A Novel Graph based Trajectory Predictor with Pseudo Oracle

Biao Yang, *Member*, *IEEE*, Guocheng Yan, Pin Wang, *Member*, *IEEE*, Chingyao Chan, *Member*, *IEEE*, Xiaofeng Liu, *Member*, Yang Chen, *Member*, *IEEE*, *IEEE*

Abstract—Pedestrian trajectory prediction in dynamic scenes remains a challenging and critical problem in numerous applications, such as self-driving cars and socially aware robots. Challenges concentrate on capturing pedestrians' social interactions and handling their future uncertainties. Pedestrians' head orientations can be used as an oracle that indicates relevant pedestrians [1], thus is beneficial to model social interactions. Moreover, latent variable distributions of pedestrians' future trajectories can be termed as another oracle. However, few works fully utilize these oracle information for an improved prediction performance. In this work, we propose GTPPO (*G*raph-based *T*rajectory *P*redictor with *P*seudo *O*racle), which is a generative model-based trajectory predictor. Pedestrians' social interactions are captured by the proposed GA^2T (*G*raph *A*ttention & social *A*ttention ne*T*work) module. Social attention is calculated on the basis of pedestrians' moving directions, which are termed as a pseudo oracle. Moreover, we propose a latent variable predictor to learn the latent variable distribution from observed trajectories. Such latent variable distribution reflects pedestrians' future trajectories, and therefore can be taken as another pseudo oracle. We compare the performance of GTPPO with several recently proposed methods on benchmarking datasets. Quantitative evaluations demonstrate that GTPPO outperforms state-of-the-art methods with lower average and final displacement errors. Qualitative evaluations show that GTPPO successfully recognizes the sudden motion changes since the estimated latent variable reflects the future trajectories.

Index Terms—trajectory prediction, latent variable predictor, social attention, graph attention network.
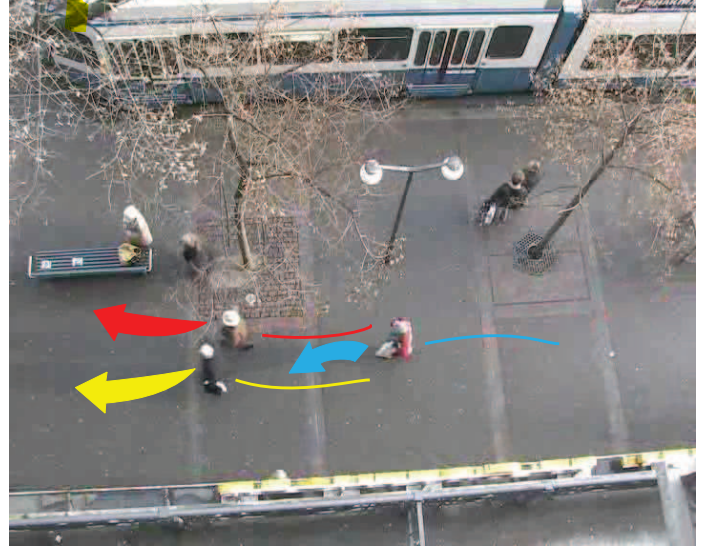
Fig. 1: Trajectory prediction. For three pedestrians in the scene, colored lines represent observed trajectories and the task is to predict their future trajectories marked with different colored arrows (best viewed in color).

I. INTRODUCTION

Pedestrian trajectory prediction in dynamic scenes remains a critical problem with numerous applications, such as self-driving cars [2] and socially aware robots [3]. As demonstrated in Fig. 1, pedestrians' future trajectories marked with different colored arrows should be predicted given the past trajectories marked with different colored lines. Such prediction is difficult due to various reasons, for example, the complex human—human and human—objects interactions existed in dynamic environments. Moreover, pedestrians' interpersonal and multi-modal properties pose critical challenges for accurate trajectory prediction.

Early works related to trajectory prediction always depend on linear, constant velocity, and constant acceleration models.

B. Yang, G. Yan and Y. Chen are with the Department of Information Science and Engineering, Changzhou University, Changzhou, 213000 China.

P. Wang and C. Chan are with the California PATH, University of California, Berkeley, Richmond, CA, 94804, USA.

X. Liu is with the College of IoT Engineering, Hohai University, Changzhou, 213000, China.

Corresponding author: X. Liu (xfliu@hhu.edu.cn)

However, these simple models show poor performance in complex environments. Later, several stochastic models, including Gaussian mixture regression [4], hidden Markov models [5], and dynamic Bayesian networks (DBNs) [6] are used to model pedestrians' motion patterns. However, it is difficult to design hand-crafted features that are useful in general cases. Recent works resort to deep neural networks, e.g. the recurrent neural network (RNN) [7], to automatically model pedestrians' motion patterns. Specifically, the long-short term memory (LSTM) network is used to encode pedestrians' motion patterns, and their future trajectories are estimated by sampling from the encoded hidden states [8]. Later, a social generative adversarial network (SGAN) is proposed to generate socially acceptable and multi-modal trajectories [9]. Many recent studies focus on better modeling pedestrians' social interactions. Meanwhile, great efforts concentrate on generating a knowledge-rich latent variable for an improved prediction performance. A common practice is learning knowledge from image data by semantic segmentation [10] or object detection [11]. However, such a learning process drastically increases computing overhead.

Though remarkable progress has been made over recent years, there are still two shortages which influence the tra-

jectory prediction performance. First, few works utilize the correlation between pedestrians' head orientations and their social interactions. A common sense is that a pedestrian's future trajectory is always influenced by pedestrians in front. We call them as relevant pedestrians. Utilizing such common sense is beneficial for social interactions modeling. Second, as far as we know, no work explores the latent variable distributions of future trajectories. Most generative model-based trajectory predictors use a random Gaussian noise as the latent variable to generate diverse outputs. Several works learn latent variable distributions from observed trajectories or image data. However, we verify that latent variable distributions learned from future trajectories are more useful for accurate trajectory prediction.

In this work, we propose GTPPO (**G**raph-based **T**rajectory **P**redictor with **P**seudo **O**racle), which is a generative model-based trajectory predictor. GTPPO consists of an LSTM-based encoder-decoder network and a latent variable predictor. In the encoder network, a GA^2T (**G**raph **A**ttention & social **A**ttention ne**T**work) module is proposed to capture pedestrians' social interactions. Specifically, the graph attention learned in the data-driven manner is used to aggregate neighbors' information. Then, the social attention calculated on the basis of pedestrians' moving directions is used to reinforce the correlations between relevant pedestrians. Outputs of the GA^2T module are denoted as interactive states. Two shared LSTMs are used to encode pedestrians' motion and interactive states, respectively. A novel latent variable predictor is proposed with two feed-forward neural networks, which estimate two latent variable distributions from observed and ground-truth trajectories, respectively. Inputs to the predictor are pedestrians' positions, velocities, and accelerations extracted from trajectories. Positions reflect the potential scene layout, while velocities and accelerations represent pedestrians' motion patterns and radicalness. At the training stage, we minimize the KL-divergence between the two latent variable distributions. Afterwards, the latent variable predictor estimates the latent variable from observed trajectories only at the testing stage. However, the estimated latent variable contains knowledge about pedestrians' future trajectories. A final vector is generated by concatenating the motion state, interactive state, estimated latent variable, and a random Gaussian noise together. The random Gaussian noise is employed to handle future uncertainties. In the decoder network, a shared LSTM is used to convert the final vector into pedestrians' relative positions, which are used to forecast future trajectories.

Generally, our main contributions are three-fold. (1) We propose a GA^2T module to model pedestrians' social interactions. We utilize pedestrians' moving directions as a pseudo oracle to calculate their social attention, and then improve the graph attention network (GAT) by highlighting the correlations between relevant pedestrians. (2) We propose a novel latent variable predictor which can estimate a knowledge-rich latent variable for an improved prediction performance. Such a latent variable contains knowledge about pedestrians' future trajectories and thus can be termed as another pseudo oracle. (3) We embed the GA^2T module and latent variable predictor into a generative model-based trajectory predictor to

handle future uncertainties. Moreover, we achieve state-of-the-art performance on ETH [12] and UCY [13] datasets.

The rest of the paper is organized as follows. Section II reviews related works. Section III describes the proposed method in detail. Section IV presents the experimental results. Section V provides a conclusion and discussion.

II. RELATED WORK

A. Trajectory prediction methods

Trajectory prediction is a time-series modeling problem that attempts to understand pedestrians' motion patterns. Early researches focus on predicting the future trajectories with the linear, constant velocity, and constant acceleration models [14]. However, such a simple model cannot understand complex motion patterns, thus are not suitable for long-term prediction. For longer prediction horizon, flow-based methods [15][16] are proposed to learn the directional flow from observed trajectories. Subsequently, trajectories are generated by recursively sampling the distribution of future motion derived from the learned directional flow.

To better understand pedestrians' motion patterns, researchers have resorted to several learning-based methods, including Gaussian mixture regression [4], Gaussian process [17], random tree searching [18], hidden Markov models [5], and dynamic Bayesian networks (DBNs) [6]. Among these methods, DBNs are commonly used since they can easily incorporate context information [19]. However, these learning-based methods are nontrivial to handle high-dimensional data. Moreover, it is difficult to design hand-crafted features that are workable in general cases.

The recent rise of deep neural networks provides a novel solution to understand pedestrians' motion patterns. Alahi et al. [8] presented Social-LSTM, which models pedestrian motions with a shared LSTM. Then, trajectory prediction was performed through sampling from the distributions of trajectory embedding. Later, Gupta et al. [9] proposed SGAN, which adversarially trains a generator and discriminator to produce socially acceptable trajectories. The latent variable was sampled from a random Gaussian noise to handle future uncertainties. Amirian et al. [20] and Ma et al. [21] used Info- and Wasserstein-GAN structures for adversarial training, respectively. The purpose of using improved GAN structures is to avoid model collapse. Other works perform improved trajectory prediction by proposing well-designed modules. Zhang et al. [22] proposed SR-LSTM, which recursively refines the hidden states of LSTM through a state-refinement module. Amir et al. [23] embedded two attention modules into the trajectory predictor for an improved prediction performance. Zhu et al. [24] improved the SGAN by adding a query module. Learning information from the scene is another way to produce improved prediction performance. Xue et al. [25] and Syed et al. [10] utilized three LSTMs to encode person, social, and scene scale information and then aggregate them for context-aware trajectory prediction. Ridel et al. [26] presented a joint representation of the scene and past trajectories by using the Conv-LSTM and LSTM, respectively. Lisotto et al. [27] improved Social-LSTM by encompassing prior knowledge

about the scene as a navigation map that embodies most frequently crossed areas. However, using high-dimensional image data drastically increase the computing overhead.

B. Graph models for modeling social interactions

One challenge of accurate trajectory prediction is how to model pedestrians' social interactions. Such interactions can be defined by hand-crafted rules, such as social forces [28] and stationary crowds' influence [29]. However, these hand-crafted rules are difficult to capture complex interactions. Graph models provide a novel manner to capture human—human and human—objects interactions. Choi et al. [30] proposed a spatio-temporal graph to explore the causal relationship between vehicle's intention and behavior. The found relationship was used for better trajectory prediction. Eiffert et al. [29] used a spatio-temporal graph to forecast the future trajectories of a social robot. Huang et al. [31] and Kosaraju et al. [32] used GAT to capture pedestrians' social interactions. Ivanovic et al. [33] modeled the multimodal aspects of pedestrian trajectories with dynamic graphs. Moreover, graph models are especially good for handling heterogeneous agents [34]. Despite the ability of modeling complex social interactions, methods mentioned above neglect the correlations between pedestrians' head orientations and trajectory prediction. As reported by Hasan et al. [1], knowing the head orientation is beneficial for modeling social interactions, specifically, future trajectories of the target person are influenced by pedestrians in front. In this work, we improve the GAT module by adding additional social attention, which is calculated on the basis of pedestrians' moving directions.

C. Latent variable learning

Generative model-based methods have gone mainstream due to their ability in handling future uncertainties. The latent variable has a strong correlation with the generated multimodal outputs. The random Gaussian noise is commonly used as the latent variable [9][23]. However, it contains little knowledge about pedestrians' motion characteristics or scene information. Lee et al. [7] performed trajectory prediction with a conditional variational auto-encoders. The latent variable was learned from observed trajectories. Yuan et al. [35] proposed a diversity sampling function to generate a diverse and likely set of latent variables. Zhang et al. [36] presented a stochastic module to generate the latent variable on the basis of pedestrians' history motions. Tang et al. [37] proposed a dynamic encoder to learn latent variables from multiple inputs, including trajectories and the environmental context. However, processing the visual context needs more computation than processing trajectory data only.

In this work, we propose a novel latent variable predictor that estimates the latent variable from trajectory data only. Moreover, we feed pedestrians' positions, velocities, and accelerations into the predictor to learn knowledge about the environmental context, pedestrians' motion patterns, and their radicalness. Unlike references [36] and [37] that only utilize observed data, we try to bridge the latent variable distribution gaps between observed and ground-truth trajectories. Our

inspiration comes from reference [38], which focused on stochastic video generation with a learned prior.

III. PROPOSED METHOD

In this work, we propose GTPPO, which is a graph model-based trajectory predictor. GTPPO can produce accurate trajectory prediction while maintaining diverse outputs. Fig. 2 illustrates the system pipeline of GTPPO.

A. Problem definition

The trajectory prediction problem is a time-series analysis. For pedestrian i, we first term his position (x_i^t, y_i^t) at time step t as p_i^t. The aim of trajectory prediction is to estimate his future trajectory $\mathcal{T}_i = \left(p_i^{t+1}, \ldots, p_i^{t+T_{obs}} \right)$, considering his motion history $\mathcal{H}_i = \left(p_i^0, \ldots, p_i^t \right)$ and human—human or huamn—object interactions. Therefore, the trajectory prediction problem is converted into training a parametric model that predicts future trajectory $\mathcal{T}_i$, which can be formulated as follows:

$$\arg\max_{\Theta} P_\theta \left(\mathcal{T}_i | \mathcal{H}_0, \ldots, \mathcal{H}_n \right), \tag{1}$$

where Θ represents learnable parameters, and n represents the number of pedestrians. Recently, the above-mentioned formulation is always converted into a sequence-to-sequence prediction problem, which can be resolved by the RNN module.

B. Encoder-decoder network

Similar to STGAT, we use an encoder-decoder network due to its ability to generate multi-modal outputs. We briefly introduce the encoder and decoder networks as follows:

Encoder network: Each pedestrian's motion patter is a time-series, which can be well modeled by an LSTM. We use a shared LSTM for each pedestrian to encode the motion pattern. Similar to STGAT, we denote this LSTM by M-LSTM. Specifically, a single layer MLP is used to convert the relative position of pedestrian i at time t ($\Delta x_i^t = x_i^t - x_i^{t-1}$, $\Delta y_i^t = y_i^t - y_i^{t-1}$) into a fixed-length vector e_i^t. Then, the vector is fed into M-LSTM to generate the motion state of pedestrian i at time t as follows:

$$e_i^t = \phi \left(\Delta x_i^t, \Delta y_i^t; W_{ee} \right), \tag{2}$$

$$m_i^t = M - LSTM \left(m_i^{t-1}, e_i^t; W_M \right), \tag{3}$$

where $\phi(\cdot)$ is an embedding function. W_{ee} and W_M are the learnable weights of $\phi(\cdot)$ and the encoder function $M - LSTM(\cdot)$, respectively.

Afterwards, pedestrians' motion states are fed into the GA^2T module to aggregate their social interactions. We use another shared LSTM to process the outputs of the GA^2T module to generate interactive states. Similar to STGAT, we denote this LSTM by G-LSTM, which is defined as follows:

$$g_i^t = G - \text{LSTM} \left(g_i^{t-1}, \hat{m}_i^t; W_G \right), \tag{4}$$

where $\hat{m}_i^t$ is the output of the GA^2T module. W_G is the learnable weight of $G - LSTM(\cdot)$.
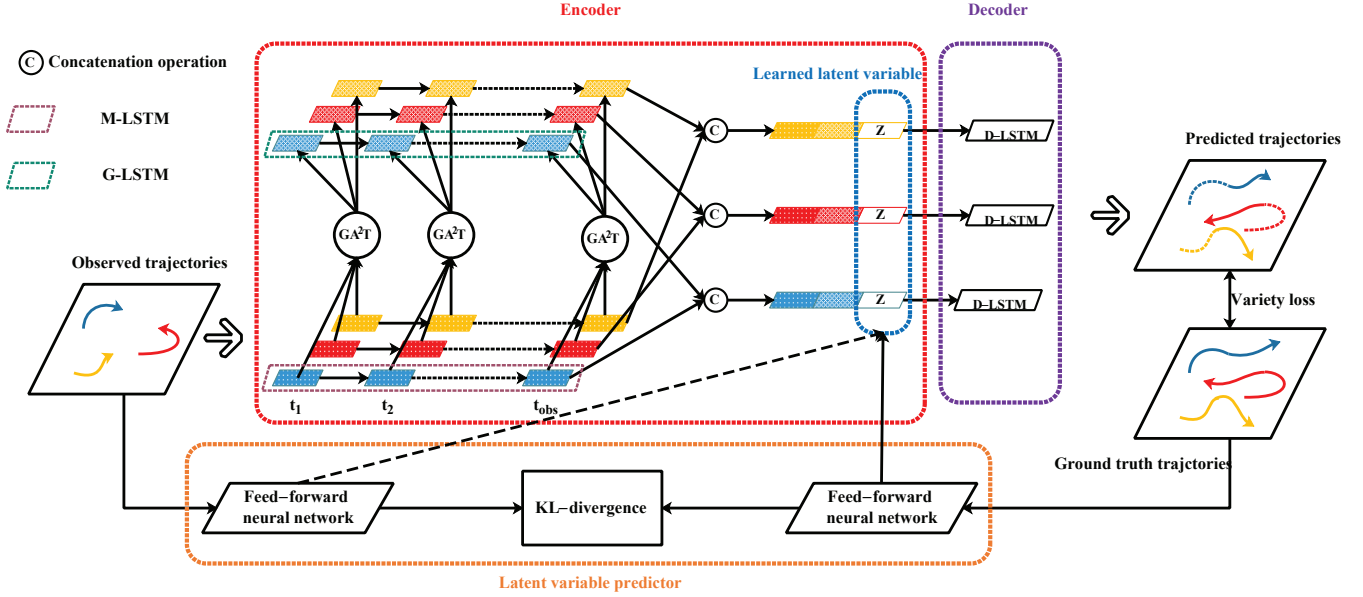
Fig. 2: System pipeline. Our model contains two key components: an encoder-decoder network and a latent variable predictor. The encoder module consists of a motion-LSTM, a graph-LSTM, and GA^2T. The latent variable predictor estimates latent variables with two feed-forward neural networks. The latent variable is learned from ground-truth and observed trajectories in the training and testing stages, respectively. The decoder module generates the future trajectories based on the concatenation of two LSTMs outputs and the learned latent variable (best viewed in color).

Decoder network: The decoder network is used for future trajectory prediction. The intermediate state vector fed into the decoder module consists of three parts: motion states of M-LSTM, interactive states of G-LSTM, and the estimated latent variables. We use a shared LSTM in the decoder module and term it as D-LSTM. Then, the predicted relative position can be generated through the following decoding operation:

$$d_i^t = D - LSTM\left(d_i^t, e_i^t; W_D\right), \tag{5}$$

$$\left(\Delta x_i^{t+1}, \Delta y_i^{t+1}\right) = \delta\left(d_i^{t+1}\right), \tag{6}$$

where W_D is the learnable weight of $D - LSTM(\cdot)$, $\delta(\cdot)$ is a linear layer that converts the embedding into relative positions. d_i^t is the concatenation of m_i^t, g_i^t, and the estimated latent variable z_i^t. e_i^t is the input embedding at time t, which is calculated from Eq(2). The subsequent inputs of D-LSTM are calculated on the basis of the embedding of the last predicted relative position.

C. GA^2T module

Graph model is an effective tool to model pedestrians' social interactions, which are critical for accurate trajectory prediction. Pedestrians in a scene are termed as nodes on the graph and their interactions can be modeled by graph neural networks. In this work, we use the GAT to aggregate information from neighbors by assigning different importance to different nodes. Details of the GAT can be found in STGAT [31].

The GAT can learn pedestrians' social interactions in a data-driven manner theoretically. However, the learning process always struggles due to lack of enough and multifarious training

data. As a common knowledge, pedestrians' future trajectories are always influenced by people in front of them. As illustrated by the upper part of Fig. 3, future trajectories of target one are only influenced by target three who is in target one's field of view (FoV). Hence, pedestrian's head orientation can be used as an oracle for an improved prediction performance [1]. Therefore, we propose the GA^2T module, which captures pedestrians' social interactions with two attentional operations. As demonstrated in the lower part of Fig. 3, the social attention contains human knowledge is used after the automatically learned graph attention.

We calculate the social attention on the basis of pedestrians' bearing angles. We use pedestrians' moving directions as a pseudo oracle since it is hard to accurately recognize pedestrians' head orientations from vision data. Then, the cosine values of pedestrians' bearing angles are calculated as follows:

$$\cos(\mathcal{B}) = \begin{bmatrix} \cos(b_{11}) & \cdots & \cos(b_{1n}) \\ \vdots & \ddots & \vdots \\ \cos(b_{n1}) & \cdots & \cos(b_{nn}) \end{bmatrix}, \tag{7}$$

where n is the number of pedestrians in a scene. b_{ij} represents the bearing angle of agent j from agent i, i.e. the angle between the velocity of agent i and the vector joining agents i and j. Afterwards, the attentional weights are calculated on the basis of the cosine values. We perform hard or soft attention operation to refine the outputs of the GAT. The hard and soft attentions are formulated as follows:

Hard attention: The influence of one pedestrian to another decreases with the increase of their bearing angle. That is to say, a large cosine value indicates a large influence between two pedestrians. Therefore, the hard attention weight is repre-

sented as a matrix H_A of the same size of $cos(\mathcal{B})$, and each element h_{ij} in H_A is set to 0 or 1 by thresholding. h_{ij} is set to 1 if $cos(b_{ij})$ is larger than an empirically threshold 0, otherwise is set to 0.

Soft attention: Unlike hard attention that calculates attention weights by thresholding, soft attention adaptively calculates the attention weights S_A, which is formulated as follows:

$$S_A = \varphi(Conv(\cos(\mathcal{B}))), \tag{8}$$

where $\varphi(\cdot)$ represents the sigmoid activation and $Conv(\cdot)$ represents the 1×1 convolutional operation.

D. Latent variable predictor

As discussed before, the latent variable plays a critical role in generative model-based trajectory predictors. GTPPO proposes a novel latent variable predictor that learns the knowledge about the scene and pedestrians' characteristics from trajectory data. Specifically, we train a latent variable predictor that can estimate similar latent variable distributions from observed and ground-truth trajectories, respectively. As shown in Fig. 2, the latent variable predictor consists of two feed-forward neural networks, which are formulated as follows:

$$\left(\mu_i^k, \sigma_i^k\right) = \Psi\left(I_i^k; W_{LP}^k\right), \tag{9}$$

$$\left(\hat{\mu}_i^k, \hat{\sigma}_i^k\right) = \hat{\Psi}\left(\hat{I}_i^k; \hat{W}_{LP}^k\right), \tag{10}$$

where $\Psi(\cdot)$ and $\hat{\Psi}(\cdot)$ are the feed-forward neural networks with learnable weights W_{LP}^k and $\hat{W}_{LP}^k$, respectively. I_i^k and $\hat{I}_i^k$ are the kth kind of input (positions, velocities, and accelerations) we extract from observed and ground-truth trajectories, respectively. $\left(\mu_i^k, \sigma_i^k\right)$ and $\left(\hat{\mu}_i^k, \hat{\sigma}_i^k\right)$ represent latent variable distributions of the kth kind of input estimated by $\Psi(\cdot)$ and $\hat{\Psi}(\cdot)$, respectively. At the training stage, the estimated latent variable for pedestrian i at time t is z_i^t, which is generated by concatenating samples from $\left(\hat{\mu}_i^k, \hat{\sigma}_i^k\right)$ (k=1, 2, 3) and a random Gaussian noise. At the testing stage, z_i^t is generated by concatenating samples from $\left(\mu_i^k, \sigma_i^k\right)$ (k=1, 2, 3) and a random Gaussian noise.

E. Loss function

The loss function used in this work consists of two parts, namely, the variety and latent variable distribution losses. The variety loss is used to fit the best-predicted trajectory in L2 loss while maintaining diverse outputs. It works as follows: for each pedestrian, the model generates multiple outputs. Then, it chooses the trajectory that has the smallest L2 distance to ground-truth to calculate the variety loss as follows:

$$\mathcal{L}_{\text{variety}} = \min_m \left\| \hat{\mathcal{T}}_i - \mathcal{T}_i^m \right\|_2, \tag{11}$$

where $\hat{\mathcal{T}}_i$ and $\mathcal{T}_i^m$ are ground-truth and the mth predicted trajectories, respectively. m is a hyper-parameter and is set to 20 according to SGAN [9].

The latent variable distribution loss is used to measure the latent variable distribution gaps between observed and ground-

truth trajectories. We use KL-divergence to calculate the loss, which is formulated as follows:

$$\mathcal{L}_{\text{LD}} = D_{KL}((\mu_i^k, \sigma_i^k) \,||\, (\hat{\mu}_i^k, \hat{\sigma}_i^k)), \tag{12}$$

Afterwards, the total loss is defined in a weighted manner as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{variety}} + \alpha \times \mathcal{L}_{\text{LD}}, \tag{13}$$

where α is set to 10 by cross validation across benchmarking datasets.

F. Implementation details

One layer LSTMs are used for encoder and decoder, in which the dimensions of the hidden states are 32. The 16-dimensional latent variable contains a four-dimensional random Gaussian noise and three four-dimensional vectors that embedded from positions, velocities, and accelerations, respectively. Details of the GAT module can be found in STGAT [31]. We train the network with a batch size of 64 for 400 epochs using Adam [39] optimizer. The learning rate of the encoder-decoder network is 0.001 and that of the latent variable predictor is 0.0001. The proposed model is built with Pytorch framework and is trained with an Intel I7 CPU and an NVIDIA GTX-1080 GPU.

IV. EXPERIMENTAL RESULTS

The proposed method is evaluated on ETH [12] and UCY [13] which are publicly available. These two datasets contain five sub-datasets, namely, ETH, HOTEL, UNIV, ZARA1, and ZARA2. All sub-datasets contain real-world pedestrian trajectories with rich human—human and human—object interaction scenarios, including people crossing each other, groups forming and dispersing, and collision avoidance. All the trajectories of 1,536 pedestrians are converted to real-world coordinates. We reduce the sampling frequency to 2.5Hz to decrease the computing overhead. We use the leave-one-out approach similar to that used by Social-LSTM [8]. Specifically, we train models on four sets and test them on the remaining set. The observed and predicted horizons are 8 (3.2 seconds) and 12 (4.8 seconds) time steps, respectively. The prediction horizon is denoted as **T**. In addition, the proposed method is evaluated with two error metrics as follows:

1. *Average Displacement Error (ADE)*: Average L2 distance between the predicted trajectory and the ground-truth trajectory over all predicted horizons.

2. *Final Displacement Error (FDE)*: The Euclidean distance between the predicted and the true final destination at the last predicted step.

A. Quantitative evaluations

Comparisons with state-of-the-art methods: Since the commonly used baselines, including the linear regressor, vanilla-LSTM, and social force model perform worse than Social-LSTM [8], we only compare the proposed method against the following state-of-the-art methods:

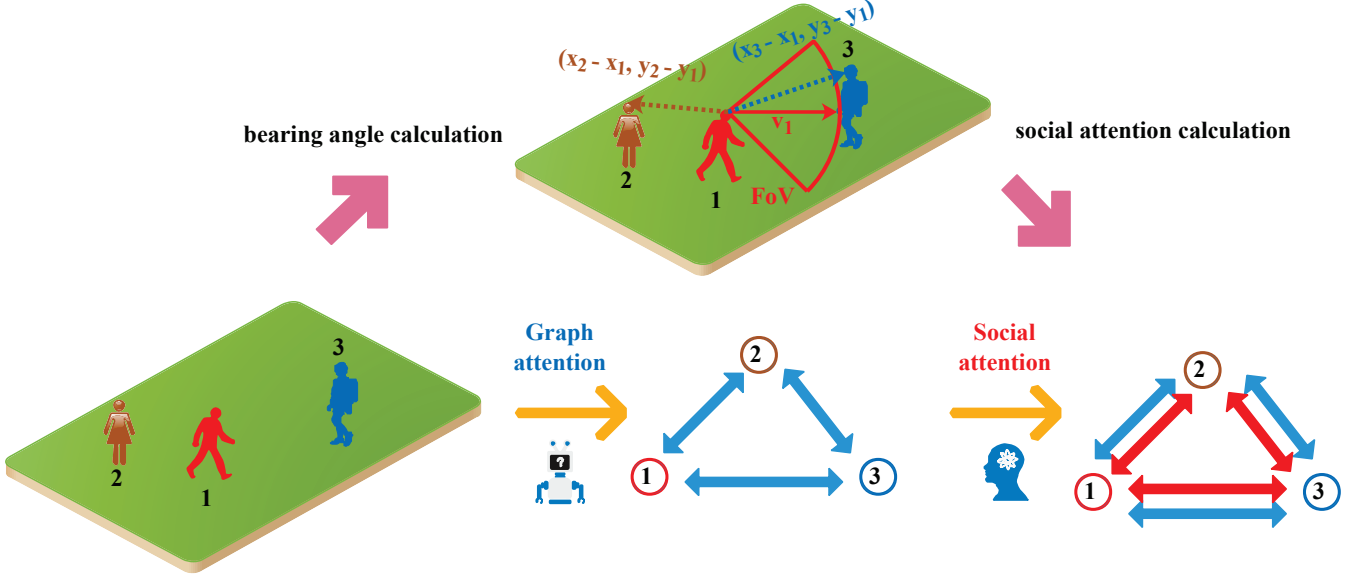1. *Social-LSTM* [8]: An improved LSTM-based trajectory prediction method by proposing a social pooling layer to

Fig. 3: The GA^2T module. We use two attentional operations to aggregate neighbors' information. The former is the graph attention learned in a data-driven manner, the latter is the social attention inspired by the fact that pedestrians' future trajectories are always influenced by people in front of them but not the people behind. Pedestrians' social attention is calculated on the basis of their bearing angles (best viewed in color).

aggregate hidden states of interested pedestrians. Future trajectories are predicted by decoding the concatenation of LSTM embedding and social pooling outputs.

2. *SGAN* [9]: An improved version of Social-LSTM by utilizing adversarial training to generate socially acceptable trajectories. Gaussian noises are used as latent variables to generate multi-modal outputs in consideration of pedestrians' future uncertainties. The model is trained using a variety loss with the hyper-parameter set to 20. At test time, 20 times are sampled from the generator and the best prediction in L2 sense is used for quantitative evaluation.

3. *SR-LSTM* [22]: An improved version of Social-LSTM by proposing a data-driven state refinement module. Such a module iteratively refines the current pedestrians' hidden states on the basis of their neighbors' intentions through message passing.

4. *Sophie* [23]: An improved version of SGAN by utilizing attention mechanisms, namely, the social and physical attention modules. The trajectory prediction performance is improved by highlighting the key information with attention operations.

5. *S-Way* [20]: An improved version of SGAN by replacing the L2-loss with the information-loss proposed in Reference [40] to avoid mode collapsing.

6. *Social-BiGAT* [32]: An improved version of SGAN by using the bicycle structure to train the generator. A graph attention network is used to model social interactions for better prediction performance.

7. *STGAT* [31]: An autoencoder-based trajectory prediction method that uses a spatio-temporal graph attention network to model pedestrians' social interactions in the scene. Specifically, the spatial interactions are captured by the graph

attention mechanism, and temporal correlations are modeled by a shared LSTM.

Table 1 presents the comparison results between ours and the state-of-the-art methods. Social-LSTM and SGAN are typical deterministic and generative model-based trajectory predictors that use deep neural networks. However, their performance is not as satisfactory as the recently proposed methods. Sophie uses two attention mechanisms in capturing social interactions and then achieves an improved prediction performance compared with SGAN. Besides the attention mechanism, S-Ways calculates the bearing angles, Euclidean distances, and the future closest distances between pedestrians and then feeds them into an improved pooling module with attentional weighting. Therefore, S-Ways performs better than other methods in the ETH dataset. However, S-Ways pays more attention to capturing social interactions than to encoding pedestrians' motion patterns. As a result, it performs poorly in ZARA1 and ZARA2 datasets which are not as challenging as the ETH dataset. SR-LSTM proposes a state refinement module to aggregate neighbors' information. It achieves similar prediction performance as that of S-Ways. Social-BiGAT and STGAT use graph models to capture social interactions. Specifically, STGAT achieves the sub-optimal average ADE and FDE because of the usage of the spatio-temporal GAT. It reveals the fact that the graph model is good at modeling social interactions which are significant for accurate trajectory prediction. The spatio-temporal mechanism used in STGAT makes it better than Social-BiGAT that also uses GAT to model social interactions. Our methods, especially the one uses the soft attention mechanism, outperform others in both predicted horizons in HOTEL, UNIV, ZARA1, and ZARA2 datasets. An ablation study will be conducted to explore the reasons

for such superiority in prediction performance.

Ablation study: For the ablation study, we investigate the effects of different modules used in GTPPO, which is an improved version of STGAT by proposing a latent variable predictor and a GA^2T module. We denote our whole model as GTPPO-MLP-Soft or GTPPO-MLP-Hard. MLP represents the latent variable predictor with multiple inputs. Soft and Hard represent soft and hard attention mechanisms, respectively. As presented in Table 2, the attention mechanism encourages the model in capturing complex social interactions. Therefore, GTPPO-Soft and GTPPO-Hard improve the prediction performance of STGAT in the crowded UNIV dataset. GTPPO-Soft performs better than GTPPO-Hard by learning pedestrians' social interactions in a data-driven manner. GTPPO-MLP encourages the model to explore the knowledge about pedestrians' future trajectories. Specifically, GTPPO-MLP estimates a latent variable that reflects the future motions. Compared with STGAT, improvements can be observed in most cases except the ADE value in the UNIV dataset. GTPPO-MLP-Soft and GTPPO-MLP-Hard leverage the abilities of modeling complex social interactions and learning knowledge from future trajectories. Specifically, GTPPO-MLP-Soft achieves the lowest average ADE and FDE values. However, it performs slightly inferior to GTPPO-MLP in ZARA1 and ZARA2 datasets. A possible reason is that GTPPO-MLP-Soft has to make a trade-off between the learning of soft attention mechanism and the latent variable predictor.

Evaluations of different sampling times: A generative model-based trajectory predictor handles future uncertainties by generating multiple outputs. However, many outputs are far away from the ground-truth due to the variety loss [41]. These outputs may influence or mislead further decisions on the basis of the predicted trajectories. We propose that a good trajectory predictor should estimate accurate future trajectories with few attempts while maintaining diversity outputs. As revealed by the ablation study, the MLP module encourages the model to investigate pedestrians' future motions. Hence, the proposed method can perform accurate trajectory prediction with few attempts on the basis of the learned knowledge from future trajectories.

We perform a comparison between STGAT and GTPPO-MLP when gradually decreasing sampling times. Fig. 4 illustrates the comparison results of average ADE and FDE between STGAT and GTPPO-MLP when using different sampling numbers. For both methods, the prediction performance goes worse when gradually decreasing sampling times. However, GTPPO-MLP can still perform satisfactory trajectory prediction with few samplings. As demonstrated in Fig. 4, the average ADE of GTPPO-MLP with five samplings is 0.43, whereas that of STGAT with 20 samplings is 0.45. The average FDE of GTPPO-MLP with one sampling is better than that of STGAT with 20 samplings. All these findings reveal the ability of GTPPO-MLP in accurate trajectory prediction with few attempts.

Computational time analysis: The computational time of trajectory prediction is a critical issue for real-time applications. We calculate the computational times of the proposed methods compared with STGAT. As presented in Table 3,
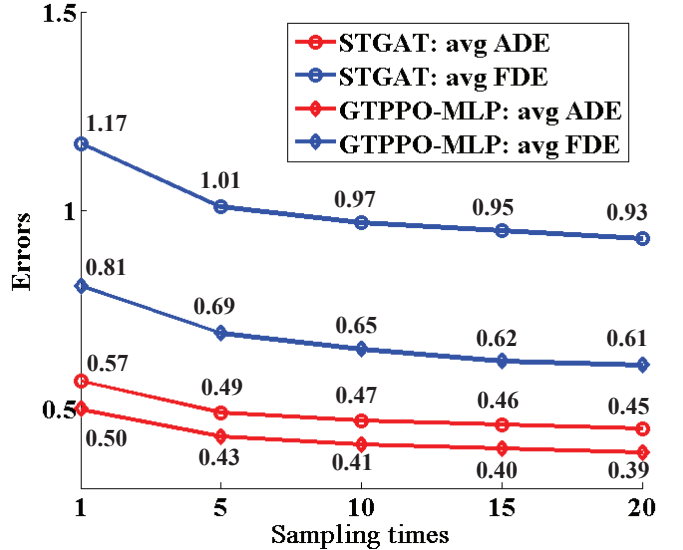

Fig. 4: Comparison results of average ADE (red lines) and FDE (blue lines) values between GTPPO-MLP and STGAT when using different sampling times across all datasets. The diamond marker represents GTPPO-MLP, and the circle marker represents STGAT (best viewed in color.)

the MLP module increases little computing overhead since only a simple feed-forward neural network is added during tests. GTPPO-MLP-Hard and GTPPO-MLP-Soft need about four times of computing overheads due to the calculation of bearing angles. Notably, we simultaneously process 64 scenes in once forward computation. Therefore, the proposed method can satisfy the needs of real-time applications. Moreover, one can use GTPPO-MLP in the scenes that are not so crowded for better real-time performance.

B. Qualitative evaluations

We perform some qualitative evaluations to get an insightful understanding of GTPPO. Fig. 5 demonstrates the trajectory prediction results by using STGAT, SR-LSTM, Sophie, GTPPO-MLP, GTPPO-MLP-Soft, and GTPPO-MLP-Hard in different datasets. Each sub-figure represents a scene with multiple labeled pedestrians. Similar to former works, the predicted trajectory is the best one that has the lowest ADE value among the 20 samples generated by each method. Generally, all methods perform accurate trajectory prediction in most cases. These methods successfully recognize the still pedestrians, for example, targets one and two in the second scene of Fig. 5(b). Our methods perform better than the selected methods by generating more close future trajectories. Except for more close outputs, our methods successfully handle the challenging sudden motion changes. As shown in the first scene of Fig. 5(a), STGAT, SR-LSTM, and Sophie fail to capture the motion changes of the target one. However, our methods recognize the motion changes due to the estimated latent variable that reflects future motions, and then perform accurate trajectory prediction. Similar results can be observed in other scenes, for example, the third scene of Fig. 5(a), the third and fourth scenes of Fig. 5(c) and Fig. 5(d). These

TABLE I: Comparison results with state-of-the-art methods across all datasets. We report ADE and FDE for T = 12 in meters. Our method outperforms state-of-the-art methods in HOTEL, UNIV, ZARA1, and ZARA2 datasets, and is specifically good for average ADE and FDE (low is preferred and is labeled with bold fonts).

Metric	Dataset	Social-LSTM	SGAN	SR-LSTM	Sophie	S-Way	Social-BiGAT	STGAT	Ours	
									Hard	Soft
ADE	ETH	1.09	0.87	0.63	0.70	**0.39**	0.69	0.65	0.68	0.66
	HOTEL	0.79	0.67	0.37	0.76	0.39	0.49	0.35	0.37	**0.21**
	UNIV	0.67	0.76	0.51	0.54	0.55	0.55	0.52	**0.38**	**0.38**
	ZARA1	0.47	0.35	0.41	0.30	0.44	0.30	0.34	**0.23**	**0.23**
	ZARA2	0.56	0.42	0.32	0.38	0.51	0.36	0.29	0.28	**0.20**
AVG		0.72	0.61	0.45	0.54	0.46	0.48	0.43	0.39	**0.34**
FDE	ETH	2.35	1.62	1.25	1.43	**0.64**	1.29	1.12	1.07	1.04
	HOTEL	1.76	1.37	0.74	1.67	0.66	1.01	0.66	0.61	**0.33**
	UNIV	1.40	1.52	1.10	1.24	1.31	1.32	1.10	**0.63**	**0.63**
	ZARA1	1.00	0.68	0.90	0.63	0.64	0.62	0.69	**0.36**	**0.36**
	ZARA2	1.17	0.84	0.70	0.78	0.92	0.75	0.60	0.38	**0.33**
AVG		1.54	1.21	0.94	1.15	0.83	1.00	0.83	0.61	**0.54**

TABLE II: Ablation study. We report ADE and FDE for T = 12 in meters across five datasets. MLP represents **m**ultiple inputs **l**atent variable **p**redictor. Soft / Hard represent soft and hard attention mechanisms, respectively (low is preferred and is labeled with bold fonts).

Metric	Dataset	STGAT	-Hard	-Soft	-MLP	-MLP-Hard	-MLP-Soft
ADE	ETH	**0.65**	0.70	0.68	0.65	0.68	0.66
	HOTEL	0.35	0.32	0.31	0.23	0.37	**0.21**
	UNIV	0.52	0.47	0.46	0.67	**0.38**	**0.38**
	ZARA1	0.34	0.34	0.32	**0.22**	0.23	0.23
	ZARA2	0.29	0.29	0.29	**0.19**	0.28	0.20
AVG		0.43	0.42	0.39	0.39	0.39	**0.34**
FDE	ETH	1.12	1.21	1.18	1.07	1.07	**1.04**
	HOTEL	0.66	0.58	0.54	0.36	0.61	**0.33**
	UNIV	1.10	1.00	0.98	0.94	**0.63**	**0.63**
	ZARA1	0.69	0.70	0.61	**0.35**	0.36	0.36
	ZARA2	0.60	0.62	0.64	**0.31**	0.38	0.33
AVG		0.83	0.82	0.79	0.61	0.61	**0.54**

TABLE III: Computational times of the proposed methods compared with STGAT. We calculate the times for one forward computation of a batch with the size 64. The unit of the time is millisecond.

Method	STGAT	GTPPO-MLP	GTPPO-MLP-Hard	GTPPO-MLP-Soft
Time	62	65($\approx$1.0X)	250($\approx$4.0X)	255($\approx$4.0X)

findings reveal the fact that the information learned from positions, velocities, and accelerations are useful for accurate trajectory prediction in most cases.

Except for the best trajectories, density maps of the predicted trajectories reveal the abilities of different methods in generating accurate and diverse outputs. Fig. 6 illustrates the density maps of the predicted trajectories in four typical scenarios selected from (a) ETH, (b) HOTEL, (c) ZARA1, and (d) ZARA2 datasets. Density maps in the UNIV dataset are not shown because there are too many trajectories in each scene. In the ETH scenario, GTPPO-MLP and GTPPO-MLP-Soft recognize the sudden motion changes whereas STGAT fails. In addition, density maps generated by GTPPO-MLP-Soft are more separate than those generated by GTPPO-MLP due to the use of the soft attention mechanism. In the HOTEL scenario, all methods generate wrong future trajectories for target one to avoid collision with target two. However, the trajectories predicted by our methods are more close to the ground-truth. In the ZARA1 scenario, GTPPO-MLP and GTPPO-MLP-Soft recognize the slow moves of targets two and three, whereas STGAT still predicts long-distance future trajectories. In the ZARA2 scenario, GTPPO-MLP and GTPPO-MLP-Soft successfully recognize the motion changes of target one, whereas STGAT fails.
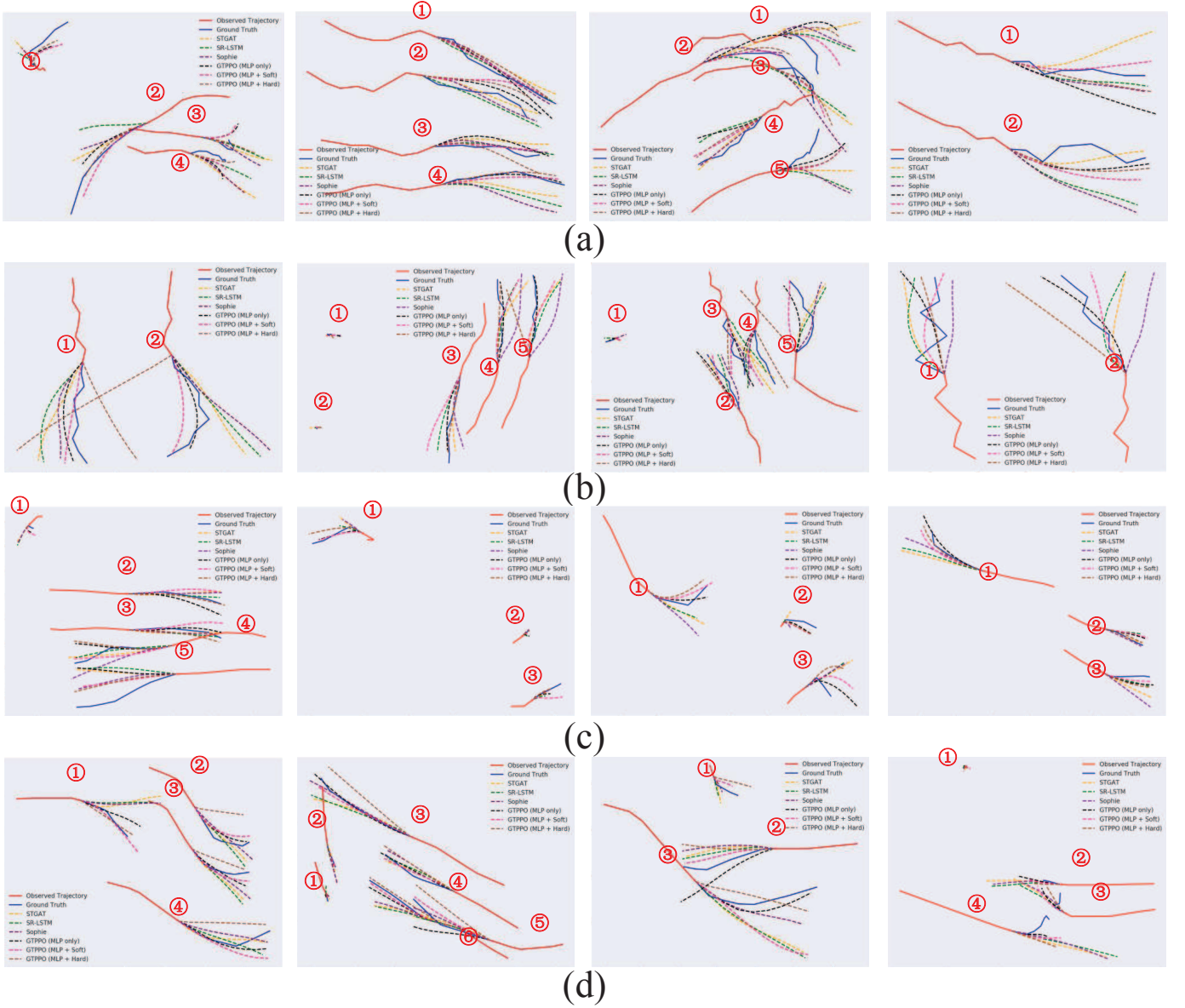
Fig. 5: Trajectory prediction results using STGAT, SR-LSTM, Sophie, GTPPO-MLP, GTPPO-MLP-Soft, and GTPPO-MLP-Hard in (a) ETH, (b) HOTEL, (c) ZARA1, and (d) ZARA2 datasets. Red and blue lines represent observed and ground-truth trajectories, respectively. Dashed lines of different colors represent trajectories predicted by different methods. We show the best trajectory with the lowest ADE value from 20 predicted samples (best viewed in color and zoom-in).

V. CONCLUSION AND DISCUSSION

In this work, we propose GTPPO, which performs trajectory prediction with two pseudo oracles. One pseudo oracle is pedestrians' moving directions at the last observed step, which are used to approximate pedestrians' head orientations. An improved GAT module, GA^2T, is proposed by adding the social attention calculated on the basis of pedestrians' moving directions. GA^2T is verified to achieve an improved prediction performance in crowded datasets. Another pseudo oracle is the latent variable estimated by a novel latent variable predictor. Such a latent variable learns knowledge about future trajectories. Random Gaussian noise is injected into the estimated latent variable to handle future uncertainty. Evaluations are performed in two commonly used metrics, namely, ADE and FDE, across five publicly available datasets. Comparisons with state-of-the-art approaches indicate the superiority of the proposed method in trajectory prediction. Ablation studies and qualitative evaluations reveal the effects of different modules, especially the ability of the latent variable predictor to recognize sudden motion changes. Besides, the proposed method only learns knowledge from trajectories, thus meets the requirement of real-time performance. Our future focuses on how to control the latent variable predictor to generate a better latent variable.
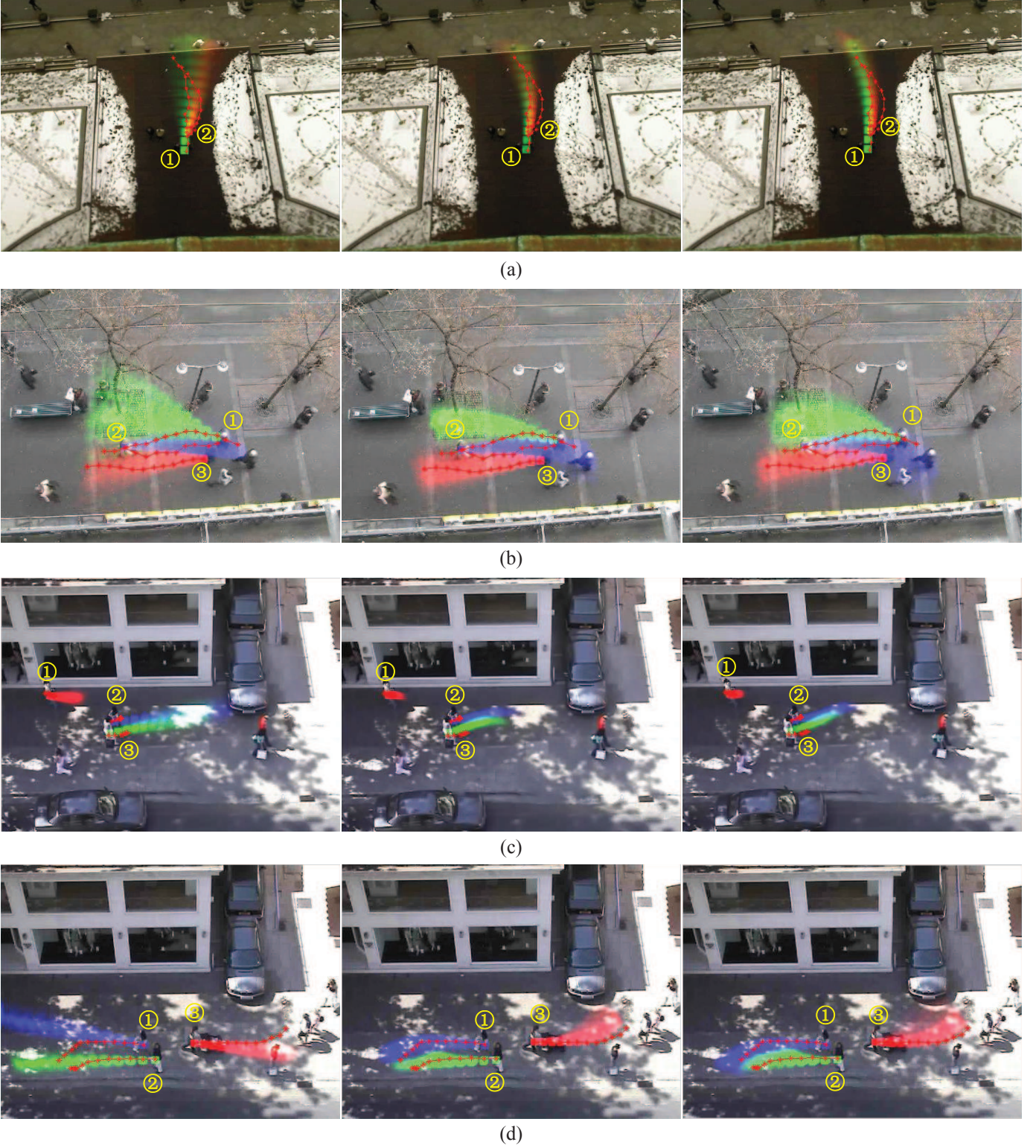
Fig. 6: Density maps of the predicted trajectories in (a) ETH, (b) HOTEL, (c) ZARA1, and (d) ZARA2 datasets. The first, second, and third columns represent density maps generated by STGAT, GTPPO-MLP, and GTPPO-MLP-Soft, respectively. The density maps are generated by sampling 300 times from the well-learned generators. The red stars represent the ground-truth future trajectories, and different colors indicate the density distributions of differently labeled pedestrians (best viewed in color and zoom-in).
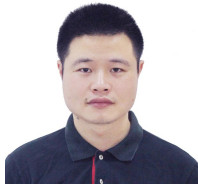
Acknowledgment

References

[1] I. Hasan, F. Setti, T. Tsesmelis, A. Del Bue, M. Cristani, and F. Galasso, "" seeing is believing": Pedestrian trajectory forecasting using visual frustum of attention," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 1178–1185.

[2] J. Hong, B. Sapp, and J. Philbin, "Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8454–8462.

[3] M. Luber, J. A. Stork, G. D. Tipaldi, and K. O. Arras, "People tracking with human motion predictions from social forces," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 464–469.

[4] J. Li, W. Zhan, and M. Tomizuka, "Generic vehicle tracking framework capable of handling occlusions based on modified mixture particle filter," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 936–942.

[5] W. Wang, J. Xi, and D. Zhao, "Learning and inferring a driver's braking action in car-following scenarios," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 3887–3899, 2018.

[6] D. Kasper, G. Weidl, T. Dang, G. Breuel, A. Tamke, A. Wedel, and W. Rosenstiel, "Object-oriented bayesian networks for detection of lane change maneuvers," *IEEE Intelligent Transportation Systems Magazine*, vol. 4, no. 3, pp. 19–31, 2012.

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

[8] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 961–971.

[9] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2255–2264.

[10] A. Syed and B. T. Morris, "Sseg-lstm: Semantic scene segmentation for trajectory prediction," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 2504–2509.

[11] S. Haddad, M. Wu, H. Wei, and S. K. Lam, "Situation-aware pedestrian trajectory prediction with spatio-temporal attention model," *arXiv preprint arXiv:1902.05437*, 2019.

[12] S. Pellegrini, A. Ess, and L. Van Gool, "Improving data association by joint modeling of pedestrian trajectories and groupings," in *European conference on computer vision*. Springer, 2010, pp. 452–465.

[13] L. Leal-Taixé, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese, "Learning an image-based motion context for multiple people tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3542–3549.

[14] S. Zernetsch, S. Kohnen, M. Goldhammer, K. Doll, and B. Sick, "Trajectory prediction of cyclists using a physical model and an artificial neural network," in *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2016, pp. 833–838.

[15] W. Zhi, R. Senanayake, L. Ott, and F. Ramos, "Spatiotemporal learning of directional uncertainty in urban environments with kernel recurrent mixture density networks," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4306–4313, 2019.

[16] S. Molina, G. Cielniak, T. Krajník, and T. Duckett, "Modelling and predicting rhythmic flow patterns in dynamic environments," in *Annual Conference Towards Autonomous Robotic Systems*. Springer, 2018, pp. 135–146.

[17] C. Laugier, I. E. Paromtchik, M. Perrollaz, M. Yong, J.-D. Yoder, C. Tay, K. Mekhnacha, and A. Nègre, "Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety," *IEEE Intelligent Transportation Systems Magazine*, vol. 3, no. 4, pp. 4–19, 2011.

[18] G. Aoude, J. Joseph, N. Roy, and J. How, "Mobile agent trajectory prediction using bayesian nonparametric reachability trees," in *Infotech@ Aerospace 2011*, 2011, p. 1512.

[19] Y. Gu, Y. Hashimoto, L.-T. Hsu, M. Iryo-Asano, and S. Kamijo, "Human-like motion planning model for driving in signalized intersections," *IATSS research*, vol. 41, no. 3, pp. 129–139, 2017.

[20] J. Amirian, J.-B. Hayet, and J. Pettré, "Social ways: Learning multi-modal distributions of pedestrian trajectories with gans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[21] H. Ma, J. Li, W. Zhan, and M. Tomizuka, "Wasserstein generative learning with kinematic constraints for probabilistic interactive driving behavior prediction," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 2477–2483.

[22] P. Zhang, W. Ouyang, P. Zhang, J. Xue, and N. Zheng, "Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 085–12 094.

[23] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, "Sophie: An attentive gan for predicting paths compliant to social and physical constraints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1349–1358.

[24] Y. Zhu, D. Qian, D. Ren, and H. Xia, "Starnet: Pedestrian trajectory prediction using deep neural network in star topology," *arXiv preprint arXiv:1906.01797*, 2019.

[25] H. Xue, D. Q. Huynh, and M. Reynolds, "Ss-lstm: A hierarchical lstm model for pedestrian trajectory prediction," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 1186–1194.

[26] D. Ridel, N. Deo, D. Wolf, and M. Trivedi, "Scene compliant trajectory forecast with agent-centric spatio-temporal grids," *arXiv preprint arXiv:1909.07507*, 2019.

[27] M. Lisotto, P. Coscia, and L. Ballan, "Social and scene-aware trajectory prediction in crowded spaces," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[28] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[29] S. Yi, H. Li, and X. Wang, "Understanding pedestrian behaviors from stationary crowd groups," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3488–3496.

[30] C. Choi, A. Patil, and S. Malla, "Drogon: A causal reasoning framework for future trajectory forecast," *arXiv preprint arXiv:1908.00024*, 2019.

[31] Y. Huang, H. Bi, Z. Li, T. Mao, and Z. Wang, "Stgat: Modeling spatial-temporal interactions for human trajectory prediction," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6272–6281.

[32] V. Kosaraju, A. Sadeghian, R. Martín-Martín, I. Reid, H. Rezatofighi, and S. Savarese, "Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks," in *Advances in Neural Information Processing Systems*, 2019, pp. 137–146.

[33] B. Ivanovic and M. Pavone, "The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2375–2384.

[34] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha, "Trafficpredict: Trajectory prediction for heterogeneous traffic-agents," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 6120–6127.

[35] Y. Yuan and K. Kitani, "Diverse trajectory forecasting with determinantal point processes," *arXiv preprint arXiv:1907.04967*, 2019.

[36] L. Zhang, Q. She, and P. Guo, "Stochastic trajectory prediction with social graph network," *arXiv preprint arXiv:1907.10233*, 2019.

[37] C. Tang and R. R. Salakhutdinov, "Multiple futures prediction," in *Advances in Neural Information Processing Systems*, 2019, pp. 15 398–15 408.

[38] E. Denton and R. Fergus, "Stochastic video generation with a learned prior," in *International Conference on Machine Learning*, 2018, pp. 1182–1191.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[40] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Advances in neural information processing systems*, 2016, pp. 2172–2180.

[41] L. A. Thiede and P. P. Brahma, "Analyzing the variety loss in the context of probabilistic trajectory prediction," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9954–9963.

Xiaofeng Liu received the B.S. degree in electronics engineering and the M.S. degree in computer science from the Taiyuan University of Technology in 1997 and 1999, respectively, and the Ph.D. degree in biomedical engineering from Xi'an Jiaotong University in 2006. From 2013 to 2014 he was a visiting scholar in the University College London. Since 2011 He has been a full Professor with the Hohai University, China, where he is also the Leader of the Cognition and Robotics Laboratory and the vice Director of Jiangsu Key laboratory of Special Robotic Technologies. He has over 16 grants as a PI and over 12 grants as a Researcher, including the National High-Tech R&D Program (863) and the National Basic Research Program (973). He has been granted 15 patents and authored 20 accredited journal papers. His current research interests focus on human robot interactions, social robotics, bioinspired navigation and neural engineering.

Biao Yang received his BS degree from Nanjing University of Technology. He received his MS and Ph.D degrees in instrument science and technology from Southeast University, Nanjing, China, in 2014. From 2018 to 2019 he was a visiting scholar in the University of California, Berkeley. Now he works at Changzhou University, China. His current research interests include machine learning and pattern recognition based on computer vision.

Guocheng Yan was born in Fuyang, china, in 1996. He received the B.S degree in applied physics from Fuyang Normal University 2019. He is currently pursuing the master degree in machine learning.

Yang Chen received his Ph.D. degree in Harbin Engineering University, Harbi, China. He is now a associate professor at the School of Information Science and Engineering, Changzhou University. His research interests include underwater array signal processing and acoustic signal processing.

Pin Wang received the Ph.D. degree in Transportation Engineering from Tongji University, China, in 2016, and was a postdoc at California PATH, University of California, Berkeley, from 2016 to 2018. She is now is a researcher and team leader at California PATH, UC Berkeley. Her current research is focused on Deep Learning algorithms and applications for Autonomous Driving, including driving behavior learning, trajectory planning, control policy learning, and pedestrian intention prediction. She also collaborates with industries on projects such as intelligent traffic control and advanced vehicular technology assessment.

Jingyao Chan received his Ph.D. degree in Mechanical Engineering from University of California, Berkeley, US, in 1988. He has three decades of research experience in a broad range of automotive and transportation systems, and now is the Program Leader at California PATH and the Co-Director of Berkeley DeepDrive Consortium (bdd.berkeley.edu).He is now leading research in several topics, including driving behavior learning, pedestrian-vehicle interaction, sensor fusion for driving policy adaptation, and supervisory control in automated driving systems, human factors in automated driving.