrics. With K set to 20, our proposed MINet achieved an impressive 18.4 % improvement on ADE and 35.2 % improvement on FDE, outperforming the leading SimAug. Moreover, even when K is set to 1, MINet still achieves superior results, surpassing the DESIRE model with $K = 5$ and even outperforming the Social GAN with $K = 20$.

The significance of the destination in pedestrian trajectory prediction is underscored by the performance of solely Ours-D, which outperformed the baseline, highlighting the critical role of the destination in the trajectory prediction process. Additionally, Ours-D-S further improved upon Ours-D, revealing the essential contribution of the scene feature encoding module. However, the performance of Ours-D-S-I was found to be comparable to Ours-D-S, likely due to the spaciousness of the SDD dataset locations, resulting in fewer "avoidance" instances and consequently, limited training and testing data.

Previous experiments [9] have shown that in the "Best of N" method, the larger the value of K, the smaller the ADE and FDE error metrics, signifying better model performance with lower ADE and FDE values under the same conditions. Our model's noteworthy performance, even under the condition of $K = 1$, outperforms DESIRE with $K = 5$ and Social GAN with $K = 20$, highlighting the significant advantages of our model in pedestrian trajectory prediction tasks.

During our ablation experiments, two significant issues emerged. Firstly, when the network is superimposed on the scene feature encoding module, the network training speed becomes very slow. This slowdown predominantly stems from the necessity of image feature extraction within the scene feature encoding module, where the choice of CNN significantly influences training speed. Excessive depth within the network prolongs both training and prediction times, jeopardizing real-time applicability. Conversely, overly shallow networks compromise the precision of scene feature extraction, consequently diminishing prediction efficacy. Hence, future endeavors should meticulously balance CNN depth with prediction accuracy.

Secondly, when the network is superimposed to the interactive feature encoding module, the improvement in prediction accuracy is relatively small. Plausible explanations for this phenomenon include the heightened complexity induced by module integration, leading to diminished generalization performance. Alternatively, insufficient interaction data within the dataset may impede the module's capacity to discern interaction cues, consequently limiting the network's performance improvement. Subsequent research could explore optimizing interaction feature extraction modules with enhanced generalization capabilities or augmenting dataset diversity by incorporating additional interaction scenarios for network training.

5 CONCLUSIONS AND FUTURE WORK

This paper proposed a novel Multi-Information fusion Network, MINet, designed for pedestrian trajectory prediction. To better simulate the real walking environment of pedestrians, we incorporate a "walkable area" that encompasses not only

the pedestrian-accessible space but also the opposing "non-walkable area", including static obstacles and buildings, to enrich scene information. Additionally, we utilize an interaction feature encoding module to capture pedestrian interactions in dense scenes. Furthermore, a destination feature encoding module is integrated to extract pedestrians' destinations and describe their intentions. In our study, we also conducted ablation experiments on these modules, and the results clearly demonstrate the superiority of MINet over existing baseline models on both the ETH/UCY and SDD datasets. The incorporation of the scene feature encoding module, the interaction feature encoding module, and the destination feature encoding module collectively contribute to the improved performance of MINet in pedestrian trajectory prediction tasks.

The proposed network in our study exhibits promising capabilities, yet there remain avenues for refinement. Firstly, the incorporation of multiple influencing factors in pedestrian trajectory prediction contributes to network bloating and prolonged computation times, potentially compromising real-time performance. Therefore, optimizing the network's computational efficiency or reducing complexity without sacrificing prediction accuracy stands as a pivotal next step. Secondly, our experimentation solely revolves around specific datasets, necessitating the collection of additional scene information for broader applicability, thereby fortifying robustness. Lastly, while our interaction module effectively captures interactions, its scope is somewhat limited, failing to encompass the diverse array of objects prevalent in modern traffic scenarios, including wheelchairs, skateboards, bicycles, carts, etc. Enhancing the interaction module to accommodate the complexities of contemporary traffic environments remains imperative. Moving forward, we are committed to addressing these shortcomings to refine our network model and present a more robust and effective solution.

REFERENCES

[1] Luo, Y.—Cai, P.—Bera, A.—Hsu, D.—Lee, W. S.—Manocha, D.: PORCA: Modeling and Planning for Autonomous Driving Among Many Pedestrians. IEEE Robotics and Automation Letters, Vol. 3, 2018, No. 4, pp. 3418–3425, doi: 10.1109/LRA.2018.2852793.

[2] Bennewitz, M.—Burgard, W.—Thrun, S.: Learning Motion Patterns of Persons for Mobile Service Robots. Vol. 4, 2002, pp. 3601–3606, doi: 10.1109/ROBOT.2002.1014268.

[3] Yao, J.—Ye, Y.: The Effect of Image Recognition Traffic Prediction Method Under Deep Learning and Naive Bayes Algorithm on Freeway Traffic Safety. Image and Vision Computing, Vol. 103, 2020, Art. No. 103971, doi: 10.1016/j.imavis.2020.103971.

[4] Luber, M.—Stork, J. A.—Tipaldi, G. D.—Arras, K. O.: People Tracking with Human Motion Predictions from Social Forces. 2010, pp. 464–469, doi: 10.1109/ROBOT.2010.5509779.

[5] Alahi, A.—Goel, K.—Ramanathan, V.—Robicquet, A.—Fei-Fei, L.—Savarese, S.: Social LSTM: Human Trajectory Prediction in Crowded Spaces. 2016, pp. 961–971, doi: 10.1109/CVPR.2016.110.

[6] Gupta, A.—Johnson, J.—Fei-Fei, L.—Savarese, S.—Alahi, A.: Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks. 2018, pp. 2255–2264, doi: 10.1109/CVPR.2018.00240.

[7] Sadeghian, A.—Kosaraju, V.—Sadeghian, A.—Hirose, N.—Rezatofighi, H.—Savarese, S.: SoPhie: An Attentive GAN for Predicting Paths Compliant to Social and Physical Constraints. 2019, pp. 1349–1358, doi: 10.1109/CVPR.2019.00144.

[8] Manh, H.—Alaghband, G.: Scene-LSTM: A Model for Human Trajectory Prediction. CoRR, 2018, doi: 10.48550/arXiv.1808.04018.

[9] Mangalam, K.—Girase, H.—Agarwal, S.—Lee, K. H.—Adeli, E.—Malik, J.—Gaidon, A.: It Is Not the Journey But the Destination: Endpoint Conditioned Trajectory Prediction. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. M. (Eds.): Computer Vision – ECCV 2020. Springer, Cham, Lecture Notes in Computer Science, Vol. 12347, 2020, pp. 759–776, doi: 10.1007/978-3-030-58536-5_45.

[10] Wang, C.—Wang, Y.—Xu, M.—Crandall, D. J.: Stepwise Goal-Driven Networks for Trajectory Prediction. IEEE Robotics and Automation Letters, Vol. 7, 2022, No. 2, pp. 2716–2723, doi: 10.1109/LRA.2022.3145090.

[11] Liang, J.—Jiang, L.—Niebles, J. C.—Hauptmann, A. G.—Fei-Fei, L.: Peeking Into the Future: Predicting Future Person Activities and Locations in Videos. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 5718–5727, doi: 10.1109/CVPR.2019.00587.

[12] Pei, Z.—Qi, X.—Zhang, Y.—Ma, M.—Yang, Y. H.: Human Trajectory Prediction in Crowded Scene Using Social-Affinity Long Short-Term Memory. Pattern Recognition, Vol. 93, 2019, pp. 273–282, doi: 10.1016/j.patcog.2019.04.025.

[13] Helbing, D.—Farkas, I.—Vicsek, T.: Simulating Dynamical Features of Escape Panic. Nature, Vol. 407, 2000, No. 6803, pp. 487–490, doi: 10.1038/35035023.

[14] Lin, K.—Chen, M.—Deng, J.—Hassan, M. M.—Fortino, G.: Enhanced Fingerprinting and Trajectory Prediction for IoT Localization in Smart Buildings. IEEE Transactions on Automation Science and Engineering, Vol. 13, 2016, No. 3, pp. 1294–1307, doi: 10.1109/TASE.2016.2543242.

[15] Morris, B. T.—Trivedi, M. M.: Trajectory Learning for Activity Understanding: Unsupervised, Multilevel, and Long-Term Adaptive Approach. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, 2011, No. 11, pp. 2287–2301, doi: 10.1109/TPAMI.2011.64.

[16] Zhang, P.—Ouyang, W.—Zhang, P.—Xue, J.—Zheng, N.: SR-LSTM: State Refinement for LSTM Towards Pedestrian Trajectory Prediction. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 12077–12086, doi: 10.1109/CVPR.2019.01236.

[17] Salzmann, T.—Ivanovic, B.—Chakravarty, P.—Pavone, M.: Trajectron++: Dynamically-Feasible Trajectory Forecasting with Heterogeneous Data. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. M. (Eds.): Computer Vision –

ECCV 2020. Springer, Cham, Lecture Notes in Computer Science, Vol. 12363, 2020, pp. 683–700, doi: 10.1007/978-3-030-58523-5_40.

[18] Ivanovic, B.—Pavone, M.: The Trajectron: Probabilistic Multi-Agent Trajectory Modeling with Dynamic Spatiotemporal Graphs. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 2375–2384, doi: 10.1109/ICCV.2019.00246.

[19] Yuan, Y.—Weng, X.—Ou, Y.—Kitani, K.: AgentFormer: Agent-Aware Transformers for Socio-Temporal Multi-Agent Forecasting. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 9793–9803, doi: 10.1109/ICCV48922.2021.00967.

[20] Helbing, D.—Molnár, P.: Social Force Model for Pedestrian Dynamics. Physical Review E, Vol. 51, 1995, No. 5, pp. 4282–4286, doi: 10.1103/PhysRevE.51.4282.

[21] Xue, H.—Huynh, D. Q.—Reynolds, M.: SS-LSTM: A Hierarchical LSTM Model for Pedestrian Trajectory Prediction. 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 2018, pp. 1186–1194, doi: 10.1109/WACV.2018.00135.

[22] Park, S. H.—Kim, B.—Kang, C. M.—Chung, C. C.—Choi, J. W.: Sequence-to-Sequence Prediction of Vehicle Trajectory via LSTM Encoder-Decoder Architecture. 2018 IEEE Intelligent Vehicles Symposium (IV), 2018, pp. 1672–1678, doi: 10.1109/IVS.2018.8500658.

[23] Mangalam, K.—An, Y.—Girase, H.—Malik, J.: From Goals, Waypoints & Paths to Long Term Human Trajectory Forecasting. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 15213–15222, doi: 10.1109/ICCV48922.2021.01495.

[24] Pfeiffer, M.—Paolo, G.—Sommer, H.—Nieto, J.—Siegwart, R.—Cadena, C.: A Data-Driven Model for Interaction-Aware Pedestrian Motion Prediction in Object Cluttered Environments. 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 5921–5928, doi: 10.1109/ICRA.2018.8461157.

[25] Vemula, A.—Muelling, K.—Oh, J.: Social Attention: Modeling Attention in Human Crowds. 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 4601–4607, doi: 10.1109/ICRA.2018.8460504.

[26] Jaipuria, N.—Habibi, G.—How, J. P.: A Transferable Pedestrian Motion Prediction Model for Intersections with Different Geometries. CoRR, 2018, doi: 10.48550/arXiv.1806.09444.

[27] Huang, Z.—Wang, J.—Pi, L.—Song, X.—Yang, L.: LSTM Based Trajectory Prediction Model for Cyclist Utilizing Multiple Interactions with Environment. Pattern Recognition, Vol. 112, 2021, Art. No. 107800, doi: 10.1016/j.patcog.2020.107800.

[28] Hasan, I.—Setti, F.—Tsesmelis, T.—Belagiannis, V.—Amin, S.—Del Bue, A.—Cristani, M.—Galasso, F.: Forecasting People Trajectories and Head Poses by Jointly Reasoning on Tracklets and Vislets. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 43, 2021, No. 4, pp. 1267–1278, doi: 10.1109/TPAMI.2019.2949414.

[29] Kothari, P.—Sifringer, B.—Alahi, A.: Interpretable Social Anchors for Human Trajectory Forecasting in Crowds. 2021 IEEE/CVF Conference on Com-

puter Vision and Pattern Recognition (CVPR), 2021, pp. 15551–15561, doi: 10.1109/CVPR46437.2021.01530.

[30] Zhong, J.—Sun, H.—Cao, W.—He, Z.: Pedestrian Motion Trajectory Prediction with Stereo-Based 3D Deep Pose Estimation and Trajectory Learning. IEEE Access, Vol. 8, 2020, pp. 23480–23486, doi: 10.1109/ACCESS.2020.2969994.

[31] Sun, D.—Roth, S.—Lewis, J. P.—Black, M. J.: Learning Optical Flow. In: Forsyth, D., Torr, P., Zisserman, A. (Eds.): Computer Vision – ECCV 2008. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 5304, 2008, pp. 83–97, doi: 10.1007/978-3-540-88690-7_7.

[32] Kothari, P.—Kreiss, S.—Alahi, A.: Human Trajectory Forecasting in Crowds: A Deep Learning Perspective. IEEE Transactions on Intelligent Transportation Systems, Vol. 23, 2022, No. 7, pp. 7386–7400, doi: 10.1109/TITS.2021.3069362.

[33] Han, Y.—Chen, P.—Meng, T.: Harris Corner Detection Algorithm at Sub-Pixel Level and Its Application. Proceedings of the 2015 International Conference on Computational Science and Engineering, Atlantis Press, Advances in Computer Science Research, 2015, pp. 133–137, doi: 10.2991/iccse-15.2015.23.

[34] Li, Y. B.—Li, J. J.: Harris Corner Detection Algorithm Based on Improved Contourlet Transform. Procedia Engineering, Vol. 15, 2011, pp. 2239–2243, doi: 10.1016/j.proeng.2011.08.419.

[35] Xia, B.—Wong, C.—Peng, Q.—Yuan, W.—You, X.: CSCNet: Contextual Semantic Consistency Network for Trajectory Prediction in Crowded Spaces. Pattern Recognition, Vol. 126, 2022, Art. No. 108552, doi: 10.1016/j.patcog.2022.108552.

[36] Pellegrini, S.—Ess, A.—Schindler, K.—van Gool, L.: You'll Never Walk Alone: Modeling Social Behavior for Multi-Target Tracking. 2009 IEEE 12th International Conference on Computer Vision, 2009, pp. 261–268, doi: 10.1109/ICCV.2009.5459260.

[37] Lerner, A.—Chrysanthou, Y.—Lischinski, D.: Crowds by Example. Computer Graphics Forum, Vol. 26, 2007, No. 3, pp. 655–664, doi: 10.1111/j.1467-8659.2007.01089.x.

[38] Robicquet, A.—Sadeghian, A.—Alahi, A.—Savarese, S.: Learning Social Etiquette: Human Trajectory Understanding in Crowded Scenes. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.): Computer Vision – ECCV 2016. Springer, Cham, Lecture Notes in Computer Science, Vol. 9912, 2016, pp. 549–565, doi: 10.1007/978-3-319-46484-8_33.

[39] Li, J.—Ma, H.—Tomizuka, M.: Conditional Generative Neural System for Probabilistic Trajectory Prediction. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019, pp. 6150–6156, doi: 10.1109/IROS40897.2019.8967822.

[40] Deo, N.—Trivedi, M. M.: Trajectory Forecasts in Unknown Environments Conditioned on Grid-Based Plans. CoRR, 2020, doi: 10.48550/arXiv.2001.00735.

[41] Zamboni, S.—Kefato, Z. T.—Girdzijauskas, S.—Norén, C.—Dal Col, L.: Pedestrian Trajectory Prediction with Convolutional Neural Networks. Pattern Recognition, Vol. 121, 2022, Art. No. 108252, doi: 10.1016/j.patcog.2021.108252.

[42] Liang, J.—Jiang, L.—Hauptmann, A.: SimAug: Learning Robust Representa-

tions from Simulation for Trajectory Prediction. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. M. (Eds.): Computer Vision – ECCV 2020. Springer, Cham, Lecture Notes in Computer Science, Vol. 12358, 2020, pp. 275–292, doi: 10.1007/978-3-030-58601-0_17.

[43] Lee, N.—Choi, W.—Vernaza, P.—Choy, C. B.—Torr, P. H. S.—Chandraker, M.: DESIRE: Distant Future Prediction in Dynamic Scenes with Interacting Agents. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2165–2174, doi: 10.1109/CVPR.2017.233.

Tao Yuan is currently studying for a Master's degree at the Taiyuan University of Technology. His major is pattern recognition and deep learning.

Xiaohong Han is Professor at the Taiyuan University of Technology and a Master's tutor. She visited the University of Texas at Dallas in the USA for a one-year scientific research exchange and cooperative research on the application of intelligent optimization algorithms. Her main research areas are big data mining, artificial intelligence, pattern recognition, image processing, etc.