Multi-Stream Representation Learning for Pedestrian Trajectory Prediction

Yuxuan Wu[1], Le Wang[1]*, Sanping Zhou[1], Jinghai Duan[2], Gang Hua[3], Wei Tang[4]

[1]Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
[2]School of Software Engineering, Xi'an Jiaotong University
[3]Wormpex AI Research
[4]University of Illinois at Chicago
{yuxuan9862, ganghua}@gmail.com, {lewang, spzhou}@xjtu.edu.cn, caesardjh98@stu.xjtu.edu.cn, tangw@uic.edu

Abstract

Forecasting the future trajectory of pedestrians is an important task in computer vision with a range of applications, from security cameras to autonomous driving. It is very challenging because pedestrians not only move individually across time but also interact spatially, and the spatial and temporal information is deeply coupled with one another in a multi-agent scenario. Learning such complex spatio-temporal correlation is a fundamental issue in pedestrian trajectory prediction. Inspired by the procedure that the hippocampus processes and integrates spatio-temporal information to form memories, we propose a novel multi-stream representation learning module to learn complex spatio-temporal features of pedestrian trajectory. Specifically, we learn temporal, spatial and cross spatio-temporal correlation features in three respective pathways and then adaptively integrate these features with learnable weights by a gated network. Besides, we leverage the sparse attention gate to select informative interactions and correlations brought by complex spatio-temporal modeling and reduce complexity of our model. We evaluate our proposed method on two commonly used datasets, *i.e.*, ETH-UCY and SDD, and the experimental results demonstrate that our method achieves state-of-the-art performance. Code: https://github.com/YuxuanIAIR/MSRL-master

Introduction

Pedestrian trajectory prediction aims to predict the future movement of each pedestrian in a traffic scenario. It is important in many intelligent applications like autonomous driving and video surveillance.

Although great progress has been made recently, predicting future trajectories of pedestrians remains challenging due to the temporal motion directed by each pedestrian's intention (Mangalam et al. 2020) and the spatial interaction among different pedestrians (Gupta et al. 2018; Huang et al. 2019; Sun, Jiang, and Lu 2020; Yu et al. 2020). By modeling the temporal dependence and the spatial interactions, previous works have achieved great advancement. Most of them learn temporal motion and spatial interaction via separate models and feed trajectories into stacked models of spatial and temporal components to get spatio-temporal features, as shown in Figure 1. Concretely, the temporal model employs the temporal sequential networks (*e.g.*, Recurrent Neural Networks and its variants (Hochreiter and Schmidhuber 1997), Temporal Convolutional Network (Bai, Kolter, and Koltun 2018), and Self-attention (Vaswani et al. 2017)) to model the temporal dependence, and meanwhile, the spatial model aggregates the messages from all other pedestrians via the pooling mechanism (Gupta et al. 2018), Graph Convolutional Networks (Sun, Jiang, and Lu 2020), or the spatial attention (Huang et al. 2019).

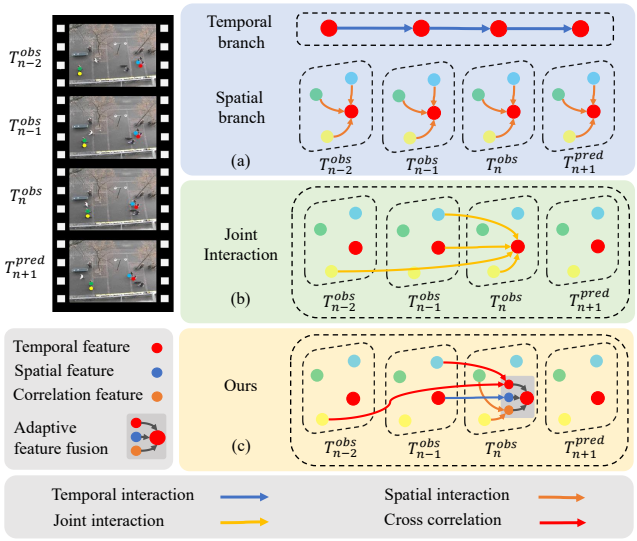

Figure 1: (a) represents previous methods that model temporal motion and spatial interaction by two separate branches. (b) represents joint modeling. (c) represents our method that learns temporal, spatial and correlation feature in three pathways and integrates them adaptively.

However, the architecture mentioned above may not capture complex spatio-temporal correlations completely because there exists interplay between the two dimensions. Although there are models like AgentFormer (Yuan et al. 2021) that learns spatio-temporal features jointly as shown in Figure 1(b), the efficiency and representation ability is sub-optimal because of the great amount of computation brought by joint modeling and interference between different kinds of features. Hence, an architecture that can effec-

*Corresponding author.

tively handle such complex spatio-temporal information is required for accurately predicting pedestrians' trajectories.

Inspired by the finding that hippocampus has different sub-regions for processing temporal, spatial and conjoint information respectively (Rangel, Quinn, and Chiba 2016) and a special region is responsible for integrating the features above (Buzsáki and Llinás 2017), we propose the multi-stream representation learning module, which learns temporal dependency, spatial interaction and multidimensional correlation in three pathways and integrates them with learnable weights by a gated network, as shown in Figure 1(c) and Figure 2. The temporal branch learns temporal motion dependency via performing self-attention on each person's trajectory sequence. The spatial branch learns spatial interaction by performing sparse self-attention on slices of each time step. Each slice contains motion states of each person in one time step. The cross branch processes spatial and temporal dimensions together and employs sparse self-attention to investigate the relationship between motion states of different persons within a certain time scale and learn possible correlations between two dimensions. Then temporal, spatial and correlation features are merged by the gated network, which enables the model to adaptively assign different weights according to the importance of different features. The proposed multi-stream representation learning module makes each pathway focus on distinct features and allows the model to integrate these features adaptively.

In addition, sparse attention gate is utilized in spatial branch and cross branch to select informative interactions and cross-space-time correlations. This enforces the model to focus on most relevant interactions or correlations and reduces complexity. Finally, we apply our proposed module into CVAE framework to achieve multi-modal prediction.

We evaluate our method on two widely used benchmarks - ETH-UCY (Pellegrini et al. 2009; Lerner, Chrysanthou, and Lischinski 2007) and SDD (Bock et al. 2020). Extensive experiments show that our method outperforms the states of the art under both metrics of Average Displacement Error (ADE) and Final Displacement Error (FDE). The ablation studies also verify the effectiveness of our proposed model.

The contributions of our work are summarized below:

- Learning complex spatial and temporal features of pedestrian trajectories as well as their correlations is not well-addressed by previous models and we propose the multi-stream representation learning module that learns temporal, spatial and conjoint features by three pathways and integrates them dynamically via gated mechanism.
- One great challenge for interaction and correlation modeling is that densely connecting the cross correlations and spatial interactions not only leads to intractable computational complexity but also introduces redundancy. To address this issue, we design a sparse attention gate that can selectively model the most relevant interactions and correlations to sparsify our model and reduce complexity.
- Extensive experiments show that learning complex spatio-temporal patterns of pedestrian trajectories via our proposed method greatly improves the prediction accuracy and achieves state-of-the-art performance.

Related Work

Trajectory Prediction. Trajectory prediction is a meaningful and insightful task in audio, finance and traffic scenes. Advancement in deep learning has greatly improved the machine's ability to recognize and predict data with sequence representations. RNN and its variant LSTM (Hochreiter and Schmidhuber 1997) networks have made lots of progress in various sequence prediction tasks and have been successfully applied to learn the temporal motion pattern of pedestrians, like Social-LSTM (Alahi et al. 2016) and STGAT (Huang et al. 2019). Moreover, some models like Social Attention (Vemula, Muelling, and Oh 2018) and Trajectron (Ivanovic and Pavone 2019) use LSTM to implement a spatio-temporal graph, which can represent structured sequence data and achieve great performances. As RNN-based models may suffer from gradient varnishing or explosion in certain conditions, some models like SGCN (Shi et al. 2021) and UNIN (Zheng et al. 2021) utilize temporal convolutional networks (TCN) (Lea et al. 2017) to learn temporal dependency. In recent years, Transformer (Vaswani et al. 2017) has quickly dominated NLP across various tasks. By replacing recurrence with self-attention mechanism, Transformer can better learn the long-term dependencies. Beyond NLP, Transformer also achieves great success in pedestrian trajectory prediction. TransformerTF (Giuliari et al. 2020) and STAR (Yu et al. 2020) apply Transformer to model temporal dependency and achieve better performance.

Social Interactions. Social interactions play an important role in determining pedestrians' future trajectories. Previous distance-based methods learn the interactions between pedestrian pairs with a distance closer than a threshold. For instance, S-GAN (Gupta et al. 2018) proposes a kind of pooling mechanism to compute interactions based on relative distance between pedestrians. With the development of graph-structured network, graph models are utilized by many models to learn interactions. RSBG (Sun, Jiang, and Lu 2020) and STGCNN (Mohamed et al. 2020) use graph convolution networks to obtain interaction features. In addition, the attention mechanism is applied by many models because of its effectiveness in extracting relationship between elements. STGAT (Huang et al. 2019) leverages the graph attention network (GAT) to assign different weights to pedestrians involved in interactions. Other than the graph attention network, self-attention (Vaswani et al. 2017) is also prevalent in modeling spatial interactions. STAR (Yu et al. 2020) regards the self-attention mechanism as message passing in an undirected fully connected graph and calculates the interaction weights to achieve weighted social interaction modeling. SGCN (Shi et al. 2021) uses self-attention to generate a sparse graph for interaction learning.

Most of the methods above learn temporal motion and spatial interaction via separate models and feed trajectories into stacked models of spatial and temporal components to get spatio-temporal features. They failed to capture the correlation between two dimensions. AgentFormer (Yuan et al. 2021) learns such correlation via agent-aware attention but the spatial interaction and cross spatio-temporal correlation may interfere with each other in their model. Besides, excessive connections across different time steps are useless.

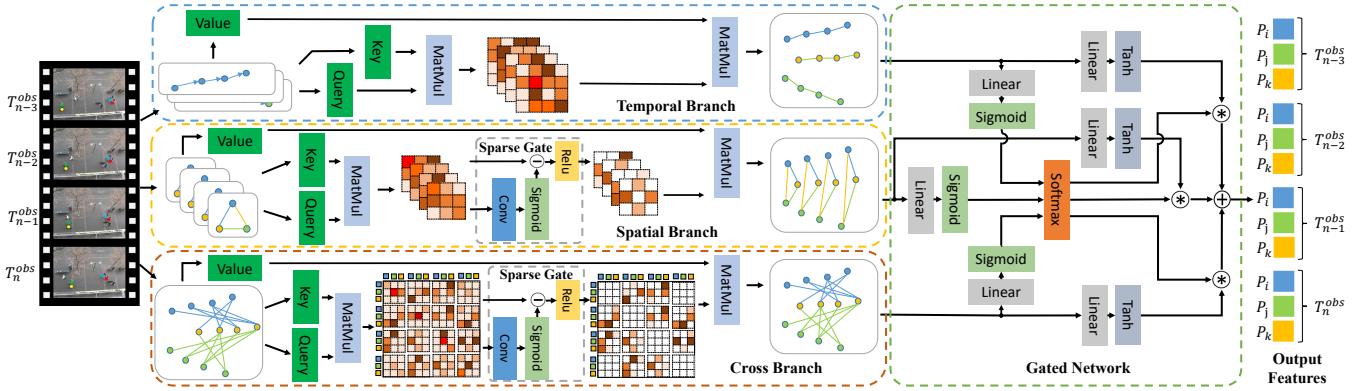

Figure 2: Architecture of the multi-stream representation learning module. Temporal, spatial and cross branches aim to learn temporal, spatial and correlation features respectively. The gated network integrates these features adaptively. Sparse attention gate is utilized in spatial branch and cross branch to enforce the model focus on important interactions and correlations.

Hence, our model is designed to learn spatial features, temporal features as well as their correlations in three pathways in which each branch can concentrate on one specific feature. In addition, sparse attention gate is used to enforce the model focus on a small number of important interactions or correlations and dismiss useless interactions or correlations.

Method

Problem Formulation

We represent the state of the n-th pedestrian at time t as X_n^t including its position and velocity. Given $n \in \{1, 2, \ldots, N\}$ pedestrians over observable time steps $t \in \{T_1, T_2, \ldots, T_{\text{obs}}\}$, the historical trajectories can be expressed as $X = \{X_n^t\}_{n=1,t=1}^{N,T_{\text{obs}}}$. Likewise, the future trajectories of N pedestrians over future time steps $t \in \{T_{\text{obs}+1}, T_{\text{obs}+2}, \ldots, T_{\text{pred}}\}$ can be represented as $Y = \{Y_n^t\}_{n=1,t=T_{\text{obs}+1}}^{N,T_{\text{pred}}}$ We would like to learn the probability distribution function $P(Y|X)$, a generative future trajectory distribution conditioned on past trajectories.

Multi-stream Representation Learning Module

Time Encoding Before learning trajectory features, the position encoding method of Transformer (Vaswani et al. 2017) is utilized to represent the temporal information of trajectory positions. The "time encodings" are concatenated with trajectory input embeddings to form the input of our multi-stream representation learning module.

Given the trajectory embeddings of all N pedestrians in T time steps as input keys $K \in \mathbb{R}^{T\times N\times D}$, queries $Q \in \mathbb{R}^{T\times N\times D}$, and values $V \in \mathbb{R}^{T\times N\times D}$, where D is the embedding dimension of the input state. As shown in Figure 2, we apply three parallel branches to extract motion tendency feature, spatial interaction feature and multidimensional correlation feature and then fuse them via gated mechanism. This makes each pathway focus on a specific kind of feature and allows the model to select important features adaptively.

Temporal Dependency The temporal branch learns temporal dependency of trajectory via self-attention. We extract their own trajectory features for each pedestrian from the input and do linear projections to obtain Q, K, V as queries $Q_t \in \mathbb{R}^{T\times D}$, keys $K_t \in \mathbb{R}^{T\times D}$ and values $V_t \in \mathbb{R}^{T\times D}$. Temporal self-attention is executed on each person's keys, queries and values. For each of the N pedestrians, we get the temporal feature as follows:

$$F_{\text{t}} = \text{Softmax}(\frac{Q_{\text{t}}K_{\text{t}}^{\text{T}}}{\sqrt{d_{\text{k}}}}) \cdot V_{\text{t}}, \tag{1}$$

where $F_{\text{t}} \in \mathbb{R}^{T\times D}$. The temporal feature of each pedestrian is then concatenated and reshaped back into $\overline{F}_{\text{t}} \in \mathbb{R}^{T\times N\times D}$.

Spatial Interaction The spatial branch learns spatial interactions via sparse self-attention. We extract spatial states of all pedestrians in each time step from the input and do linear projections to obtain Q, K, V as queries $Q_s \in \mathbb{R}^{N\times D}$, keys $K_s \in \mathbb{R}^{N\times D}$ and values $V_s \in \mathbb{R}^{N\times D}$. For each time step, we get a dense attention weight matrix as follows:

$$W_{\text{s}} = \text{Softmax}(\frac{Q_{\text{s}}K_{\text{s}}^{\text{T}}}{\sqrt{d_{\text{k}}}}), \tag{2}$$

where $W_{\text{s}} \in \mathbb{R}^{N\times N}$ is for each time step. Then we proceed to select important interactions and generate the sparse interaction weight matrix via sparse attention gate:

$$\widetilde{W}_{\text{s}} = \text{Relu}(W_{\text{s}} - \text{Sigmoid}(\text{CNN}(W_{\text{s}}))). \tag{3}$$

Convolution is applied on lines of the dense attention weight matrix to get high-level features of interactions and the output of sigmoid layer is between 0 and 1, which can serve as the gate for dense attention weight matrix. The equation above preserves informative interactions with weight generated by self-attention and the weights of interactions to be dismissed become 0. Finally, we design the "Zero-Softmax" function to normalize the sparse attention weights and prevent zero-weights from being back into non-zero weights. Specifically, given a flattened matrix $\mathbf{x} = [x_1, x_2, \ldots, x_{\mathcal{D}}]$,

$$\text{Zero-Softmax}(x_i) = \frac{\exp(x_i) - 1}{\sum_j^{\mathcal{D}} (\exp(x_j) - 1) + \epsilon}, \tag{4}$$

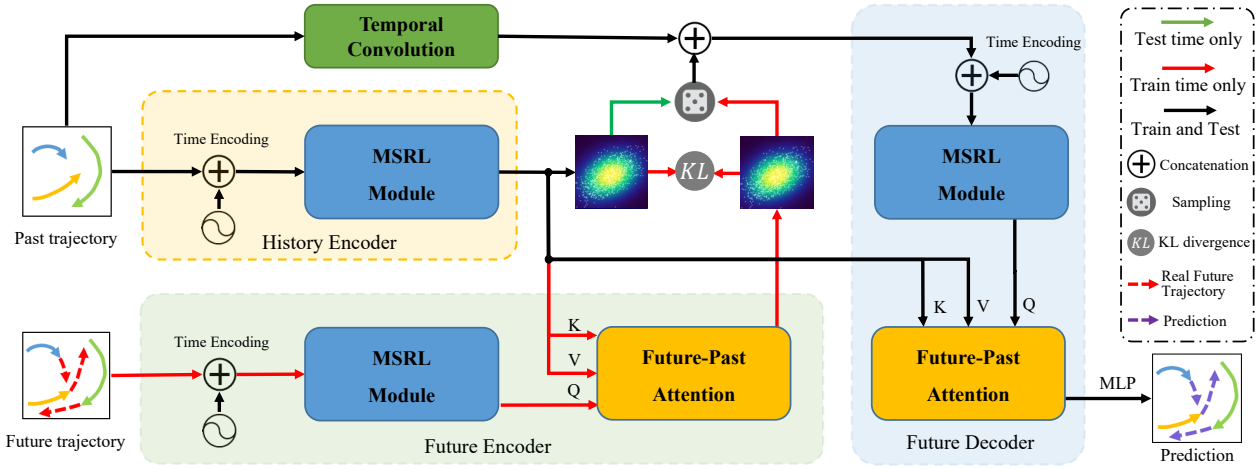

Figure 3: Multi-Stream Representation Learning Module (MSRL Module) based trajectory prediction framework

where ϵ is a neglectable small constant for numerical stability, $\mathcal{D}$ is the dimensionality of the input vector. The normalized sparse interaction weight matrix is then multiplied with V_s to produce the spatial feature for each time step as:

$$F_s = \text{Zero-Softmax}(\widetilde{W}_\text{s}) \cdot V_s, \tag{5}$$

where $F_\text{s} \in \mathbb{R}^{N\times D}$. The spatial feature of each time step is then concatenated and reshaped back into $\overline{F}_\text{s} \in \mathbb{R}^{T\times N\times D}$.

Cross Correlation The cross branch learns correlation between spatial and temporal dimensions through cross-space-time connections. We first flatten the spatio-temporal dimensions of all pedestrians to obtain the trajectory feature sequence as $\widetilde{X} = (X_1^1, ..., X_N^1, X_1^2, ..., X_N^2, ..., X_1^{T_\text{obs}}, ..., X_N^{T_\text{obs}})$ and do linear projection to obtain Q, K, V as queries $Q_c \in \mathbb{R}^{(T\times N)\times D}$, keys $K_c \in \mathbb{R}^{(T\times N)\times D}$ and values $V_c \in \mathbb{R}^{(T\times N)\times D}$. Self-attention is performed on flattened sequence to get primitive cross attention weights:

$$W_\text{c} = \text{Softmax}(\frac{Q_\text{c}K_\text{c}^\text{T}}{\sqrt{d_\text{k}}}). \tag{6}$$

As the cross spatio-temporal interaction makes more connections, it inevitably introduces noise. Thus, learning sparse attention weight is more important because sparsity can not only make the model concentrate on important attentions but also remove noise. Similar with the spatial branch, we apply convolution on lines of the primitive cross attention weight matrix and generate the sparse attention weight matrix via gated mechanism as:

$$\widetilde{W}_\text{c} = \text{Relu}(W_\text{c} - \text{Sigmoid}(\text{CNN}(W_\text{c}))). \tag{7}$$

In addition, we add proper masks to make this branch focus on cross-space-time connections. Finally, the correlation feature is obtained by multiplying normalized sparse attention weights with V_c as:

$$F_c = \text{Zero-Softmax}(\widetilde{W}_\text{c}) \cdot V_c. \tag{8}$$

$F_c \in \mathbb{R}^{(T\times N)\times D}$ is then rearranged into $\overline{F}_\text{c} \in \mathbb{R}^{T\times N\times D}$.

Gated Network With temporal feature $\overline{F}_\text{t}$, spatial feature $\overline{F}_\text{s}$ and correlation feature $\overline{F}_\text{c}$, we fuse them via gated network to allow the model to identify important features and fuse these features adaptively. For features of each branch, the intermediate feature $\widetilde{F}$ and weight of gated mechanism G are calculated as:

$$\begin{aligned} \widetilde{F} &= Tanh(\phi(\overline{F}, W^F)), \\ G &= Sigmoid(\phi(\overline{F}, W^G)), \end{aligned} \tag{9}$$

where W^F and W^G are learnable weights of linear projection ϕ and the three pathways use different weights for linear projection. In this way, the intermediate feature $\widetilde{F}_t, \widetilde{F}_s, \widetilde{F}_c$ for each branch and the fusion weight G_t, G_s, G_c for each branch can be obtained. The fusion weight is then normalized via Softmax and the integrated feature F is obtained by summing features from each pathway with normalized weights, where $[,,]$ indicates concatenation:

$$\begin{aligned} [\hat{G}_t, \hat{G}_s, \hat{G}_c] &= Softmax([G_t, G_s, G_c]), \\ F &= \widetilde{F}_t \odot \hat{G}_t + \widetilde{F}_s \odot \hat{G}_s + \widetilde{F}_c \odot \hat{G}_c. \end{aligned} \tag{10}$$

Multi-modal Prediction

Having introduced our multi-stream representation learning module, we apply it into CVAE framework shown in Figure 3 to achieve multi-modal prediction. We introduce the Gaussian latent variable $Z = \{z_1, z_2, .., z_n\}$ for the scene with n pedestrians and the latent distribution can be expressed as:

$$P(Y|X) = \int P(Y|X, Z)P(Z|X)dZ, \tag{11}$$

where $P(Z|X)$ is a Gaussian conditional prior factorized over pedestrians and $P(Y|Z, X)$ is a conditional likelihood model. The optimization can be performed by minimizing the negative evidence-based lower bound (ELBO) as:

$$\begin{aligned} \mathcal{L}_{ELBO} = &- E_{q(Z|Y,X)}[logP(Y|Z, X)] \\ &+ KL(Q(Z|Y, X)||P(Z|X)). \end{aligned} \tag{12}$$

History Encoder The History Encoder aims to encode the past trajectories into a latent distribution and generate the Gaussian parameters of the conditional prior distribution $P(Z|X)$ for each pedestrian. It takes the multi-stream representation learning module followed by standard feed-forward and norm layers in Transformer to get the features of past trajectories. Mean-pooling is used to get pedestrian-wise past trajectory feature, which is then fed into an MLP to generate the Gaussian parameters of $P(Z|X)$.

Future Encoder The Future Encoder models the posterior distribution $Q(Z|Y,X)$ conditioned on past trajectories. Firstly, the proposed multi-stream representation learning module is utilized to learn the feature of future trajectory. Then, we use a Future-Past Attention module to update the future trajectory feature. It receives history trajectory feature from the History Encoder as K,V and future trajectory feature produced by the previous multi-stream representation learning module as Q to learn the relationship between past trajectory and future trajectory. The Future-Past Attention module shares the same architecture with the motion branch in the multi-stream representation learning module, except that K,V and Q are from different sources, as shown in Figure 4. In this way, the connection between the observable trajectory and future trajectory is built, which helps the network learn more effective future trajectory features and further excavate pedestrian motion patterns. Similar with the History Encoder, standard feed-forward and norm layers are followed. Mean-pooling is then used to get pedestrian-wise future trajectory feature, which is then fed into an MLP to generate the Gaussian parameters of $Q(Z|Y,X)$.

Future Decoder The Future Decoder takes the sampled result Z concatenated with the result of temporal convolution as input and decodes the input back into predicted future trajectories. The samples are obtained from posterior distribution $Q(Z|Y,X)$ in training but from prior distribution $P(Z|X)$ in testing. Besides, a sampler plugin like LDS (Ma et al. 2021) is utilized to improve sample diversity within a fixed number of sample amount. We use this plugin because improved sample diversity and quality can better help our model play advantages of spatio-temporal feature modeling. More details about this plugin can be seen in supplemental materials. The architecture of Future Decoder is similar with that of Future Encoder. The multi-stream representation learning module learns possible interactions in prediction horizon and the Future-Past Attention module further refines the future trajectory features with past features. In addition, we add proper masks to prevent the model from attending to subsequent positions when predicting. Standard feed-forward and norm layers are followed. The predicted future trajectory $\hat{Y}$ is finally generated via MLP.

Training the Model

The CVAE model can be trained by minimizing the negative evidence-based lower bound (ELBO). The first term in Eq.(12) is the reconstruction loss, or expected negative log-likelihood. As we use a Gaussian to model the conditional likelihood $P(Y|X,Z) = \mathcal{N}(\hat{Y}, I/\beta)$, where I is the identity matrix and β is a weighting factor, this term equals to the mean squared error: $\mathcal{L}_{MSE} = \frac{1}{2}\beta||Y - \hat{Y}||^2$ The second term in Eq.(12) is the KL divergence between prior and posterior, which is calculated as follows:

$$\mathcal{L}_{KL} = KL(Q(Z|Y,X)||P(Z|X)). \tag{13}$$

Besides the negative evidence-based lower bound (ELBO), we add a variety loss proposed by S-GAN (Gupta et al. 2018) to encourage the model to produce diverse results. For each scene we generate k possible predictions and choose the best prediction to calculate the variety loss:

$$\mathcal{L}_{VAR} = min||Y_i - \hat{Y}_i^{(k)}||^2. \tag{14}$$

Finally, the total loss function is defined as:

$$\mathcal{L} = \mathcal{L}_{MSE} + \mathcal{L}_{KL} + \mathcal{L}_{VAR}. \tag{15}$$

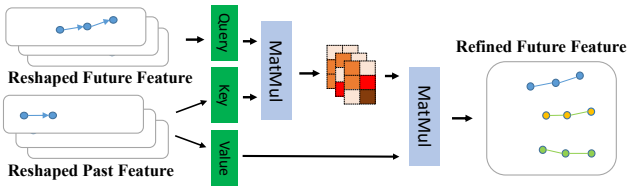

Figure 4: Future-Past Attention module for strengthening the relationship between future trajectories and past trajectories

Experiments

Evaluation Dataset. We evaluate our model on the ETH-UCY (Pellegrini et al. 2009; Lerner, Chrysanthou, and Lischinski 2007) and Stanford Drone Dataset (SDD) (Bock et al. 2020), which are widely-used benchmarks for pedestrian trajectory prediction. ETH-UCY contains five scenarios named ETH, HOTEL, UNIV, ZARA1 and ZARA2. Following prior work, we use the leave-one-out cross validation strategy on ETH-UCY. SDD contains 20 scenes and the coordinate of trajectory is recorded in pixel coordinate system with the pixel as the unit. An observation length of 8 timesteps (3.2s) and a prediction horizon of 12 timesteps (4.8s) is used for evaluation.

Evaluation Metrics. We employ two evaluation metrics: Average Displacement Error (ADE) and Final Displacement Error (FDE). ADE is the average L2 distance between ground truth and prediction over all predicted time steps. FDE is the L2 distance between predicted final destination and real final destination at the end of the prediction horizon. Following previous works, for each model compared, 20 trajectories are generated and the trajectory closest to ground truth is used to calculate ADE and FDE.

Quantitative Analysis

We compare our method with state-of-the-art models in recent years. As we do not take scene context as input, models using scene context are not listed for fair comparison. The experimental results on ADE/FDE are presented in Table 1 and Table 2, showing that our model achieves state-of-the-art performance on both ETH-UCY and SDD.

Model		ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
S-GAN	(Gupta et al. 2018)	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	(Sadeghian et al. 2019)	0.70/1.43	0.76/1.67	0.54/1.24	0.30/0.63	0.38/0.78	0.51/1.15
NEXT	(Liang et al. 2019)	0.73/1.65	0.30/0.59	0.60/1.27	0.38/0.81	0.31/0.68	0.46/1.00
STGAT	(Huang et al. 2019)	0.68/1.29	0.68/1.40	0.57/1.29	0.29/0.60	0.37/0.75	0.52/1.07
STGCNN	(Mohamed et al. 2020)	0.64/1.11	0.49/0.85	0.44/0.79	0.34/0.53	0.30/0.48	0.44/0.75
STAR	(Yu et al. 2020)	0.36/0.65	0.17/0.36	0.31/0.62	0.26/0.55	0.22/0.46	0.26/0.53
PECNet	(Mangalam et al. 2020)	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	(Shi et al. 2021)	0.63/1.03	0.32/0.55	0.37/0.70	0.29/0.53	0.25/0.45	0.37/0.65
LB-EBM	(Pang et al. 2021)	0.30/0.52	0.13/0.20	0.27/0.52	0.20/0.37	0.15/0.29	0.21/0.38
AgentFormer	(Yuan et al. 2021)	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	(Sun et al. 2021)	**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	(Duan et al. 2022)	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	(Shi et al. 2022)	0.39/0.61	0.13/0.22	0.29/0.49	0.19/0.31	0.15/0.29	0.23/0.38
MemoNet	(Xu et al. 2022b)	0.40/0.61	**0.11/0.17**	**0.24/0.43**	0.18/0.32	**0.14**/0.24	0.21/0.35
GroupNet	(Xu et al. 2022a)	0.46/0.73	0.15/0.25	0.26/0.49	0.21/0.39	0.17/0.33	0.25/0.44
Ours	AAAI 2023	**0.28/0.47**	0.14/0.22	**0.24/0.43**	**0.17/0.30**	**0.14/0.23**	**0.19/0.33**

Table 1: Comparison with state-of-the-art methods on ETH-UCY for ADE/FDE. Lower is better.

Model		ADE	FDE
S-GAN	(Gupta et al. 2018)	27.23	41.44
Sophie	(Sadeghian et al. 2019)	16.27	29.38
MATF	(Zhao et al. 2019)	22.59	33.53
CF-VAE	(Bhattacharyya et al. 2019)	12.60	22.30
Multiverse	(Liang et al. 2020)	14.78	27.09
MANTRA	(Marchetti et al. 2020)	8.96	17.76
SimAug	(Liang and Hauptmann 2020)	10.27	19.71
EvolveGraph	(Li et al. 2020)	13.90	22.90
PECNet	(Mangalam et al. 2020)	9.96	15.88
LB-EBM	(Pang et al. 2021)	8.87	15.61
PCCSNet	(Sun et al. 2021)	8.62	16.16
CAGN	(Duan et al. 2022)	9.42	15.93
SIT	(Shi et al. 2022)	8.59	15.27
MemoNet	(Xu et al. 2022b)	<u>8.56</u>	**12.66**
GroupNet	(Xu et al. 2022a)	9.31	16.11
Ours	AAAI 2023	**8.22**	<u>13.39</u>

Table 2: Comparison with state-of-the-art methods on SDD for ADE and FDE. Lower is better. The bold/underlined font represent the best/second best result.

	Model	ADE	FDE
(1)	T	0.24	0.39
(2)	T+S	0.22	0.36
(3)	T+S+C	0.21	0.34
(4)	Stacked T and S	0.22	0.37
(5)	Joint	0.26	0.43
(6)	T+(SC)	0.21	0.35
(7)	T+S+C+FG	0.20	0.34
(8)	T+S+C+FG+SG	0.19	0.33

Table 3: Ablation studies. T, S, C are temporal branch, spatial branch and cross branch respectively. FG is gated network for feature fusion and SG is sparse attention gate. Stacked T and S means spatial model after temporal model. T+(SC) means temporal branch is preserved and spatial and correlative interactions are learned together.

Ablation Studies

We conduct extensive ablative experiments to show the effectiveness of each component in our proposed method. The average performance of five sub-sets in ETH-UCY is shown in Table 3. T, S and C refer to temporal branch, spatial branch and cross branch respectively. FG is feature fusion gated network and SG is sparse attention gate for spatial and cross branches. For control groups without fusion gate, element-wise sum is used as fusion function.

The comparison between (1), (2) and (3) indicates that spatial interactions and correlation features are both useful in trajectory prediction. In instance (4), we use temporal transformer followed by spatial transformer to replace our proposed multi-stream representation learning module, and the result is worse than (2) and (3). This reveals that our proposed model can better handle spatio-temporal features compared with common methods used by previous models.

Then we investigate whether the spatio-temporal features can be learned together without disentangling. In (5), two dimensions are not distinguished and the spatio-temporal features are learned by directly applying self-attention on flattened sequence. The performance is much worse. This indicates each branch has different features and learning these features via multiple pathways is better. In (6), self-motion tendency and interaction with others are separated but interactions within the same frame and cross-frame interactions are mixed together. This setting is also similar with the agent-aware attention in AgentFormer. (3) is better than (6), showing that out proposed model can better handle spatio-temporal features of trajectories. Comparison between (3) and (7) reveals that fusing features by gated network can help the model adaptly focus on important features. The effectiveness of sparse attention gate is verified by (7) and (8).

Qualitative Analysis

Predicted Trajectory Visualization. We visualize the predicted trajectories in several scenarios to illustrate the effec-

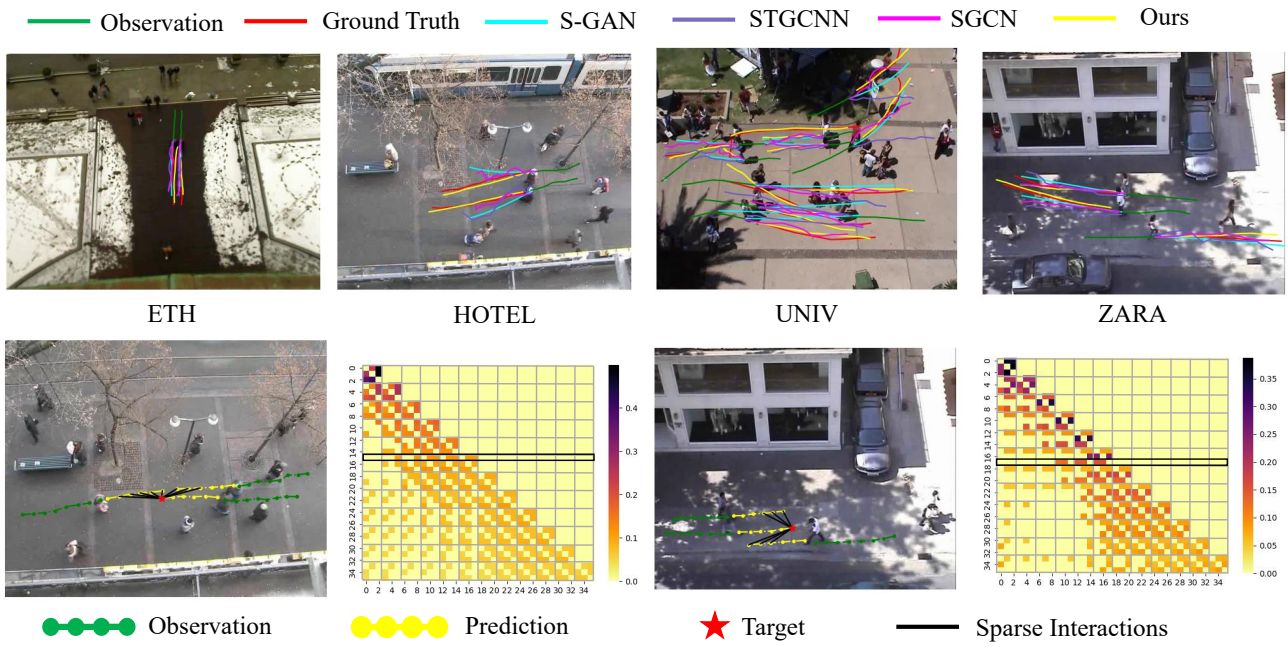

Figure 5: Visualizations of predicted future trajectory and sparse attention in typical scenarios

tiveness of our multi-stream representation learning module. The visualization results are shown in Figure 5. We compare our model with S-GAN, STGCNN and SGCN. The visualization result on ETH and HOTEL shows that our model can better handle the situation where pedestrians avoid collision or walk together. The prediction result is also smoother. This reveals that our model can capture the interplay between two dimensions. In UNIV, there are a great number of pedestrians. The interaction is presumably very complex and the motion tendency is hard to estimate. In this scenario, our model can still outperform prior models. The prediction visualization reveals that our model can learn more complex interactions effectively. As for ZARA, our model is able to produce trajectories closer to ground truth when abrupt changes in moving direction occur, which is shown in the top-right corner. Conceivably, our model can better learn the impacts of changes in motion trends on interactions. The visualizations indicate that our model can better capture spatio-temporal features of pedestrian trajectories and the correlation between spatial and temporal dimensions.

Visualization of interactions and correlations. We also visualize spatial interactions and cross-space-time correlations in typical scenarios in Figure 5. The sparse interactions and correlations for the target pedestrian in the 6^{th} prediction time step and the sparse attention weight matrices for corresponding scenarios are shown in the figure. The target pedestrian walks in group with another pedestrian while needing to avoid collision with the coming pedestrian. Apparently, our model can learn informative spatial interactions and correlations while picking out connections that have a major impact on the pedestrian's trajectory. Moreover, we visualize the sparse attention weight matrix of the multi-stream representation learning module in the future decoder, which shows the interaction and correlation weights within 12 future time steps. The spatial interaction weights and the cross correlation weights are put into one matrix. The weight matrix consists of T*T sub-blocks and each sub-block has the shape of N*N. The sub-blocks on the diagonal line show spatial interaction weights in each time step and other sub-blocks show the weights for cross-spatio-temporal correlations. The weights of spatial interactions and multidimensional correlations for target pedestrian at the 6^{th} time step are shown in the line marked by black rectangle in the matrix. As shown in Figure 5, besides spatial interactions, there are also some cross correlations distributed in sub-blocks that are not on the diagonal line. The weights in sub-blocks of cross correlations are more sparse than the weights in sub-blocks of spatial interactions. This reveals that spatial interactions and cross correlations are different and two kinds of connections should be processed in different pathways. Overall, our model can adaptively pay attention to spatial interactions and cross correlations and get better representations of spatio-temporal features in pedestrian trajectory.

Conclusion

In this paper, we propose to learn spatial, temporal and correlation features of pedestrian trajectory in three respective pathways and then adaptively integrate these features with learnable weights by a gated network. Sparse attention gate is also used to make the model focus on informative interactions and correlations. Extensive experiments show that our model achieves state-of-the-art performance and our multi-stream representation learning module can accurately and effectively model the trajectory features. It is expected that our framework can also be applied in more diversified prediction tasks other than pedestrian trajectory prediction.

Acknowledgments

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 Grants 2022JC-41 and 2021JQ-054, China Postdoctoral Science Foundation under Grant 2020M683490, Fundamental Research Funds for the Central Universities under Grants XTR042021005 and XTR072022001, and Applied Technology R&D Project of Beilin under Grant GX2129.

References

Alahi, A.; Goel, K.; Ramanathan, V.; Robicquet, A.; Fei-Fei, L.; and Savarese, S. 2016. Social LSTM: Human trajectory prediction in crowded spaces. In *CVPR*, 961–971.

Bai, S.; Kolter, J. Z.; and Koltun, V. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. In *ICLRW*.

Bhattacharyya, A.; Hanselmann, M.; Fritz, M.; Schiele, B.; and Straehle, C.-N. 2019. Conditional flow variational autoencoders for structured sequence prediction. In *NeurIPSW*.

Bock, J.; Krajewski, R.; Moers, T.; Runde, S.; Vater, L.; and Eckstein, L. 2020. The inD dataset: A drone dataset of naturalistic road user trajectories at german intersections. In *IV*, 1929–1934.

Buzsáki, G.; and Llinás, R. 2017. Space and time in the brain. *Science*, 358(6362): 482–485.

Duan, J.; Wang, L.; Long, C.; Zhou, S.; Zheng, F.; Shi, L.; and Hua, G. 2022. Complementary Attention Gated Network for Pedestrian Trajectory Prediction. In *AAAI*, 542–550.

Giuliari, F.; Hasan, I.; Cristani, M.; and Galasso, F. 2020. Transformer Networks for Trajectory Forecasting. In *ICPR*, 10335–10342.

Gupta, A.; Johnson, J.; Fei-Fei, L.; Savarese, S.; and Alahi, A. 2018. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *CVPR*, 2255–2264.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780.

Huang, Y.; Bi, H.; Li, Z.; Mao, T.; and Wang, Z. 2019. STGAT: Modeling spatial-temporal interactions for human trajectory prediction. In *ICCV*, 6272–6281.

Ivanovic, B.; and Pavone, M. 2019. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *ICCV*, 2375–2384.

Lea, C.; Flynn, M. D.; Vidal, R.; Reiter, A.; and Hager, G. D. 2017. Temporal convolutional networks for action segmentation and detection. In *CVPR*, 156–165.

Lerner, A.; Chrysanthou, Y.; and Lischinski, D. 2007. Crowds by example. *Computer Graphics Forum*, 26(3): 655–664.

Li, J.; Yang, F.; Tomizuka, M.; and Choi, C. 2020. EvolveGraph: Multi-agent trajectory prediction with dynamic relational reasoning. In *NeurIPS*, 19783–19794.

Liang, J.; and Hauptmann, A. 2020. Simaug: Learning robust representations from simulation for trajectory prediction. In *ECCV*, 275–292.

Liang, J.; Jiang, L.; Murphy, K.; Yu, T.; and Hauptmann, A. 2020. The garden of forking paths: Towards multi-future trajectory prediction. In *CVPR*, 10508–10518.

Liang, J.; Jiang, L.; Niebles, J. C.; Hauptmann, A. G.; and Fei-Fei, L. 2019. Peeking into the future: Predicting future person activities and locations in videos. In *CVPR*, 5725–5734.

Ma, Y. J.; Inala, J. P.; Jayaraman, D.; and Bastani, O. 2021. Likelihood-based diverse sampling for trajectory forecasting. In *ICCV*, 13279–13288.

Mangalam, K.; Girase, H.; Agarwal, S.; Lee, K.-H.; Adeli, E.; Malik, J.; and Gaidon, A. 2020. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In *ECCV*, 759–776.

Marchetti, F.; Becattini, F.; Seidenari, L.; and Del Bimbo, A. 2020. MANTRA: Memory Augmented Networks for Multiple Trajectory Prediction. In *CVPR*, 7143–7152.

Mohamed, A.; Qian, K.; Elhoseiny, M.; and Claudel, C. 2020. Social-STGCNN: A Social Spatio-Temporal Graph Convolutional Neural Network for Human Trajectory Prediction. In *CVPR*, 14424–14432.

Pang, B.; Zhao, T.; Xie, X.; and Wu, Y. N. 2021. Trajectory Prediction with Latent Belief Energy-Based Model. In *CVPR*, 11814–11824.

Pellegrini, S.; Ess, A.; Schindler, K.; and Van Gool, L. 2009. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*, 261–268.

Rangel, L. M.; Quinn, L. K.; and Chiba, A. A. 2016. *Space, Time, and the Hippocampus*, 59–75. Cham: Springer International Publishing. ISBN 978-3-319-15759-7.

Sadeghian, A.; Kosaraju, V.; Sadeghian, A.; Hirose, N.; Rezatofighi, H.; and Savarese, S. 2019. SoPhie: An attentive GAN for predicting paths compliant to social and physical constraints. In *CVPR*, 1349–1358.

Shi, L.; Wang, L.; Long, C.; Zhou, S.; Zheng, F.; Zheng, N.; and Hua, G. 2022. Social Interpretable Tree for Pedestrian Trajectory Prediction. In *AAAI*, 2235–2243.

Shi, L.; Wang, L.; Long, C.; Zhou, S.; Zhou, M.; Niu, Z.; and Hua, G. 2021. SGCN: Sparse graph convolution network for pedestrian trajectory prediction. In *CVPR*, 8994–9003.

Sun, J.; Jiang, Q.; and Lu, C. 2020. Recursive Social Behavior Graph for Trajectory Prediction. In *CVPR*, 660–669.

Sun, J.; Li, Y.; Fang, H.-S.; and Lu, C. 2021. Three steps to multimodal trajectory prediction: Modality clustering, classification and synthesis. In *ICCV*, 13250–13259.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*, 5998–6008.

Vemula, A.; Muelling, K.; and Oh, J. 2018. Social attention: Modeling attention in human crowds. In *ICRA*, 4601–4607.

Xu, C.; Li, M.; Ni, Z.; Zhang, Y.; and Chen, S. 2022a. GroupNet: Multiscale Hypergraph Neural Networks for Trajectory Prediction with Relational Reasoning. In *CVPR*, 6498–6507.

Xu, C.; Mao, W.; Zhang, W.; and Chen, S. 2022b. Remember Intentions: Retrospective-Memory-based Trajectory Prediction. In *CVPR*, 6488–6497.

Yu, C.; Ma, X.; Ren, J.; Zhao, H.; and Yi, S. 2020. Spatio-Temporal Graph Transformer Networks for Pedestrian Trajectory Prediction. In *ECCV*, 507–523.

Yuan, Y.; Weng, X.; Ou, Y.; and Kitani, K. 2021. AgentFormer: Agent-Aware Transformers for Socio-Temporal Multi-Agent Forecasting. In *ICCV*, 9813–9823.

Zhao, T.; Xu, Y.; Monfort, M.; Choi, W.; Baker, C.; Zhao, Y.; Wang, Y.; and Wu, Y. N. 2019. Multi-agent tensor fusion for contextual trajectory prediction. In *CVPR*, 12126–12134.

Zheng, F.; Wang, L.; Zhou, S.; Tang, W.; Niu, Z.; Zheng, N.; and Hua, G. 2021. Unlimited Neighborhood Interaction for Heterogeneous Trajectory Prediction. In *ICCV*, 13148–13157.