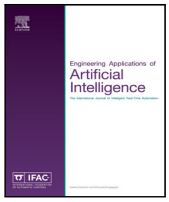
Engineering Applications of Artificial Intelligence 106 (2021) 104487

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



STENet: A hybrid spatio-temporal embedding network for human trajectory forecasting

Bo Zhang [a,b], Chengzhi Yuan [b], Tao Wang [b], Hongbo Liu [a,b,*]

[a] College of Artificial Intelligence, Dalian Maritime University, Dalian, 116026, China
[b] School of Information Science and Technology, Dalian Maritime University, Dalian, 116026, China

ARTICLE INFO

Keywords:
Trajectory forecasting
1D-CNN
Graph attention mechanism
Pedestrian grouping strategy
Hierarchical structure
Wasserstein distance

ABSTRACT

In this paper, we present a hybrid spatio-temporal embedding network (named as STENet) for human trajectory forecasting, which is built upon a GAN-based hierarchical framework. Differently from traditional approaches that only use LSTM for trajectory modeling, we exploit the 1D Convolutional Neural Network (1D-CNN) to embed position features at multiple temporal scales. Moreover, we propose a two-stage graph attention mechanism, which can better describe mutual interactions among pedestrians in the crowd. Additionally, group influences at every time step are taken into account as well. The overall framework is designed using a hierarchical manner, and trained using the Wasserstein distance. We carry out our experiments on the ETH and the UCY datasets. The corresponding results demonstrate the effectiveness of the proposed framework.

1. Introduction

Trajectory forecasting in the crowd has emerged a relevant topic in the domain of behavior analysis, which has been widely investigated by researchers in this field. Given the past motion histories, the task is to predict plausible future paths of pedestrians for successive time steps, which can be potentially applied in the areas of visual surveillance, navigation, abnormal event detection, and early warning, etc.

Trajectory forecasting remains challenging nowadays due to varied external factors, such as camera viewpoints, crowd densities, scene dynamics, mutual occlusions, and background clutters, etc. Furthermore, human behaviors tend to demonstrate multi-modal characteristics considering the specific situations they are facing. For example, in the traffic junction, people usually have multiple choices (such as stop, turn left/right, go straight, etc.) depending on their goals and personal preferences. Thus, a good predictive model requires not only proper tools for long-term sequential modeling, but also efficient ways to describe behavior uncertainties. Additionally, in realistic environments, people are not only driven by their personal intentions while walking, but also need to follow a series of implicit social rules that comply with common sense (such as standing in line at a ticket counter, avoiding collisions in an immediate distance, and following people in their front in the presence of over-crowded situations). Thus, the capability of modeling these rules is of sufficient importance in forecasting human trajectories in socially aware systems.

Currently, the mainstream approaches largely leverage on the standard encoder–decoder diagram, where recurrent neural networks (such as Recurrent Neural Networks (RNN), Gated Recurrent Unit (GRU), and Long–Short Term Memory (LSTM)) are exploited to model the temporal structures of trajectories. With the help of modern deep learning techniques, the uncertainties of trajectory data can be modeled by exploiting generative models (such as Generative Adversarial Network (GAN), Variational Auto-Encoder (VAE)), which are able to implicitly/explicitly infer the multi-modal distributions of the future paths. In addition, environmental cues (Sadeghian et al., 2019) can be taken into account as well, which will further promote the prediction performances. Although a lot of effort has been spent by the research community in the recent years, the work that fully addressed the spatial–temporal characteristics of human trajectories is still limited. Moreover, group information has always been ignored. Therefore, we propose a hybrid spatio-temporal embedding network for human trajectory prediction in this work, where the novel aspects are summarized as follows:

- As for temporal feature embedding, we combine the 1D-CNN and the LSTM modules to describe trajectory evolution, which can embed position features at multiple temporal scales.
- As for spatial feature embedding, we introduce a two-stage graph attention mechanism, which considers not only mutual interactions in the crowd, but also the self-influence with respect to each individual. Moreover, we incorporate a pedestrian clustering strategy into our framework, which is able to provide the spatial layout of pedestrians at the group level, since group information is critical for path planning as well.

* Correspondence to: Linghai Road 1, HighTech Area, Dalian, 116026, China.
E-mail addresses: bzhang@dlmu.edu.cn (B. Zhang), ycz1192607613@dlmu.edu.cn (C. Yuan), wangtao_@dlmu.edu.cn (T. Wang), lhb@dlmu.edu.cn (H. Liu).

https://doi.org/10.1016/j.engappai.2021.104487
Received 31 May 2021; Received in revised form 2 September 2021; Accepted 24 September 2021
Available online 11 October 2021

- We exploit the Wasserstein distance to train the model, which is a better alternative as compared to the Jensen–Shannon divergence (JS divergence) (Goodfellow et al., 2014) that adopted in the standard GAN-based models. It is helpful to deal with the phenomenon of *mode collapse* (Arjovsky and Bottou, 2017).

The rest of the paper is organized as follows: In Section 2, we review the recent literature in trajectory forecasting. In Section 3, we present the details of the proposed framework, including the generator, the discriminator, and how to train the model. Experimental results are demonstrated comprehensively in Section 4, where we provide ablation studies on the main components of the proposed framework, and make comparisons with other state-of-the-art methods. Finally, we conclude our work in Section 5.

2. Related work

Trajectory forecasting has been widely investigated by the research community, with a wide range of applications including crowd analysis (Wang et al., 2021), event prediction (Shi et al., 2019; Yuen and Torralba, 2010), and object tracking (Fernández-Sanjurjo et al., 2019; Fu et al., 2021), etc. In the following, we will briefly present the new progress in the recent years. More comprehensive overviews on the relevant topics can be found in the survey papers (Ahmed et al., 2018; Rudenko et al., 2020).

2.1. Trajectory clustering

Trajectory clustering provides an effective way to deal with group behaviors in the crowd. Typical methods include: (1) clustering by spatial distances (Junejo and Foroosh, 2007; Wang et al., 2011), and (2) clustering by spatial distributions (Wang et al., 2006). Ferreira et al. (2013) clustered trajectories by encoding the similarities among trajectories via the vector fields. AlZoubi et al. (2017) presented the so-called qualitative trajectory calculus (QTC) to encode the similarities among trajectories. Brankovic et al. (2020) presented a new algorithm for trajectory clustering based on the Fréchet distance. In Wang et al. (2017) and Portugal et al. (2017), spatio-temporal trajectory patterns were taken into account. Hu et al. (2013) proposed an incremental clustering algorithm by leveraging on the Dirichlet Process Mixture Model (DPMM). Xu et al. (2015) proposed a shrinkage-based framework for trajectory clustering, which runs in an unsupervised manner. Lawal et al. (2016) presented a support vector clustering algorithm for motion clustering. Mahrsi and Rossi (2012) proposed a modularity-based clustering algorithm, which leverages on hierarchical graphs. Yue et al. (2020) proposed a deep embedded trajectory clustering network, which introduces a new oriented loss to constrain the clustering assignment.

As for applications, trajectory clustering can be effectively used in many down-streaming tasks. Ge et al. (2012) integrated a bottom-up hierarchical clustering method into the object tracking framework, achieving very promising performances. Zhong et al. (2015) exploited the k-means algorithm to cluster trajectories, which is able to learn behavior patterns for crowd simulations. Chen and Corso (2015) exploited the spatio-temporal trajectory clustering algorithms for action detection. Shen et al. (2018) presented an effective clustering method by utilizing submodular maximization, which has been successfully applied to motion segmentation. Tokmakov et al. (2020) utilized trajectory clustering for action recognition. Doshi and Yilmaz (2020) proposed an unsupervised framework for anomaly detection in traffic videos, which can locate suspicious regions by k-means clustering.

2.2. Sequence modeling and trajectory prediction

Sequence modeling is the pre-request for a good long-term predictive model. In its early stages, researchers can only deal with one-step prediction by exploiting the first-order Markov model. However, this is not sufficient in many realistic applications. In the last decade, with the help of recurrent neural networks (such as RNN, LSTM, and GRU), long-term sequence modeling becomes applicable, which have been considered as the fundamental tools for trajectory prediction. The most popular way is to design an encoder–decoder framework, where the motion history and the future trajectory can be described by recurrent neural networks properly.

Moreover, it is also assumed that social influences have significant impacts on an individual' decision. Among the off-the-shelf solutions, the so-called social force model (SFM) (Helbing and Molnar, 1995) was the first attempt for human–human interaction modeling, which can be applied to multi-target tracking in the crowd (Pellegrini et al., 2009). The only limitation is that the features in this model are handcrafted, which need proper priors. In the recent years, due to the rapid development of attention mechanisms, social interactions can be readily embedded into the prediction framework in a data-driven manner. Alahi et al. (2016) proposed the Social-LSTM for trajectory prediction, where the social pooling layer was introduced to model mutual interactions in the neighborhood of a target pedestrian. Bartoli et al. (2018) presented a context-aware recurrent neural network, where both human–human and human–space interactions were jointly taken into account. Vemula et al. (2018) exploited the structural RNN for trajectory prediction, which can capture the importance of each person via spatial–temporal graphs. Mohamed et al. (2020) introduced a novel spatio-temporal graph CNN, allowing interaction modeling through graphical models. Giuliari et al. (2020) leveraged on the transformer model (Vaswani et al., 2017) for path prediction, which utilized the temporal attention mechanism to build the internal structure of a trajectory.

Another essential aspect is the multi-modal characteristics of behaviors, since people could have varied choices towards the situations in their front. This kind of uncertainty can be well described by probabilistic models. Particularly, generative models are widely used nowadays, such as the generative adversarial networks, the variational auto-encoders, and the flow-based generative models. Gupta et al. (2018) presented the Social GAN model, which exploited generative adversarial networks to produce diverse future paths. In Sadeghian et al. (2019), the so-called Sophie model was proposed, which adopted a similar GAN-based mechanism, and further took environmental cues into consideration. Amirian et al. (2019) embedded the attention pooling mechanism into the Info-GAN framework (Chen et al., 2016). Lee et al. (2017) proposed the so-called DESIRE model, which utilized the conditional variational auto-encoder (CVAE) for multi-modal trajectory prediction. Yuan and Kitani (2019) presented another VAE-based framework, which can produce multiple future paths by learning the diversity sampling function (DSF). Li et al. (2019), proposed a conditional generative neural system (CGNS) to estimate future trajectory distributions through adversarial training. Ma et al. (2021) presented a hierarchical discriminative framework, which can learn multi-modal trajectory distributions explicitly. Ouyang et al. (2018) utilized generative adversarial network for trajectory generation, which can capture semantic motion patterns among pedestrians.

3. Methodology

In this section, we first present the overall framework, and then introduce the details of the generator and the discriminator (including the pedestrian grouping strategy, the 1D-CNN module, and the proposed graph attention mechanism). Finally, we discuss about how to train the model.

3.1. Framework

Let $\mathbf{X}=\{X_1, X_2, \ldots, X_n\}$ be the observation sequences of all the pedestrians in the scene. The trajectory of person i is defined as: $X_i=\{(x_i^t, y_i^t); t=1, 2, \ldots, T_{obs}\}$, where T_{obs} is the length of the observation. The objective is to generate the future trajectories $\hat{\mathbf{Y}}=\{\hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_n\}$ of all the pedestrians, where $\hat{Y}_i=\{(x_i^t, y_i^t); t = T_{obs}+1, T_{obs}+2, \ldots, T_{obs}+T_{pred}\}$. T_{pred} represents the length of the prediction sequence. Moreover, we use $\mathbf{Y}=\{Y_1, Y_2, \ldots, Y_n\}$ to indicate the corresponding ground truths, allowing the model to be trained in a supervised manner.

The whole framework is presented in Fig. 1, which is comprised by the generator and the discriminator. The generator consists of the encoder and the decoder, respectively, where the former is used to encode the observation sequences and the later is used to produce plausible future paths from the noise. The discriminator is able to measure the similarity between the generated trajectories and the real ones.

3.2. Generator

As for the encoder, the observation sequences are first processed by the fully-connected layer, which can map a 2D coordinate (x_i^t, y_i^t) with respect to person i at the time step t into a high-dimensional space as shown in Eq. (1). We use W_e to indicate the embedding weights, and use V_i^t to represent the result of this position embedding. On one hand, V_i^t will be used for pedestrian grouping, which is able to provide the spatial layout of the crowd at the group level. On the other hand, V_i^t will be processed by the 1D-CNN and the LSTM modules in sequence, for the purpose of trajectory modeling along the temporal dimension. Furthermore, we introduce a two-stage graph attention mechanism into the encoder, which can better account for social interactions in the crowd.

$$V_i^t = \mathrm{FC}((x_i^t, y_i^t); W_e). \tag{1}$$

As for the decoder, it consists of a series of LSTMs that organized hierarchically. The hidden state of the bottom LSTM is initialized using the concatenation of the random noise z and the output of the encoder. The input of the bottom LSTM is initialized using the coordinates of the last time step in the observation sequence, namely $(x_i^{T_{obs}}, y_i^{T_{obs}})$. During the decoding procedure, the output of the bottom LSTM will be first processed by the graph attention and the grouping modules, and then further propagated to the next time step along the horizontal direction. At the vertical direction, the output of the LSTM will be processed from bottom to top, where at the top layer the hidden state will be finally decoded as the future position at the next time step.

In order to better clarify the whole process, we provide a detailed description of the main components in the generator hereafter.

3.2.1. Pedestrian grouping

Pedestrian grouping is formulated using Eq. (2), where N represents the number of agents in the scene. Grouping(·) indicates the clustering strategy at a given time step t, whose outputs are the cluster centers, annotated as $(c_1^t, c_2^t, \ldots, c_k^t)$.

$$(c_1^t, c_2^t, \ldots, c_k^t) = \mathrm{Grouping}(V_1^t, V_2^t, \ldots, V_N^t). \tag{2}$$

In our implementation, we exploit the MeanShift (Comaniciu and Meer, 1999) algorithm as the clustering strategy. Examples of the grouping results are shown in Fig. 2. Comparing to the standard k-means approach, the MeanShift algorithm does not need to fix the number of clusters in advance. We find that embedding the spatial layout at the group level can promote the prediction performances, which will be further validated in the experimental section.

3.2.2. Trajectory modeling

Trajectory evolution is modeled using Eq. (3). We use 1D-CNN(·) to indicate the 1D convolutional neural network, whose weight matrix is annotated by W_{cnn}. The 1D-CNN model takes the position embedding of person i at multiple time steps as the input, namely $\{V_i^{t-r}, V_i^{t-r+1}, \ldots, V_i^t, \ldots, V_i^{t+r}\}$, where r indicates the window size. The obtained result is denoted as p_i^t, which will be used as the input of the LSTM in the encoder. We use LSTM(·) to indicate the LSTM model, whose weight matrix is annotated by W_{lstm}. h_i^t is the hidden state of the LSTM.

$$\begin{aligned} p_i^t &= \text{1D-CNN}((V_i^{t-r}, \ldots, V_i^t, \ldots, V_i^{t+r}); W_{\mathrm{cnn}}); \\ h_i^t &= \mathrm{LSTM}(h_i^{t-1}, p_i^t; W_{\mathrm{lstm}}). \end{aligned} \tag{3}$$

In the following, we will present the implementation details of the 1D-CNN model, which is able to integrate the position embedding at multiple temporal scales. At the time step t, we apply multiple 1D convolution kernels to process the observation sequence as shown in Eq. (4), where w_{k1}, w_{k3}, w_{k5} are the weights corresponding to kernel sizes 1, 3, and 5. A_i^t, B_i^t, C_i^t are the obtained results, which will be further concatenated in order to generate the final output p_i^t.

$$\begin{aligned} A_i^t &= w_{k1}^t \times V_i^t, \\ B_i^t &= \textstyle\sum_{m=-1}^{1} w_{k3}^{t-m} \times V_i^{t+m}, \\ C_i^t &= \textstyle\sum_{m=-2}^{2} w_{k5}^{t-m} \times V_i^{t+m}, \\ p_i^t &= \mathrm{concate}(A_i^t, B_i^t, C_i^t). \end{aligned} \tag{4}$$

3.2.3. Social interaction modeling

According to the standard graph attention implementation (Veličković et al., 2017), the mutual influence $\alpha_{i,j}^t$ with respect to any pair of persons i and j at the time step t can be computed using Eq. (5), where h_i^t and h_j^t are the hidden state of the LSTM, respectively. W_{FC} indicates the weight matrix of the fully-connected layer.

$$\alpha_{i,j}^t = \mathrm{softmax}(\mathrm{FC}(\mathrm{concate}(h_i^t, h_j^t); W_{\mathrm{FC}})), \ (i \neq j). \tag{5}$$

In our framework, we exploit a two-stage graph attention mechanism. At the first stage, we consider the overall impact γ_i^t that surrounding agents exert on person i, which can be computed as in Eq. (6):

$$\gamma_i^t = \textstyle\sum_{j=1}^{N} \alpha_{i,j}^t, \ (i \neq j). \tag{6}$$

At the second stage, we incorporate the self-influence with respect to each person, which can be further formulated as in Eq. (7), where W_{FC} indicates the weight matrix of the fully-connected layer, and β_i^t indicates the attention value.

$$\beta_i^t = \mathrm{softmax}(\mathrm{FC}(\mathrm{concate}(h_i^t, \gamma_i^t); W_{\mathrm{FC}})). \tag{7}$$

We use GAT(·) to describe the operations mentioned above as shown in Eq. (8), where h_1^t, h_2^t, …, h_N^t are the hidden representations of all the N pedestrians in the scene.

$$\beta_i^t = \mathrm{GAT}(h_1^t, h_2^t, \ldots, h_N^t). \tag{8}$$

Finally, we concatenate the outputs of the grouping and the social attention modules, in order to generate the output o_i^t of the encoder as shown in Eq. (9), where g_i^t is the cluster center that person i belongs to. When $t = T_{obs}$, the obtained $o_i^{T_{obs}}$ can be viewed as the compressed representation of the motion history.

$$o_i^t = \mathrm{FC}(\mathrm{concate}(g_i^t, \beta_i^t \cdot h_i^t); W_{\mathrm{FC}}). \tag{9}$$

3.3. Discriminator

The discriminator $D(\cdot)$ is used to measure the similarity between the predicted trajectory and the corresponding ground truth. In our implementation, the discriminator is built upon the 1D-CNN and the LSTM modules as shown in Fig. 1. We apply another fully-connected layer on top of the LSTM, and the corresponding result is considered as the output of the discriminator.

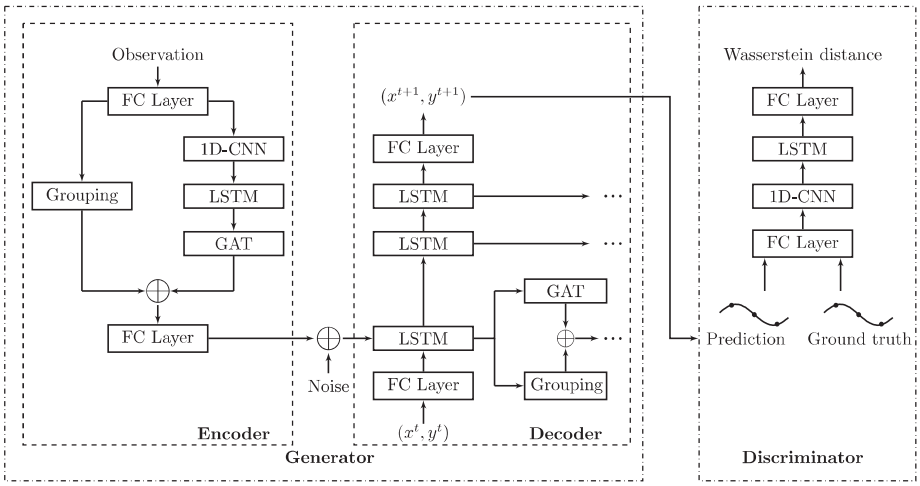


Fig. 1. The overall framework. The generator consists of the encoder and the decoder, where the encoder is designed using the pedestrian grouping module, the 1D-CNN module, and the two-stage graph attention mechanism (annotated as GAT). The decoder is built upon a series of LSTMs that organized hierarchically. The discriminator is built upon the 1D-CNN and the LSTM modules, which can be used to measure the similarity between the predicted trajectories and the real ones.

Fig. 2. Pedestrian grouping. The pictures are taken from the ETH and the UCY datasets in realistic environments.

3.4. Model training

The training objective is to minimize the difference between the real data distribution and the distribution of the generated trajectories. Motivated by the recent progress in image generation (Arjovsky et al., 2017), we exploit the so-called Wasserstein distance (or the *earth-mover* distance) to design the loss functions. The advantages are summarized as follows: (1) the JS divergence is a constant when the overlap of the two distributions is negligible, whereas the Wasserstein distance does not have this problem; and (2) the JS divergence will lead to the phenomenon of *mode collapse*.

According to Arjovsky et al. (2017), the Wasserstein distance can be approximated by maximizing L_W, which is defined as in Eq. (10), where $p(Y)$ and $p(\hat{Y})$ represent the real data distribution and the distribution of the generated trajectories. $f_w(\cdot)$ can be viewed as the discriminator $D(\cdot)$.

$$L_W = \mathbb{E}_{Y\sim p(Y)}[f_w(Y)] - \mathbb{E}_{\hat{Y}\sim p(\hat{Y})}[f_w(\hat{Y})]. \tag{10}$$

The model can be trained in an alternative way: (a) keep the discriminator $D(\cdot)$ fixed, and then minimize the loss function with respect to the generator as shown in Eq. (11); (b) keep the generator fixed, and then minimize the loss function with respect to the discriminator as shown in Eq. (12).

$$Loss_{Generator} = -E_{\hat{Y}\sim p(\hat{Y})}[D(\hat{Y})] + min_k \|Y - \hat{Y}^{(k)}\|_2. \tag{11}$$

$$Loss_{Discriminator} = -E_{Y\sim p(Y)}[D(Y)] + E_{\hat{Y}\sim p(\hat{Y})}[D(\hat{Y})]. \tag{12}$$

Particularly, we add the term $min_k \|Y - \hat{Y}^{(k)}\|_2$ in Eq. (11), which selects the prediction $\hat{Y}$ that is the closest to the ground truth Y among all the k hypotheses. More detailed descriptions with respect to the training procedure can be found in Algorithm 1. We use θ and w to represent the weights of the generator and discriminator, respectively. g_θ and g_w are the corresponding gradients. The function clip$(\cdot)$ and RMSprop$(\cdot)$ represent the weight clipping operation (Arjovsky et al., 2017) and the optimizer used in our work.

Algorithm 1: Model Training Procedure

Input: the learning rate of the generator α_θ; the learning rate of the discriminator α_w; the clipping parameter c; the number of epochs M; the current time step t.
Output: θ, w.
while $t < M$ **do**
  Update the discriminator using the following equations:
  $g_w \leftarrow \nabla_w[-E_{Y\sim p(Y)}[D(Y)] + E_{\hat{Y}\sim p(\hat{Y})}[D(\hat{Y})]]$;
  $w \leftarrow w - \alpha_w \cdot \text{RMSprop}(w, g_w)$;
  $w \leftarrow \text{clip}(w, -c, c)$;
  Update the generator using the following equations:
  $g_\theta \leftarrow -\nabla_\theta[-E_{\hat{Y}\sim p(\hat{Y})}[D(\hat{Y})] + min_k||Y - \hat{Y}^{(k)}||_2]$;
  $\theta \leftarrow \theta - \alpha_\theta \cdot \text{RMSprop}(\theta, g_\theta)$;
  $t = t + 1$;
end while

4. Experimental results

In this section, we first introduce the benchmark datasets, the evaluation protocols, and the hyper-parameters used in our experiments. Then, we demonstrate the dynamics of the training process. Next, we provide ablation studies on the main components of the framework. Finally, we compare our model with other baseline approaches, and visualize the prediction results.

4.1. Benchmark datasets

In this work, the ETH and the UCY datasets are used as the benchmark datasets. The ETH dataset (Pellegrini et al., 2009) has two subsets (eth and hotel), and the UCY dataset (Lerner et al., 2007) has three subsets (zara-1, zara-2, and univ). These two datasets contain thousands of walking pedestrians in the real-world scenes. All the videos are

Table 1
Prediction performances on the ETH and the UCY datasets (ADE/FDE), where the JS divergence and the Wasserstein distance are used for comparisons.

Dataset	T_{pred} = 8		T_{pred} = 12	
	JS distance	Wasserstein distance	JS distance	Wasserstein distance
eth	0.33/**0.48**	**0.28**/0.53	0.45/0.83	**0.38/0.79**
hotel	0.24/0.39	**0.21/0.32**	0.27/0.50	**0.24/0.46**
univ	0.24/0.54	**0.22/0.47**	0.31/0.64	**0.28/0.54**
zara-1	0.21/0.44	**0.17/0.39**	0.26/0.46	**0.24/0.45**
zara-2	0.15/0.33	**0.14/0.29**	0.21/0.43	**0.16/0.31**
avg	0.23/0.44	**0.20/0.40**	0.30/0.57	**0.26/0.51**

Table 2
Evaluations on the 1D-CNN for trajectory prediction in terms of ADE/FDE metrics, where we set T_{obs} = 8, and test T_{pred} = 8 and 12, respectively. The experiments are carried out on the ETH and the UCY datasets.

Dataset	T_{pred} = 8		T_{pred} = 12	
	without 1D-CNN	with 1D-CNN	without 1D-CNN	with 1D-CNN
eth	0.36/0.58	**0.28/0.53**	0.44/0.90	**0.38/0.79**
hotel	0.25/0.45	**0.21/0.32**	0.33/0.60	**0.24/0.36**
univ	0.28/0.66	**0.22/0.47**	0.30/0.56	**0.28/0.54**
zara-1	0.22/0.45	**0.17/0.39**	0.30/0.54	**0.24/0.45**
zara-2	0.21/0.44	**0.14/0.29**	0.26/0.35	**0.16/0.31**
avg	0.26/0.52	**0.20/0.40**	0.33/0.59	**0.26/0.51**

collected from the bird-eye viewpoints, including crowd behaviors like people walking together, group movements, and collision avoidance, etc. We use the 5-fold cross-validation strategy for model training and validation.

4.2. Evaluation protocols and parameter settings

The average displacement error (ADE) and the final displacement error (FDE) are exploited as the evaluation metrics (the lower, the better). We set T_{obs} to 8, which implies that 8 discrete time steps are used for observation. T_{pred} is set to 8 and 12, respectively, where we would like to evaluate the performances with respect to varied prediction lengths. k is set to 20 in Eq. (11), which uses the same value as other baseline approaches do (Giuliari et al., 2020; Gupta et al., 2018; Mangalam et al., 2020). Regarding the hidden state of the LSTM modules in the encoder, we set its hidden dimension to 64. As for the stacked LSTMs (3 layers in total) in the decoder, we set the hidden dimension to 64 for every LSTM block. The learning rates of the generator α_θ and the discriminator α_w are set to 1e-3 and 1e-4, respectively. The number of epochs M is set to 100, and the batch size is set to 64.

4.3. Model training

The RMSprop algorithm is used as the optimizer in our experiment. In Fig. 3, we demonstrate the loss functions with respect to the generator and the discriminator, showing that the training procedure converges consistently at every fold of the cross-validation.

Moreover, we also compare the prediction results that use the JS divergence and the Wasserstein distance for model training, respectively, where the Wasserstein distance shows better performances as shown in Table 1.

4.4. 1D-CNN

In this section, we would like to demonstrate the advantages of using the 1D-CNN module. For comparison, we present the experimental results by removing it from the overall framework. It can be seen clearly in Table 2 that the 1D-CNN can promote the prediction accuracy on all the subsets significantly in both settings.

Table 3
Evaluations on the pedestrian grouping strategy for trajectory prediction in terms of ADE/FDE metrics, where we set T_{obs} = 8, and test T_{pred} = 8 and 12, respectively. The experiments are carried out on the ETH and the UCY datasets.

Dataset	T_{pred} = 8		T_{pred} = 12	
	without grouping	with grouping	without grouping	with grouping
eth	0.33/0.67	**0.28/0.53**	0.42/0.83	**0.38/0.79**
hotel	0.23/0.47	**0.21/0.32**	0.25/**0.46**	**0.24/0.46**
univ	0.27/0.63	**0.22/0.47**	0.35/0.65	**0.28/0.54**
zara-1	0.19/**0.36**	**0.17**/0.39	0.25/**0.45**	**0.24/0.45**
zara-2	0.17/0.37	**0.14/0.29**	0.21/0.40	**0.16/0.31**
avg	0.24/0.50	**0.20/0.40**	0.30/0.56	**0.26/0.51**

Table 4
Evaluations on the proposed graph attention mechanism in terms of ADE/FDE metrics (T_{pred} = 12), where GAT represents *graph attention*. The experiments are carried out on the ETH and the UCY datasets.

Dataset	without GAT	social pooling	naive GAT	the proposed GAT
eth	0.40/0.84	0.43/0.89	0.41/0.85	**0.38/0.79**
hotel	0.26/0.47	0.30/0.59	0.25/0.54	**0.24/0.46**
univ	0.29/0.57	0.30/0.66	0.31/0.57	**0.28/0.54**
zara-1	0.27/0.50	0.29/0.51	0.24/0.47	**0.24/0.45**
zara-2	0.25/0.32	0.27/0.53	0.24/0.33	**0.16/0.31**
avg	0.29/0.54	0.32/0.64	0.29/0.55	**0.26/0.51**

Table 5
Evaluations on the proposed graph attention mechanism in terms of ADE/FDE metrics (T_{pred} = 8), where GAT represents *graph attention*. The experiments are carried out on the ETH and the UCY datasets.

Dataset	without GAT	social pooling	naive GAT	the proposed GAT
eth	0.32/0.63	0.39/0.71	0.36/0.65	**0.28/0.53**
hotel	0.25/0.51	0.26/0.47	0.22/0.34	**0.21/0.32**
univ	0.25/0.61	0.25/0.63	0.23/0.58	**0.22/0.47**
zara-1	0.23/0.51	0.24/0.55	0.21/0.45	**0.17/0.39**
zara-2	0.20/0.40	0.23/0.46	0.20/0.37	**0.14/0.29**
avg	0.25/0.53	0.27/0.56	0.24/0.50	**0.20/0.40**

4.5. Pedestrian grouping

In this section, we will evaluate the influences of pedestrian grouping in the prediction task. To this end, we conduct an ablation study. The corresponding results are reported in Table 3, showing that the group clustering strategy is effective in the prediction task as well.

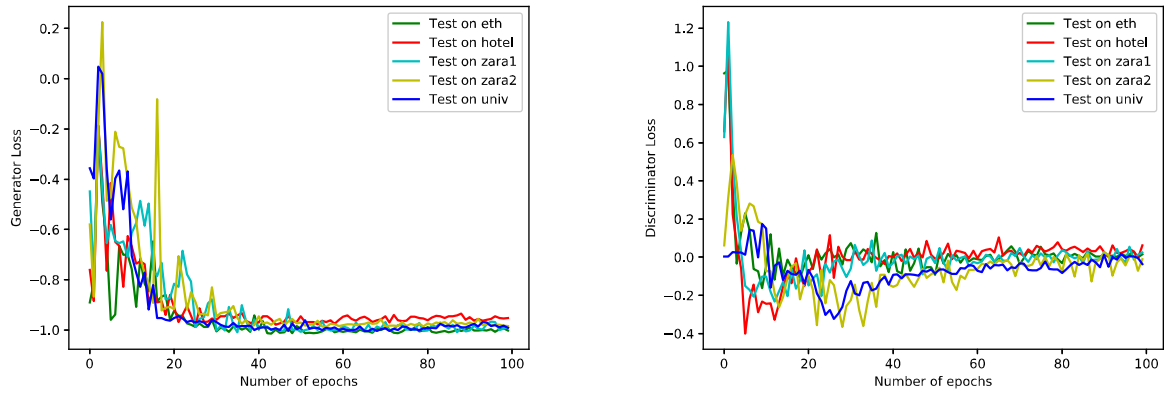
4.6. Graph attention mechanism

In this section, we will evaluate on the role of the proposed graph attention mechanism. To this aim, we arrange the experiments as follows: (1) as a baseline, we remove the graph attention module from the framework; (2) for comparison, we use the so-called social pooling (Alahi et al., 2016) as the attention mechanism, which is a typical spatial attention implementation; (3) in addition, we also compare the prediction performances using the naive graph attention implementation that presented in Veličković et al. (2017).

The corresponding results are reported in Table 4 (T_{pred} = 12) and Table 5 (T_{pred} = 8), respectively, from where we can find that the proposed graph attention mechanism can achieve better performances consistently on all the subsets.

4.7. Hierarchical architecture

In our framework, the decoder is designed using the stacked LSTMs. Although there is no theoretically proof that why multi-layer structure can promote the prediction performances, it has been found on some specific applications (such as machine translation (Barone et al., 2017; Goldberg, 2016; Meng and Zhang, 2019; Prakash et al., 2016), traffic state forecasting (Cui et al., 2020; Li et al., 2021), and action recognition (Azzam et al., 2020), etc.) that deep recurrent neural networks can



(a) (b)

Fig. 3. The loss functions in the training procedure. It can be seen clearly that both the losses of the generator and the discriminator will converge after a certain number of epochs. (a) the loss of the generator; (b) the loss of the discriminator.

Table 6
Evaluations on the hierarchical structure of the decoder in terms of ADE/FDE. The experiments are carried out on the ETH and the UCY datasets.

Dataset	$T_{pred} = 8$			$T_{pred} = 12$		
	1-layer	2-layers	3-layers	1-layer	2-layers	3-layers
eth	0.31/0.61	0.28/0.55	**0.28/0.53**	0.45/0.87	0.40/0.81	**0.38/0.79**
hotel	0.25/0.47	0.22/0.34	**0.21/0.32**	0.28/0.53	0.25/0.50	**0.24/0.36**
univ	0.28/0.63	0.25/0.56	**0.22/0.47**	0.31/0.59	0.29/0.55	**0.28/0.54**
zara-1	0.22/0.43	0.18/0.41	**0.17/0.39**	0.27/0.50	0.26/0.47	**0.24/0.45**
zara-2	0.17/0.35	0.15/0.31	**0.14/0.29**	0.19/0.37	0.18/0.32	**0.16/0.31**
avg	0.25/0.50	0.22/0.43	**0.20/0.40**	0.30/0.57	0.28/0.53	**0.26/0.51**

Table 7
Comparisons with other state-of-the-art approaches in terms of ADE/FDE metrics ($T_{pred} = 12$) on the ETH and the UCY datasets.

Dataset	eth	hotel	zara-1	zara-2	univ	avg
Social LSTM (Alahi et al., 2016)	1.09/2.35	0.79/1.76	0.47/1.00	0.56/1.17	0.67/1.40	0.72/1.54
Social GAN (Gupta et al., 2018)	0.81/1.52	0.72/1.61	0.34/0.69	0.42/0.84	0.60/1.26	0.56/1.13
Sophie (Sadeghian et al., 2019)	0.70/1.43	0.76/1.67	0.30/0.63	0.38/0.78	0.55/1.31	0.54/1.15
CGNS (Li et al., 2019)	0.62/1.40	0.70/0.93	0.32/0.59	0.35/0.71	0.48/1.22	0.49/0.97
Social-BiGAT (Kosaraju et al., 2019)	0.69/1.29	0.49/1.01	0.30/0.62	0.36/0.75	0.55/1.32	0.48/1.00
TF (Giuliari et al., 2020)	0.61/1.12	**0.18**/0.30	**0.22/0.38**	0.17/0.32	0.35/0.65	0.31/0.55
PECNet (Mangalam et al., 2020)	0.54/0.87	**0.18/0.24**	**0.22**/0.39	0.17/0.30	0.35/0.60	0.29/**0.48**
STENet (Ours)	**0.38/0.79**	0.24/0.46	0.24/0.45	**0.16/0.31**	**0.28/0.54**	**0.26**/0.51

Table 8
Comparisons with other state-of-the-art approaches in terms of ADE/FDE metrics ($T_{pred} = 8$) on the ETH and the UCY datasets. Particularly, Sophie (Sadeghian et al., 2019), CGNS (Li et al., 2019), Social-BiGAT (Kosaraju et al., 2019), TF (Giuliari et al., 2020), and PECNet (Mangalam et al., 2020) do not provide the preliminary results in this setting.

Dataset	eth	hotel	zara-1	zara-2	univ	avg
Linear	0.84/1.60	0.35/0.60	0.41/0.74	0.53/0.95	0.56/1.01	0.54/0.98
LSTM	0.70/1.45	0.55/1.17	0.25/0.53	0.31/0.65	0.36/0.77	0.43/0.91
Social LSTM (Alahi et al., 2016)	0.73/1.48	0.49/1.01	0.27/0.56	0.33/0.70	0.41/0.84	0.45/0.91
Social GAN (Gupta et al., 2018)	0.60/1.19	0.48/0.95	0.21/0.42	0.27/0.54	0.36/0.73	0.38/0.77
STENet (Ours)	**0.28/0.53**	**0.21/0.32**	**0.17/0.39**	**0.14/0.29**	**0.22/0.47**	**0.20/0.40**

achieve better results than shallower structures. In order to highlight the impacts of the hierarchical architecture, we compare the prediction performances using varied number of LSTM layers. The corresponding results are presented as in Table 6. As compared to the single layer structure, increasing the number of LSTM layers can promote the prediction performances gradually.

4.8. Comparisons

In this section, we compare the proposed framework with other baseline approaches (such as Social GAN (Gupta et al., 2018), Sophie (Sadeghian et al., 2019), CGNS (Li et al., 2019), Social-BiGAT (Kosaraju et al., 2019), TF (Giuliari et al., 2020), and PECNet (Mangalam et al., 2020)). We present the corresponding results in Table 7 ($T_{pred} = 12$) and Table 8 ($T_{pred} = 8$), respectively, showing that the proposed framework can achieve promising prediction performances. Particularly, when $T_{pred} = 12$, we obtain competitive results on the subsets of eth, zara-2, and univ. As for hotel and zara-1, PECNet (Mangalam et al., 2020) is a better option. For $T_{pred} = 8$, our approach performs the best.

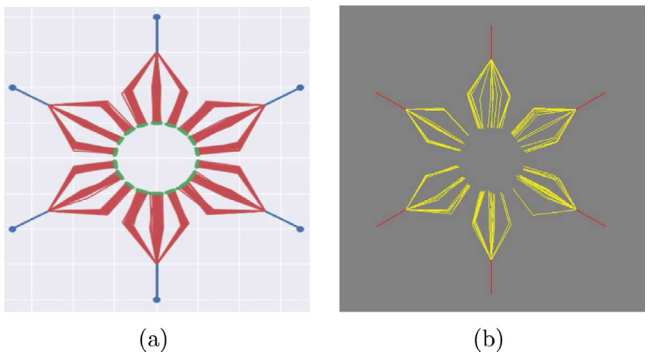


Fig. 4. Multi-modal trajectory prediction on the *toy* dataset. (a) the toy example. (b) the prediction results.

4.9. Visualization

First, we exploit the synthetic *toy* dataset (Amirian et al., 2019) for demonstration, showing that the proposed framework is able to generate multiple future trajectories. There are six groups of synthetic trajectories in this dataset, all starting from one specific point located along a circle. When approaching the circle center, they will further split into three different sub-directions, thus generating 18 varied motion modalities in total. The experimental results are shown in Fig. 4, from where we can observe that the proposed framework can produce trajectories moving towards three different directions with respect to each group successfully.

Secondly, we provide the qualitative results on the ETH and the UCY datasets in Fig. 5.

Finally, we visualize the prediction results of other typical benchmark methods for comparisons, including Social LSTM (Alahi et al., 2016), Social GAN (Gupta et al., 2018), and TF (Giuliari et al., 2020), etc, which are demonstrated in Fig. 6.

5. Conclusions and future work

In this work, we present a hybrid spatio-temporal embedding network for human trajectory forecasting, which is built upon the GAN-based encoder–decoder architecture. The main advantages of the proposed framework are summarized as follows: (1) we combine the 1D-CNN with the standard LSTM to model trajectory evolution along the temporal dimension, which can capture position features at varied temporal scales; (2) we integrate the pedestrian grouping strategy into our framework, for the purpose of better describing the spatial layout of agents in the scene; (3) as for interaction modeling, a two-stage graph attention mechanism is proposed, which can better account for social interactions among pedestrians; (4) the model is trained via the Wasserstein distance, which allows generating trajectories with multiple motion modalities. Additionally, the hierarchical architecture is exploited in the generator, which is able to further promote the prediction accuracy. Extensive experiments are carried out on the ETH and the UCY datasets, where we evaluate the roles of the main technical components (such as the 1D-CNN module, the grouping strategy, the attention mechanism, etc.) by ablation studies. Experimental results demonstrate the effectiveness of the proposed framework.

As for the future work, we would like to focus on the following two issues in order to further promote the model performances: (1) more efficient clustering strategy, which is able to group pedestrians in highly dense crowds accurately; (2) integrating environmental cues (such as roads, buildings) into the framework, which is able to better interpret the scene semantics and facilitate trajectory forecasting when avoiding obstacles.

CRediT authorship contribution statement

Bo Zhang: Conceptualization, Methodology, Writing. **Chengzhi Yuan:** Methodology, Software. **Tao Wang:** Validation. **Hongbo Liu:** Supervision, Reviewing and editing.

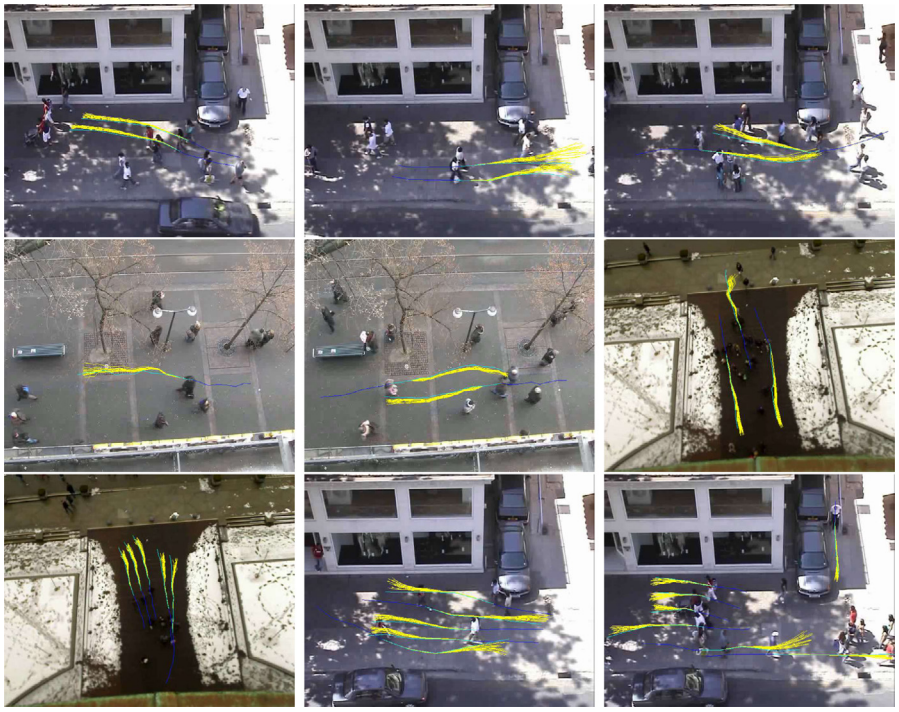
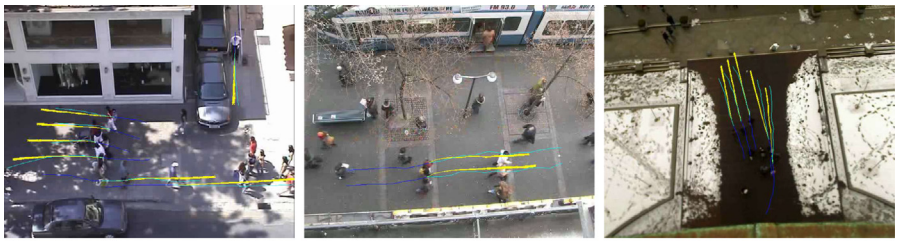
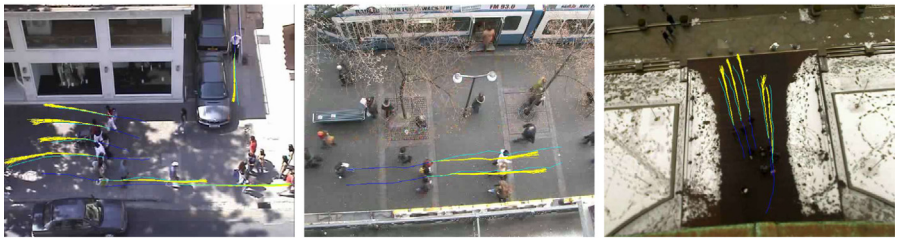


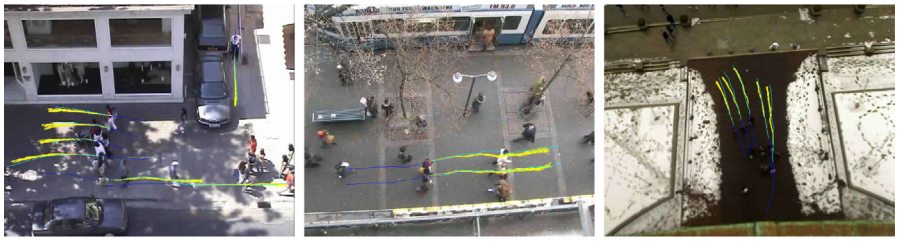
Fig. 5. Multi-modal trajectory prediction on the ETH and the UCY datasets.



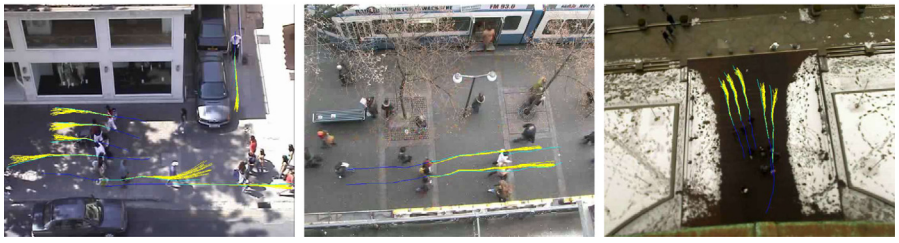
(a) Social LSTM



(b) Social GAN



(c) TF



(d) The proposed model

Fig. 6. Comparisons with other benchmark approaches, where 3 different scenarios are taken for demonstration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 61702073, 61772102, 62176036), the China Postdoctoral Science Foundation (Grant No. 2019M661079), and the Liaoning Collaborative Fund, China (Grant No. 2020-HYLH-17).

References

Ahmed, S.A., Dogra, D.P., Kar, S., Roy, P.P., 2018. Trajectory-based surveillance analysis: A survey. IEEE Trans. Circuits Syst. Video Technol. 29, 1985–1997.

Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S., 2016. Social LSTM: Human trajectory prediction in crowded spaces. In: The Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition. IEEE, pp. 961–971.

AlZoubi, A., Al-Diri, B., Pike, T., Kleinhappel, T., Dickinson, P., 2017. Pair-activity analysis from video using qualitative trajectory calculus. IEEE Trans. Circuits Syst. Video Technol. 28, 1850–1863.

Amirian, J., Hayet, J.B., Pettre, J., 2019. Social ways: Learning multi-modal distributions of pedestrian trajectories with GANs. In: The Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition Workshop. IEEE, pp. 2964–2972.

Arjovsky, M., Bottou, L., 2017. Towards principled methods for training generative adversarial networks. arXiv preprint arXiv:1701.04862.

Arjovsky, M., Chintala, S., Bottou, L., 2017. Wasserstein generative adversarial networks. In: The Proceedings of the International Conference on Machine Learning. PMLR, pp. 214–223.

Azzam, R., Alkendi, Y., Taha, T., Huang, S., Zweiri, Y., 2020. A stacked LSTM-based approach for reducing semantic pose estimation error. IEEE Trans. Instrum. Meas. 70, 1–14.

Barone, A.V.M., Helcl, J., Sennrich, R., Haddow, B., Birch, A., 2017. Deep architectures for neural machine translation. arXiv preprint arXiv:1707.07631.

Bartoli, F., Lisanti, G., Ballan, L., Del Bimbo, A., 2018. Context-aware trajectory prediction. In: The Proceedings of the IEEE International Conference on Pattern Recognition. IEEE, pp. 1941–1946.

Brankovic, M., Buchin, K., Klaren, K., Nusser, A., Popov, A., Wong, S., 2020. (k, l)-medians clustering of trajectories using continuous dynamic time warping. In: the Proceedings of the International Conference on Advances in Geographic Information Systems, pp. 99–110.

Chen, W., Corso, J.J., Action detection by implicit intentional motion clustering. In: the Proceedings of the IEEE International Conference on Computer Vision, pp. 3298–3306.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P., 2016. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. arXiv preprint arXiv:1606.03657.

Comaniciu, D., Meer, P., 1999. Mean shift analysis and applications. In: The Proceedings of the IEEE International Conference on Computer Vision. IEEE, pp. 1197–1203.

Cui, Z., Ke, R., Pu, Z., Wang, Y., 2020. Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values. Transp. Res. C 118, 102674.

Doshi, K., Yilmaz, Y., Fast unsupervised anomaly detection in traffic videos. In: the Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition Workshop, pp. 624–625.

Fernández-Sanjurjo, M., Bosquet, B., Mucientes, M., Brea, V.M., 2019. Real-time visual detection and tracking system for traffic monitoring. Eng. Appl. Artif. Intell. 85, 410–420.

Ferreira, N., Klosowski, J.T., Scheidegger, C.E., Silva, C.T., 2013. Vector field k-means: Clustering trajectories by fitting multiple vector fields. In: Computer Graphics Forum. Wiley Online Library, pp. 201–210.

Fu, C., Ding, F., Li, Y., Jin, J., Feng, C., 2021. Learning dynamic regression with automatic distractor repression for real-time UAV tracking. Eng. Appl. Artif. Intell. 98, 104116.

Ge, W., Collins, R.T., Ruback, R.B., 2012. Vision-based analysis of small groups in pedestrian crowds. IEEE Trans. Pattern Anal. Mach. Intell. 34, 1003–1016.

Giuliari, F., Hasan, I., Cristani, M., Galasso, F., 2020. Transformer networks for trajectory forecasting. arXiv preprint arXiv:2003.08111.

Goldberg, Y., 2016. A primer on neural network models for natural language processing. J. Artificial Intelligence Res. 57, 345–420.

Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial networks. arXiv preprint arXiv:1406.2661.

Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A., 2018. Social GAN: Socially acceptable trajectories with generative adversarial networks. In: The Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition. IEEE, pp. 2255–2264.

Helbing, D., Molnar, P., 1995. Social force model for pedestrian dynamics. Phys. Rev. E 51 (4282).

Hu, W., Li, X., Tian, G., Maybank, S., Zhang, Z., 2013. An incremental dpmm-based method for trajectory clustering, modeling, and retrieval. IEEE Trans. Pattern Anal. Mach. Intell. 35, 1051–1065.

Junejo, I.N., Foroosh, H., 2007. Trajectory rectification and path modeling for video surveillance. In: The Proceedings of the IEEE International Conference on Computer Vision. IEEE, pp. 1–7.

Kosaraju, V., Sadeghian, A., Martín-Martín, I., Rezatofighi, S.H., Savarese, S., 2019. Social-BiGAT: Multimodal trajectory forecasting using bicycle-GAN and graph attention networks. arXiv preprint arXiv:1907.03395.

Lawal, I.A., Poiesi, F., Anguita, D., Cavallaro, A., 2016. Support vector motion clustering. IEEE Trans. Circuits Syst. Video Technol. 27, 2395–2408.

Lee, N., Choi, W., Vernaza, P., Choy, C.B., Torr, P.H., Chandraker, M., 2017. DESIRE: Distant future prediction in dynamic scenes with interacting agents. In: the Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, pp. 336–345.

Lerner, A., Chrysanthou, Y., Lischinski, D., 2007. Crowds by example. Comput. Graph. Forum 26, 655–664.

Li, J., Guo, F., Sivakumar, A., Dong, Y., Krishnan, R., 2021. Transferability improvement in short-term traffic prediction using stacked LSTM network. Transp. Res. C 124, 102977.

Li, J., Ma, H., Tomizuka, M., 2019. Conditional generative neural system for probabilistic trajectory prediction. arXiv preprint arXiv:1905.01631.

Ma, Y., Zhang, B., Conci, N., Liu, H., 2021. A hierarchical framework for motion trajectory forecasting based on modality sampling. In: IEEE ICPR Workshops and Challenges: Virtual Event, January (2021) 10–15, Part IV, pp. 235–249.

Mahrsi, M.K.E., Rossi, F., 2012. Modularity-based clustering for network-constrained trajectories. arXiv preprint arXiv:1205.2172.

Mangalam, K., Girase, H., Agarwal, S., Lee, K.H., Adeli, E., Malik, J., Gaidon, A., 2020. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In: The Proceedings of the European Conference on Computer Vision. Springer, pp. 759–776.

Meng, F., Zhang, J., 2019. DTMT: A novel deep transition architecture for neural machine translation. In: the Proceedings of the AAAI Conference on Artificial Intelligence, pp. 224–231.

Mohamed, A., Qian, K., Elhoseiny, M., Claudel, C., 2020. Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In: the Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, pp. 14424–14432.

Ouyang, K., Shokri, R., Rosenblum, D.S., Yang, W., 2018. A non-parametric generative model for human trajectories. In: the Proceedings of the International Joint Conference on Artificial Intelligence, pp. 3812–3817.

Pellegrini, S., Ess, A., Schindler, K., Gool, L.V., 2009. You'll never walk alone: Modeling social behavior for multi-target tracking. In: The Proceedings of the IEEE International Conference on Computer Vision. IEEE, pp. 261–268.

Portugal, I., Alencar, P., Cowan, D., 2017. Developing a spatial–temporal contextual and semantic trajectory clustering framework. arXiv preprint arXiv:1712.03900.

Prakash, A., Hasan, S.A., Lee, K., Datla, V., Qadir, A., Liu, J., Farri, O., 2016. Neural paraphrase generation with stacked residual LSTM networks. arXiv preprint arXiv:1610.03098.

Rudenko, A., Palmieri, L., Herman, M., Kitani, K.M., Gavrila, D.M., Arras, K.O., 2020. Human motion trajectory prediction: A survey. Int. J. Robot. Res. 39, 895–935.

Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Savarese, S., 2019. Sophie: An attentive GAN for predicting paths compliant to social and physical constraints. In: The Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition. IEEE, pp. 1349–1358.

Shen, J., Peng, J., Shao, L., 2018. Submodular trajectories for better motion segmentation in videos. IEEE Trans. Image Process. 27, 2688–2700.

Shi, D., Zurada, J., Karwowski, W., Guan, J., Çkıt, E., 2019. Batch and data streaming classification models for detecting adverse events and understanding the influencing factors. Eng. Appl. Artif. Intell. 85, 72–84.

Tokmakov, P., Hebert, M., Schmid, C., 2020. Unsupervised learning of video representations via dense trajectory clustering. In: The Proceedings of the European Conference on Computer Vision. Springer, pp. 404–421.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. arXiv preprint arXiv:1706.03762.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., 2017. Graph attention networks. arXiv preprint arXiv:1710.10903.

Vemula, A., Muelling, K., Oh, J., 2018. Social attention: Modeling attention in human crowds. In: The Proceedings of the IEEE International Conference on Robotics and Automation. IEEE, pp. 1–7.

Wang, X., Ma, K.T., Ng, G.W., Grimson, W.E.L., 2011. Trajectory analysis and semantic region modeling using nonparametric hierarchical bayesian models. Int. J. Comput. Vis. 95, 287–312.

Wang, D., Ma, Q., Wang, N., Fan, X., Lu, M., Liu, H., 2021. AONet: Active offset network for crowd flow prediction. Eng. Appl. Artif. Intell. 97, 104022.

Wang, W., Shen, J., Xie, J., Porikli, F., 2017. Super-trajectory for video segmentation. In: the Proceedings of the IEEE International Conference on Computer Vision, pp. 1671–1679.

Wang, X., Tieu, K., Grimson, E., 2006. Learning semantic scene models by trajectory analysis. In: The Proceedings of the European Conference on Computer Vision. Springer, pp. 110–123.

Xu, H., Zhou, Y., Lin, W., Zha, H., 2015. Unsupervised trajectory clustering via adaptive multi-kernel-based shrinkage. In: the Proceedings of the IEEE International Conference on Computer Vision, pp. 4328–4336.

Yuan, Y., Kitani, K., 2019. Diverse trajectory forecasting with determinantal point processes. arXiv preprint arXiv:1907.04967.

Yue, M., Li, Y., Yang, H., Ahuja, R., Chiang, Y.Y., Shahabi, C., 2020. Detect: Deep trajectory clustering for mobility-behavior analysis. arXiv preprint arXiv:2003.01351.

Yuen, J., Torralba, A., 2010. A data-driven approach for event prediction. In: The Proceedings of the European Conference on Computer Vision. Springer, pp. 707–720.

Zhong, J., Cai, W., Luo, L., Yin, H., 2015. Learning behavior patterns from video: A data-driven framework for agent-based crowd modeling. In: the Proceedings of the International Conference on Autonomous Agents and Multiagent Systems, pp. 801–809.