# Synchronous Bi-Directional Pedestrian Trajectory Prediction with Error Compensation

Ce Xie[1,2], Yuanman Li[1,2](✉), Rongqin Liang[1,2], Li Dong[3], and Xia Li[1,2]

[1] College of Electronics and Information Engineering, Shenzhen University, Shenzhen, China
[2] Guangdong Key Laboratory of Intelligent Information Processing
{2100432087, 1810262064}@email.szu.edu.cn, {yuanmanli,lixia}@szu.edu.cn
[3] Department of Computer Science, Ningbo University, Ningbo, China

**Abstract.** Pedestrian trajectory prediction as an essential part of reasoning human motion behaviors, has been deployed in a number of vision applications, such as autonomous navigation and video surveillance. Most existing methods adopt autoregressive frameworks to forecast the future trajectory, where the trajectory is iteratively generated based on the previous outputs. Such a process will suffer from large accumulated errors over the long-term forecast horizon. To address this issue, in this paper, we propose a Synchronous Bi-Directional framework (SBD) with error compensation for pedestrian trajectory prediction, which can greatly alleviate the error accumulation during prediction. Specifically, we first develop a bi-directional trajectory prediction mechanism, and force the predicting procedures for two opposite directions to be synchronous through a shared motion characteristic. Different from previous works, the mutual constraints inherent to our framework from the synchronous opposite-predictions can significantly prevent the error accumulation. In order to reduce the possible prediction error in each timestep, we further devise an error compensation network to model and compensate for the positional deviation between the ground-truth and the predicted trajectory, thus improving the prediction accuracy of our scheme. Experiments conducted on the Stanford Drone dataset and the ETH-UCY dataset show that our method achieves much better results than existing algorithms. Particularly, by resorting to our alleviation methodology for the error accumulation, our scheme exhibits superior performance in the long-term pedestrian trajectory prediction.

## 1 Introduction

Pedestrian trajectory prediction aims to forecast the future trajectory based on the observed history trajectory. As one of the most important human behavior prediction tasks, it plays an important role in many related fields, such as autonomous navigation [1, 2] and video surveillance [3, 4].

Although the pedestrian trajectory prediction has been analyzed and researched in a variety of ways, it remains to be a challenging task because of the inherent properties of human. First, human behaviors are full of indeterminacy,

thus there could be several plausible but distinct future trajectories under the same historical trajectory and scene. Second, pedestrians are highly affected by their neighbors. However, modeling the underlying complex inter-personal interactions is still challenging in real scenarios. Given the historical trajectory of the target pedestrian, a pedestrian trajectory prediction method should effectively model both the temporal motion patterns and the possible spatial interactions, and then forecast the positions or distribution of the future trajectory based on the modeled features.

The pioneering methods [5–8] mainly focus on the human motions and human-human interactions by using handcrafted features. Recently, the attention mechanism and the recurrent neural networks (RNNs), which show outstanding ability in extracting temporal dependencies and spatial interactions among adjacent pedestrians, have been applied to many methods [9–15] and achieve a great success in pedestrian trajectory prediction. However, most of these methods use the single autoregressive frameworks to forecast the future trajectory. For examples, approaches like [9, 10] generate trajectory at a timestep and feed the predicted trajectory back into the model to produce the trajectory for the next timestep. Alternatively, methods like [11–13] forecast the spatial position at a future time and then feed the currently predicted position back into the model to produce the next spatial position. These frameworks would suffer from the huge accumulated errors over the long-term forecast horizon [16], and thus their performance may tend to degrade rapidly over time.

In this paper, we propose a novel Synchronous Bi-Directional framework (SBD) with error compensation for pedestrian trajectory prediction to alleviate the problem of error accumulation. SBD first models the spatial-temporal feature through a simple temporal motion extractor and a spatial interaction extractor. Meanwhile, SBD incorporates a conditional variational autoencoder (CVAE) module to produce the multi-modality of the future trajectory. We then propose a synchronous bi-directional trajectory generator to alleviate the error accumulation in trajectory prediction process. Specifically, we devise a shared characteristic between two opposite predictions, by resorting to which, the generator performs mutually constrained synchronous bi-directional prediction to greatly prevent the error accumulation. Different from previous methods, such as [12], the trajectory generator in SBD implements predictions for two opposite directions synchronously, while maintaining relative independence to prevent from the error propagation between the two branches. Besides, to further reduce possible errors in the predicted trajectory, we design an error compensation network to model and compensate for the positional deviation between the ground-truth and predicted trajectory. The main contributions of our work can be summarized as follows:

- We propose a synchronous bi-directional framework (SBD) for pedestrian trajectory prediction. Different from existing approaches, our predicting procedures for two opposite directions are designed to be synchronous through a shared motion characteristic, and the mutual constraints from the syn-

chronous opposite-predictions can significantly prevent the error accumulation.
- Through modeling the spatial deviation between the predicted trajectory and the ground-truth, we further devise an error compensation network to compensate the prediction error at each timestep, thus improving the final prediction accuracy.
- Our method achieves the state-of-the-art performance on two benchmark pedestrian trajectory prediction datasets. Particularly, thanks to the alleviation scheme for the error accumulation, our method exhibits excellent performance in the long-term pedestrian trajectory prediction.

## 2 Related Work

Pedestrian trajectory prediction aims to estimate the future positions base on the observed paths, which can be roughly categorized into methods based on hand-crafted features and methods based on deep learning. In this section, we give a brief review of related work.

**Pedestrian trajectory prediction based on hand-crafted features.** Traditional methods [5–8] heavily rely on the hand-crafted rules to describe human motions and human-human interactions. For examples, the Social Force [5] employs a dynamic system to model the human motions as attractive force towards a destination and repulsive forces to avoid collision. The Linear Trajectory Avoidance is proposed in [8] for short-term pedestrian trajectory prediction through jointly modeling the scene information and the dynamic social interactions among pedestrians. However, these hand-crafted methods are difficult to generalize in more complex real scenes.

**Pedestrian trajectory prediction based on deep learning.** Thanks to the powerful representation of deep learning, many methods design ingenious networks for pedestrian trajectory prediction. For examples, Social-LSTM [14] extracts the motion feature for each pedestrian through dividual Long Short Term Memory networks (LSTMs) and devises a social pooling layer to aggregate the interaction information among nearby pedestrians. SR-LSTM [17] refines the current states of all pedestrians in a crowd by timely capturing the changes of their neighbors and modeling the social interactions within the same moment.

The graph convolutional networks (GCNs) [18] are also introduced by many trajectory prediction methods to extract the cooperative interactions among pedestrians [10, 19–22]. For instance, Social-STGCNN [19] learns the spatial context and temporal context using a spatio-temporal graph convolution neural network. SGCN [21] introduces a sparse graph to model the sparse directed interactions among pedestrians. In addition, VDRGCN [22] devises three directed graph topologies to exploit different types of social interactions.

The attention based approaches have been devised for pedestrian trajectory prediction to model the temporal dependencies and spatial interactions among

pedestrians. For examples, Social-BiGAT [20] combines the graph model and attention mechanism to model the social interactions. TPNSTA [13] adaptively extracts important information in both spatial and temporal domains through a unified spatial-temporal attention mechanism. Agentformer [9] simultaneously learns representations from the time and social dimensions and proposes a agent-aware attention mechanism for multi-agent trajectory prediction. More recently, the work CAGN [23] designs a complementary dual-path attention architecture to capture the frequent and peculiar modals of the trajectory.

Due to the inherent multi-modality of human behaviors, many stochastic prediction methods are proposed to learn the distribution of trajectory based on the deep generative model, such as generative adversarial networks (GANs) [13, 15, 24, 25], conditional variational autoencoders (CVAEs) [9, 11, 12, 26–28]. For examples, Social-GAN [15] incorporates the LSTM model with the GANs to produce multiple plausible trajectories. PECNet [26] concatenates the features of historical trajectory and predicted multi-modal end-points to predict the whole trajectories. BiTraP [12] predicts future trajectories from two directions based on multi-modal goal estimation. DisDis [27] further studies the latent space and proposes to learn the discriminative personalized latent distributions to represent personalized future behaviors. In addition, SIT [29] builds a hand-craft tree and uses the branches in the tree to represent the multi-modal future trajectories.

## 3 Proposed Method

In this section, we introduce our SBD, which performs mutually constrained synchronous bi-directional trajectory prediction based on a shared motion characteristic to alleviate the problem of error accumulation. We describe the architecture of our method in Fig. 1, which mainly consists of three components: 1) a spatial-temporal encoder; 2) a synchronous bi-directional decoder and 3) an error compensation network.

### 3.1 Problem Formulation

Pedestrian trajectory prediction task aims to generate plausible future trajectory for the target pedestrian based on the historical trajectories of target and target's neighboring pedestrians. Mathematically, let $x^t \in \mathbb{R}^2$ be the spatial coordinate of a target pedestrian at the timestamp $t$, and denote $X = [x^{-H+1}, x^{-H+2}, ..., x^0] \in \mathbb{R}^{H\times 2}$ as the observed history trajectory, where $H$ is observation horizon and the current location is $x^0$. Let $\mathcal{N}$ represent the neighbor set and $\mathbb{X}_\mathcal{N} = [X_{\mathcal{N}_1}, X_{\mathcal{N}_2}, ..., X_{\mathcal{N}_N}] \in \mathbb{R}^{N\times H\times 2}$ be the historical trajectories of neighbors, where the $X_{\mathcal{N}_i} \in \mathbb{R}^{H\times 2}$ belongs to the $i$-th neighbor. We use $Y = [y^1, y^2, ..., y^{T_f}] \in \mathbb{R}^{T_f\times 2}$ to represent the ground-truth future trajectory of the target pedestrian, where $y^t \in \mathbb{R}^2$ denotes the spatial coordinate at the future timestamp $t$, and $T_f$ is the prediction horizon. Similarly, we use $\hat{Y} = [\hat{y}^1, \hat{y}^2, ..., \hat{y}^{T_f}] \in \mathbb{R}^{T_f\times 2}$ to indicate the predicted future trajectory. The overall goal is to learn a trajectory prediction model $\mathcal{F}$, which predicts a future trajectory $\hat{Y} = \mathcal{F}(X, \mathbb{X}_\mathcal{N})$ close to $Y$.

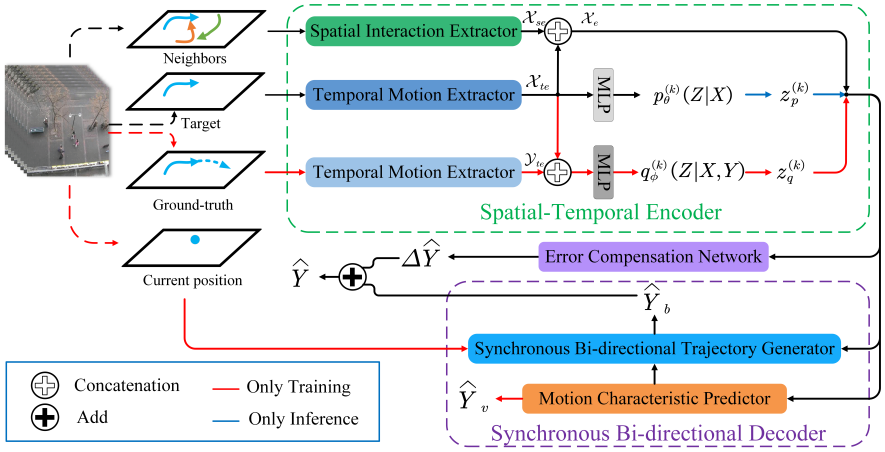


**Fig. 1.** The framework of SBD. The blue trajectory belongs to the target pedestrian and the orange/green trajectories are neighbours.

### 3.2 Spatial-Temporal Encoder

**Modeling trajectories.** To capture the temporal motion feature of a target pedestrian, we adopt a simple temporal motion extractor as proposed in [12]. We first embed the positions of the target pedestrian through a fully connected layer (FC) with ReLU activation as

$$e^t = FC(x^t; \Theta_e) \ , \tag{1}$$

where $t = -H+1, ..., 0$ and $\Theta_e$ represents the parameters of FC. The embedded feature $e^t \in \mathbb{R}^{1\times d_{te}}$ is then fed into a GRU block to produce the hidden state at the time step $t$:

$$h_{te}^t = GRU(h_{te}^{t-1}, e^t) \ , \tag{2}$$

We obtain the temporal motion feature $\mathcal{X}_{te} \in \mathbb{R}^{1\times d_{te}}$ as $\mathcal{X}_{te} = h_{te}^0$.

As for the modeling of social interactions among surrounding pedestrians, in this paper, we propose to capture the social influence of the neighbors to the target using the attention mechanism [30]. Specifically, we first embed the current states of all pedestrians including the target and neighbors as:

$$r = FC(x^0; \Theta_r) \ , \quad r_{\mathcal{N}} = FC(x_{\mathcal{N}}^0; \Theta_r) \ , \tag{3}$$

where $r \in \mathbb{R}^{1\times d_{se}}$ and $r_{\mathcal{N}} \in \mathbb{R}^{N\times d_{se}}$ contains the features of the target and neighbors, respectively, and $\Theta_r$ represents the parameters of FC. According to the dot-product attention strategy [30], we calculate the spatial interaction feature as:

$$\mathcal{X}_{se} = softmax(\frac{QK^T}{\sqrt{d_{se}}})V \ \in \mathbb{R}^{1\times d_{se}} \ . \tag{4}$$

$$Q = rW_Q \ , \quad K = (r \oplus r_{\mathcal{N}})W_K \ , \quad V = (r \oplus r_{\mathcal{N}})W_V \ , \tag{5}$$

where $\oplus$ serves as the concatenation operation, and $W_Q, W_K, W_V \in \mathbb{R}^{d_{se} \times d_{se}}$ represent the trainable parameters of dividual linear transformations to generate the query $Q \in \mathbb{R}^{1 \times d_{se}}$, key $K \in \mathbb{R}^{(N+1) \times d_{se}}$ and value $V \in \mathbb{R}^{(N+1) \times d_{se}}$.

Finally, we produce the spatial-temporal feature $\mathcal{X}_e$ of the target as:

$$\mathcal{X}_e = (\mathcal{X}_{te} \oplus \mathcal{X}_{se}) W_e \ , \tag{6}$$

where $W_e \in \mathbb{R}^{(d_{te}+d_{se}) \times d_e}$ is the trainable weight matrices. In the training stage, the ground-truth $Y$ is also encoded by another temporal motion extractor yielding $\mathcal{Y}_{te}$.

**Generating distributions of trajectory.** Considering the multi-modality of future trajectory, similar to the previous methods [9, 11, 12, 27], SBD incorporates a conditional variational autoencoder (i.e., CVAE [31]) to estimate the future trajectory distribution $p(Y|X)$. Based on the [31], we introduce a latent variable $z$ to represent the high-level latent intent of the target pedestrian and rewrites $p(Y|X)$ as:

$$p(Y|X) = \int p(Y|X, Z) p_\theta(Z|X) dZ \ , \tag{7}$$

where $p_\theta(Z|X)$ is the Gaussian distribution based on the observed trajectory.

In this work, we use a multilayer perceptron (MLP) to map the temporal feature $\mathcal{X}_{te}$ to the Gsussian parameters $(\mu_p, \sigma_p)$ of the distribution $p_\theta(Z|X) = N(\mu_p, \sigma_p^2)$. According to the [31], in the training stage, another MLP is adopted to produce the distribution $q_\phi(Z|X, Y) = N(\mu_q, \sigma_q^2)$ with the inputs of $\mathcal{X}_{te}$ and $\mathcal{Y}_{te}$. The latent variable $z$ is sampled from $q_\phi(Z|X, Y)$. In the inference stage, we directly obtain different latent variables from $p_\theta(Z|X)$ to generate the multi-modality of trajectory.

To produce diverse and plausible trajectories, we stack $K$ parallel pairs of MLP to obtain the diverse latent variables. Therefore, we take the accumulation of negative evidence lower bound in the [31] as the corresponding loss function:

$$\begin{aligned} \mathcal{L}_{elbo} = \sum_{k=1}^{K} \Big\{ & -\mathbb{E}_{q_\phi^{(k)}(Z|X,Y)} \left[ \log p^{(k)}(Y \mid X, Z) \right] \\ & + KL \left( q_\phi^{(k)}(Z \mid X, Y) \| p_\theta^{(k)}(Z \mid X) \right) \Big\} \ . \end{aligned} \tag{8}$$

### 3.3 Synchronous Bi-Directional Decoder

In order to alleviate the problem of error accumulation in the trajectory prediction process, we propose a novel synchronous bi-directional decoder as shown in Fig. 2. The proposed decoder is a two-phase trajectory prediction system, where the first step is to generate a series of motion characteristics shared by two opposite directions, and the second step is to perform the mutually constrained simultaneous bi-directional prediction based on the motion characteristic.

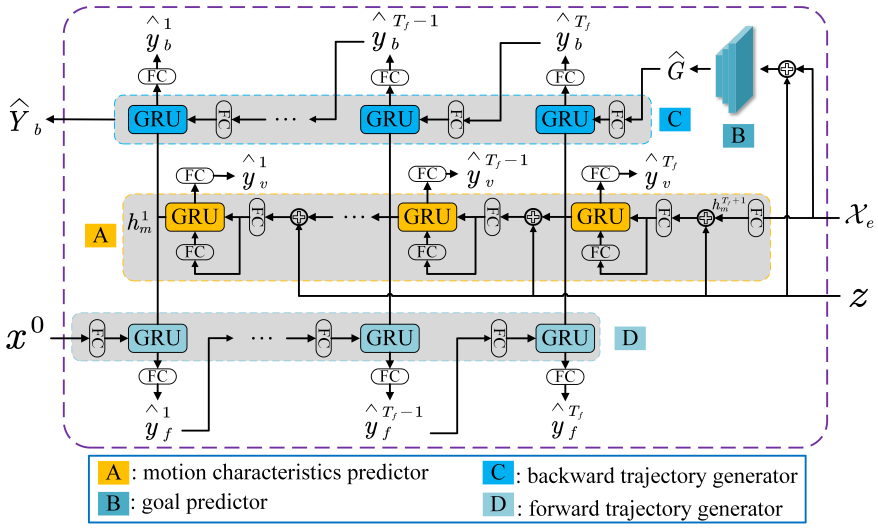


**Fig. 2.** The structure of the synchronous bi-directional decoder. The proposed decoder is a two-phase trajectory prediction system, where the first step is to generate motion characteristics through a motion characteristic predictor and the second step is to perform the mutually constrained simultaneous bi-directional prediction based on the motion characteristic. Finally, the decoder output the backward trajectory $\hat{Y}_b$ as the preliminary predicted result.

We should note that the existing work [12] also adopts a bi-directional trajectory prediction structure. However, it predicts two opposite directions sequentially, where it first performs forward trajectory prediction, and the results are then used as an assistance for the backward trajectory prediction. Such a sequential process can not solve the problem of error propagation. One may also use a vanilla independent bi-directional framework, which independently predicts trajectories in opposite directions. Nevertheless, the error accumulation would occur in both directions since no interactions between them.

Compared with the above strategies, our bi-directional decoder performs synchronous bi-directional prediction, where the mutual constraints from two directions can significantly prevent the error accumulation. The experiments shown in the Section 4 will also show the superiority of our scheme.

**Motion characteristic predictor.** In order to prevent the error propagation in each prediction direction, we devise a motion characteristic predictor, where the generated features are shared by two opposite predictions. Since the prediction procedures for two opposite predictions rely on the same feature, they would be affected by the mutual constraint from each other, thus greatly alleviating the error propagation in both directions. Intuitively, the characteristic should reflect the common real-time state of the target pedestrian in the physical

space. In this paper, we specialize motion characteristic as the velocity feature of the pedestrian, and supervise the motion characteristic predictor using the real velocity.

Specifically, the spatial-temporal feature $\mathcal{X}_e$ and the latent vector $z$ are fed into the motion characteristic predictor to generate the motion characteristics. As shown in Fig. 2, a GRU is used as the basic model of the predictor. We first adopt a FC to map $\mathcal{X}_e$ to the initial hidden state $h_m^{T_f+1}$ and produce the hidden states for the timesteps from $T_f$ to 1:

$$r_m^{t+1} = FC(h_m^{t+1} \oplus z; \Theta_m) \ , \tag{9}$$

$$h_m^t = GRU\left(r_m^{t+1}, FC(r_m^{t+1}; \Theta_m^{in})\right) \ , \tag{10}$$

where $\Theta_m, \Theta_m^{in}$ are parameters, and the hidden state $h_m^t$ represents the motion characteristic at the time step $t$. Then, we propose to forecast the velocity vector $\hat{Y}_v = [\hat{y}_v^1, \hat{y}_v^2, ..., \hat{y}_v^{T_f}]$ based on the motion characteristics $h_m^t$ $(t = 1, ..., T_f)$ as:

$$\hat{y}_v^t = FC(h_m^t; \Theta_v) \ , \tag{11}$$

where $\Theta_v$ are parameters to be learned. In the training stage, we force $\hat{Y}_v$ to approximate the true velocity $\widetilde{Y}_v = \frac{\partial Y}{\partial t}$, and the corresponding loss function is formulated as:

$$\mathcal{L}_{motion} = \left\| \hat{Y}_v - \widetilde{Y}_v \right\|_2 \ . \tag{12}$$

**Synchronous bi-directional trajectory generator.** In order to alleviate the error accumulation in the trajectory prediction process, we devise a synchronous bi-directional trajectory generator, which consists of a goal predictor, a backward trajectory generator and a forward trajectory generator.

*Goal predictor.* The goal predictor aims to forecast the goal position of the trajectory based on the feature $\mathcal{X}_e$ and the latent vector $z$, which will be used to guide the backward trajectory generation as shown in Fig. 2. The loss of the goal predictor is defined as:

$$\mathcal{L}_{goal} = \left\| \hat{G} - \tilde{y}^{T_f+1} \right\|_2 \ . \tag{13}$$

Here, $\tilde{y}^{T_f+1}$ represents the next position after the endpoint of the target trajectory, which is approximately calculated as:

$$\tilde{y}^{T_f+1} \approx y^{T_f} + (y^{T_f} - y^{T_f-1}) \ . \tag{14}$$

*Bi-directional trajectory generator.* As depicted in Fig. 2, the two opposite prediction branches are synchronous at each timestep with a shared motion characteristic $h_m^t$ $(t = 1, ..., T)$. The GRU is adopted as the basic model of two opposite trajectory generators. The current position $x^0$ and predicted goal $\hat{G}$ act as the initial states for the forward and backward branches, respectively. The procedure of the forward prediction is formulated as:

$$\begin{aligned} h_f^t &= GRU(h_m^t, FC(\hat{y}_f^{t-1}; \Theta_f) \ , \\ \hat{y}_f^t &= FC(h_f^t; \Theta_f') \ , \end{aligned} \tag{15}$$

while the procedure of the backward prediction is

$$\begin{aligned} h_b^t &= GRU(h_m^t, FC(\hat{y}_b^{t+1}; \Theta_b)) \ , \\ \hat{y}_b^t &= FC(h_b^t; \Theta_b') \ , \end{aligned} \tag{16}$$

Here, $t = 1, ..., T_f$; $\Theta_f, \Theta_f', \Theta_b, \Theta_b'$ are learnable parameters; $h_f^t, h_b^t$ represent the hidden states, and $\hat{y}_f^t, \hat{y}_b^t$ denote the predicted positions at timestep $t$ by the forward generator and the backward generator, respectively.

As illustrated in Fig. 2, the motion characteristic $h_m^t$ plays as a shared feature, which is used to predict the position $\hat{y}_b^t$ based on the position at $t+1$ in the backward branch while participating in the prediction of $\hat{y}_f^t$ with the input of $\hat{y}_f^{t-1}$ in the forward branch. This design lets the two opposite prediction branches mutually constrained from each other, thus preventing the error accumulation. For instance, as for the prediction of $\hat{y}_b^1$, the output of the backward generator not only relies on the previous state $\hat{y}_b^2$, but also the shared motion feature $h_m^1$, which is constrained by the forward generator.

Denote $\hat{Y}_f = [\hat{y}_f^1, \hat{y}_f^2, ..., \hat{y}_f^{T_f}]$ and $\hat{Y}_b = [\hat{y}_b^1, \hat{y}_b^2, ..., \hat{y}_b^{T_f}]$ as the predicted trajectory by the forward and backward trajectory generators, respectively. The loss of our synchronous bi-directional trajectory generator is defined as:

$$\mathcal{L}_{traj} = \alpha_1 \left\| \hat{Y}_f - Y \right\|_2 + \alpha_2 \left\| \hat{Y}_b - Y \right\|_2 \ , \tag{17}$$

where $\alpha_1$ and $\alpha_2$ are two hyper-parameters used to balance two prediction branches.

### 3.4 Error Compensation Network.

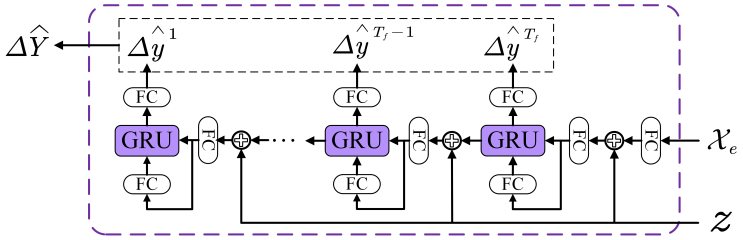


**Fig. 3.** The structure of the error compensation network.

Despite that our synchronous bi-directional framework can effectively prevent the error accumulation in the sequentially prediction process. However, due to the complex multi-modality property inherent to human motion behaviors, there may still exist prediction deviations for some certain contexts. In order to further reduce the possible prediction error at each time-step, we introduce an

error compensation network, which aims to compensate for the spatial deviations between the ground-truth trajectory and the predicted one based on the current context.

As shown in Fig. 3, with the context of the spatial-temporal feature $\mathcal{X}_e$ and the latent vector $z$, the error compensation network predicts a compensation value for the target pedestrian at each timestep, which is formulated as:

$$r_e^{t+1} = FC(h_e^{t+1} \oplus z; \Theta_e) \ , \tag{18}$$

$$h_e^t = GRU\left(r_e^{t+1}, FC(r_e^{t+1}; \Theta_e^{in})\right) \ , \tag{19}$$

$$\Delta\hat{y}^t = FC(h_e^t; \Theta_e^{out})) \ , \tag{20}$$

where $t = 1, ..., T_f$, and the initial hidden state $h_e^{T_f+1}$ is generated by a FC based on the spatial-temporal feature $\mathcal{X}_e$. Besides, $\Theta_e, \Theta_e^{in}, \Theta_e^{out}$ are parameters, and $\Delta\hat{y}^t$ represents the compensation value at the timestep $t$. Letting $\Delta\hat{Y} = [\Delta\hat{y}^1, \Delta\hat{y}^2, ..., \Delta\hat{y}^{T_f}]$ be the predicted compensation value, we take the $\hat{Y}_b + \Delta\hat{Y}$ as the final predicted trajectory. The loss of error compensation network can be written as:

$$\mathcal{L}_{error} = \left\| \hat{Y}_b + \Delta\hat{Y} - Y \right\|_2 \ . \tag{21}$$

Finally, SBD is trained end-to-end by minimizing the following loss function:

$$\mathcal{L} = \beta_1\mathcal{L}_{goal} + \beta_2\mathcal{L}_{traj} + \beta_3\mathcal{L}_{error} + \beta_4\mathcal{L}_{motion} + \beta_5\mathcal{L}_{elbo} \ , \tag{22}$$

where the $\beta_1, \beta_2, \beta_3, \beta_4$ and $\beta_5$ are used to balance different terms.

## 4 Experiments

In this section, we evaluate the performance of our proposed SBD, which is implemented using the PyTorch framework. All the experiments are conducted on a desktop equipped with an NVIDIA RTX 3090 GPU.

### 4.1 Experimental Setup

**Datasets.** We evaluate our method on two public trajectories datasets: the Stanford Drones Dataset (SDD) [32] and ETH-UCY [8, 33].

SDD is a well established benchmark for pedestrian trajectory prediction in bird's eye view. The dataset consists of 20 scenes containing several moving agents and the coordinates of trajectory is recorded at 2.5Hz in pixel coordinate system in pixels.

ETH-UCY contains of five sub-datasets: ETH, HOTEL, UNIV, ZARA1 and ZARA2. All the pedestrian trajectory data in these datasets are captured by fixed surveillance cameras at 2.5Hz and recorded in world-coordinates.

**Evaluation Metric.** For the sake of fairness, we use the standard history-future split, which segment the first 3.2 seconds (8 frames) of a trajectory as historical trajectory to predict the next 4.8 seconds (12 frames) future trajectory. For the ETH-UCY, we follow the leave-one-out strategy [14] with 4 scenes for training and the remaining one for testing. Following prior works [8, 14, 22], we adopt the two widely-used error metrics to evaluate the performance of different pedestrian trajectory prediction models, including: 1) Average Displacement Error (ADE): The average Euclidean distance between the ground-truth trajectory and the predicted one; and 2) Final Displacement Error (FDE): The Euclidean distance between the endpoints of the ground-truth trajectory and the predicted one. To be consistent with previous works [9, 15, 23], we adopt the best-of-K (K = 20) strategy to compute the final ADE and FDE.

**Implementation Details.** In our experiments, the embedding dimension $d_{te}$ and $d_{se}$ in encoder are set to 256 and 32, respectively. The dimension of hidden dimensions in the temporal motion extractor, synchronous bi-directional decoder and error compensation network are 256. The length of the latent vector is 32. Besides, the number of prior nets and posterior nets in encoder is 20. We employ the Adam optimizer [34] to train model and use cosine annealing schedule as in [35] to adjust the learning rate. Beside, we train the entire network with the following hyper-parameter settings: initial learning rate of $10^{-3}$, batch size is 128, $\alpha_1$, $\alpha_2$ in (17) are 0.2, 0.4, the $\beta_1$, $\beta_2$, $\beta_3$, $\beta_4$, $\beta_5$ in (21) are 3, 1, 0.6, 0.4, 0.1, and the number of epochs is 100.

### 4.2 Quantitative Evaluation

We compare our SBD with several generative baselines, including the GAN based methods [15, 25, 36–38], GCN based methods [19, 21], TransFormer based methods [9, 10], CVAE based methods [11, 12, 26], and other generative methods [23, 29, 39–41].

**Performance on standard trajectory prediction.** Table 1 reports the results of our SBD and existing methods [15, 25, 26, 39–41, 29] on SDD. We observe that our method significantly outperforms all the competitive approaches under standard 20 samplings. Specifically, our method reduces the ADE from 8.59 to 7.78 compare to the previous state-of-the-art-20-samplings method, i.e., SIT [29], achieving 9.4% relative improvement. As for FDE metric, SBD is better than Y-Net with 20 samplings by 18.1%. Besides, compared with the method Y-Net [41]+Test Time Sampling Trick (TTST) with 10000 sampling, our method still achieves performance gains on the ADE metric. Notice that our method does not use any scene context, while Y-Net models the additional image information and thus suffers from huge computational costs.

In Table 2, we summarize the results of SBD and existing methods [15, 36, 19, 26, 10, 11, 37, 21, 38, 9, 12, 41, 23, 29] on ETH-UCY. We can still observe that our method achieves the best or second best rank for each dataset. Besides, the

**Table 1.** Comparison with different methods on the SDD (Lower is better). † indicates that the results are reproduced by [42] with the official released code. The values highlighted by red and blue represent the best and second best results, respectively.

| Methods | SGAN | Goal-GAN | PECNet | LB-EBM | PCCSNET | Y-net† | Y-Net+TTST | SIT | SBD |
|---|---|---|---|---|---|---|---|---|---|
| Samping | 20 | 20 | 20 | 20 | 20 | 20 | 10000 | 20 | 20 |
| ADE | 27.23 | 12.2 | 9.96 | 8.87 | 8.62 | 8.97 | 7.85 | 8.59 | 7.78 |
| FDE | 41.44 | 22.1 | 15.88 | 15.61 | 16.16 | 14.61 | 11.85 | 15.27 | 11.97 |

**Table 2.** Comparison with baselines on the ETH-UCY (Lower is better). The values highlighted by red and blue represent the best and second best results, respectively.

| Method | Sampling | ETH | HOTEL | UNIV | ZARA1 | ZARA2 | AVG |
|---|---|---|---|---|---|---|---|
| SGAN | 20 | 0.81/1.52 | 0.72/1.61 | 0.60/1.26 | 0.34/0.69 | 0.42/0.84 | 0.58/1.18 |
| STGAT | 20 | 0.65/1.12 | 0.35/0.66 | 0.52/1.10 | 0.34/0.69 | 0.29/0.60 | 0.43/0.83 |
| Social-STGCNN | 20 | 0.64/1.11 | 0.49/0.85 | 0.44/0.79 | 0.34/0.53 | 0.30/0.48 | 0.44/0.75 |
| PECNet | 20 | 0.54/0.87 | 0.18/0.24 | 0.35/0.60 | 0.22/0.39 | 0.17/0.30 | 0.29/0.48 |
| STAR | 20 | 0.36/0.65 | 0.17/0.36 | 0.31/0.62 | 0.26/0.55 | 0.22/0.46 | 0.26/0.53 |
| Trajectron++ | 20 | 0.39/0.83 | 0.12/0.21 | 0.20/0.44 | 0.15/0.33 | 0.11/0.25 | 0.19/0.41 |
| TPNMS | 20 | 0.52/0.89 | 0.22/0.39 | 0.55/1.13 | 0.35/0.70 | 0.27/0.56 | 0.38/0.73 |
| SGCN | 20 | 0.63/1.03 | 0.32/0.55 | 0.37/0.70 | 0.29/0.53 | 0.25/0.45 | 0.37/0.65 |
| STSF-Net | 20 | 0.63/1.13 | 0.24/0.43 | 0.28/0.52 | 0.23/0.45 | 0.21/0.41 | 0.32/0.59 |
| AgentFormer | 20 | 0.45/0.75 | 0.14/0.22 | 0.25/0.45 | 0.18/0.30 | 0.14/0.24 | 0.23/0.39 |
| BiTraP-NP | 20 | 0.37/0.69 | 0.12/0.21 | 0.17/0.37 | 0.13/0.29 | 0.10/0.21 | 0.18/0.35 |
| Y-Net+TTST | 10000 | 0.28/0.33 | 0.10/0.14 | 0.24/0.41 | 0.17/0.27 | 0.13/0.22 | 0.18/0.27 |
| CAGN | 20 | 0.41/0.65 | 0.13/0.23 | 0.32/0.54 | 0.21/0.38 | 0.16/0.33 | 0.25/0.43 |
| SIT | 20 | 0.39/0.61 | 0.13/0.22 | 0.29/0.49 | 0.19/0.31 | 0.15/0.29 | 0.23/0.38 |
| SBD | 20 | 0.32/0.54 | 0.10/0.17 | 0.15/0.32 | 0.12/0.25 | 0.09/0.18 | 0.16/0.29 |

proposed method outperforms competitive methods in terms of the average ADE and FDE under standard 20 samplings. Compared with the previous state-of-the-art-20-samplings method, i.e., BiTraP-NP [12], our algorithm achieves 11.1% and 17.1% relative improvements in terms of the average ADE and FDE.

**Performance on long-term trajectory prediction.** In order to further demonstrate the effectiveness of our scheme in alleviating the problem of error accumulation, we conduct additional experiments for the long-term trajectory prediction on ETH-UCY. Following the setting in [29], we keep the observed trajectory for 3.2 seconds (8 frames) and set the longer future trajectory to 6.4 seconds (16 frames), 8.0 seconds (20 frames) and 9.6 seconds (24 frames), respectively. As shown in Table 3, our method outperforms all baselines on all long-term prediction lengths by a big margin. For example, when the prediction horizon is extended to 9.6 seconds, our SBD is better than the second best rank method BiTraP-NP [12] by 26.9% in ADE, which is significant. The reason behind is that our framework benefits from the synchronous bi-directional prediction via mutual constraints from two opposite branches, endowing it capability to alleviate error accumulation in long-term trajectory prediction.

**Table 3.** Long-term prediction results on ETH-UCY in ADE/FDE. ‡ denotes that the results are from [29]. † represents that the results are reproduced with the official released code.

| | T=16 | T=20 | T=24 |
|---|---|---|---|
| | ADE/FDE | ADE/FDE | ADE/FDE |
| SGAN‡ | 2.16/3.96 | 2.40/4.52 | 2.79/4.66 |
| PECNet‡ | 2.89/2.63 | 3.02/2.55 | 3.16/2.53 |
| Social-STGCNN‡ | 0.54/1.05 | 0.71/1.30 | 0.92/1.76 |
| BiTraP-NP† | 0.29/0.57 | 0.38/0.74 | 0.52/1.07 |
| SIT | 0.49/1.01 | 0.55/1.12 | 0.68/1.22 |
| SBD | 0.22/0.41 | 0.30/0.54 | 0.38/0.71 |

**Table 4.** Ablation study of each component on the SDD dataset in ADE/FDE.

| | SD | viBD | sBD | ECN | ADE | FDE |
|---|---|---|---|---|---|---|
| group-1 | ✓ | × | × | × | 8.71 | 13.13 |
| group-2 | × | ✓ | × | × | 8.77 | 12.94 |
| group-3 | × | × | ✓ | × | 8.06 | 12.45 |
| group-4 | × | × | ✓ | ✓ | 7.78 | 11.97 |

### 4.3 Ablation Studies

In this subsection, we perform ablation experiments to explore the contribution of each component of our method. The results are detailed in Table 4. The "SD" denotes that model uses the single directional generator (backward trajectory generator) as the prediction module. The "viBD" indicates that using the vanilla independent bi-directional trajectory generator. The "sBD" represents the proposed synchronous bi-directional prediction based on a share motion characteristic. The "ECN" denotes the error compensation network. According to the results of group-1 and group-2 in Table 4, we observe that vanilla independent bi-directional prediction cannot effectively alleviate the error accumulation and improve pedestrian prediction. The results of group-1, group-2 and group-3 in Table 4 show that the proposed mutually constrained simultaneous bi-directional prediction by the synchronous bi-directional decoder could effectively alleviate the limitation of error accumulation and improve pedestrian prediction. Besides, the error compensation network can further reduce the positional deviation between the ground-truth and predicted trajectory as shown in group-3 and group-4 in Table 4.

### 4.4 Qualitative Evaluation

We conclude this section by conducting the qualitative comparisons. Due to the page limit, we only compare with the recent BiTraP-NP method [12], which also adopts a (sequential) bi-directional prediction. As shown in Fig. 4, we visualize the best-of-20 predicted trajectories of our SBD and BiTraP-NP in different real

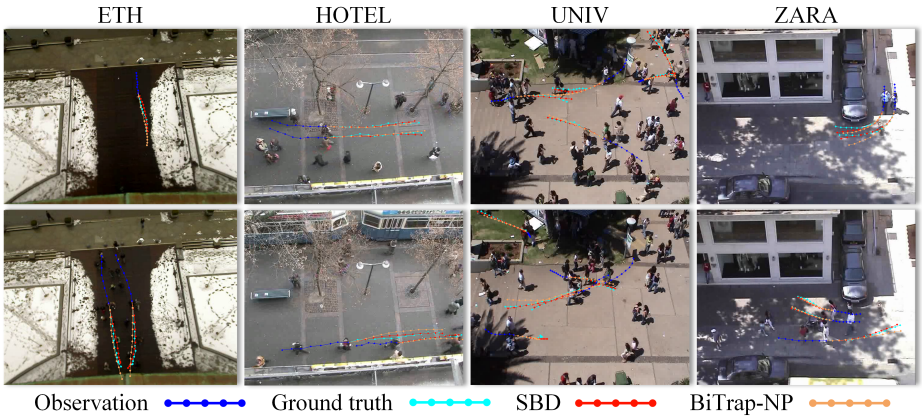


**Fig. 4.** Visualization of predicted trajectories on the ETH-UCY Dataset by our SBD and BiTrap-NP [12]. The best one of the 20 generated trajectories are plotted.

traffic scenes on the ETH-UCY datasets. We observe that our method is able to accurately predict the future trajectory in various traffic scenes. For example, the visualization results of the first row in Fig. 4 show that BiTraP-NP performs similar to SBD for short-term prediction yet a little deviates from the ground truth paths over time, and our SBD still exhibits better performance in longer prediction.

## 5 Conclusion

In this paper, we propose a synchronous bi-directional framework (SBD) with error compensation for pedestrian trajectory prediction. Our method performs the mutually constrained synchronous bi-directional prediction based on a shared motion characteristic, which can greatly alleviate the problem of error accumulation. Besides, we have introduced an error compensation network to reduce the spatial deviation for certain contexts in the predicted trajectory, further improving the prediction accuracy. Experimental results are provided to demonstrate the superiority of our method on Stanford Drone Dataset and ETH-UCY. Furthermore, we have also shown that our method with alleviating error accumulation performs significantly better than existing algorithms for long-term pedestrian trajectory prediction.

**Acknowledgements** This work was supported in part by the Natural Science Foundation of China under Grant 62001304, Grant 61871273, Grant 61901237 and Grant 62171244; in part by the Foundation for Science and Technology Innovation of Shenzhen under Grant RCBS20210609103708014, the Guangdong Basic and Applied Basic Research Foundation under Grant 2022A1515010645 and the Shenzhen College Stability Support Plan (Key Project).

## References


1. 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. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5725–5734 (2019)
2. 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 **3**(4), 3418–3425 (2018)
3. Luber, M., Stork, J.A., Tipaldi, G.D., Arras, K.O.: People tracking with human motion predictions from social forces. In: Proceedings of the IEEE International Conference on Robotics and Automation. pp. 464–469 (2010)
4. Bastani, V., Marcenaro, L., Regazzoni, C.S.: Online nonparametric bayesian activity mining and analysis from surveillance video. IEEE Transactions on Image Processing **25**(5), 2089–2102 (2016)
5. Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. Physical review E **51**(5), 4282 (1995)
6. Tay, M.K.C., Laugier, C.: Modelling smooth paths using gaussian processes. In: Proceedings of the International Conference on Field and Service Robotics. pp. 381–390 (2008)
7. Treuille, A., Cooper, S., Popović, Z.: Continuum crowds. ACM Transactions on Graphics (TOG) **25**(3), 1160–1168 (2006)
8. Pellegrini, S., Ess, A., Schindler, K., Van Gool, L.: You'll never walk alone: Modeling social behavior for multi-target tracking. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 261–268 (2009)
9. Yuan, Y., Weng, X., Ou, Y., Kitani, K.M.: Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9813–9823 (2021)
10. Yu, C., Ma, X., Ren, J., Zhao, H., Yi, S.: Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In: Proceedings of the European Conference on Computer Vision. pp. 507–523 (2020)
11. Salzmann, T., Ivanovic, B., Chakravarty, P., Pavone, M.: Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In: Proceedings of the European Conference on Computer Vision. pp. 683–700 (2020)
12. Yao, Y., Atkins, E., Johnson-Roberson, M., Vasudevan, R., Du, X.: Bitrap: Bi-directional pedestrian trajectory prediction with multi-modal goal estimation. IEEE Robotics and Automation Letters **6**(2), 1463–1470 (2021)
13. Li, Y., Liang, R., Wei, W., Wang, W., Zhou, J., Li, X.: Temporal pyramid network with spatial-temporal attention for pedestrian trajectory prediction. IEEE Transactions on Network Science and Engineering (2021)
14. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social lstm: Human trajectory prediction in crowded spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 961–971 (2016)
15. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social gan: Socially acceptable trajectories with generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2255–2264 (2018)
16. Fragkiadaki, K., Levine, S., Felsen, P., Malik, J.: Recurrent network models for human dynamics. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4346–4354 (2015)

17. Zhang, P., Ouyang, W., Zhang, P., Xue, J., Zheng, N.: Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 12085–12094 (2019)
18. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
19. Mohamed, A., Qian, K., Elhoseiny, M., Claudel, C.: Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 14424–14432 (2020)
20. Kosaraju, V., Sadeghian, A., Martín-Martín, R., Reid, I., Rezatofighi, H., Savarese, S.: Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. In: Proceedings of the Advances in Neural Information Processing Systems **32** (2019)
21. Shi, L., Wang, L., Long, C., Zhou, S., Zhou, M., Niu, Z., Hua, G.: Sgcn: Sparse graph convolution network for pedestrian trajectory prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8994–9003 (2021)
22. Su, Y., Du, J., Li, Y., Li, X., Liang, R., Hua, Z., Zhou, J.: Trajectory forecasting based on prior-aware directed graph convolutional neural network. IEEE Transactions on Intelligent Transportation Systems (2022)
23. Duan, J., Wang, L., Long, C., Zhou, S., Zheng, F., Shi, L., Hua, G.: Complementary attention gated network for pedestrian trajectory prediction (2022)
24. 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. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1349–1358 (2019)
25. Dendorfer, P., Osep, A., Leal-Taixé, L.: Goal-gan: Multimodal trajectory prediction based on goal position estimation. In: Proceedings of the Asian Conference on Computer Vision (2020)
26. 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: Proceedings of the European Conference on Computer Vision. pp. 759–776 (2020)
27. Chen, G., Li, J., Zhou, N., Ren, L., Lu, J.: Personalized trajectory prediction via distribution discrimination. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 15580–15589 (2021)
28. Wang, C., Wang, Y., Xu, M., Crandall, D.: Stepwise goal-driven networks for trajectory prediction. IEEE Robotics and Automation Letters (2022)
29. Shi, L., Wang, L., Long, C., Zhou, S., Zheng, F., Zheng, N., Hua, G.: Social interpretable tree for pedestrian trajectory prediction (2022)
30. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Proceedings of the Advances in Neural Information Processing Systems pp. 5998–6008 (2017)
31. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
32. Robicquet, A., Sadeghian, A., Alahi, A., Savarese, S.: Learning social etiquette: Human trajectory understanding in crowded scenes. In: Proceedings of the European Conference on Computer Vision. pp. 549–565 (2016)
33. Lerner, A., Chrysanthou, Y., Lischinski, D.: Crowds by example. In: Computer graphics forum. vol. 26, pp. 655–664 (2007)

34. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
35. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016)
36. Huang, Y., Bi, H., Li, Z., Mao, T., Wang, Z.: Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6272–6281 (2019)
37. Liang, R., Li, Y., Li, X., Tang, Y., Zhou, J., Zou, W.: Temporal pyramid network for pedestrian trajectory prediction with multi-supervision. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 2029–2037 (2021)
38. Wang, Y., Chen, S.: Multi-agent trajectory prediction with spatio-temporal sequence fusion. IEEE Transactions on Multimedia (2021)
39. Pang, B., Zhao, T., Xie, X., Wu, Y.N.: Trajectory prediction with latent belief energy-based model. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11814–11824 (2021)
40. Sun, J., Li, Y., Fang, H.S., Lu, C.: Three steps to multimodal trajectory prediction: Modality clustering, classification and synthesis. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 13250–13259 (2021)
41. Mangalam, K., An, Y., Girase, H., Malik, J.: From goals, waypoints & paths to long term human trajectory forecasting. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 15233–15242 (2021)
42. Gu, T., Chen, G., Li, J., Lin, C., Rao, Y., Zhou, J., Lu, J.: Stochastic trajectory prediction via motion indeterminacy diffusion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 17113–17122 (2022)