Sparse Instance Conditioned Multimodal Trajectory Prediction

Yonghao Dong[1] Le Wang[1*] Sanping Zhou[1] Gang Hua[2]
[1]National Key Laboratory of Human-Machine Hybrid Augmented Intelligence,
National Engineering Research Center for Visual Information and Applications,
and Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
[2]Wormpex AI Research

Abstract

Pedestrian trajectory prediction is critical in many vision tasks but challenging due to the multimodality of the future trajectory. Most existing methods predict multimodal trajectories conditioned by goals (future endpoints) or instances (all future points). However, goal-conditioned methods ignore the intermediate process and instance-conditioned methods ignore the stochasticity of pedestrian motions. In this paper, we propose a simple yet effective Sparse Instance Conditioned Network (SICNet), which gives a balanced solution between goal-conditioned and instance-conditioned methods. Specifically, SICNet learns comprehensive sparse instances, i.e., representative points of the future trajectory, through a mask generated by a long short-term memory encoder and uses the memory mechanism to store and retrieve such sparse instances. Hence SICNet can decode the observed trajectory into the future prediction conditioned on the stored sparse instance. Moreover, we design a memory refinement module that refines the retrieved sparse instances from the memory to reduce memory recall errors. Extensive experiments on ETH-UCY and SDD datasets show that our method outperforms existing state-of-the-art methods. In addition, ablation studies demonstrate the superiority of our method compared with goal-conditioned and instance-conditioned approaches.

1. Introduction

Pedestrian trajectory prediction aims to predict pedestrians' future paths given their observed trajectories. It plays an important role in autonomous driving [22, 43, 5], human motion prediction [6, 12, 19], video surveillance [57, 48, 18, 32], and visual recognition [38, 42, 46]. Despite the recent advancement [30, 49, 8, 36, 10], pedestrian trajectory prediction is still challenging due to the multimodality of future trajectories. As the pedestrian's motions are stochastic and indeterminate [29], multiple future paths are possible given their current states.

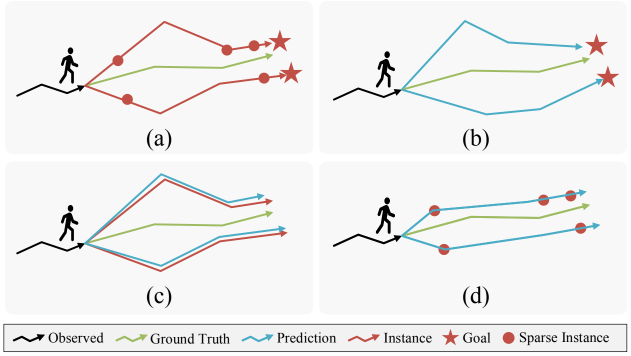


Figure 1. Illustration of different multimodal prediction methods. **(a)** Goals, instances and sparse-instances prepared for the prediction. **(b)** *Goal-conditioned* methods. **(c)** *Instance-conditioned* methods. **(d)** *Sparse-instance-conditioned* methods. The sparse-instance-conditioned method is more accurate in in-between states than the goal-conditioned method and more flexible than the instance-conditioned method, thus making the better prediction.

To handle multimodality, many methods [27, 49, 55] transform the multimodal trajectories into multimodal goals. They first predict various goals as modality representations and then sample multiple goals to generate diverse future trajectories. However, they only focus on goals while neglecting the intermediate movement process. As illustrated in Figure 1(b), although with accurate goals, the intermediate process could be far from the ground truth. Thus, the multimodal goals are not equal to the multimodal future trajectories. It is necessary to consider the in-between states to comprehensively describe the multimodality of trajectory. Moreover, some other work [27, 28] predicts multimodal trajectories conditioned by instances. However, the instance-conditioned methods will lead to motion stochasticity loss and modality redundancy, resulting in performance degradation and computational resource waste. As shown in Figure 1(c), due to the reduced randomness of the instance-conditioned approach, the prediction retains the same bias

[*] Corresponding author.

when the instance deviates from the ground truth.

To close this gap, we present a Sparse Instance Conditioned Network (SICNet). In our model, a sparse instance, which can give a comprehensive and adaptive description of future modality, is constructed by element-wise multiplication between the future trajectory and a learned mask. Hence, SICNet can predict the future trajectory more precisely guided by the sparse instance. To obtained the related sparse instancs during inference, we adopt the memory mechanism to store non-redundant sparse instances corresponding to all observed trajectories. Therefore, SICNet can recall its modality's most related sparse instance and jointly generate the prediction during inference. Moreover, when the memory meets an unfamiliar observed trajectory, *i.e.*, a trajectory quite different from stored items, the memory mechanism may recall false experiences. Hence we propose a memory refinement module with a refinement loss to reduce such recall errors by bridging the gap between the recall and the ground truth. For multimodal trajectories generation, we can retrieve a certain number of the most likely sparse instances from the external memory and then generate corresponding multimodal trajectories. In addition, to further remove the redundancy from the retrieved sparse instances, we use the cluster mechanism as a post-process to obtain succinct multimodal proposals for the prediction instead of the sparse instance during inference.

Extensive experiments and ablation studies on two popular pedestrian trajectory prediction datasets, *i.e.*, ETH-UCY [31, 21] and Stanford Drone Dataset (SDD) [33], demonstrate the superiority of our method over prior state-of-the-art methods. In summary, our contributions are four-fold: (1) We give a balanced solution, *i.e.*, a novel sparse instance, between the goal and instance to guide the multimodal trajectory prediction. (2) We propose a simple yet effective sparse instance conditioned network for the multimodal trajectory prediction. (3) We propose a memory refinement module for the external memory to reduce recall errors when meeting unfamiliar observed trajectories. (4) Extensive experiments and ablation studies demonstrate the superiority and flexibility of our method compared with the goal-conditioned and instance-conditioned approaches.

2. Related Works

Multimodal Pedestrian Trajectory Prediction. Pedestrian trajectory prediction [2, 23, 3, 39, 45] aims to predict pedestrians' future trajectories based on their observed trajectories. There are some early studies on the deterministic trajectory prediction [1, 44, 14, 50]. Due to the strong randomness and uncertainty of pedestrian motions, there is no single correct future trajectory prediction. Motivated by this, Social GAN [13] proposes the concept of multimodal pedestrian trajectory prediction and emphasizes its importance.

Earlier work generates multimodal trajectories through latent variables. Social GAN [13] proposes a Generative Adversarial Network based on a social pooling mechanism to generate a multimodal trajectory distribution. SGCN [37] generates multimodal trajectories by modeling the future predictions as a bi-variate Gaussian distribution. The CVAE-based methods [15, 35, 8] generate multimodal trajectories by sampling latent variables from a learned latent space. However, the latent variable conditioned method is persecuted by the problem of model collapse, thus resulting in performance degradation. Moreover, the latent variable conditioned methods also suffer in interpretability.

To address the model collapse and interpretability issues, PECNet [27] proposes a goal-conditioned approach, which transforms multimodal trajectories into multimodal goals and then samples multiple goals to generate corresponding diverse predictions. After that, amounts of goal-conditioned studies [55, 8, 49] emerge and show superior performance. However, goal-conditioned methods represent different modalities with different possible goals while neglecting the intermediate states, thus suffering in performance improvement. In addition, PCCSNet[27] and MANTRA [28] propose instance-conditioned approaches, which use all points of the future instance to describe each modality precisely. However, it could lead to stochasticity loss and modality redundancy, resulting in performance degradation and computational resource waste. Moreover, Y-net [26] use goals and waypoints to make the prediction. However, the waypoints' number and time step locations are manually determined in Y-net [26], which may decrease the prediction accuracy by manually choosing errors.

In contrast, we give a balanced way, *i.e.*, a novel sparse instance, between the goal and instance to guide the multimodal trajectory prediction. The sparse instance is constructed by the learned representative points of the instance, which can improve modality representation accuracy compared with goal-conditioned methods and preserve stochasticity compared with instance-conditioned methods.

Memory Network. The memory-augmented network [51, 11, 25, 24, 53] uses an external memory module to store critical information and has been widely applied in many areas like detection [7, 4], tracking [52, 9], segmentation [47], image generation [17], and video summarization [20]. Memory GAN [17] proposed a novel end-to-end unsupervised GAN network with a learnable memory network that effectively learns a highly multimodal latent space without suffering from structural discontinuity and forgetting problems. OGEMN [7] proposes the first object-guided external memory network for the online video object detection. STMTrack [52] proposes a novel tracking framework built on a space-time memory network, which can fully use historical information related to the target for better adapting to appearance variations during tracking. RMNet [47] memorizes the target object regions, effectively alleviating

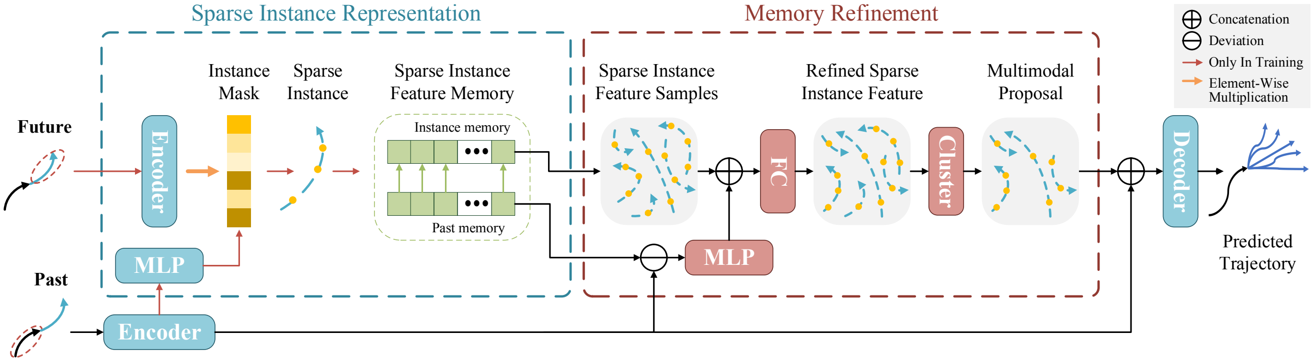


Figure 2. The framework of our proposed SICNet. SICNet adopts a two-stage training strategy. The sparse instance representation module generates the sparse instances by a learned mask and then trains an external memory to store and retrieve them. The memory refinement module refines the retrieved sparse instances from memory to reduce the searching errors. We cluster the refined sparse instances into multimodal proposals to remove redundancy further. The cluster only exists in inference to save training costs.

objects' ambiguity and reducing computational complexity.

Two previous memory-based work, MANTRA [28] and MemoNet [49], also use the memory mechanism in multimodal trajectory prediction. However, the difference are mainly in three aspects: (1) The contribution of our proposed SICNet focuses on a novel representation of the trajectory modality, *i.e.*, a learned sparse instance, while MANTRA [28] uses the whole instance and MemoNet [49] uses the goal to represent modalities of multimodal trajectories; (2) We propose a memory refinement module to reduce recall errors when meeting unfamiliar observed trajectories during inference, while MANTRA [28] and MemoNet [49] ignore such circumstances. (3) To demonstrate the superiority of our proposed sparse instance, we do not use extra social interaction and map information, while MANTRA [28] use extra map information and MemoNet [49] use social interaction information. Despite this, we also achieve the best performance compared with the above two methods, which indicates the superiority of our method.

3. Our Method

As aforementioned, traditional goal-conditioned methods ignore the intermediate process and instance-conditioned methods ignore the stochasticity of pedestrian motions, which leads to performance degradation. In this section, we present a sparse-instance-conditioned network (SICNet) to close this gap. Our SICNet adopts a two-stage training strategy. In the first stage, we learn the sparse instance, which is used to reconstruct the future trajectory jointly with the observed trajectory. Meanwhile, an external memory is trained to store and retrieve the sparse instance. In the second stage, we refine the retrieved sparse instances from the external memory through a refinement module with a refinement loss to reduce the recall errors. In addition, we use the cluster mechanism to remove redundancy of the retrieved sparse instances further during inference.

3.1. Problem Formulation

The objective of pedestrian trajectory prediction is to predict possible future trajectory coordinates based on the observed trajectories. For each target pedestrian, given an observed trajectory $X = \{(x_t, y_t)\}_{t=1}^{T_{obs}}$, where (x_t, y_t) is the 2D coordinate at time t, we aim to predict the future trajectory coordinates $Y = \{(x_t, y_t)\}_{t=T_{obs}+1}^{T_{pred}}$. Note that multimodal pedestrian trajectory prediction requires the model to predict K future trajectories $\{Y_i\}_{i=1}^{K}$ to account for multimodality, while only one future trajectory (ground truth) is provided for training in the dataset.

3.2. Sparse Instance Representation

In the first training stage, we generate the sparse instance features to represent future modalities and store them by memory mechanism. As shown in Figure 3, we first train the encoder-decoder with a reconstruction loss in this stage. Then we train the memory module to store the sparse instances generated during reconstruction.

Feature Encoder. As the trajectory is a temporal sequence, we use two LSTM [56] blocks to build the past encoder $f_p(\cdot)$ and the future encoder $f_y(\cdot)$, respectively. The hidden states of the two encoders are used as their corresponding features, as follows:

$$\begin{aligned} \mathbf{F}_p &= f_p(X), \\ \mathbf{F}_y &= f_y(Y), \end{aligned} \tag{1}$$

where $\mathbf{F}_p \in \mathbb{R}^{1\times d}$ is the past feature extracted from the observed trajectory X, and $\mathbf{F}_y \in \mathbb{R}^{1\times d}$ is the future feature extracted from the future trajectory Y. d is the dimension of hidden state in LSTM.

Sparse Instance. When inputting an observed trajectory, we want to obtain the corresponding sparse instance of its future modality so that we can construct future trajectories jointly using the observed trajectory and the sparse instance.

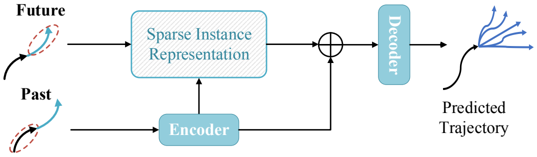


Figure 3. The first stage of the training process. In this stage, we train the sparse instance by reconstructing the future trajectory. Then we train an external memory to store and retrieve the learned sparse instances.

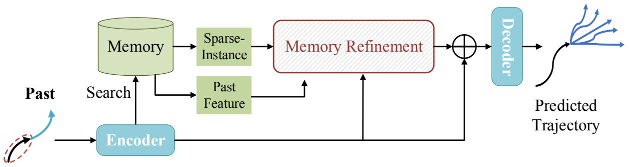


Figure 4. The second stage of the training process. In this stage, we refine the retrieved sparse instances from memory through the memory refinement module to reduce the recall errors.

Hence, we first learn an instance mask from the past feature $\mathbf{F}_p$ through an MLP layer, as follows:

$$\begin{aligned} \mathbf{F}_m &= \phi(\mathbf{F}_p, \theta_M), \\ M &= \mathbb{I}(\delta(\mathbf{F}_m) \le \mu)), \end{aligned} \tag{2}$$

where $\phi(\cdot,\cdot)$ denotes the MLP layer. θ_M is a learnable parameter. $\mathbf{F}_m$ is the learned feature which is used to construct the mask. M denotes the instance mask. $\delta(\cdot)$ is sigmoid function to map $\mathbf{F}_m$ to the mask value $[0,1]$. $\mathbb{I}$ is an indicator function which equals 0 if the inequality holds, otherwise 1. μ is the mask threshold. Compared with directly masking instances, masking instances in the high-level feature dimension is more efficient and effective. Hence we generate the sparse instance $\mathbf{F}_{ins}$ in the feature level dimension, as:

$$\mathbf{F}_{ins} = \mathbf{F}_y \odot M, \tag{3}$$

where $\odot$ denotes the element-wise multiplication.

Feature Decoder. The sparse instance feature should be representative enough for the future trajectory. In addition, our goal is to predict the future trajectory based on the observed trajectory conditioned on the sparse instance. Therefore, we use an LSTM [56] decoder to reconstruct the future trajectory with the combination of the past feature and the sparse instance feature, as follows:

$$Y_r = f_d(\mathbf{F}_p \oplus \mathbf{F}_{ins}), \tag{4}$$

where $f_d(\cdot)$ is the decoder implemented by an LSTM block, $\oplus$ is a concatenation operation, and Y_r is the reconstructed future trajectory. The trajectory ℓ_2-norm loss is used as the reconstruction loss, as:

$$\mathcal{L}_1 = \|Y - Y_r\|_2, \tag{5}$$

where $||\cdot||_2$ denotes the ℓ_2-norm distance.

Memory Architecture. The key idea of the memory module is to use the encoding of the observed trajectory as a memory key to retrieve possible sparse instances and jointly generate the multimodal predictions. To store the sparse instances generated during the reconstruction process, we first randomly initializes two memory banks, *i.e.*, the past memory bank $\mathbf{M}_p \in \mathbb{R}^{m\times d}$, and the instance memory bank $\mathbf{M}_i \in \mathbb{R}^{m\times d}$ to store the past and the sparse instance features obtained from the reconstruction process, respectively. m denotes the storage size of the memory bank, and d is the dimension of the memory item. To combine the past and sparse instance features, the items in these two memory banks are aligned. Specifically, for the j-th past feature in $\mathbf{M}_p[j]$, there is a unique sparse instance feature in $\mathbf{M}_i[j]$ corresponding to it.

Memory Update. Due to the abundant similar trajectories in the dataset, it brings high redundancy to store all of them in the memory bank. Hence we need to remove modality redundancy by memory updating process. Moreover, traditional memory mechanism [16] updates the memory bank with the stored items' labels, while such labels are not provided in trajectory prediction to distinguish trajectories with different modalities. Therefore, we design an unsupervised memory update strategy based on the distance between the stored and current items, as shown in Figure 5. Inspired by [16], an additional mark memory bank $\mathbf{A} \in \mathbb{R}^{m\times 1}$ tracks the age of items stored in memory without being used. All items in $\mathbf{A}$ are initialized to 1 and aligned to their corresponding items in $\mathbf{M}_p$ and $\mathbf{M}_i$. Given a pair of inputs $(\mathbf{F}_p, \mathbf{F}_{ins})$, we first calculate the feature similarities between $\mathbf{F}_p$ and each item in $\mathbf{M}_i$. Then, the k-th ($k \in \{1,...,m\}$) item $\mathbf{M}_p[k]$ with the highest similarity is used to retrieve the corresponding sparse instance feature, *i.e.*, $\mathbf{M}_i[k]$, through the one-to-one architecture of memory banks.

Specifically, the distance $\mathcal{D}$ between $\mathbf{F}_{ins}$ and $\mathbf{M}_i[k]$ is defined as:

$$\mathcal{D} = ||\mathbf{M}_i[k] - \mathbf{F}_{ins}||_2. \tag{6}$$

We compare $\mathcal{D}$ with a threshold ξ to judge whether F_{ins} and $\mathbf{M}_i[k]$ have the same modality. When $\mathcal{D} > \xi$, which means that F_{ins} and $\mathbf{M}_i[k]$ have different modalities. Hence we update $\mathbf{M}_p$, $\mathbf{M}_i$ with the oldest age (*i.e.*, corresponding $\mathbf{A}$ with the maximum value) by $\mathbf{F}_p$, $\mathbf{F}_{ins}$, respectively, as follows:

$$\mathbf{M}_i[s] \leftarrow \mathbf{F}_{ins}, \quad \mathbf{M}_p[s] \leftarrow \mathbf{F}_p, \quad \mathbf{A}[s] \leftarrow 0, \tag{7}$$

where s is the index with the oldest age. Since the s-th item in the past memory bank has been updated, the s-th item in $\mathbf{A}$ is set to 0 to prevent the newly added item from overwriting it. When $\mathcal{D} \le \xi$, which means that F_{ins} and $\mathbf{M}_i[k]$ have the same modality. Hence we only update $\mathbf{M}_p[k]$ to approach

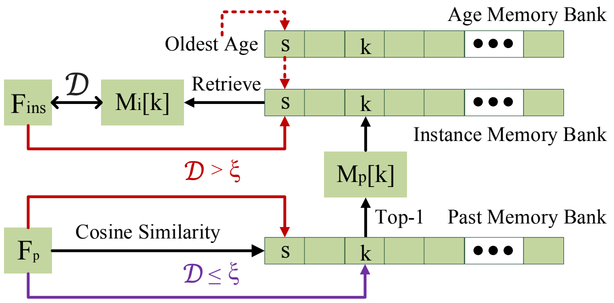


Figure 5. An illustration of the memory training process in SICNet. Red and purple lines denote the memory writing process.

$\mathbf{F}_p$ by taking the normalized average of $\mathbf{M}_p[k]$ and $\mathbf{F}_p$, as:

$$\mathbf{M}_p[k] \leftarrow \frac{\mathbf{F}_p + \mathbf{M}_p[k]}{\|\mathbf{F}_p + \mathbf{M}_p[k]\|}, \quad \mathbf{A}[k] \leftarrow 0. \tag{8}$$

Thus, we can retrieve the sparse instance feature $\mathbf{M}_i[k]$ easily, when we receive the past feature similar to $\mathbf{F}_p$. Since the k-th item in the past memory bank has been updated, the k-th item in $\mathbf{A}$ is set to 0 to prevent the newly added item from covering it. All values in $\mathbf{A}$ will add the same constant after an epoch.

3.3. Memory Refinement

The input observed trajectory encoding could be unfamiliar to the well-trained memory, which could be caused by: (1) modality loss due to removing redundancy during the memory update process; (2) the input modality does not exist in the train set. Such unfamiliarity could lead to the recalled sparse instance not matching the input trajectory. To reduce such recall errors, we design a memory refinement module with a refinement loss in the second training stage, as shown in Figure 4.

We first consider the difference between the past feature $\mathbf{F}_p$ and its corresponding closest past memory item $\mathbf{M}_p[k]$ with the same modality. To avoid extra optimization similar to Eq. (8), we do not optimize $\mathbf{F}_p$ to approach $\mathbf{M}_p[k]$ again but encode the difference between $\mathbf{F}_p$ and $\mathbf{M}_p[k]$ as an additional deviation feature $\mathbf{F}_v$, which is formulated as:

$$\mathbf{F}_v = \phi(\mathbf{F}_p \oplus \mathbf{M}_p[k], \theta_V), \tag{9}$$

where θ_V is a learnable parameter, and $\oplus$ is the concatenation operation. Then, the retrieved sparse instance feature is refined as $\mathbf{F}_r$, as follows:

$$\mathbf{F}_r = \psi(\mathbf{F}_v \oplus \mathbf{M}_i[k], \theta_R), \tag{10}$$

where $\psi(\cdot,\cdot)$ is a linear projection with a learnable parameter θ_R, $\mathbf{F}_r$ is the refined sparse instance feature. In this case, the model can give a reasonable sparse instance considering the difference between the input observed feature and the stored observed feature in the memory bank. The loss function of the memory refinement module is:

$$\mathcal{L}_2 = \|\mathbf{F}_{ins} - \mathbf{F}_r\|_2. \tag{11}$$

Finally, the past feature $\mathbf{F_p}$ is concatenated with the refinement feature $\mathbf{F}_r$ to predict the future trajectory. Concretely, we use the pre-trained decoder $f_d(\cdot)$ in the first stage to achieve the prediction, as follows:

$$\hat{Y} = f_d(\mathbf{F}_p \oplus \mathbf{F}_r), \tag{12}$$

where $\hat{Y}$ is the predicted future trajectory.

Clustering. We adopt the clustering mechanism during inference to further remove the modality redundancy of the retrieved sparse instances. By retrieving C ($C >> k$) sparse instances corresponding to the past trajectory from the well-trained memory, we can generate C refined sparse instances through the refinement network. Then we cluster the C refined sparse instances into K multimodal proposals. The clustered multimodal proposals will guide the multimodal trajectory prediction instead of sparse instances in inference. Ablation studies verify the effectiveness of this post-process.

3.4. Model Training & Inference

Training. SICNet is trained by a two-stage strategy. In the first stage, we train the encoders $f_p(\cdot)$, $f_y(\cdot)$ and the decoder $f_d(\cdot)$ in the sparse instance representation module with the reconstruction loss $\mathcal{L}_1$. Then we use the memory mechanism to store the sparse instances generated during the reconstruction process. In the second stage, we fix the parameters of $f_p(\cdot)$, $f_y(\cdot)$ and $f_d(\cdot)$, and train the memory refinement module with the refinement loss $\mathcal{L}_2$.

Inference. We aim to predict K trajectories to cover the multimodality of future trajectories. During inference, we select the top-C past memory items by feature similarity and then retrieve corresponding C sparse instance items from the instance memory bank, where $C >> K$ is to improve the prediction diversity. Then we obtained C refinement sparse instances through the memory refinement module. Subsequently, the C sparse instance items are clustered into K multimodal proposals. Finally, we feed the combination of the past feature and multimodal proposals into the decoder to generate K predicted future trajectories.

4. Experiments and Discussions

4.1. Experimental Setting

Evaluation Datasets. We evaluate our proposed method on two benchmark datasets, *i.e.*, ETH-UCY [31, 21] and Stanford Drone Dataset (SDD) [33]. ETH-UCY is the most commonly used pedestrian trajectory prediction dataset captured in the bird's eye view. It contains five subsets, where the ETH [31] includes ETH and HOTEL subsets, and the

Method	Venue/Year	Input	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
Social LSTM [1]	CVPR2016	P+S	1.09/2.35	0.79/1.76	0.67/1.40	0.47/1.00	0.56/1.17	0.72/1.54
Social GAN [13]	CVPR2018	P+S	0.87/1.62	0.67/1.37	0.76/1.52	0.35/0.68	0.42/0.84	0.61/1.21
SoPhie [34]	CVPR2019	P+S	0.70/1.43	0.76/1.67	0.54/1.24	0.30/0.63	0.38/0.78	0.51/1.15
MANTRA [28]	CVPR2020	P+S	0.48/0.88	0.17/0.33	0.37/0.81	0.27/0.58	0.30/0.67	0.32/0.65
PECNet [27]	ECCV2020	P+S	0.54/0.87	0.18/0.24	0.35/0.60	0.22/0.39	0.17/0.30	0.29/0.48
SGCN [37]	CVPR2021	P+S	0.63/1.03	0.32/0.55	0.37/0.70	0.29/0.53	0.25/0.45	0.37/0.65
AgentFormer [54]	ICCV2021	P+S	0.45/0.75	0.14/0.22	0.25/0.45	**0.18/0.30**	0.14/**0.24**	0.23/0.39
PCCSNet [41]	ICCV2021	P	0.28/0.54	**0.11**/0.19	0.29/0.60	0.21/0.44	0.15/0.34	0.21/0.42
CAGN [8]	AAAI2022	P+S	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 [36]	AAAI2022	P+S	0.39/0.62	0.14/0.22	0.27/0.47	0.19/0.33	0.16/0.29	0.23/0.38
MemoNet [49]	CVPR2022	P+S	0.40/0.61	**0.11**/0.17	**0.24/0.43**	**0.18**/0.32	0.14/**0.24**	0.21/0.35
Ours	-	P	**0.27/0.45**	**0.11/0.16**	0.26/0.46	0.19/0.33	**0.13**/0.26	**0.19/0.33**

Table 1. Comparison with state-of-the-art methods on ETH-UCY in ADE/FDE. P and S indicate inputting the observed history points and social interaction information, respectively. All methods input the observed 8 time steps and output the predicted 12 time steps. Bold indicates the best performance. Underline indicates the second best performance. The Lower the better.

Method	Venue/Year	Input	ADE/FDE
Social LSTM [1]	CVPR2016	P+S	31.19/56.97
Social GAN [13]	CVPR2018	P+S	27.23/41.44
SoPhie [34]	CVPR2019	P+S	16.27/29.38
MANTRA [28]	CVPR2020	P+S	8.96/17.76
PECNet [27]	ECCV2020	P+S	9.96/15.88
PCCSNet [41]	ICCV2021	P	8.62/16.16
CAGN [8]	AAAI2022	P+S	9.42/15.93
SIT [36]	AAAI2022	P+S	9.13/15.42
MemoNet [49]	CVPR2022	P+S	8.56/**12.66**
Ours	-	P	**8.44**/13.65

Table 2. Comparison with state-of-the-art methods on SDD in ADE/FDE. P and S indicate inputting the observed history points and social interaction information, respectively. All methods input the observed 8 time steps and output the predicted 12 time steps. The Lower the better.

UCY [21] includes UNIV, ZARA1, and ZARA2 subsets. On ETH-UCY, we use the leave-out-one method for training on four subsets and testing on the other one following previous efforts [40, 54, 36]. SDD [33] is a large pedestrian trajectory prediction dataset also captured in bird's eye view. It contains 20 scenes of more than 10,000 trajectories collected from college campuses and is much larger than ETH-UCY. On SDD, we use a prior train-test split for training and testing according to previous methods [13, 27, 36]. We observe the historical trajectories of 8 time steps and predict the subsequent 12 time steps on both ETH-UCY and SDD.

Evaluation Metrics. Following previous methods [49, 36, 8, 41, 27], we employ two commonly used metrics, *i.e.*, Average Displacement Error (ADE) and Final Displacement Error (FDE), to evaluate the trajectory prediction performance. ADE measures the average ℓ_2-norm distance between all points of the ground truth and predicted trajectory. FDE measures the ℓ_2-norm distance between the destination points of the ground truth and predicted trajectory. We use the best-of-20 metrics following previous methods [49, 41].

Implementation Details. In our experiments, the two encoders are all 3-layer BiLSTM [56] architecture with hidden sizes of 48. The decoder is 3-layer BiLSTM [56] architecture with hidden size of 96. The dimensions of θ_M, θ_V and θ_R are set to 48, 48 and 96, respectively. The parameter μ of mask threshold is 0.2. The parameter ξ in memory training is 0.0001. The Adam optimizer is used to train our model by 100 epochs with a learning rate of 0.0005, decaying by 0.5 with an interval of 50.

4.2. Quantitative Analysis

On ETH-UCY. As shown in Table 1, we compare our method with eleven state-of-the-art methods on the ETH-UCY dataset in past six years. Among them, MemoNet [49], PECNet [27], and CAGN [8] are goal-conditioned methods. MANTRA [28] and PCCSNet [41] are instance-conditioned methods. SoPhie [34], SGCN [37] and AgentFormer [54] generate multimodal trajectories by latent variables. The experiment results show that generating multimodal predictions by goals or instance is better than by latent variables.

Moreover, the results indicate that our method significantly outperforms all the competing methods on both ETH and UCY. Compared with the previous best method MemoNet [49], our method further improves the performance on ADE and FDE. Meanwhile, Compared with the other instance-conditioned methods, PCCSNet [41], and MANTRA [28], our method also achieve the best performance. The underlying reason could be that our method captures the effective intermediate process compared with

Method	K=15	K=10	K=5
PECNet [27]	10.66/17.85	11.79/21.00	14.49/28.08
PCCSNet [41]	9.27/17.62	10.36/20.12	**12.57**/24.74
CAGN [8]	10.01/17.49	11.06/19.89	13.85/26.87
MemoNet [49]	9.62/15.36	11.26/19.64	14.43/27.45
Ours	**9.13/15.31**	**10.27/18.13**	12.82/**23.96**

Table 3. Experiments on different best-of-K predictions on SDD in ADE/FDE. The lower the better.

Method	SI	REF	CL	ADE/FDE
Baseline				0.46/0.94
(1)		✓	✓	0.25/0.50
(2)	✓		✓	0.27/0.56
(3)	✓	✓		0.22/0.39
(4)	✓	✓	✓	**0.19/0.33**

Table 4. Ablation study of each component of our method on ETH-UCY dataset in ADE/FDE. The lower the better.

the goal-conditioned methods and is more flexible compared with the instance-conditioned methods.

To show the superiority of our method more clearly, we do not use extra information, such as social interaction and map information, as previous approaches. Nevertheless, our method still achieves the best performance, clearly demonstrating the superiority of our method using sparse instance.

On SDD. We compare our method with nine state-of-the-art methods on the SDD dataset in Table 2. For ADE, our method outperforms all the competing methods. For FDE, our method also achieves comparable performance with the best method MemoNet [49] and exceeds all other methods. The results manifest that the sparse instance representation can comprehensively describe the trajectory multimodality rather than goal-conditioned methods and thus achieves competitive performance. We speculate that our method underperforms MemoNet [49] on FDE because the sparse instance benefits more on the whole trajectory prediction (*i.e.*, ADE) than goal prediction (*i.e.*, FDE).

4.3. Ablation Study

Different Best-of-K Predictions. Previous methods commonly use best-of-K (usually $K = 20$) as the quantified metric of multimodal trajectory prediction. To further validate the adaptability and effectiveness of our proposed sparse instance representation, we conduct an experiment on various best-of-K predictions with $K = 5, 10, 15$ on SDD. We compare our method with the top three performance goal-conditioned methods, *i.e.*, MemoNet [49], CAGN [8] and PECNet [27] and the best performance instance-conditioned method PCCSNet [41]. As shown in Table 3, our method achieves significant performance on all experimental settings. Interestingly, our method surpasses the previous best method MemoNet [49] when $K = 5, 10, 15$, despite underperforming MemoNet [49] when $K = 20$ in Table 2. It shows the adaptability of our method of using the sparse instance, which also performs significantly with a small K.

μ	0	0.1	0.2	0.3	0.4	0.5
ADE	0.21	0.21	**0.19**	0.20	0.20	0.21
FDE	0.34	0.33	**0.33**	0.33	0.34	0.37
μ	0.6	0.7	0.8	0.9	1.0	-
ADE	0.25	0.28	0.26	0.24	0.21	-
FDE	0.45	0.54	0.48	0.43	0.38	-

Table 5. Ablation study of different mask threshold values on the ETH-UCY dataset. The lower the better.

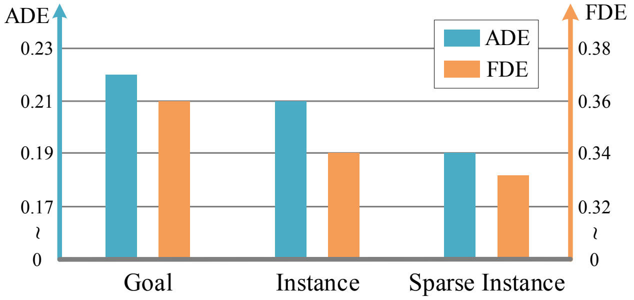
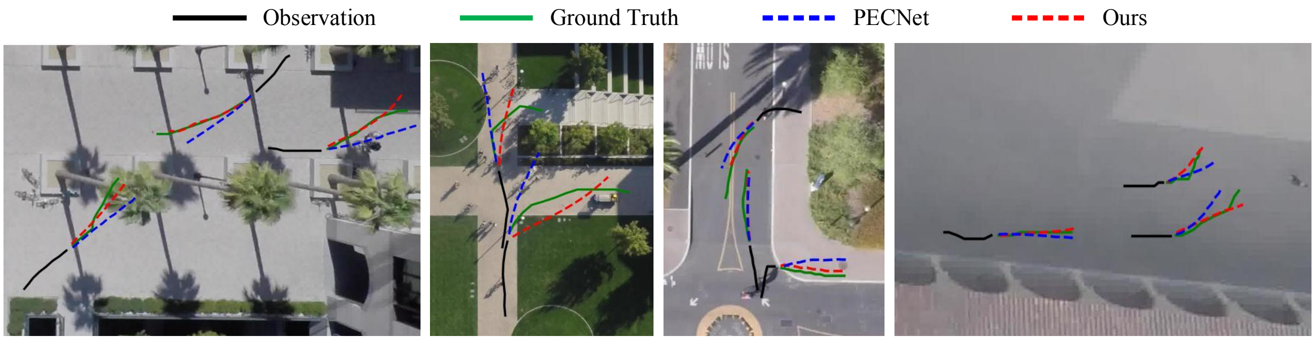


Figure 6. Experiments of the sparse instance analysis on the ETH-UCY dataset. We replace our proposed sparse instance with the goal and instance for the multimodal prediction.

Contribution of Each Component. We divide our proposed SICNet into three components, *i.e.*, the sparse instance representation (SI), the memory refinement module (REF), and the clustering post-process (CL). Ablation experiments of different combinations of the three components are evaluated on the ETH-UCY dataset. As shown in Table 4, all of the three components contribute to the performance improvement, which demonstrates their effectiveness.

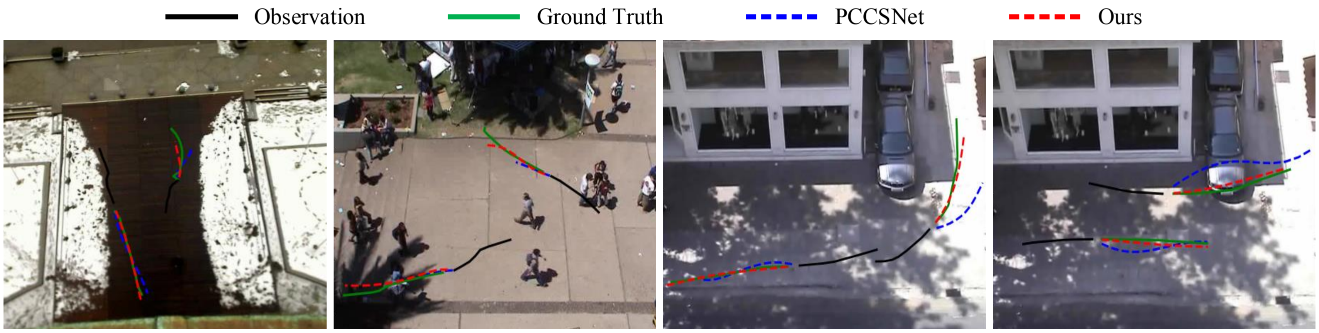
Analysis of Mask Threshold. As shown in Table 5, we conduct experiments for different mask threshold values μ on the ETH-UCY dataset. It indicates that the performance will decrease when the mask threshold value μ is too small or too large. When $\mu = 0$, the mask is useless and the whole instance is reserved, thus causing the stochasticity loss. When $\mu = 1$, it denotes the whole instance is masked, thus causing no guidance information for the multimodal prediction. When $\mu = 0.2$, it achieves the best performance.

Analysis of Sparse Instance. As shown in Table 6, we replace the sparse instance in SICNet with goal-conditioned and instance-conditioned representations on the ETH-UCY dataset. Experiment results show that using the sparse instance achieves the best performance, which indicates the



Coupa Hyang Little Quad

Figure 7. Qualitative comparison of our method with the goal-conditioned method PECNet [27] on the SDD dataset. Given the observed trajectories, we illustrate the ground truth paths and predicted trajectories by SICNet and the goal-conditioned method PECNet [27] for four different scenes. It shows our prediction is closer to the ground truth than PECNet [27], especially for the intermediate process.



ETH UNIV ZARA1 ZARA2

Figure 8. Qualitative comparison of our method with the instance-conditioned method PCCSNet [41] on the ETH-UCY dataset. Given the observed trajectories, we illustrate the ground truth paths and predicted trajectories by SICNet and the instance-conditioned method PCCSNet [41] for four different scenes. It shows our prediction is closer to the ground truth than PCCSNet [41].

superiority of our method compared with goal-conditioned and instance-conditioned approaches.

4.4. Qualitative Analysis

We compare the visualized trajectories with the well-accepted goal-conditioned method PECNet [27] on SDD in Figure 7, and the instance-conditioned method PCCSNet [41] on ETH-UCY in Figure 8. As shown in Figure 7, trajectories in Coupa show that our method achieves more precise trajectory predictions than PECNet. Trajectories in Little and Quad show that both PECNet and our method can predict destinations close to the ground truth, while our method can predict more precise intermediate trajectories due to our proposed sparse instance representation. Trajectories in Hyang show that our method can also give better predictions, although the ground truth trajectories are irregular. The results verify the superiority of our method compared with goal-conditioned methods. Moreover, as shown in Figure 8, the visualization results show that our prediction is closer to the ground truth than PCCSNet [41], which indicates the superiority of our SICNet compared with the instance-conditioned approaches.

5. Conclusion

In this paper, we present a simple yet effective sparse-instance-conditioned network for pedestrian multimodal trajectory prediction, which leverages sparse instances to guide multimodal prediction. Moreover, we propose a memory refinement module to reduce recall errors when meeting unfamiliar trajectories during inference. Extensive experiments show that our method performs better than previous methods, and ablation studies demonstrate that our sparse-instance-conditioned method can generate multimodal predictions more accurately and adaptively compared with goal-conditioned and instance-conditioned approaches.

Acknowledgement

This work was supported partly by National Key R&D Program of China under Grant 2021YFB1714700, NSFC under Grants 62088102 and 62106192, Natural Science Foundation of Shaanxi Province under Grant 2022JC-41, and Fundamental Research Funds for the Central Universities under Grant XTR042021005.

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social LSTM: Human trajectory prediction in crowded spaces. In *CVPR*, pages 961–971, 2016.

[2] Inhwan Bae, Jin-Hwi Park, and Hae-Gon Jeon. Non-probability sampling network for stochastic human trajectory prediction. In *CVPR*, pages 6477–6487, 2022.

[3] Guangyi Chen, Junlong Li, Jiwen Lu, and Jie Zhou. Human trajectory prediction via counterfactual analysis. In *ICCV*, pages 9824–9833, 2021.

[4] Xinlei Chen and Abhinav Gupta. Spatial memory for context reasoning in object detection. In *ICCV*, pages 4086–4096, 2017.

[5] Chiho Choi, Joon Hee Choi, Jiachen Li, and Srikanth Malla. Shared cross-modal trajectory prediction for autonomous driving. In *CVPR*, pages 244–253, 2021.

[6] Lingwei Dang, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. MSR-GCN: Multi-scale residual graph convolution networks for human motion prediction. In *ICCV*, pages 11447–11456, 2021.

[7] Hanming Deng, Yang Hua, Tao Song, Zongpu Zhang, Zhengui Xue, Ruhui Ma, Neil Robertson, and Haibing Guan. Object guided external memory network for video object detection. In *ICCV*, pages 6677–6686, 2019.

[8] Jinghai Duan, Le Wang, Chengjiang Long, Sanping Zhou, Fang Zheng, Liushuai Shi, and Gang Hua. Complementary attention gated network for pedestrian trajectory prediction. In *AAAI*, pages 542–550, 2022.

[9] Zhihong Fu, Qingjie Liu, Zehua Fu, and Yunhong Wang. STMTrack: Template-free visual tracking with space-time memory networks. In *CVPR*, pages 13769–13778, 2021.

[10] Harshayu Girase, Haiming Gang, Srikanth Malla, Jiachen Li, Akira Kanehara, Karttikeya Mangalam, and Chiho Choi. LOKI: Long term and key intentions for trajectory prediction. In *ICCV*, pages 9803–9812, 2021.

[11] Dong Gong, Frederic Z. Zhang, Javen Qinfeng Shi, and Anton van den Hengel. Memory-augmented dynamic neural relational inference. In *ICCV*, pages 11843–11852, 2021.

[12] Tianpei Gu, Guangyi Chen, Junlong Li, Chunze Lin, Yongming Rao, Jie Zhou, and Jiwen Lu. Stochastic trajectory prediction via motion indeterminacy diffusion. In *CVPR*, pages 17113–17122, 2022.

[13] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: socially acceptable trajectories with generative adversarial networks. In *CVPR*, pages 2255–2264, 2018.

[14] Adam Houenou, Philippe Bonnifait, Véronique Cherfaoui, and Wen Yao. Vehicle trajectory prediction based on motion model and maneuver recognition. In *IROS*, pages 4363–4369, 2013.

[15] Boris Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *ICCV*, pages 2375–2384, 2019.

[16] Łukasz Kaiser, Ofir Nachum, Aurko Roy, and Samy Bengio. Learning to remember rare events. In *ICLR*, 2017.

[17] Youngjin Kim, Minjung Kim, and Gunhee Kim. Memorization precedes generation: Learning unsupervised gans with memory networks. In *ICLR*, 2018.

[18] Sumith Kulal, Jiayuan Mao, Alex Aiken, and Jiajun Wu. Programmatic concept learning for human motion description and synthesis. In *CVPR*, pages 13843–13852, 2022.

[19] Mihee Lee, Samuel S. Sohn, Seonghyeon Moon, Sejong Yoon, Mubbasir Kapadia, and Vladimir Pavlovic. MUSE-VAE: Multi-scale vae for environment-aware long term trajectory prediction. In *CVPR*, pages 2221–2230, 2022.

[20] Sangho Lee, Jinyoung Sung, Youngjae Yu, and Gunhee Kim. A memory network approach for story-based temporal summarization of 360° videos. In *CVPR*, pages 1410–1419, 2018.

[21] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. *Comput. Graph. Forum*, 26(3):655–664, 2007.

[22] Peizhao Li, Pu Wang, Karl Berntorp, and Hongfu Liu. Exploiting temporal relations on radar perception for autonomous driving. In *CVPR*, pages 17071–17080, 2022.

[23] Shijie Li, Yanying Zhou, Jinhui Yi, and Juergen Gall. Spatial-temporal consistency network for low-latency trajectory forecasting. In *ICCV*, pages 1940–1949, 2021.

[24] Jialun Liu, Yifan Sun, Feng Zhu, Hongbin Pei, Yi Yang, and Wenhui Li. Learning memory-augmented unidirectional metrics for cross-modality person re-identification. In *CVPR*, pages 19366–19375, 2022.

[25] Zhian Liu, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. A hybrid video anomaly detection framework via memory-augmented flow reconstruction and flow-guided frame prediction. In *ICCV*, pages 13588–13597, 2021.

[26] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From goals, waypoints & paths to long term human trajectory forecasting. In *ICCV*, pages 15233–15242, 2021.

[27] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In *ECCV*, pages 759–776, 2020.

[28] Francesco Marchetti, Federico Becattini, Lorenzo Seidenari, and Alberto Del Bimbo. Mantra: Memory augmented networks for multiple trajectory prediction. In *CVPR*, pages 7143–7152, 2020.

[29] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *CVPR*, pages 14424–14432, 2020.

[30] Alessio Monti, Angelo Porrello, Simone Calderara, Pasquale Coscia, Lamberto Ballan, and Rita Cucchiara. How many observations are enough? knowledge distillation for trajectory forecasting. In *CVPR*, pages 6553–6562, 2022.

[31] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*, pages 261–268, 2009.

[32] Zheng Qin, Sanping Zhou, Le Wang, Jinghai Duan, Gang Hua, and Wei Tang. Motiontrack: Learning robust short-term and long-term motions for multi-object tracking. In *CVPR*, pages 17939–17948, 2023.

[33] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning Social Etiquette: Human trajectory understanding in crowded scenes. In *ECCV*, pages 549–565, 2016.

[34] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. SoPhie: An attentive gan for predicting paths compliant to social and physical constraints. In *CVPR*, pages 1349–1358, 2019.

[35] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *ECCV*, pages 683–700, 2020.

[36] Liushuai Shi, Le Wang, Chengjiang Long, Sanping Zhou, Fang Zheng, Nanning Zheng, and Gang Hua. Social interpretable tree for pedestrian trajectory prediction. In *AAAI*, pages 2235–2243, 2022.

[37] Liushuai Shi, Le Wang, Chengjiang Long, Sanping Zhou, Mo Zhou, Zhenxing Niu, and Gang Hua. SGCN: Sparse graph convolution network for pedestrian trajectory prediction. In *CVPR*, pages 8990–8999, 2021.

[38] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. In *CVPR*, pages 16519–16529, 2021.

[39] Hao Sun, Zhiqun Zhao, and Zhihai He. Reciprocal learning networks for human trajectory prediction. In *CVPR*, pages 7416–7425, 2020.

[40] Jianhua Sun, Qinhong Jiang, and Cewu Lu. Recursive social behavior graph for trajectory prediction. In *CVPR*, pages 660–669, 2020.

[41] Jianhua Sun, Yuxuan Li, Hao-Shu Fang, and Cewu Lu. Three steps to multimodal trajectory prediction: Modality clustering, classification and synthesis. In *ICCV*, pages 13250–13259, 2021.

[42] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *IEEE T-PAMI*, 43(10):3349–3364, 2020.

[43] Jingke Wang, Tengju Ye, Ziqing Gu, and Junbo Chen. LTP: Lane-based trajectory prediction for autonomous driving. In *CVPR*, pages 17134–17142, 2022.

[44] Jürgen Wiest, Matthias Höffken, Ulrich Kreßel, and Klaus Dietmayer. Probabilistic trajectory prediction with gaussian mixture models. In *IV*, pages 141–146, 2012.

[45] Yuxuan Wu, Le Wang, Sanping Zhou, Jinghai Duan, Gang Hua, and Wei Tang. Multi-stream representation learning for pedestrian trajectory prediction. In *AAAI*, pages 2875–2882, 2023.

[46] Zifeng Wu, Chunhua Shen, and Anton Van Den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *PR*, 90:119–133, 2019.

[47] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Shengping Zhang, and Wenxiu Sun. Efficient regional memory network for video object segmentation. In *CVPR*, pages 1286–1295, 2021.

[48] Bo Xu, Han Huang, Cheng Lu, Ziwen Li, and Yandong Guo. Virtual multi-modality self-supervised foreground matting for human-object interaction. In *ICCV*, pages 428–437, 2021.

[49] Chenxin Xu, Weibo Mao, Wenjun Zhang, and Siheng Chen. Remember intentions: Retrospective-memory-based trajectory prediction. In *CVPR*, pages 6488–6497, 2022.

[50] Hao Xue, Du Q. Huynh, and Mark Reynolds. SS-LSTM: A hierarchical lstm model for pedestrian trajectory prediction. In *WACV*, pages 1186–1194, 2018.

[51] Gang Yang, Man Zhou, Keyu Yan, Aiping Liu, Xueyang Fu, and Fan Wang. Memory-augmented deep conditional unfolding network for pan-sharpening. In *CVPR*, pages 1788–1797, 2022.

[52] Tianyu Yang and Antoni B. Chan. Visual tracking via dynamic memory networks. *IEEE T-PAMI*, 43(1):360–374, 2021.

[53] Jiyang Yu, Jingen Liu, Liefeng Bo, and Tao Mei. Memory-augmented non-local attention for video super-resolution. In *CVPR*, pages 17834–17843, 2022.

[54] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris Kitani. AgentFormer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *ICCV*, pages 9793–9803, 2021.

[55] He Zhao and Richard P. Wildes. Where are you heading? dynamic trajectory prediction with expert goal examples. In *ICCV*, pages 7609–7618, 2021.

[56] Zheng Zhao, Weihai Chen, Xingming Wu, Peter CY Chen, and Jingmeng Liu. LSTM network: a deep learning approach for short-term traffic forecast. *IET Intell. Transp. Syst.*, 11(2):68–75, 2017.

[57] Desen Zhou, Zhichao Liu, Jian Wang, Leshan Wang, Tao Hu, Errui Ding, and Jingdong Wang. Human-object interaction detection via disentangled transformer. In *CVPR*, pages 19568–19577, 2022.