STGlow: A Flow-Based Generative Framework With Dual-Graphormer for Pedestrian Trajectory Prediction

Rongqin Liang, Student Member, IEEE, Yuanman Li, Member, IEEE, Jiantao Zhou, Senior Member, IEEE, and Xia Li, Member, IEEE

Abstract— The pedestrian trajectory prediction task is an essential component of intelligent systems. Its applications include but are not limited to autonomous driving, robot navigation, and anomaly detection of monitoring systems. Due to the diversity of motion behaviors and the complex social interactions among pedestrians, accurately predicting their future trajectory is challenging. Existing approaches commonly adopt generative adversarial networks (GANs) or conditional variational auto-encoders (VAEs) to generate diverse trajectories. However, GAN-based methods do not directly model data in a latent space, which may make them fail to have full support over the underlying data distribution. CVAE-based methods optimize a lower bound on the likelihood of observations, which may cause the learned distribution to deviate from the underlying distribution. The above limitations make existing approaches often generate highly biased or inaccurate trajectories. In this article, we propose a novel generative flow-based framework with dual-graphormer for pedestrian trajectory prediction (STGlow). Different from previous approaches, our method can more precisely model the underlying data distribution by optimizing the exact log-likelihood of motion behaviors. Besides, our method has clear physical meanings for simulating the evolution of human motion behaviors. The forward process of the flow gradually degrades complex motion behavior into simple behavior, while its reverse process represents the evolution of simple behavior into complex motion behavior. Furthermore, we introduce a dual-graphormer combined with the graph structure to more adequately model the temporal dependencies and the mutual spatial interactions. Experimental results on several benchmarks demonstrate that our method achieves much better performance compared to previous state-of-the-art approaches.

Index Terms— Attention mechanism, deep neural network, generative flow, graph learning, trajectory prediction.

I. INTRODUCTION

TRAJECTORY prediction, as one of the most important future behavior modeling tasks, aims to predict the future trajectory based on the observed trajectory. It plays an important role in applications such as self-driving vehicles [1], autonomous navigation robots [2], anomaly behavior detection [3], [4], video surveillance [5], [6], [7], and so on. Despite that significant advances have been achieved recently [8], [9], [10], [11], [12], [13], [14], [15], accurately predicting future trajectories of pedestrians remains challenging due to the inherent properties of pedestrians. First, due to the differences in human intent and unique behavior patterns, the future trajectories of pedestrians are full of diversity, even when they

Manuscript received 17 November 2022; revised 21 May 2023; accepted 10 July 2023. Date of publication 26 July 2023; date of current version 30 October 2024. This work was supported in part by the Key Project of Shenzhen Science and Technology Plan under Grant 20220810180617001, in part by the Foundation for Science and Technology Innovation of Shenzhen under Grant RCBS20210609103708014, in part by the Natural Science Foundation of China under Grant 62001304, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2022A1515010645, and in part by the Open Research Project Program of the State Key Laboratory of Internet of Things for Smart City (University of Macau) under Grant SKLioTSC(UM)-2021-2023/ORP/GA04/2022. (Corresponding author: Yuanman Li.)

Rongqin Liang, Yuanman Li, and Xia Li are with the Guangdong Key Laboratory of Intelligent Information Processing, College of Electronics and Information Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: 1810262064@email.szu.edu.cn; yuanmanli@szu.edu.cn; lixia@szu.edu.cn).
Jiantao Zhou is with the State Key Laboratory of Internet of Things for Smart City and the Department of Computer and Information Science, University of Macau, Macau (e-mail: jzhtou@um.edu.mo).
Digital Object Identifier 10.1109/TNNLS.2023.3294998

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
share the same historical trajectory. Second, influenced by surrounding agents, there are highly complex social interactions among pedestrians, which drive pedestrians to make decisions such as walking parallel, walking in groups, or changing direction/speed to avoid collisions. Faced with the challenge of diverse future trajectories, most previous works [8], [11], [12], [13], [14], [16] applied generative models to model the multimodality of human motion behaviors. For instance, some studies [8], [11], [13] employed generative adversarial networks (GANs) [17] to predict the distribution of future trajectories. However, GAN-based methods do not directly model data in a latent space, which may make them fail to have full support over the underlying data distribution, thus generating highly biased trajectories. In addition, the training process of GANs is often unstable due to adversarial learning. Alternatively, some works exploited conditional variational autoencoders (CVAEs) [16], [18], [19], [20], [21] or diffusion model [14] to model the diversity of future trajectories. However, both of them optimize the variational lower bound on the log-likelihood of observations [22], which may cause the learned distribution to deviate from the underlying distribution. This means that using a lower bound criterion may yield a suboptimal solution with respect to the true log-likelihood, resulting in inaccurate future trajectories. Therefore, how to more precisely model the underlying distribution of pedestrian trajectories is very important for pedestrian trajectory prediction tasks.

To straightforwardly model the social interactions among pedestrians, some researches [9], [23], [24], [25], [26] proposed to represent the social interactions between pedestrians utilizing the topology of graphs. However, graph-based methods may suffer from over-smoothing problems on node features [26], which means that when constructing graphs for crowded environments, the features of nodes will be smoothed by the aggregation of nodes. This may lead to the loss of unique behavior characteristics of pedestrians. Therefore, how to effectively model social interactions while maintaining the unique behavioral features of pedestrians is still a challenge for pedestrian trajectory prediction tasks.

Faced with the above challenges on diverse trajectories and social interactions, in this article, we propose a novel flow-based generative framework with a dual-graphormer for pedestrian trajectory prediction (STGlow). First, to more precisely model the underlying distribution, we propose a generative flow framework with pattern normalization (Glow-PN) to produce multiple reasonable future trajectories. In contrast to previous GAN- and CVAE-based models, our framework optimizes the exact log-likelihood of observations by mapping a complex data distribution into a simple and tractable one through a series of invertible transformations. Similar to the process of artists creating artworks, complex motion behaviors of pedestrians are not accomplished at one stroke, but based on simple motion behaviors (e.g., every step of walking or every action) combined with individual behavior habits originating from various “biases” such as people’s walking habits, traffic awareness, and potential intentions. Hence, we propose to characterize the evolution of motion behavior from simple to complex by modeling the “biases” progressively utilizing the generative flow with a series of simple and tractable functions. Second, to model the social interactions more effectively, we further propose a dual-graphormer to extract the representations of motion behaviors and model the temporal dependencies and the mutual spatial interactions. The proposed dual-graphormer combines graph structure with transformers, where the designed attention mechanism can adaptively focus on all other nodes. Our method not only enables intuitive and effective modeling of social interactions, but also greatly alleviates the problem of over-smoothing of nodes. Specifically, in the training phase, a Glow-PN is applied to learn the distribution of the motion behavior conditioned on the representation of social interactions. In the inference phase, simple behaviors are sampled from the standard normal distribution and formed into “evolved” representations of complex motion behaviors using the reverse process of Glow-PN conditioned on the representation of social interactions. These representations of complex motion behaviors are eventually decoded into predicted trajectories through a bidirectional trajectory prediction module. The main contributions of our work can be summarized as follows.

1) We present a novel diverse trajectory prediction framework based on generative flow, simulating the evolution of human motion behaviors from simple to complex. In contrast to previous approaches, our method can more precisely model the underlying distribution by optimizing the exact log-likelihood of motion behaviors. Besides, a PN is carefully developed to normalize the unique behavioral pattern of pedestrians, greatly improving the prediction accuracy.

2) We further propose a dual-graphormer to extract the representations of motion behaviors and model the social interactions in both temporal and spatial domains. Different from previous Transformer-based methods, our dual-graphormer combined with the graph structure more adequately models the temporal dependencies and the mutual spatial interactions.

3) The proposed framework achieves state-of-the-art performance on widely used pedestrian trajectory prediction benchmarks, providing a promising direction for generating diverse and reasonable trajectories.

The remainder of this article is organized as follows. Section II gives a brief review of related works. Section III details our proposed STGlow for pedestrian trajectory prediction. Extensive experimental results are presented in Section IV, and we finally draw a conclusion in Section V.

II. RELATED WORKS

A. Pedestrian Trajectory Prediction

Traditional trajectory prediction methods mainly rely on designing handcrafted rules to model human interactions [27], [28], [29], [30], [31]. For instance, Social Force [27] introduced attractive and repulsive forces to avoid collisions. Although these methods demonstrate the importance of interaction modeling, they are limited by the handcrafted features and perform poorly in trajectory prediction.
With the great success of deep neural networks, the recurrent neural network (RNN) and its variants are widely applied in trajectory prediction task [1], [32], [33], [34] and motion prediction task [35], [36], on the basis of their good performance on sequence learning [37], [38], [39]. Wherein, Social-long short-term memory (LSTM) [32] employed an LSTM to encode pedestrian trajectory and designed a Social Pooling to aggregate the global representation of neighboring pedestrians. To enhance the representation ability of social interaction features, many studies [1], [34], [40], [41] have followed this idea of transmitting information between pedestrians and proposed different effective message-passing mechanisms. Although the RNN-based approaches approach the trajectory prediction task in a data-driven manner, they ignore the important fact that the future trajectories of pedestrians are full of diversity due to the differences in human intents and unique behavior patterns.

Besides, graph networks are utilized in various tasks such as action understanding [42], [43], recommendation systems [44], and text classification [45], due to their capability of modeling non-Euclidean structured data. Recently, the intuitive modeling power of graph models has been applied to represent complex social interactions among pedestrians [9], [23], [24], [46], [47], [48]. For instance, the work GTPPO [24] explored a social graph attention module that combines specific obstacle avoidance experiences (OAEs) with graph attention to capture pedestrians’ social interactions. DMRGCGN [48] proposed a disentangled multiscale aggregation to better represent social interactions between pedestrians on a weighted graph. Though the topology of graphs seems to be a straightforward way to represent social interactions, graph-based methods may suffer from over-smoothing problems on node features [26], [48], which may lead to the loss of the unique behavior characteristics of pedestrians.

To generate diverse future trajectories, some researchers suggested employing generative models to model the diversity of human motion behaviors [8], [49], [50]. Part of these works were based on GANs [8], [11], [13]. Among them, Social-GAN [8] applied GANs for the first time to generate diverse future trajectories and designed a pooling module to aggregate social interactions. Furthermore, TPNNMS [11] proposed a temporal pyramid structure to model both global and local contexts of human motion behaviors. Another part of the works [12], [16], [18], [20], [21] applied CVAEs to explicitly encode the distribution of diverse future trajectories. For instance, Trajectron++ [16] utilized the latent variable framework of CVAEs to explicitly encode diversity and modeled social interactions in combination with a graph-structured recurrent model, while PECNet [18] embedded the distant trajectory endpoints into a latent space to assist in long-range diverse trajectory prediction. More recently, MID [14] devised a Transformer-based diffusion model for trajectory prediction with a reverse process of motion indeterminacy diffusion. Though previous generative models have achieved promising performance in modeling the diversity of human behaviors, these approaches still have inherent limitations, for example, methods based on GANs may not fully support the data distribution due to the lack of encoding latent variables, while methods based on CVAEs and diffusion models optimize a lower bound on the log-likelihood of observations. Such limitations could make them generate biased or inaccurate trajectories.

B. Normalizing Flow

Normalizing flows (NFs) are invertible generative models that map complex data distributions to simple and tractable ones. Recently, NFs have been successfully applied to a variety of generation tasks such as image generation [22], [51], [52], [53], video generation [54], and speech synthesis [55], [56]. For example, Glow [52] proposed an invertible 1 × 1 convolution for the generative flow and showed the efficiency of realistic-looking synthesis and manipulation of large images. WaveGlow [55] proposed a flow-based network for generating high-quality speech from mel-spectrograms. In our task, we propose a flow-based method for pedestrian trajectory prediction. Our proposed scheme has clear physical meanings to simulate the evolution of human motion behaviors.

C. Transformer

Transformer [57], which relies entirely on self-attention mechanisms to model global dependencies of the serialization inputs, has recently made remarkable progress in a variety of natural language processing (NLP) tasks [58], vision tasks [59], [60], [61], [62], and speech recognition [63]. For instance, in vision tasks [64], ViT [59] sequentialized the image into a series of tokens and modeled the global dependencies of the image through the Transformer encoder. More recently, some works [10], [12], [14] have applied Transformer to pedestrian trajectory prediction tasks. Among them, STAR [10] employed a temporal transformer and spatial transformer, respectively, to extract temporal dependencies and spatial interactions, while AgentFormer [12] exploited an agent-aware Transformer to learn representations from both temporal and spatial dimensions. Different from prior works, in this work, we devise a dual-graphomer that can more adequately model the temporal dependencies and the mutual spatial interactions.

III. PROPOSED APPROACH: STGLOW

The overall framework of the STGlow model is illustrated in Fig. 1. It primarily consists of three components: 1) a dual-graphomer to extract the representations of motion behaviors and model the temporal dependencies and the mutual spatial interactions; 2) Glow-PN to learn the underlying distribution of complex motion behaviors, conditioned on the social interactions; and 3) a bidirectional trajectory prediction module to forecast diverse future trajectories. To facilitate check, we assemble the primary denotations and their accompanying explanations in the Nomenclature section.

A. Problem Formulation

Given \( N \) pedestrians with observed trajectories \( \mathcal{X} = \{X_1^{T_{t-1}} \ldots X_{N}^{T_{t+1}}\} \) from time steps \( T_{t-1} \) to \( T_0 \) in the scene, the trajectory prediction algorithm aims to predict the future trajectories \( \mathcal{Y} = \{Y_1^{T_{t+1}}, \ldots, Y_{N}^{T_{t+2}}\} \) of all pedestrians.
in the upcoming time steps \( T_1 \) to \( T_p \), where \( X_i^t = (x_i^t, y_i^t) \in \mathbb{R}^2 \) is the position of the \( i \)-th pedestrian at the time step \( t \). The trajectory prediction algorithm takes as input the observed trajectories with \( t_o \) time steps of all pedestrians in a scene and aims to predict their future trajectories in the next \( t_p \) time steps by a model \( f(\cdot) \), denoted by

\[
\hat{Y} = f(X; W^*)
\]

where \( \hat{Y} \) is the set of future trajectories predicted by \( f(\cdot) \) and \( W^* \) represents the set of learnable parameters in the model.

For the sake of brevity, we hereafter drop the superscript when there is no ambiguity, that is, \( X_i \triangleq X_i^t \) and \( Y_i \triangleq Y_i^{(t_p)} \). We further use \( X, Y \) to represent a generic history trajectory and the corresponding future trajectory, respectively.

B. Dual-Graphormer

Influenced by surrounding agents, highly complex social interactions among pedestrians may force them to make decisions such as walking parallel, walking in groups, or changing direction and speed to avoid collisions. Obviously, such social interactions contain both temporal dependencies and spatial interactions, which are fundamentally important for accurately predicting trajectories. In this work, we design a dual-graphormer to more adequately model the temporal dependencies and the mutual spatial interactions. As shown in Fig. 1, our dual-graphormer primarily consists of two components: 1) a temporal graphormer (TG) and 2) a spatial graphormer (SG).

1) Temporal Graphormer: In this work, we decouple temporal dependencies into behavior-independent and -dependent temporal dependencies. The former type reveals the importance of each previous time step to the future trajectory. The other type of dependencies models the relationships among different previous motion behaviors across the temporal domain.

Assume that there are \( T \) time steps of each trajectory, denoted by \( X_i^{1:T} \). As shown in Fig. 2, the TG takes \( X_i^{1:T} \) as input and outputs a set of embeddings \( TH_i^{1:T} \) with temporal dependencies, which we define as temporal embedding. Specifically, we first construct a temporal graph by treating each time step as a node of the graph

\[
G_{imp} = (V_{imp}, A_{imp}).
\]

Here, \( V_{imp} = \{V_i | t = 1, \ldots, T\} \) represents nodes of \( G_{imp} \). \( A_{imp} \) is the adjacency matrix describing the temporal dependencies, that is, the motion behavior at a given time can only be affected by previous motion states rather than future motion states. Based on this fact, we define \( A_{imp} \) as

\[
A_{imp}^{ij} = \begin{cases} 1, & i \geq j, i, j \in [1, T] \\ -\infty, & i < j \end{cases}
\]

To model the behavior-independent temporal dependencies, we design a CE that learns a centrality embedding based on the length of the influence duration of each time step, which is formulated as

\[
C_i^t = \text{Linear}(\text{Deg}(V_i), \Theta_C); \quad t = 1, T
\]
where $C_t \in \mathbb{R}^D$, $\Theta_C$ is the set of learnable parameters, and $\text{Deg}(V_t)$ calculates the outdegree of the node $V_t$. The CE adaptively explores the importance of different time steps based on the influence duration.

For behavior-dependent temporal dependencies, we first adopt a nonlinear multilayer perceptron (MLP) to embed the position $X_i^t$ at each time step as

\[
    H_i^t = \text{MLP}(X_i^t, \Theta_N)
\]

(5)

where $\Theta_N$ contains the learnable parameters, and $H_i^t \in \mathbb{R}^D$. Besides, a positional embedding $P_i^t \in \mathbb{R}^D$ is applied to label the position of the motion state at each time step in a trajectory. In this article, we encode the time step positions in a learnable way, as proposed in [65]. Then, we update the node embedding by

\[
    H_i^t := H_i^t + C_i^t + P_i^t.
\]

(6)

According to the Transformer encoder [57], we further map $H_i^t$ into three values

\[
    Q_i = H_i^{1:T} W_Q, \quad K_i = H_i^{1:T} W_K, \quad V_i = H_i^{1:T} W_V
\]

(7)

where $W_Q, W_K, W_V$ are the parameters corresponding to the query $Q_i$, key $K_i$, and value $V_i$ of the pedestrian $i$. The output of TG can be further computed as

\[
    \text{Att}(Q_i, K_i, V_i) = \text{Softmax}\left( \frac{Q_i (K_i)^T}{\sqrt{d_k}} \odot \Theta_{\text{mp}} \right) V_i
\]

(8)

where $\Theta_{\text{mp}}$ is the adjacency matrix, $\odot$ is the operation of dot product, and $d_k$ is the dimension of $Q_i$. Equation (8) characterizes the behavior-dependent temporal dependencies, which builds the relationships among different motion behaviors across the temporal domain. For brevity, we write the TG as

\[
    \text{TH}_i^{1:T} = \text{TG}(X_i^{1:T}; \Theta_{\text{tg}})
\]

(9)

where $\text{TH}_i^{1:T}$ is the temporal embedding of the TG, and $\Theta_{\text{tg}}$ contains learnable parameters. By resorting to the TG, both the behavior-independent and -dependent temporal dependencies can be characterized.

2) Spatial Graphormer: Relative positions of other pedestrians are crucial for a target pedestrian to make decisions, such as adjusting velocity or direction to avoid collisions. However, consider only the relative positions of the neighbors is not sufficient for making a reasonable decision. For example, when a neighbor in the rear is walking in the opposite direction from the target pedestrian, the target pedestrian generally does not need to make adjustments, despite their close proximity. In reality, the relative steering angle of a neighbor to the target pedestrian and the walking direction of the target pedestrian are also key elements that drive pedestrians to change their motion states.

To fully model the mutual spatial interactions between pedestrians, in this work, we propose an SG by utilizing the relative motion states of the neighbors (i.e., the relative position and the relative steering angle to the target pedestrian) and the walking direction of the target pedestrian. Specifically, we take pedestrians in the scene as nodes and construct an undirected graph based on the walking direction of the target pedestrian and the relative motion states of the neighbors. First, at time step $t$, we can construct the spatial graph as

\[
    G_{\text{spa}} = (V_{\text{spa}}, A_{\text{spa}})
\]

(10)

where $V_{\text{spa}}$ represents the nodes of $G_{\text{spa}}$, and $A_{\text{spa}}$ is the adjacency matrix of $G_{\text{spa}}$ describing the mutual spatial relationships among pedestrians. To better capture spatial interactions, we endow each node $V_i^t \in V_{\text{spa}}$ with the information of both relative positions and relative steering angles, namely

\[
    V_i^t = R_i^t + S_i^t + \text{TH}_i^t
\]

(11)

where $\text{TH}_i^t$ denotes the temporal embedding of pedestrian $i$ at time step $t$, and $R_i^t$ and $S_i^t$ represent the embedding of the relative position and relative steering angle of pedestrian $i$ to the target pedestrian, respectively. As shown in Fig. 3, we adopt a single-layer MLP with ReLU activation to embed the relative position and relative steering angle of neighbors as

\[
    R_j^t = \text{MLP}(x_{j^\rightarrow_{i}, j^\rightarrow_{i}}; \Theta_R)
\]

\[
    x_{j^\rightarrow_{i}, j^\rightarrow_{i}} = x_{j} - x_{i}^t; \quad y_{j^\rightarrow_{i}, j^\rightarrow_{i}} = y_{j} - y_{i}^t
\]

(12)

\[
    S_j^t = \text{MLP}(\Delta s_{j}^t; \Theta_S)
\]

\[
    \Delta s_{j}^t = \frac{\Delta X_{j}}{\Delta X_{j}^t}
\]

(13)

where $(x_{j}^t, y_{j}^t)$ is the relative position of neighbor $j$ to the target pedestrian $i$ at time step $t$, $\Delta s_j^t$ denotes the corresponding relative steering angle, $\Delta X_{j}$ (i.e., $\Delta v_{x_{j}}$, $\Delta v_{y_{j}}$) = $(x_{j}^t - x_{j}^t_{-1}, y_{j}^t - y_{j}^t_{-1})$ denotes the walking direction of pedestrian $i$ at time step $t$, and $\Theta_R$ and $\Theta_S$ are the parameters of MLP. We refer to $R_j^t$ as the spatial embedding (SE), and $S_j^t$ as the steering embedding (HE). Note that we do not employ positional encoding when modeling spatial interactions since there is no natural order for pedestrians in the scene.
Intuitively, people commonly pay little attention to pedestrians outside their field of vision. For simplicity, we set the maximum binocular field of view to 180° in the horizontal position for a pedestrian. Based on this fact, we design the adjacency matrix $A_{\text{spa}}$ as

$$A_{\text{spa}}^{ij} = \begin{cases} 1, & \text{if } x_{i,j}^t, \Delta uv^t_{i} \geq 0 \text{ and } y_{i,j}^t, \Delta uv^t_{j} \geq 0 \text{ or } \text{inf}, & \text{Otherwise} \end{cases} \quad (14)$$

where $(\Delta uv^t_{i}, \Delta uv^t_{j})$ is the walking direction of pedestrian $i$ along the x- and y-axes, and $(x_{i,j}^t, y_{i,j}^t)$ is the relative position of neighbor $j$ to pedestrian $i$. $A_{\text{spa}}$ characterizes the mutual spatial relationships among pedestrians.

Similar to the TG, the output of the SG can be further computed as

$$Q^t = \text{SH}_{1:N}^{t}WQ^t, \quad K^t = \text{SH}_{1:N}^{t}WK^t,$$

$$V^t = \text{SH}_{1:N}^{t}WV^t,$$

$$\text{Att}(Q^t, K^t, V^t) = \text{Softmax}\left(\frac{Q^t(K^t)^T}{\sqrt{d_k}} \odot A_{\text{spa}}\right)V^t \quad (15)$$

where $N$ is the number of pedestrians in the scene, and $WQ^t, WK^t, WV^t$ are the parameters. For brevity, we write the SG as

$$\text{SH}_{1:N}^{t} = \text{SG}(X_{1:N}^{t-1}, \text{TH}_{1:N}^{t}; \Theta_{\text{sh}}) \quad (16)$$

where $\text{SH}_{1:N}^{t}$ is the output of the SG, which we define as spatial-temporal embedding. $\Theta_{\text{sh}}$ is the set of parameters. Similar to [57], the multihead attention mechanism is employed in our framework. Note that different from graph-based methods, our dual-graphormer takes advantage of the self-attention mechanism to avoid direct aggregation of connected nodes, which could greatly alleviate the over-smoothing problem. Such a conclusion has been carefully justified in [26].

3) Proposed Dual-Graphormer in Trajectory Prediction: To model the behavior pattern of pedestrians, we first apply the dual-graphormer to extract the deep representations of motion behaviors and social interactions. As shown in Fig. 1, we primarily extract representatives of two types of inputs, that is, the full trajectory and the observed historical trajectory. Note that the full trajectories are used only in the training phase.

The full trajectory reflects the motion behavior of each pedestrian in a time period, such as where to go and how to go. In our STGlow model, we define the representation of the full trajectory as the motion behavior of each pedestrian, which can be formulated as

$$\text{MB}_{\text{target}} = \text{TG}((\text{Concat}(X_{\text{target}}, Y_{\text{target}}), \Theta_{\text{tg}}) \quad (17)$$

where $\Theta_{\text{tg}}$ is the TG.

Besides, the observed historical trajectories of all pedestrians in the scene are encoded via the temporal and SG to extract the social interactions of the target pedestrian, which can be written as

$$\text{TH}_{1:N} = \text{TG}(X_{1:N}; \Theta_{\text{tg}})$$

$$\text{SH}_{\text{target}} = \text{SG}(X_{1:N}^{t-1}, \text{TH}_{1:N}^{t}; \Theta_{\text{sh}})$$

$$\text{TH}_{\text{target}} = \text{TG}(X_{\text{target}}; \Theta_{\text{tg}})$$

$$\text{ST}_{\text{target}} = \text{TH}_{\text{target}} + \text{SH}_{\text{target}} \quad (18)$$

where $\Theta_{\text{tg}}, \Theta_{\text{sh}}$, and $\Theta_{\text{tg}}$ are parameters. So far, we have obtained the representation $\text{MB}_{\text{target}}$ of motion behavior of the target pedestrian and the mutual social interaction $\text{ST}_{\text{target}}$. In the inference phase, we first predict the latent motion behavior $\text{MB}_{\text{target}}$ conditioned on $\text{ST}_{\text{target}}$. The $\text{MB}_{\text{target}}$ is then fed into a decoder to generate future trajectories.

C. Proposed Glow-PN

Due to the differences in human intent and unique behavior patterns, the motion behaviors of pedestrians are of high diversity. Namely, the future trajectories could be very different given the same historical trajectory. Thus, learning the underlying distribution of motion behaviors conditioned on social interactions is important to predict the diverse trajectories of pedestrians. Existing approaches commonly use GANs or CVAEs to generate diverse trajectories, where their limitations have been carefully discussed in Section I.

In our work, we simulate the process of degradation and evolution between the complex motion behavior and a simple behavior as follows:

$$\text{MB}_{\text{target}} \xrightarrow{f_1} h_1 \xrightarrow{h_2} \ldots \xrightarrow{h_{k-1}} z \quad (19)$$

where $f_1, f_2, \ldots, f_k$ denote a set of invertible transformations to simulate the degrading and evolving process, and $h_1, \ldots, h_{k-1}$ denote the intermediate motion behaviors. We can see that the forward process of formula (19) gradually degrades the complex motion behavior $\text{MB}_{\text{target}}$ into a simple behavior $z$, while its reverse process represents the evolution of a simple behavior to the complex motion behavior.

In reality, the complex motion behavior $\text{MB}_{\text{target}}$ of pedestrians is not accomplished at one stroke, but based on simple motion behaviors $z$ (e.g., every step of walking or every action) combined with individual behavior habits which may originate from various aspects such as people’s walking habits, traffic awareness, and potential intentions. For the sake of simplicity, we assume that $z$ follows a standard normal distribution, that is

$$z \sim N(z; 0, I) \quad (20)$$

where I is an identity matrix. Letting $x \triangleq \text{MB}_{\text{target}}$, the log-likelihood of complex motion behavior can be written as

$$\log p_{\theta}(x) = \log(p_{\theta}(z)|\text{det}(dz/dx)|) \quad (21)$$

$$= \log p_{\theta}(z) + \sum_{i=1}^{K} \log \text{det}(dh_i/dh_{i-1}) \quad (22)$$

$$= \log p_{\theta}(z) + \sum_{i=1}^{K} \log \text{det}(f_i^{-1}(x)) \quad (23)$$

Equation (21) holds because $\int p_{\theta}(x)dx = \int p_{\theta}(z)dz$, and (23) holds because $h_i = f_i^{-1}(h_{i-1})$. The first term in (23)
is the log-likelihood of the standard normal distribution and the scalar value \( \log |\det(J(f_i^{-1}(x)))| \) is the logarithm of the absolute value of the determinant of the Jacobian matrix \( J(f_i^{-1}(x)) \). This value reflects the transformation from \( h_{i-1} \) to \( h_i \) of the motion behavior under the transformation \( f_i \). Then, our framework optimizes the parameters by minimizing the negative log-likelihood function, where the loss is

\[
L_p = \min - \sum_{i=1}^{N} \log p_0(x_i). \tag{24}
\]

In contrast to previous approaches, our method can more precisely model the underlying data distribution by optimizing the exact log-likelihood of motion behaviors as shown in (24). Obviously, how to design the invertible transformation functions \( f_1, \ldots, f_k \) is essential for the optimization of (24). We should bear in mind that \( f_1, \ldots, f_k \) should be differentiable thus allowing the end-to-end training.

In our work, we employ Glow [52] to learn the invertible transformations of motion behaviors from complex to simple. As shown in Fig. 4, for the forward process, we take motion behavior \( \text{MB}_{\text{target}} \) as input of Glow-PN. After several “steps of flow,” the complex motion behavior is degraded into a simple behavior that can be represented by a simple distribution. A step of flow here consists of a PN, an invertible \( 1 \times 1 \) convolution, and an affine coupling layer, described below.

1) **Pattern Normalization:** Considering the fact that diverse motion behaviors share the same behavior pattern (e.g., the individual walking habits), we propose a PN for trajectory prediction as shown in Fig. 5. Different from the Actnorm adopted in the original Glow, PN performs normalization for multiple motion behaviors jointly at the sample axis. As will be shown in the experiments, our proposed PN greatly improves the prediction performance.

Specifically, the forward function, reverse function, and log-determinant of our proposed PN are calculated as

Function: \( y = s_g \odot x + b \)

Reverse Function: \( x = (y - b)/s_g \)

Log determinant: \( \log |s_g| \) \tag{25}

where \( x \) indicates the input of PN, and \( y \) signifies its output. Both \( x \) and \( y \) are tensors of shape \((K \times C \times 1)\) with \( K \) motion behaviors and channel dimension \( C \). \( s_g, b \in \mathbb{R}^{K \times C} \) are learnable scales and bias parameters whose initialization depends on the input data so that the data has zero mean and unit variance after PN. After initialization, the scale and bias are treated as regular trainable parameters that are independent of the data. Note that different from layer normalization [66], to ensure that the transformation \( f \) is invertible, we design the forward and reverse process of normalization in PN. The forward process normalizes the behavior pattern of each pedestrian to promote model convergence, while the reverse process denormalizes multiple motion behaviors to restore its original motion characteristics.

2) **Invertible \( 1 \times 1 \) Convolution:** In order that all channels of input in the forward transformation can be updated in subsequent coupling layers, we following Glow apply an invertible \( 1 \times 1 \) convolution layer before coupling layers. The weights \( W \) of convolution are initialized as a random rotation matrix and hence invertible. Thus, the log-determinant of this transformation is easy to compute

\[
\begin{align*}
    f_{\text{conv}}^{-1} &= W y \\
    \log |\det(J(f_{\text{conv}}^{-1}(y)))| &= \log |\det(W)| \tag{26}
\end{align*}
\]

where the log-determinant starts at zero and after one SGD step, the values start to diverge from zero.

3) **Affine Coupling Layer:** Generally, computing the determinants of high-dimensional Jacobian and large matrices is very expensive. Following Glow, we reduce the complexity by designing tractable and flexible invertible transformations. Specifically, we introduce an affine coupling layer, which can efficiently compute the forward function, reverse function, and log-determinant

\[
\begin{align*}
    x_a, x_b &= \text{split}(x_{\text{affine}}) \\
    (\log s_f, t) &= \text{Linear}(\text{MLP} (\text{concat}(x_a, \text{ST}_{\text{target}}))) \tag{29} \\
    x'_b &= s_f \odot x_b + t \tag{30} \\
    x' &= \text{concat}(x_a, x'_b) \tag{31}
\end{align*}
\]

where \( x_{\text{affine}} \in \mathbb{R}^{1 \times C \times K} \) represents the input of affine coupling layer, the \( \text{split}(\cdot) \) function splits \( x \) into two halves along the channel dimension, while the \( \text{concat}(\cdot) \) operation performs the restore operation. Since the inputs \( x_a \) of formula (29) remain unchanged during the affine transformation operation, formula (29) can be arbitrary transformation. Accordingly, in the reverse process of the affine coupling layer, \( s_f \) and \( t \) can be obtained from the output \( x_a \) through formula (29), and \( x'_b \) can be obtained by the reverse of \( x_b \). Note that our affine coupling layer is conditioned on the social interaction \( \text{ST}_{\text{target}} \) to establish the mapping of complex to simple motion behaviors.
As we can see, only the $s_f$ term changes the volume of the mapping in the affine coupling layer and adds a change of variables term to the loss. The log-determinant of the affine coupling layer and the final likelihood are hence computed as follows:

$$
\log \det \left( J \left( f^{-1}_{\text{coupling}}(x) \right) \right) = \log |s_f |
$$

(32)

$$
\log p_\theta(x) = \log p_\theta(z) + \sum_{j=0}^{n} \left( \log |s_j^f| + \log \det |W_j| + \log |s_j^f| \right)
$$

(33)

where $p_\theta(z) = \mathcal{N}(z; 0, I)$, and its log-likelihood can be easily computed as $-z(x)^T z(x)/2\sigma^2$, where $\sigma = I$. So far, we have eventually completed the design of the invertible transformation of the “step of flow.” Note that, different from CVAE-based methods, our method optimizes the exact log-likelihood of motion behaviors.

D. Bidirectional Trajectory Prediction

Due to the invertible design of Glow-PN, a simple behavior drawn from the standard normal distribution can be directly evolved into a complex motion behavior $\text{MB}_{\text{target}}$ through the reverse process of Glow-PN, conditioned on the social interaction $\text{ST}_{\text{target}}$. Then, we feed the obtained $\text{MB}_{\text{target}}$ into the decoder to predict the future trajectory.

1) Bidirectional Decoder With Goal Estimation: To alleviate the error accumulation issue caused by the RNN in trajectory prediction, as shown in Fig. 6, we adopt the decoder proposed in Bitrap [20], which is designed in a bidirectional manner. We additionally predict and supervise the forward and backward future trajectory to strengthen the representation learning during bidirectional prediction. Specifically, we first predict the goal of each pedestrian as

$$
\hat{G}'_i = \text{MLP}(\text{MB}'_{\text{target}}; \Theta_s)
$$

(34)

where $\text{MB}'_{\text{target}}$ is the representation of complex motion behaviors evolved by the reverse process of Glow-PN from simple behaviors. Then, the forward trajectory $\hat{Y}_f^t$ is predicted as

$$
\begin{align*}
\hat{f}_i^{t-1} & = \text{MLP}(\hat{f}_i^{t-1}; \Theta_{h}) \\
\hat{f}_h & = \text{GRU}(\hat{f}_h^{t-1}, \hat{f}_i^{t-1}; \Theta_{h}) \\
\hat{Y}_f^t & = \text{FC}(\text{concat}(\hat{f}_i^{t}, \text{MB}'_{\text{target}}); \Theta_{fy})
\end{align*}
$$

(35)

where $t$ from 1 to $t_p$ represents the time step of the forward prediction, and $\hat{f}_h^0 = \text{MLP}(\text{MB}'_{\text{target}}; \Theta_{f_h})$. Furthermore, the backward prediction $\hat{Y}_b^t$ is formulated as

$$
\begin{align*}
\hat{b}_h^t & = \text{GRU}(\hat{b}_h^{t+1}, \hat{b}_h^t; \Theta_{h}) \\
\hat{Y}_b^t & = \text{FC}(\text{concat}(\hat{b}_h^t, \text{MB}'_{\text{target}}); \Theta_{fy})
\end{align*}
$$

(36)

where $\hat{b}_h^t$ from $t_p$ to 1 denotes the time step of the backward prediction, and $\hat{b}_h^{t+1} = \text{MLP}(\text{MB}'_{\text{target}}; \Theta_{h})$. Last, the bidirectional prediction $\hat{Y}_b^t$ considering both forward and backward trajectories is predicted as

$$
\hat{Y}_{\text{Both}}^t = \text{FC}(\text{concat}(\hat{b}_h^t, \hat{f}_h^t); \Theta_{both})
$$

(37)

where $\hat{b}_h^t = \text{MLP}(\hat{G}'_i, \Theta_{h})$. Besides, $\Theta_s$ indicates the set of parameters. We supervise the predicted forward, backward, and bidirectional trajectories as

$$
L_{\text{tra}} = \min_k \left\langle \sum_{i=1}^{t_p} \left( \lambda_1 \|G'_k - \hat{G}'_i\| + \min_{k \in K} \sum_{t=1}^{t_p} \left( \lambda_2 \|Y'_k - \hat{Y}'_t\| + \lambda_3 \|Y'_i - \hat{Y}'_t\| \right) \right) \right\rangle
$$

(38)

where $K$ represents the number of samples of future trajectories, and $\lambda_1, \lambda_2, \lambda_3$ are the coefficients that balance different losses. Combined with the log-likelihood of the motion behaviors, our final loss is formulated as

$$
L_{\text{total}} = L_p + \sum_{i=1}^{N} L_{\text{tra}}.
$$

(39)

2) Inference: In the inference phase, we can easily generate the diverse trajectories by simple behaviors sampled from the standard normal distribution through the reverse process of Glow-PN conditioned on social interactions.

IV. EXPERIMENTS

In this section, we evaluate the performance of our proposed STGlow, which is implemented using the PyTorch framework. All the experiments are performed on Ubuntu 18.04 with an NVIDIA 3090 GPU. Our source code and trained models will be publicly available upon acceptance.

A. Implementation Details

The dimension of the node embedding is set as 256, and the number of coupling layers and invertible $1 \times 1$ convolutions are empirically set as 16. We also output 64 of the channels after every four coupling layers. In the training phase, the number of samples of predicted trajectories is set to 20 and the coefficients of the final loss are set to be 1, 0.25, 0.25, and 0.5, respectively. We adopt the Adam algorithm [68] to optimize the loss function (39) and train our network with the following hyperparameter settings: batch size is 128, learning rate is 1e-3, betas are 0.9 and 0.999, weight decay is 1e-6, and the number of epochs is 400.
TABLE I

Quantitative Results on the ETH/UCY Dataset With the Best-of-20 Strategy in ADE/FDE Metric. Lower Is Better

| Methods          | Datasets | Sampling | ETH ADE | ETH FDE | Hotel ADE | Hotel FDE | Univ ADE | Univ FDE | Zara1 ADE | Zara1 FDE | Zara2 ADE | Zara2 FDE | AVG ADE | AVG FDE |
|------------------|----------|----------|---------|---------|-----------|-----------|---------|---------|-----------|---------|-----------|---------|---------|---------|
| S-GAN [8]        | 20       | 0.87 1.02| 0.67 1.37| 0.76 1.52| 0.35 0.68 | 0.42 0.84 | 0.61 1.21|
| Social-STGCNN    | 20       | 0.64 1.11| 0.40 0.85| 0.44 0.79| 0.34 0.53 | 0.30 0.48 | 0.44 0.75|
| GTPPO [24]       | 20       | 0.63 0.98| 0.19 0.30| 0.35 0.60| 0.20 0.32 | 0.18 0.31 | 0.31 0.50|
| TPMS [11]        | 20       | 0.52 0.89| 0.22 0.39| 0.55 1.13| 0.35 0.70 | 0.27 0.56 | 0.38 0.73|
| TPNSTA [13]      | 20       | 0.51 0.87| 0.22 0.39| 0.52 1.09| 0.34 0.68 | 0.26 0.54 | 0.37 0.71|
| STAR [10]        | 20       | 0.56 1.11| 0.26 0.50| 0.52 1.15| 0.41 0.90 | 0.31 0.71 | 0.41 0.87|
| Trajectory++     | 20       | 0.39 0.83| 0.12 0.21| 0.20 0.44| 0.15 0.33 | 0.11 0.25 | 0.19 0.41|
| AgentFormer      | 20       | 0.45 0.75| 0.14 0.22| 0.25 0.45| 0.18 0.30 | 0.14 0.24 | 0.23 0.39|
| PECNet [18]      | 20       | 0.54 0.87| 0.18 0.24| 0.35 0.60| 0.22 0.39 | 0.17 0.30 | 0.29 0.48|
| SGCN [23]        | 20       | 0.63 1.03| 0.32 0.55| 0.37 0.70| 0.29 0.53 | 0.25 0.45 | 0.37 0.65|
| DMRGCN [48]      | 20       | 0.60 1.09| 0.21 0.30| 0.35 0.63| 0.29 0.47 | 0.25 0.41 | 0.34 0.58|
| Y-Net+tTST [19]  |         | 1000 0.28 0.35| 0.10 0.14| 0.24 0.41| 0.17 0.27 | 0.13 0.22 | 0.18 0.27|
| CAGN [67]        | 20       | 0.41 0.65| 0.13 0.23| 0.32 0.54| 0.21 0.38 | 0.16 0.33 | 0.25 0.43|
| MID [14]         | 20       | 0.39 0.66| 0.13 0.22| 0.22 0.45| 0.17 0.30 | 0.15 0.27 | 0.21 0.38|
| BiTrAP [20]      | 20       | 0.37 0.69| 0.12 0.21| 0.17 0.37 | 0.13 0.29 | 0.10 0.21 | 0.18 0.35|
| STGlow (ours)    | 20       | 0.31 0.49| 0.09 0.14| 0.16 0.33| 0.12 0.24 | 0.09 0.19 | 0.15 0.28|

Improvement 20 10% 25% 25% 33% 6% 11% 8% 17% 10% 10% 13% 19%

B. Datasets and Metrics

Datasets: We evaluate our method on two benchmark public pedestrian trajectory prediction benchmarks including the ETH/UCY dataset [31], [69] and the Stanford Drone Dataset (SDD) [70]. The ETH and UCY dataset group consists of five video sequences: ETH & HOTEL (from ETH) and UNIV, ZARA1, and ZARA2 (from UCY). All trajectories are converted to world coordinates, so the results we report are in meters. SDD comprises more than 11 000 unique pedestrians across 20 top-down scenes captured on the Stanford University campus in a bird’s eye view containing several moving agents like humans and vehicles. We use the standard test train split as used in [18] and other previous works.

Metrics: For the sake of fairness, we adopt the evaluation metrics average displacement error (ADE) and final displacement error (FDE) which are commonly used in literature [8], [9], [12], [32]. ADE computes the average $\ell_2$ distance between the predictions and the ground-truth future, while FDE computes the $\ell_2$ distance between the predicted and ground truth at the last observed point. The number of observed time steps is 8 (3.2 s) of each person and the upcoming trajectory of 12 time steps (4.8 s) is used to predict. For the ETH/UCY dataset, we use the widely adopted leave-one-out approach evaluation methodology such that we train our model on four scenes and test on the remaining one [8], [11], [13]. Considering the diversity of future trajectories, we use the best-of-K strategy to compute the final ADE and FDE with $K = 20$.

C. Baselines

We compare with the following baselines including previous state-of-the-art methods.

GAN-Based Methods: S-GAN [8]: a model that employs GANs with a global pooling module to generate diverse pedestrian trajectories; TPMS [11] and TPNSTA [13]: methods based on the temporal pyramid network to model the global and local context of motion behavior, the latter further designs the spatial–temporal attention mechanism.

CVAE-Based Methods: Trajectron++ [16]: a recurrent graph-based forecasting model incorporating dynamic constraints; PECNet [18]: a goal-conditioned trajectory prediction network; BiTrAP [20]: a goal-conditioned bidirectional trajectory prediction method based on the CVAEs.

Transformer-Based Methods: AgentFormer [12]: a transformer-based approach that models the time and spatial dimensions simultaneously; STAR [10]: a spatial–temporal graph transformer framework.

Other Methods: Y-Net [19]: a method based on position and visual image information. “tTST” stands for the test-time sampling trick (tTST) in postprocessing, which first samples 10 000 trajectories and then clusters them into 20 trajectories; CAGN [67]: a complementary attention-gated network for pedestrian trajectory prediction; SIT [71]: a tree-based method for pedestrian trajectory prediction; MID [14]: a method based on the diffusion model.

D. Quantitative Analysis

We quantitatively compare our STGlow with a wide range of current methods. Table I compares our method with existing algorithms on the ETH/UCY dataset. Besides the performance on each dataset, we report the average results for each method in the last two columns. Note that we also report the sampling number since adding the sampling number can effectively promote the performance [14]. Based on the results, we draw the following conclusions.

1) In general, with the same sampling number of 20, our method STGlow outperforms all the previous approaches in terms of ADE and FDE for all datasets. The last row of Table I shows the performance improvement of our method over the previous best methods (marked with red underline), where our method...
Table II: Quantitative Results on the SDD Dataset With the Best-of-20 Strategy in ADE/FDE Metric. * Means the Results Are Reproduced by [14] With the Official Released Code. Lower Is Better

| Methods         | Sampling | ADE  | FDE  |
|-----------------|----------|------|------|
| S-GAN [8]       | 20       | 27.23| 41.44|
| PECNet [18]     | 20       | 9.96 | 15.88|
| Y-Net + TTST [19]| 10000   | 7.85 | 11.85|
| Y-Net* [19]     | 20       | 8.97 | 14.61|
| GTFFO [24]      | 20       | 10.13| 15.35|
| Trajectory++* [16]| 20  | 8.98 | 19.02|
| SIT [71]        | 20       | 8.59 | 15.27|
| MID [14]        | 20       | 7.61 | 14.30|
| STGlow (ours)   | 20       | 7.20 | 11.20|

improves the ADE/FDE metrics by an average of 13%/19% on ETH/UCY datasets, respectively.

2) Compared with the GAN-based method [8], [11], [13], our method achieves significant performance gains on ADE and FDE metrics. For example, our method achieves 59% and 61% relative improvements in average ADE and FDE metrics over the GAN-based method TPNSTA. In addition, compared with the methods based on CVAEs [12], [16], [18], [20] and diffusion model [14], our method also achieves a greater performance improvement due to the optimization of the exact log-likelihood of motion behavior rather than the variational lower bound. For instance, our method improves the average ADE and FDE metrics by 17% and 20%, respectively, compared with the best CV AE-based method BiTraP [20], and 29% and 26%, respectively, compared with MID [14].

3) Compared with the graph-based approach [9], [23], [24], [48] that models social interactions in an intuitive way, we integrate the graph structure with the Transformer structure in modeling social interactions and achieve significant performance improvements. For example, compared with DMRCGCN [48], our STGlow improves 56% and 52% in average ADE and FDE metrics, respectively.

4) Despite the unfair experimental settings, our method still achieves 17% and 4% performance improvement over Y-Net + TTST on average ADE/FDE metrics, respectively.

We can draw similar conclusions above on the SDD dataset, as shown in Table II, which indicates that the proposed STGlow has better generalization performance. It is worth noting that STGlow achieved a relative improvement of 20% and 23% in ADE and FDE metrics compared to Y-Net [19] without using “TTST” postprocessing. Furthermore, compared with the current state-of-the-art method MID [14], our method achieves the best performance on ADE and FDE metrics.

E. Qualitative Analysis

In this section, we present visual examples to further illustrate the ability of our STGlow to fully explore complex social interactions between pedestrians and generate reasonable and diverse future trajectories.

1) Components of Our Architecture: We first explore the impact of each component of our architecture, including...
Fig. 7. Visualization of predicted trajectories on the ETH/UCY Dataset. Given the history trajectories (blue line), we illustrate the ground-truth paths (red line) and predicted future trajectories by our STGlow (green line) and the previous state-of-the-art BiTrap (orange line) for five different scenes, and dots are the locations of pedestrians at different time steps. Best viewed in color and zoomed-in view for more clarity.

Fig. 8. Examples of diverse predictions of our method with and without dual-graphormer (ST) on SDD. The black dash represents the pedestrian historical trajectory, the red dash denotes the ground-truth trajectories, the thin blue solid line represents the predicted diverse trajectories, and the green area represents the walkable area, which is visualized by the kernel density estimation map drawn from the predicted 20 trajectories. Better viewed in color.

TABLE III
ADE and FDE Performance of Variants on the SDD Dataset. ○ and △ Indicate GRU and Transformer, Respectively

| Components | Variants |
|------------|----------|
| St | Tg | Pn | BID |
| ✓ | ✓ | ✓ | ✓ |
| ✓ | ✓ | ✓ | ✓ |
| ✓ | ✓ | ✓ | ✓ |
| ✓ | ✓ | ✓ | ✓ |
| ✓ | ✓ | ✓ | ✓ |
| ✓ | ✓ | ✓ | ✓ |
| ✓ | ✓ | ✓ | ✓ |

TABLE IV
Performance of Variants of Dual-Graphormer on the SDD Dataset

|                   | Results | Results |
|-------------------|---------|---------|
| CE PE A_{geo}     |         |         |
| ✓ × ×             | 7.27/11.42 | ✓ × × | 7.25/11.35 |
| × ✓ ×             | 7.22/11.43 | × ✓ × | 7.25/11.45 |
| × × ✓             | 7.24/11.48 | × × ✓ | 7.24/11.36 |
| × × ×             | 7.30/11.46 | × × × | 7.29/11.47 |
| ✓ ✓ ✓             | 7.24/11.38 | ✓ ✓ ✓ | 7.23/11.42 |
| ✓ ✓ ×             | 7.20/11.20 | ✓ ✓ × | 7.20/11.20 |

TABLE V
Ablation Experiments on Hyperparameters of the Loss Function on the SDD Dataset

| α | A1 | A2 | A3 | ADE/FDE |
|---|----|----|----|---------|
| 0.0 | 0.0 | 1.0 | 1.0 | 7.26/11.39 |
| 0.0 | 1.0 | 0.0 | 0.0 | 7.25/11.50 |
| 0.25 | 0.25 | 0.5 | 0.73 | 7.23/11.35 |
| 0.5 | 0.5 | 0.0 | 0.73 | 7.30/11.47 |
| 0.0 | 0.0 | 1.0 | 1.0 | 7.24/11.30 |
| 0.0 | 1.0 | 0.0 | 0.73 | 7.26/11.45 |
| 0.25 | 0.25 | 0.5 | 0.73 | 7.23/11.28 |
| 0.5 | 0.5 | 0.0 | 0.73 | 7.27/11.44 |
| 0.0 | 0.0 | 1.0 | 1.0 | 7.25/11.25 |
| 0.0 | 1.0 | 0.0 | 0.73 | 7.27/11.28 |
| 1.0 | 0.0 | 0.0 | 0.73 | 7.25/11.40 |
| 0.25 | 0.25 | 0.5 | 0.73 | 7.20/11.20 |
| 0.5 | 0.5 | 0.0 | 0.73 | 7.25/11.25 |

SG, TG, PN, and BiDs. In the variations of our approach, we choose GRU or Transformer to replace our TG to extract the representation of motion behaviors and use a widely used forward decoder to replace our BiD. The ablation results are summarized in Table III. Obviously, each component in our framework improves performance to some extent. Compared with the third and fifth rows, it can be seen that our TG can extract temporal dependencies better than GRU due to considering the temporal dependencies in both behavior-independent and -dependent situations. Besides, considering the uniqueness of the behavior pattern, our PN normalizes the behavior pattern of each pedestrian, bringing a huge performance gain compared to the last two rows. Note that our spatial and TG is significantly better than the standard.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Transformer on modeling social interactions in the fourth row, which further demonstrates the effectiveness of our proposed dual-graphormer. Other ablation results also demonstrated the importance of our proposed components.

2) Dual-Graphormer Analysis: To investigate the impact of each component in the developed TG and SG on trajectory prediction, including the CE, PE, adjacency matrix in temporal (A_{\text{tmp}}), SE, HE, and adjacency matrix in spatial (A_{\text{spa}}), we conducted corresponding ablation experiments which are summarized in Table IV. As shown by the left and right groups in the table, in general, each component designed in TG and SG helps to better model social interactions in both temporal and spatial domains, resulting in improved performance. Note that when no components are adopted, our TG and SG degenerate into commonly used Transformers, where the time step nodes and pedestrians in the scene are treated as fully connected undirected graphs. Obviously, our SG and TG significantly outperform the widely used Transformer that does not consider temporal dependencies and mutual spatial interactions.

3) Hyperparameter Analysis: To realize the rationality of the hyperparameter settings in the loss function \( L_{\text{traj}} \), we further performed ablation experiments for hyperparameters \( \alpha \), \( \lambda_1 \), \( \lambda_2 \), and \( \lambda_3 \). The experimental results are summarized in Table V. Since the forward and backward trajectory predictions have the same importance, we generally keep the \( \lambda_1 \) and \( \lambda_2 \) consistent for both in ablation experiments. Note that when \( \lambda_1 = 1.0 \), we adopt the forward predicted trajectories as the final trajectories. Similarly, when \( \lambda_2 = 1.0 \), the backward predicted trajectories served as the final trajectories. In addition, we adopt the forward predicted trajectories as the final trajectories when \( \lambda_1 = 0.5 \), \( \lambda_3 = 0.0 \) and the bidirectional predicted trajectories as the final trajectories when \( \lambda_3 \neq 0.0 \). As shown in Table V, in general, each part of the loss function \( L_{\text{traj}} \) (i.e., goal estimation, forward trajectory prediction, backward trajectory prediction, and bidirectional trajectory prediction) contributes to the proposed model. Furthermore, we observed that bidirectional trajectory prediction (\( \lambda_3 = 1.0 \)) is slightly superior to forward trajectory prediction (\( \lambda_1 = 1.0 \)) and backward trajectory prediction (\( \lambda_2 = 1.0 \)). When supervising unidirectional and bidirectional trajectory prediction simultaneously, the model achieves the best performance. Based on the experimental results, we finally adopt: \( \alpha = 1.0 \), \( \lambda_1 = 0.25 \), \( \lambda_2 = 0.25 \), and \( \lambda_3 = 0.5 \) in our experiments.

G. Inference Time Analysis

To verify the efficiency of our proposed method, we conduct a comparison experiment on inference time with existing mainstream trajectory prediction frameworks. As demonstrated in Table VI, our method is inferior only to BiTrap in inference time but has significant performance improvements.

### Table VI

| Methods       | BiTrap | Trajectron++ | Y-Net+TTST | AgentFormer | DMRGCN | Ours |
|---------------|--------|--------------|------------|-------------|--------|------|
| Time (ms/step)| 0.014  | 0.864        | 13.294     | 1.882       | 0.852  | 0.174|

![Fig. 9. Cases with poor performance on the ETH/UCY dataset. (a)–(c) Three examples of different scenarios. Better viewed in color.](image)

**H. Discussion**

In this section, we discuss some limitations of our approach. First, we experimentally find that our method fails to accurately predict the scenes when the motion behaviors of pedestrians change drastically in a short period of time. Fig. 9 provides several examples where pedestrians abruptly transition their motion behaviors. Specifically, pedestrian A in Fig. 9(a) was in a standing state for the first six time steps within the eight observed time steps and started to move slowly in the last two time steps. In Fig. 9(b), pedestrian A walked to the right at a constant speed for the first five time steps of the observation, but suddenly reduced her speed and changed her walking direction in the last three time steps. In Fig. 9(c), pedestrian B walked at a very slow speed for the first seven time steps of the observation and then started walking at a normal speed in the last time step. Meanwhile, pedestrian A was stationary until the last time step of the observation, when it suddenly began walking. Though our approach can still cope with such scenarios somewhat better than existing methods such as BiTrap and AgentFormer, it still has a large error compared to the ground truth trajectory. How to address the scenes when the motion behaviors of pedestrians change drastically still needs more effort in future research.

Besides, modeling complex social interactions between pedestrians increases the time cost in the inference phase. When evaluated on the ETH/UCY dataset, our method required 0.174 ms/step, BiTrap required 0.014 ms/step, Trajectron++ required 0.864 ms/step, and DMRGCN required 0.852 ms/step. Although our method is faster than Trajectron++ and DMRGCN, it is much slower than BiTrap. Fortunately, there has been a lot of recent work focusing on improving the efficiency of Transformers [72], [73]. We leave it as future work to build more efficient interaction modules.

V. Conclusion

In this article, we have introduced a novel STGlow framework for trajectory prediction. Different from previous approaches, our method can more precisely model the underlying data distribution by optimizing the exact log-likelihood of observations. Besides, our method has clear physical meanings to simulate the evolution of human motion behaviors.
where the forward process of the flow gradually decouples the complex motion behavior into a series of simple behaviors, while its reverse process represents the evolution of simple behaviors to the complex motion behavior. In addition, we have designed a novel dual-graphformer to extract the global social interaction of pedestrians in both temporal and spatial domains. Both quantitative and qualitative experimental results demonstrate the superiority of our approach under various situations.

REFERENCES

[1] J. Liang, L. Jiang, J. C. Niebles, A. G. Hauptmann, and L. Fei-Fei, “Peeking into the future: Predicting future person activities and locations in videos,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 5718–5727.

[2] P. T. Szemes, H. Hashimoto, and P. Korondi, “Pedestrian-behavior-based mobile agent control in intelligent space,” IEEE Trans. Instrum. Meas., vol. 54, no. 6, pp. 2250–2237, Dec. 2005.

[3] B. Musleh, E. García, R. Otamendi, J. M. Arignold, and A. De la Escalera, “Identifying and tracking pedestrians based on sensor fusion and motion stability predictions,” Sensors, vol. 10, no. 9, pp. 8028–8053, Aug. 2010.

[4] C. Huang et al., “Self-supervised attentive generative adversarial networks for video anomaly detection,” IEEE Trans. Neural Netw. Learn. Syst., early access, Apr. 5, 2022, doi: 10.1109/TNNLS.2022.3159538.

[5] V. Bastani, L. Marcenaro, and C. S. Regazzoni, “Online nonparametric Bayesian activity mining and analysis from surveillance video,” IEEE Trans. Image Process., vol. 25, no. 5, pp. 2089–2102, May 2016.

[6] L. Lin, Y. Lu, Y. Pan, and X. Chen, “Integrating graph partitioning and matching for trajectory analysis in video surveillance,” IEEE Trans. Image Process., vol. 21, no. 12, pp. 4844–4857, Dec. 2012.

[7] F. Jiang, Y. Wu, and A. K. Katsaggelos, “A dynamic hierarchical clustering method for trajectory-based unusual video event detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2018, pp. 2255–2264.

[8] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, “Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 14412–14420.

[9] C. Yu, X. Ma, J. Ren, H. Zhao, and S. Yi, “Spatio-temporal graph transformer networks for pedestrian trajectory prediction,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 507–523.

[10] R. Liang, Y. Li, X. Li, Y. Tang, J. Zhou, and W. Zou, “Temporal pyramid network for pedestrian trajectory prediction with multi-supervision,” in Proc. ACM Conf. Art. Intell., vol. 35, 2021, pp. 2029–2037.

[11] Y. Yuan, X. Weng, Y. Ou, and A. Alahi, “Social GAN: Socially acceptable trajectories with generative adversarial networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 9089–9097.

[12] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, “Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 14412–14420.

[13] C. Yu, X. Ma, J. Ren, H. Zhao, and S. Yi, “Spatio-temporal graph transformer networks for pedestrian trajectory prediction,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 507–523.

[14] R. Liang, Y. Li, X. Li, Y. Tang, J. Zhou, and W. Zou, “Temporal pyramid network for pedestrian trajectory prediction with multi-supervision,” in Proc. ACM Conf. Art. Intell., vol. 35, 2021, pp. 2029–2037.

[15] Y. Yuan, X. Weng, Y. Ou, and A. Kitani, “AgentFormer: Agent-aware transformers for socio-temporal multi-agent forecasting,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 9793–9803.

[16] L. Shi et al., “SGCN: Sparse graph convolution network for pedestrian trajectory prediction,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 8990–8999.

[17] B. Huang, G. Yan, P. Wang, X. Chen, X. Song, and Y. Chen, “A novel graph-based trajectory predictor with pseudo-oracle,” IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 12, pp. 7064–7078, Dec. 2022.

[18] M. Li, S. Chen, Y. Shen, G. Liu, I. W. Tsang, and Y. Zhang, “Online multi-agent forecasting with interpretable collaborative graph neural networks,” IEEE Trans. Neural Netw. Learn. Syst., early access, Mar. 4, 2022, doi: 10.1109/TNNLS.2022.3152251.

[19] Y. Yang et al., “Do trajectory-based forecasts truly perform better for graph representation?” in Proc. Adv. Neurol. Inf. Process. Syst., vol. 34, 2021, pp. 28877–28888.

[20] D. Helbing and P. Molnár, “Social force model for pedestrian dynamics,” Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top., vol. 51, no. 5, pp. 4282–4286, May 1995.

[21] J. Martinez, M. J. Black, and J. Romero, “On human motion prediction models for human motion,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 2, pp. 283–298, Feb. 2008.

[22] J. M. K. Cay and C. Laugier, “Modelling smooth paths using Gaussian processes,” in Proc. Field Service Robot., 2008, pp. 381–390.

[23] R. Mehran, A. Oyama, and M. Shah, “Abnormal crowd behavior detection using social force model,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2009, pp. 935–942.

[24] S. Pellegrinì, A. Ess, K. Schindler, and L. van Gool, “You’ll never walk alone: Modeling social dynamics for multi-target tracking,” in Proc. IEEE 12th Int. Conf. Comput. Vis., Sep. 2009, pp. 261–268.

[25] A. Alahi, K. Goel, Y. Ramathan, A. Robicquet, L. Fei-Fei, and S. Savarese, “Social LSTM: Human trajectory prediction in crowded spaces,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 961–971.

[26] H. Xue, D. Q. Huyhn, and M. Reynolds, “PoPPL: Pedestrian trajectory prediction by LSTM with social force as route class clustering,” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 1, pp. 77–90, Jan. 2021.

[27] P. Zhang, W. Ouyang, P. Zhang, J. Xue, and N. Zheng, “SR-LSTM: State refinement for LSTM towards pedestrian trajectory prediction,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 12077–12086.

[28] J. Martinez, M. J. Black, and J. Romero, “On human motion prediction using recurrent neural networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 4673–4684.

[29] X. Shu, L. Zhang, G.-J. Qi, W. Liu, and J. Tang, “Spatiotemporal co-attention recurrent neural networks for human-skeleton motion prediction,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 6, pp. 3300–3315, Jun. 2022.

[30] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, “A recurrent latent variable model for sequential data,” in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 2980–2988.

[31] A. Graves and N. Jaitly, “Towards end-to-end speech recognition with recurrent neural networks,” in Proc. Int. Conf. Mach. Learn., 2014, pp. 1764–1772.

[32] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, “Show and tell: A neural image caption generator,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 3156–3164.

[33] Y. Xu, Y. Chen, Y. Zhang, and X. Gu, “Collaborative motion prediction via neural motion message passing,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 6318–6327.

[34] Y. Xu, J. Yang, and S. Du, “CF-LSTM: Cascaded feature-based long-term short-term networks for predicting pedestrian trajectory,” in Proc. AAAI Conf. Art. Intell., 2020, vol. 34, no. 7, pp. 12541–12548.

[35] B. Xu, X. Shu, and Y. Song, “X-invariant contrastive augmentation and representation learning for semi-supervised skeleton-based action recognition,” IEEE Trans. Image Process., vol. 31, pp. 3852–3867, 2022.
[43] X. Shu, B. Xu, L. Zhang, and J. Tang, “Multi-granularity anchor-contrastive representation learning for semi-supervised skeleton-based action recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 7559–7576, Jun. 2023.

[44] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, “Graph convolutional neural networks for web-scale recommender systems,” in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 974–983.

[45] H. Cai, S. Lv, G. Lu, and T. Li, “Graph convolutional networks for fast text classification,” in *Proc. 4th Int. Conf. Natural Lang. Process. (ICNLP)*, vol. 33, no. 1, Mar. 2019, pp. 420–425.

[46] A. Vemula, K. Muehling, and J. Oh, “Social attention: Modeling attention in human crowds,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 4601–4607.

[47] Y. Huang, H. Bi, Z. Li, T. Mao, and Z. Wang, “STGAT: Modeling spatial-temporal interactions for human trajectory prediction,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6271–6280.

[48] I. Bae and H.-G. Jeon, “Disentangled multi-relational graph convolutional network for pedestrian trajectory prediction,” in *Proc. AAAI Conf. Art. Intell.*, vol. 35, 2021, pp. 911–919.

[49] T. Zhao et al., “Multi-agent tensor fusion for contextual trajectory prediction,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12118–12126.

[50] J. Li, H. Ma, and M. Tomizuka, “Conditional generative neural system for probabilistic trajectory prediction,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 6150–6156.

[51] L. Dinh, J. Sohl-Dickstein, and S. Bengio, “Density estimation using real NVP,” *arXiv:1605.08803*, 2016.

[52] D. P. Kingma and P. Dhariwal, “Glow: Generative flow with invertible 1 × 1 convolutions,” in *Proc. Adv. Neu. Inf. Process. Syst.*, vol. 31, 2018, pp. 10236–10245.

[53] A. Lugmayr, M. Danelljan, L. V. Gool, and R. Timofte, “SRFlow: Learning the super-resolution space with normalizing flow,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 715–732.

[54] M. Kumar et al., “Videoflow: A conditional flow-based model for stochastic video generation,” *arXiv:1903.01434*, 2019.

[55] R. Prenger, R. Valle, and B. Catanzaro, “Waveglow: A flow-based generative network for speech synthesis,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, May 2019, pp. 3617–3621.

[56] S. Kim, S.-G. Lee, J. Song, J. Kim, and S. Yoon, “FloWaveNet: A generative flow for raw audio,” *arXiv:1811.02155*, 2018.

[57] A. Vaswani et al., “Attention is all you need,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[58] T. B. Brown et al., “Language models are few-shot learners,” in *Proc. Adv. Neu. Inf. Process. Syst.*, vol. 33, 2020, pp. 1877–1901.

[59] A. Dosovitskiy et al., “An image is worth 16×16 words: Transformers for image recognition at scale,” *2020*, arXiv:1911.01721.

[60] L. Yuan et al., “Tokens-to-Token VIT: Training vision transformers from scratch on ImageNet,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 538–547.

[61] J. You, Y. Li, J. Zhou, Z. Hua, W. Sun, and X. Li, “A transformer based approach for image manipulation chain detection,” in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 3510–3517.

[62] Y. Li, J. You, J. Zhou, W. Wang, X. Liao, and X. Li, “Image operation chain detection with machine translation framework,” *IEEE Trans. Multimedia*, early access, Oct. 17, 2022, doi: 10.1109/TMM.2022.3215000.

[63] Q. Song, B. Sun, and S. Li, “Multimodal sparse transformer network for bi-visual speech recognition,” *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Apr. 12, 2022. doi: 10.1109/TTNLS.2021.3163771.

[64] Y. Li, J. Zhou, J. Tian, X. Zheng, and Y. Y. Tang, “Weighted error entropy-based information theoretic learning for robust subspace representation,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 9, pp. 4228–4242, Sep. 2022.

[65] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, “Deformable DETR: Deformable transformers for end-to-end object detection,” 2020, arXiv:2010.04159.

[66] J. Lei Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” 2016, arXiv:1607.06450.

[67] J. Duan et al., “Complementary attention gated network for pedestrian trajectory prediction,” in *Proc. AAAI Conf. Art. Intell.*, 2022, pp. 1–9.

[68] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980.

[69] A. Lerner, Y. Chrysanthou, and D. Lischinski, “Crowds by example,” *Comput. Graph. Forum*, vol. 26, no. 3, pp. 655–664, Sep. 2007.

---

Rongxin Liang (Student Member, IEEE) received the B.Eng. degree in communication engineering from Wuyi University, Guangdong, China, in 2018, and the M.S. degree in information and communication engineering from Shenzhen University, Shenzhen, China, in 2021, where he is currently pursuing the Ph.D. degree with the College of Electronics and Information Engineering. His current research interests include trajectory prediction, anomaly detection, computer vision, and deep learning.

Jiantao Zhou (Senior Member, IEEE) received the B.Eng. degree from the Department of Electronic Engineering, Dalian University of Technology, Dalian, China, in 2002, the M.Phil. degree from the Department of Radio Engineering, Southeast University, Dhaka, Bangladesh, in 2005, and the Ph.D. degree from the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, in 2009. He held various research positions with the University of Illinois at Urbana-Champaign, Champaign, IL, USA, Hong Kong University of Science and Technology, and McMaster University, Hamilton, ON, Canada. He is an Associate Professor with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, and also the Interim Head of the newly established Center for Artificial Intelligence and Robotics. He holds four granted U.S. Patents and two granted Chinese Patents. His research interests include multimedia security and forensics, computer vision, and artificial intelligence, and big data.

Dr. Zhou has coauthored two articles that received the Best Paper Award at the IEEE Pacific-Rim Conference on Multimedia in 2007 and the Best Student Paper Award at the IEEE International Conference on Multimedia and Expo in 2016. He is serving as the Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING and the IEEE TRANSACTIONS ON MULTIMEDIA.

Yuanman Li (Member, IEEE) received the B.Eng. degree in software engineering from Chongqing University, Chongqing, China, in 2012, and the Ph.D. degree in computer science from the University of Macau, Macau, in 2018. From 2018 to 2019, he was a Post-Doctoral Fellow with the State Key Laboratory of Internet of Things for Smart City, University of Macau. He is currently an Assistant Professor with the College of Electronics and Information Engineering, Shenzhen University, Shenzhen, China. His current research interests include multimedia security and forensics, data representation, computer vision, and machine learning.

Xia Li (Member, IEEE) received the B.S. and M.S. degrees in electronic engineering and signal and information processing (SIP) from Xidian University, Xi’an, China, in 1989 and 1992, respectively, and the Ph.D. degree from the Department of information engineering, Chinese University of Hong Kong, Hong Kong, in 1997. Currently, she is a member of the Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen University, Shenzhen, China. Her research interests include intelligent computing and its applications, image processing, and pattern recognition.