Multi-Agent Trajectory Prediction With Spatio-Temporal Sequence Fusion

Yu Wang and Shiwei Chen

***Abstract*—Accurate trajectory prediction of surrounding agents is an important issue for building up an intelligent transportation system. Frequent interactions among agents have a major impact on their movement patterns. Current research mainly relies on agents' spatial structure associated with the last frame of the observation to model social interactions, while paying less attention to structure information from previous moments. In addition, existing methods merely consider temporal features of a single trajectory sequence, while neglecting temporal dependencies across multiple trajectories. In this work, we endeavor to capture comprehensively social interactions among agents with the proposed Spatio-Temporal Sequence Fusion Network (STSF-Net). Specifically, we construct a spatio-temporal sequence that encodes contextual information taking explicitly spatial distributions of agents during movement into account while capturing socially temporal dependencies across multiple trajectory sequences. Besides, a social recurrent mechanism is introduced to explicitly capture temporal correlations between interactions by concerning spatial structure at each time-step. Finally, our model is evaluated on datasets covering pedestrian, vehicle, and heterogeneous multi-agent trajectories. Experimental evidence manifests that our method achieves excellent performance.**

***Index Terms*—Multi-modal trajectory prediction, saptio-temporal sequence fusion, generative adversarial networks, sequence-to-sequence.**

I. Introduction

HIGH-PRECISION trajectory prediction is extremely important to various applications, *e.g.*, autonomous driving [1], [2], and agent navigation [3], as well as many downstream tasks, including object tracking [4]–[7], and person re-identification [8]–[11]. However, trajectory prediction is fairly sophisticated and formidable since it involves various factors such as social norms, environmental constraints, scene rules, etc. Specifically, interactions among agents and changes in context frequently occur, and thus agents have to consider these factors in time for safe moving. Adequate and appropriate physical spacing between agents is also necessary to be maintained for collision avoidance under emergency. Furthermore, physical constraints of the environment, *e.g.*, road conditions, and geographical context, also have a potential impact on agent trajectories. For instance, we have to avoid obstacles and move on a feasible terrain. Another intractable issue involved in trajectory prediction is multi-modal properties [12], [13], which means that there may be multiple feasible future trajectories given an observation. This phenomenon results in difficulty in modeling and calculations.

An increasing number of works have been investigated to tackle the challenges mentioned above. Conventional methods depend on manually extracting salient and discriminative features by leveraging domain-specific knowledge [14], [15] and devising efficient learning criteria for specific scenarios [16], [17]. Since deep learning-based methods have been widely proven to be effective in resolving computer vision tasks, numerous research has attempted to solve trajectory prediction issues through deep learning methods [12], [13], [18]–[22]. These methods try to investigate social behaviors among agents based on a data-driven paradigm, while it is agnostic for us to understand what kind of interactive information the model learned. Consequently, it is challenging to sufficiently exploit such knowledge for effective inference. Some recent studies [23], [24] pay much attention to capture social behaviors among agents based on their geographic locations, but only the spatial structure associated with the last frame of the observation is considered. Here, we point out two factors that are quite important for trajectory prediction tasks:

1) **The spatial distribution of agents throughout the observation is critical for future trajectory prediction, not just the observed final spatial structure.**
   Some works model social behaviors from nearby agents based on the location distribution at the end of the observation [23], [24]. These approaches ignore topology information during agent movement and merely focus on the final moment of the observation. Since social behaviors with varying degrees of importance have different potential effects on subsequent motion patterns, existing methods fail to model this aspect. As a consequence, we investigate this factor by explicitly considering spatial structures of agents across the entire observation window to help capture social behaviors.
2) **The temporal dependency across multiple trajectory sequences involved in the scene has a non-trivial effect on accurate inference.**
   Motion patterns of agents are potentially affected by neighbors around them throughout the movement. Over

Manuscript received 19 June 2021; revised 11 September 2021; accepted 5 October 2021. Date of publication 19 October 2021; date of current version 13 January 2023. The Associate Editor coordinating the review of this manuscript and approving it for publication was Prof. Ramanathan Subramanian. *(Corresponding author: Yu Wang.)*

Yu Wang is with the College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China (e-mail: yuwangtj@yeah.net).

Shiwei Chen is with Bilibili, Shanghai 200433, China (e-mail: chenshiwei@bilibili.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMM.2021.3120535.

Digital Object Identifier 10.1109/TMM.2021.3120535

time, early weak interactions may be amplified and have a significant impact on subsequent actions. Hence, the temporal dependency across multiple trajectory sequences is crucial for formulating social interactions. But most previous methods [12], [13] overlook this factor. They independently extract the temporal dynamics of each trajectory without paying attention to the temporal dependencies of these sequences.

In general, most previous approaches capture social interactions among agents only via fusing multi-agent trajectory features based on their final positions without considering the spatial distribution throughout the movement [24], [25], as well as temporal dependencies across multiple trajectory sequences [13], [26], [27]. Therefore, to address these concerns, we construct a novel spatio-temporal sequence, which jointly encodes spatial structures of agents throughout the movement and temporal dependencies across multiple trajectories in a unified way. Besides, we adopt a social recurrent mechanism to explicitly model temporal correlations between interactions when extracting temporal dynamics of a single trajectory. Our method is evaluated on publicly available datasets covering pedestrian and vehicle trajectories. Quantitative results and qualitative analyses demonstrate that our method is promising.

II. Related Work

Social trajectory prediction has attracted considerable attention due to its importance in the decision-making of robot navigation and autonomous driving vehicles. Previous works mainly focus on how to extract discriminative features. These feature extraction strategies require to capture various kinds of interactions among agents and constraints from the scene. Conventional methods mostly depend on hand-crafted features or well-engineered learning criteria for specific scenarios [14], [17], [28], thus lacking universality. Recently, plenty of deep learning-based strategies have been explored for social trajectory prediction. They concentrate on automatically capturing latencies affecting movement [12], [13], [24], thereby learning informative representations.

Agent-centric methods attempted to discover a merge function for fusing features of multiple agents. S-LSTM [12] exploited a max-pooling scheme to integrate agent interactions within a specified local range. But it failed to model the global context and temporal correlations between interactions. S-GAN [13] combined an encoder-decoder paradigm and the generative adversarial network to capture multi-modal trajectory distribution, and also proposed a global pooling mechanism to capture interactions. Nevertheless, S-GAN employed vanilla LSTM to independently extract representations of each trajectory, without concerning temporal dependencies across multiple sequences. Besides, attention mechanisms have recently been introduced into social trajectory prediction issues to mine latencies by concentrating on important clues [18], [29].

Another line of research focused on encoding interactions by explicitly considering the spatial relationships of agents. CS-LSTM [23] pre-designed a local grid and proposed a convolutional social pooling mechanism that considers the local spatial structure of agents involved in the pre-defined grid. ChauffeurNet [26] maintained the spatial structure of multiple agents by means of introducing a bounding box area. MATF-GAN [24] constructed a social tensor that jointly encodes static scene context and trajectory features of agents. Although MATF-GAN considered the global spatial structure of agents of the observed final locations, it failed to investigate temporal dynamics across trajectories and ignored the spatial distribution of agents throughout the motion.

Beyond that, some studies also attempted to apply graph neural networks for modeling social interactions among agents [30]–[32]. Social-BiGAT [30] utilized a graph attention network to extract reliable representations for generating multi-modal trajectory distributions. SR-LSTM [31] introduced a state refinement module to encode interactions from adjacent pedestrians through a graph-based message passing mechanism. Social-STGCNN [32] substituted the need of aggregation methods by formulating interactions as a spatio-temporal graph.

In conclusion, current prominent strategies on capturing social interactions among agents are to independently extract temporal features of each trajectory and then discover a fusion mechanism to aggregate these features. Some methods also insufficiently utilized the spatial structure in the process of feature fusion. Nevertheless, these approaches neglected spatial structures of agents throughout motion and also failed to consider temporal dependencies across multiple trajectories.

III. Notations and Problem Formulation

In this section, some notations used in this paper and the problem formulation are defined. For the observed n agents involved in the scenario, t_{obs} frames' historical trajectories, and t_{fut} frames' future trajectories, we define (x_i^t, y_i^t) as coordinates of the ith agent at the tth frame. The trajectory prediction task is formulated to exploit past moving experience to reason about future trajectories (x_i^t, y_i^t), where $t = t_{obs}+1, t_{obs}+2, \ldots, t_{obs}+t_{fut}$. We denote the predicted future trajectory coordinates by $(\hat{x}_i^t, \hat{y}_i^t)$. In general, the trajectory prediction task requires the network to take as input the historical trajectories of all dynamic agents involved in the scene and is required to infer their future trajectories.

IV. Proposed STSF-Net Framework

In this section, details about the proposed Spatio-Temporal Sequence Fusion Network (STSF-Net) are provided. STSF-Net comprehensively captures social behaviors by means of investigating spatio-temporal properties of all trajectories involved in the scene. The complete architecture is presented in Fig. 1. Our method is able to predict the future trajectories of all agents involved in the scene simultaneously. Next, we will discuss each module in detail.

A. Spatio-Temporal Sequence Construction

LSTM is widely regarded as competent to learn sequential correlations, and thus it is popular in trajectory prediction tasks. Nevertheless, a vanilla LSTM reasons about future trajectories

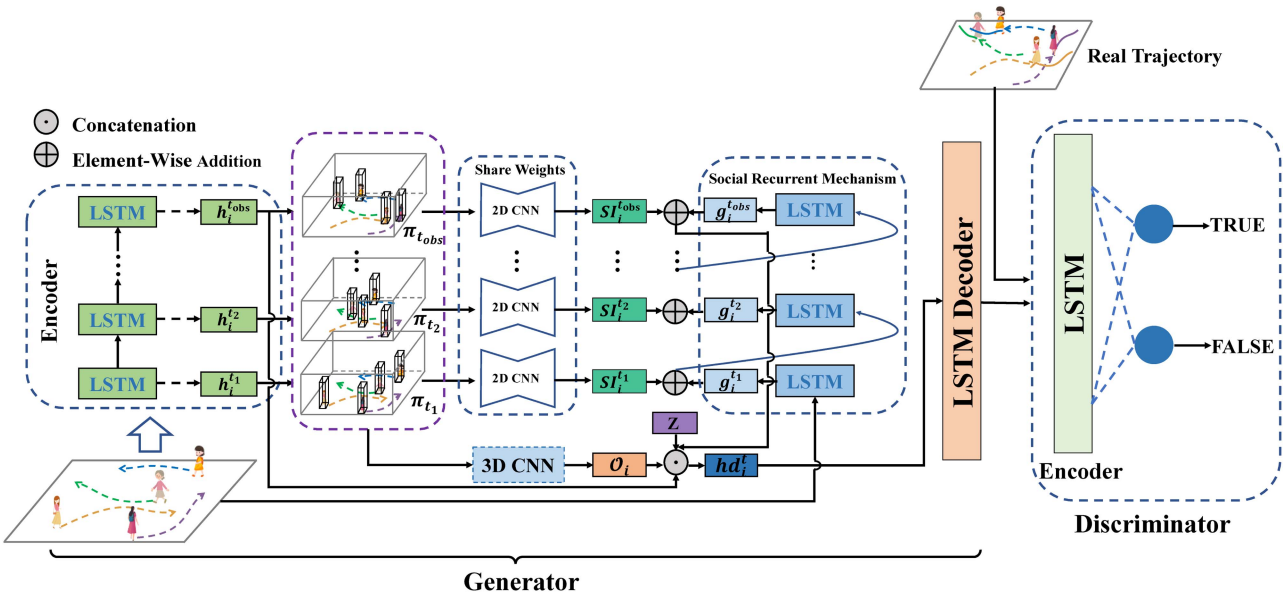

Fig. 1. The overall architecture of STSF-Net. The spatio-temporal sequence is constructed to model both temporal dependencies among multiple trajectories and the spatial structure of agents throughout the movement. Additionally, a fancy mechanism is incorporated into our framework to capture temporal correlations between interactions. The overall model is formulated in an end-to-end fashion.

independently without an ability of capturing social behaviors among agents. Before we utilize LSTM to learn trajectory representations, the coordinates of the ith agent at time-step t are encoded as a vector $\xi_i^t = \varphi(x_i^t, y_i^t)$, where φ is a linear embedding function. Then the coordinate embedding ξ_i^t is input into LSTM encoder to extract state representations as follows:

$$
\begin{aligned}
u_i^t &= \delta(W^u \xi_i^t + V^u h_i^{t-1} + b^u), \\
f_i^t &= \delta(W^f \xi_i^t + V^f h_i^{t-1} + b^f), \\
o_i^t &= \delta(W^o \xi_i^t + V^o h_i^{t-1} + b^o), \\
c_i^t &= f_i^t \odot c_{i-1}^t + u_i^t \odot tanh(W^c \xi_i^t + V^c h_i^{t-1} + b^c), \\
h_i^t &= o_i^t \odot tanh(c_i^t),
\end{aligned} \tag{1}
$$

where $W^\sharp$, $V^\sharp$ and $b^\sharp$, $\sharp \in \{u, f, o, c\}$, are learnable parameters in LSTM cells, and u, f, o and c denote input gate, forget gate, output gate and memory cell, respectively. δ is an activation function, *e.g.*, sigmoid function, and the symbol $\odot$ denotes element-wise product. Note that $h_i^t \in \mathbb{R}^{d_h}$ is the extracted state representation at time-step t.

In our framework, all LSTM encoders share parameters, and thus the model is able to handle any number of agents. The above mechanism is first employed to extract all agents' state representations $\{h_1^t, h_2^t, \ldots, h_n^t\}$ for each time-step $t = t_1, t_2, \ldots, t_{obs}$. These representation vectors reflect temporal features at each time-step.

To preserve the spatial structure of agents throughout the movement, we construct a global spatial tensor from a bird's-eye view for each time-step, where each element is initialized to 0. Then feature encodings of the corresponding time-step with the same dimension as the channel of the tensor are placed into the spatial tensor. Specifically, for a certain time-step t, we extract feature encodings of all agents $h_1^t, h_2^t, \ldots, h_n^t$, which

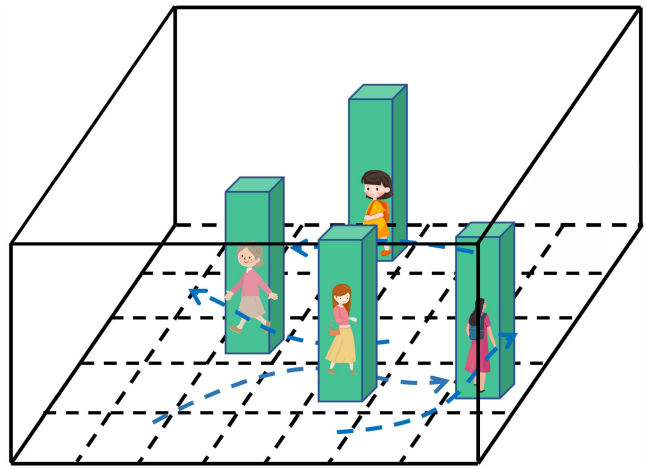

Fig. 2. The illustration of tensor construction for the time-step t. First, a bird's eye view scene is discretized into a series of cells whose elements are initialized to 0. The extracted features $h_1^t, h_2^t, \ldots, h_n^t$ (shown in green cubes) of all agents involved in the scene are placed into cells at their coordinates $(x_1^t, y_1^t), \ldots, (x_n^t, y_n^t)$ respectively to form a spatial tensor.

are respectively placed into the spatial tensor at coordinates $(x_1^t, y_1^t), (x_2^t, y_2^t), \ldots, (x_n^t, y_n^t)$. Such rasterization and alignment operation constructs a spatial feature map π_t, as shown in Fig. 2. If multiple feature encodings are placed into a same cell due to discretization, element-wise max-pooling is conducted. Likewise, we execute rasterization and alignment operations on each time-step, *i.e.*, $t_1, \ldots, t_{obs}$, and finally acquire a spatio-temporal sequence $\{\pi_1, \pi_2, \ldots, \pi_{t_{obs}}\}$.

B. Spatio-Temporal Sequence Fusion

In order to capture temporal dependencies across multiple trajectory sequences from the constructed spatio-temporal

sequence above, the 3D convolutional neural network is a natural choice. As a consequence, we stack 3D convolutional layers in a bottleneck structure, where the number of channels in the input and output layers of the entire structure is 32, and in the bottleneck layer is 16. Each convolution layer is followed by a ReLU nonlinearity and a 3D max-pooling layer for all datasets. A batch-normalization layer is also used on the vehicle trajectory dataset. For all convolution layers, kernel size is 3, padding is 1, and stride is 1. All pooling operations use a $2 \times 1 \times 1$ window. In this way, we encode spatio-temporal properties that existed in the constructed trajectory sequence and derive a fused tensor $\mathcal{O}$ of the same size as $\pi_{t_{obs}}$. Afterwards, fused vectors for each agent $\mathcal{O}_i$ are sliced out from $\mathcal{O}$ according to their coordinates $(x_i^{t_{obs}}, y_i^{t_{obs}})$.

C. Modeling Temporal Correlations Between Interactions

We utilize vanilla LSTMs to independently extract temporal dynamics of a single trajectory, and then capture temporal dependencies across multiple trajectories through the above spatio-temporal sequence construction and fusion mechanisms. However, when extracting temporal features of a single trajectory, interactions among agents are ignored. As a consequence, we further introduce a Social Recurrent Mechanism (SRM) by employing an extra LSTM to explicitly capture temporal correlations between interactions:

$$\begin{aligned} g_i^t &= LSTM(g_i^{t-1} \oplus SI_i^t), \\ SI_i^t &= [SI(\pi'_{t-1})]_{x_i^t, y_i^t}, \end{aligned} \tag{2}$$

where $\oplus$ is an element-wise addition operation, [ ] is a slice operation from the given coordinates, and π'_t is the generated spatial tensor using g_i^t with the same strategy as described in Section IV-A. SI is a spatial interaction module that consists of three 2D-convolutional layers in a bottleneck structure with kernel size 3, stride size 1, and padding size 1. The number of channels in the input and output layers is the same as π'_{t-1}, and in the bottleneck layer is half of π'_{t-1}. The first two convolutional layers are followed by a ReLU layer, and the last is followed by a Sigmoid nonlinearity. The social feature SI_i^t for the i-th agent at time-step t is sliced out from the generated tensor $SI^t(\pi'_{t-1})$ according to coordinates (x_i^t, y_i^t).

D. Multi-Modal Trajectory Prediction

Inspired by advancements in sequence generation, STSF-Net adopts an encoder-decoder structure to reason about future trajectories. Besides, motion behaviors of agents are inherently multi-modal, which means that there exist multiple socially-acceptable future trajectories when given an observed part. For this reason, the generative adversarial network, as employed in S-GAN [13], is integrated into our framework to generate multi-modal trajectory distribution, which covers the space of feasible trajectories while being in accordance with the observations. Specifically, we concatenate the state vector $h_i^{t_{obs}}$, the agent-specific spatio-temporal encoding $\mathcal{O}_i$, the socially recurrent feature $g_i^{t_{obs}}$, and a random noise z sampled from a standard Gaussian distribution, as the input to an LSTM decoder for future trajectory prediction. Besides, we also utilize a discriminator D that consists of an encoder followed by a Multi-Layer Perceptron (MLP) to distinguish the real trajectory $[X_i, Y_i]$ from the generated one $[X_i, \hat{Y}_i]$, where X_i is observed sequence, Y_i is the ground truth, and $\hat{Y}_i$ is the predicted future sequence.

E. Loss Function

In this work, the generative adversarial network is employed for multi-modal trajectory generation, and thus the adversarial loss is introduced:

$$\begin{aligned} \mathcal{L}_{gan} = \min_G \max_D \mathbb{E}_{x \sim p_{data}(x)}[logD(x)] \quad + \\ \mathbb{E}_{z \sim p(z)}[log(1 - D(G(z)))], \end{aligned} \tag{3}$$

where G and D are generator and discriminator, respectively. In addition, to measure the error between the predicted trajectory and the ground truth, the mean squared error of coordinates across the observation window is also considered:

$$\mathcal{L}_{mse} = \frac{1}{n} \sum_{i=1}^{n} \sum_{t=t_{obs}+1}^{t_{obs}+t_{fut}} (x_i^t - \hat{x}_i^t)^2 + (y_i^t - \hat{y}_i^t)^2, \tag{4}$$

Generally, we minimize the following loss function for the trajectory prediction task in STSF-Net:

$$\mathcal{L}_{all} = \mathcal{L}_{mse} + \lambda \mathcal{L}_{gan}, \tag{5}$$

where λ controls the importance of $\mathcal{L}_{mse}$ and $\mathcal{L}_{gan}$.

F. Implementation Details

We use an SGD optimizer to train STSF-Net on ETH and UTY datasets for 40 epochs with an initial learning rate of 0.001, with 0.5 times decay every 10 epochs. The size of hidden states extracted by LSTM is set to 32 for both encoder and decoder, and we follow the same data preprocessing strategy and training mechanism as S-GAN. Besides, STSF-Net is trained on NGSIM and Stanford Drone datasets using Adam optimizer [33] with a batch size of 64 for 10 epochs and 100 epochs, respectively. The learning rate is initialized as 0.001, using an exponential decay schedule with a rate of 0.5 every 5 epochs and 20 epochs respectively for NGSIM and Stanford Drone datasets. The size of hidden states in LSTM is set to 64 for both encoder and decoder. For all datasets, the random noise is a 16-dimensional vector sampled from a standard Gaussian distribution. MLP used in the discriminator consists of two fully-connected layers, of which the first layer has 1024 nodes followed by a ReLU nonlinearity, and the second has one node followed by a Sigmoid nonlinearity indicating whether the output is true or false. Our model is implemented on the PyTorch framework and trained on the Nvidia Titan V GPU.

V. Experiment

In this section, the proposed STSF-Net model is evaluated on publicly available benchmark datasets: ETH [41], UCY [42], Stanford Drone [43], NGSIM US-101 [44] and NGSIM I-80 [45]. The performance is compared with some state-of-the-art methods.

TABLE I
COMPARISON WITH DIFFERENT BASELINES USING ADE/FDE METRICS ON FIVE SCENARIOS FROM ETH AND UCY DATASETS

Dataset	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
Linear	1.33 / 2.94	0.39 / 0.72	0.82 / 1.59	0.62 / 1.21	0.77 / 1.48	0.79 / 1.59
Vanilla LSTM	1.09 / 2.41	0.86 / 1.91	0.61 / 1.31	0.41 / 0.88	0.52 / 1.11	0.70 / 1.52
S-Force [15]	0.67 / 1.52	0.52 / 1.03	0.74 / 1.12	0.40 / 0.60	0.40 / 0.68	0.54 / 0.99
S-LSTM [12]	1.09 / 2.35	0.79 / 1.76	0.67 / 1.40	0.47 / 1.00	0.56 / 1.17	0.72 / 1.54
S-GAN [13]	0.87 / 1.62	0.67 / 1.37	0.76 / 1.52	0.35 / 0.68	0.42 / 0.84	0.61 / 1.15
SoPhie [18]	0.70 / 1.43	0.76 / 1.67	0.54 / 1.24	0.30 / 0.63	0.38 / 0.78	0.54 / 1.15
PIF [19]	0.73 / 1.65	0.30 / 0.59	0.60 / 1.27	0.38 / 0.81	0.31 / 0.68	0.46 / 1.00
MATF-GAN [24]	1.01 / 1.75	0.43 / 0.80	0.44 / 0.91	0.26 / 0.45	0.26 / 0.57	0.48 / 0.90
Social-BiGAT [30]	0.69 / 1.29	0.49 / 1.01	0.55 / 1.32	0.30 / 0.62	0.36 / 0.75	0.48 / 1.00
STGAT [27]	0.70 / 1.35	0.37 / 0.67	0.59 / 1.23	0.35 / 0.69	0.31 / 0.64	0.47 / 0.92
RSBG [34]	0.80 / 1.53	0.33 / 0.64	0.59 / 1.25	0.40 / 0.86	0.30 / 0.65	0.48 / 0.99
Social-STGCNN [32]	0.64 / **1.11**	0.49 / 0.85	0.44 / 0.79	0.34 / 0.53	0.30 / 0.48	0.44 / 0.75
Social-PEC [35]	0.61 / **1.11**	0.31 / 0.52	0.47 / 0.82	0.43 / 0.77	0.35 / 0.60	0.43 / 0.76
SILA [36]	**0.56** / 1.23	0.27 / 0.63	0.55 / 1.25	0.29 / 0.63	0.32 / 0.72	0.39 / 0.89
STSF-Net (ours)	0.63 / 1.13	**0.24** / **0.43**	**0.28** / **0.52**	**0.23** / **0.45**	**0.21** /**0.41**	**0.32** / **0.59**

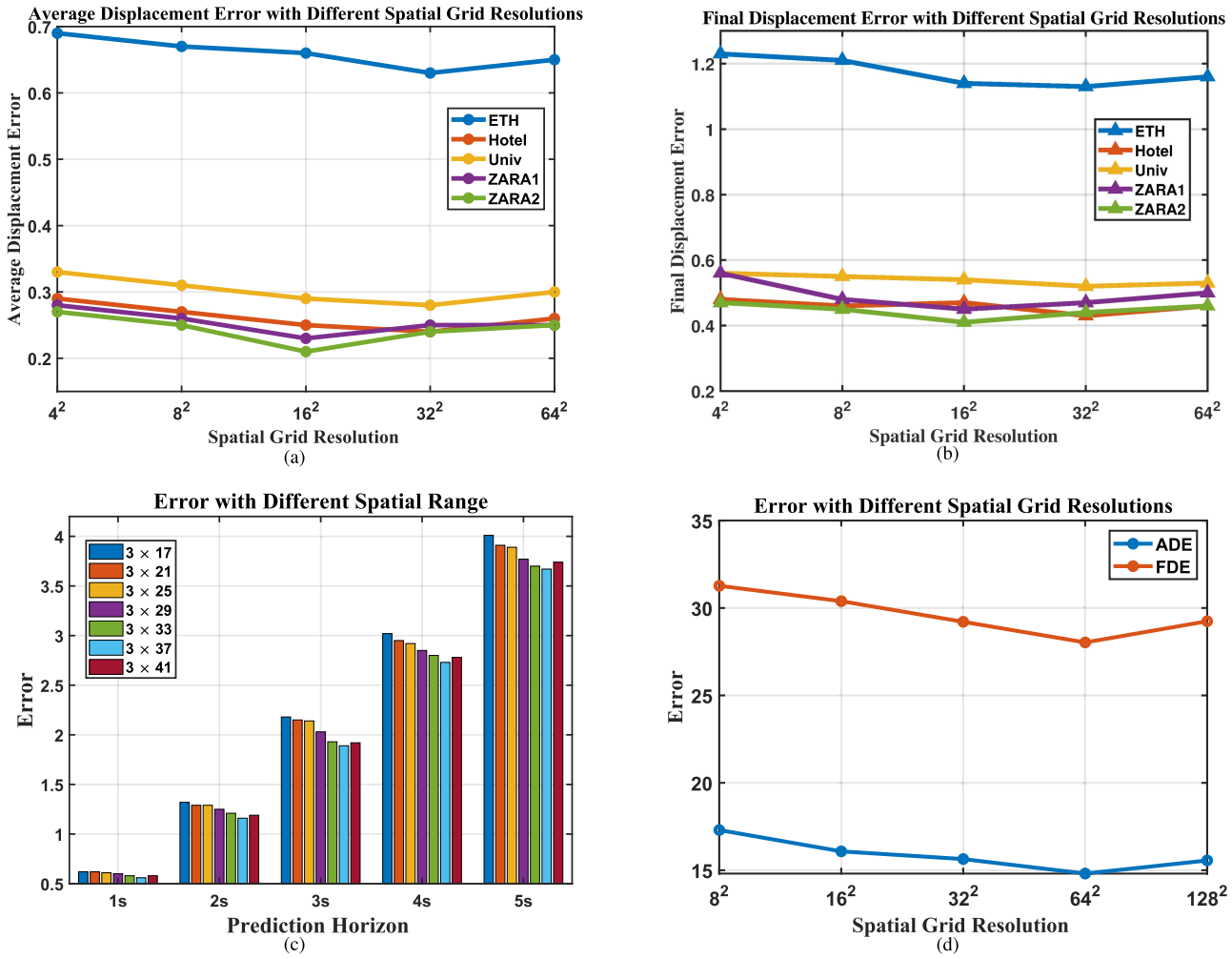

Fig. 3. With different resolutions of spatial tensor, quantitative results on (a) five scenarios of ETH, HOTEL, UNIV, ZARA1, and ZARA2 using ADE as the evaluation criteria; (b). five scenarios of ETH, HOTEL, UNIV, ZARA1, and ZARA2 using FDE as the evaluation criteria; (c). NGSIM US-101 and I-100 datasets using root squared mean error (d). Stanford Drone dataset using ADE and FDE.

A. Quantitative Evaluation

1) Pedestrian Trajectory Dataset: We first evaluate STSF-Net on ETH, UCY, and Stanford Drone datasets, which involve abundant social interactions among pedestrians. Specifically, ETH and UCY are composed of pedestrian trajectories in five scenes termed ETH, HOTEL, UNIV, ZARA1, and ZARA2, which are composed of more than 1500 pedestrians in crowded scenarios. All data are converted to coordinates by sampling at 0.4-second intervals. In experiments, we utilize the observation of 3.2 seconds to reason about trajectories for the next 4.8 seconds.

Evaluation Metric. For pedestrian trajectory datasets, we use the same evaluation rule as S-GAN for measuring the accuracy of predictions. Particularly, the Average Displacement Error (ADE) and the Final Displacement Error (FDE) metrics are employed for performance evaluations. The ADE calculates the average ℓ_2-distance between the predicted coordinates and the ground truth cross the prediction window:

$$ADE = \frac{\sum_{i=1}^{n}\sum_{t=t_{obs}+1}^{t_{obs}+t_{fut}}\sqrt{(x_i^t-\hat{x}_i^t)^2+(y_i^t-\hat{y}_i^t)^2}}{n*t_{fut}}, \quad (6)$$

and the FDE calculates the ℓ_2-distance between the predicted coordinates and the ground truth at the last frame:

$$FDE = \frac{\sum_{i=1}^{n}\sqrt{(x_i^t-\hat{x}_i^t)^2+(y_i^t-\hat{y}_i^t)^2}}{n}, \quad (7)$$

where $t = t_{obs} + t_{fut}$.

Baseline Methods. We compare STSF-Net on ETH and UCY datasets with some popular methods listed in Table I, some of these demonstrate state-of-the-art performance.

- **Linear**. A simple Kalman Filter is adopted for trajectory prediction.
- **Vanilla LSTM**. As a baseline, the vanilla Long Short-Term Memory that is qualified for sequence generation is adopted for trajectory prediction without any special mechanism for modeling social interactions.
- **S-Force**. An agent-based model, which formulates an explicit energy function that encodes social interactions and environmental factors, is developed to mine potential personal properties for effective behavior inference.
- **S-LSTM** [12]. It designed a social-pooling mechanism that generates a compact representation that facilitates information fusion from adjacent agents. Social-LSTM captures social interactions within a specified distance. Nevertheless, it fails to model global spatial structure and even only uses a simple pooling operation for feature aggregation.
- **S-GAN** [13]. It combined sequence generation tools and generative adversarial networks for multi-modal trajectory generation. Besides, S-GAN introduced a global pooling mechanism that encodes cues among agents.
- **Sophie** [18] incorporated a social attention mechanism and a physical attention mechanism to capture cues from both scene context and social interactions among agents;

TABLE II
QUANTITATIVE RESULTS ON NGSIM US-101 AND NGSIM I-80 DATASETS

Prediction horizon	1s	2s	3s	4s	5s
Linear	0.73	1.78	3.13	4.78	6.68
Vanilla LSTM	0.68	1.65	2.91	4.46	6.27
CV-GMM [17]	0.66	1.56	2.75	4.24	6.27
S-LSTM [12]	0.65	1.31	2.16	3.25	4.55
GAIL-GRU [22]	0.69	1.51	2.55	3.65	4.71
CS-LSTM [23]	0.61	1.27	2.09	3.10	4.37
S-GAN [13]	0.72	1.68	2.83	4.08	5.46
MATF-GAN [24]	0.66	1.34	2.08	2.97	4.13
M-LSTM [37]	0.58	1.26	2.12	3.24	4.66
ST-LSTM [38]	**0.56**	1.19	1.93	2.78	3.76
STSF-Net (ours)	**0.56**	**1.16**	**1.89**	**2.73**	**3.67**

TABLE III
QUANTITATIVE RESULTS ON STANFORD DRONE DATASETS

Method	ADE	FDE
Vanilla LSTM	37.35	77.13
Social Force [39]	36.38	58.14
S-LSTM [12]	31.19	56.97
S-GAN [13]	27.25	41.44
MATF-GAN [24]	22.59	33.53
Desire [21]	19.25	34.05
Sophie [18]	16.27	29.38
CGNS [40]	15.60	28.20
STSF-Net (ours)	**14.81**	**28.03**

- **PIF** [19] developed a multi-task framework to jointly predict future paths and activities by encoding human behavior and social interactions with neighbors;
- **MATF-GAN** [24]. It implemented a multi-agent tensor fusion system that jointly encodes social interactions and environment constraints. MATF-GAN combined the advantages of agent-centric and spatial-centric methods, but only focused on the last frame of the observation.
- **Social-PEC** [35] utilized a temporal CNN with a novel operation to extract social patterns;
- Recent work adopted graph-based methods because the topology of graphs is a natural way to describe social behaviors: Social-BiGAT [30], RSBG [34], Social-STGCNN [32], and SILA [36].

We compare STSF-Net against baselines on pedestrian trajectory datasets, *i.e.*, ETH and UCY. ADE and FDE metrics are exploited for performance evaluation. Quantitative results are reported in Table I. Our approach achieves a significant boost and comprehensively outperforms baselines on four scenarios of Hotel, UNIV, ZARA1, and ZARA2. The best baseline methods achieving the lowest ADE and FDE are SILA and Social-STGCNN, respectively. Compared with SILA, the performance is increased by 17.9% and 33.7% for ADE and FDE

TABLE IV
ABLATION STUDY USING ADE/FDE METRICS ON FIVE SCENARIOS FROM ETH AND UCY DATASETS

Method	STS*	STS	SRM	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
STSF-Net				1.09 / 2.41	0.86 / 1.91	0.61 / 1.31	0.41 / 0/88	0.52 / 1.11	0.70 / 1.52
STSF-Net	✓			0.80 / 1.29	0.46 / 0.92	0.59 / 1.08	0.39 / 0.71	0.28 / 0.49	0.50 / 0.90
STSF-Net	✓		✓	0.68 / 1.16	0.35 / 0.64	0.31 / 0.58	0.25 / 0.49	0.23 / 0.44	0.36 / 0.66
STSF-Net			✓	0.69 / 1.16	0.40 / 0.80	0.32 / 0.58	0.31 / 0.59	0.25 / 0.44	0.39 / 0.71
STSF-Net		✓		0.65 / 1.15	0.44 / 0.88	0.30 / 0.55	0.30 / 0.57	0.25 / 0.43	0.39 / 0.72
STSF-Net		✓	✓	**0.63 / 1.13**	**0.24 / 0.43**	**0.28 / 0.52**	**0.23 / 0.45**	**0.21 / 0.41**	**0.32 / 0.59**

TABLE V
ABLATION STUDY ON NGSIM US-101 AND NGSIM I-80 DATASETS

Method	STS*	STS	SRM	NGSIM				
				1s	2s	3s	4s	5s
STSF-Net				0.68	1.65	2.91	4.46	6.27
STSF-Net	✓			0.66	1.36	2.10	3.01	4.16
STSF-Net	✓		✓	0.59	1.26	1.98	2.83	3.81
STSF-Net			✓	0.61	1.33	2.10	2.99	4.04
STSF-Net		✓		0.60	1.24	1.96	2.78	3.76
STSF-Net		✓	✓	**0.56**	**1.16**	**1.89**	**2.73**	**3.67**

respectively. Compared with Social-STGCNN, ADE and FDE are also increased by 27.3% and 21.3%, respectively. SILA and Social-PEC derive better performance on the scene ETH. One possible reason is that ETH is more sparsely populated, so the graph structure with an attention mechanism employed by SILA and Social-PEC is more suitable for modeling relationships between agents. But our method models spatial structures through a grid-based manner, which is more suitable for crowded scenes and may cause overfitting to sparse settings. Besides, SILA utilizes two additional data (ZARA3 and uni examples from UCY) for training, which is a potential factor leading to unfair comparisons.

Also, we further analyze another important factor that affects the performance of the algorithm, *i.e.* spatial resolution of the grid due to rasterization operation in the process of constructing spatio-temporal sequence. Various resolutions are set in experiments for ADE and FDE as illustrated in Fig. 3(a) and (b). Due to under-fitting caused by low resolution and over-fitting caused by high resolution, the error first decreases and then increases. With the ADE criterion, we found that 32×32 is the ideal setting for ETH, Univ, and Hotel, and 16×16 is the suitable choice for ZARA1 and ZARA2. Likely, with the FDE criterion, 32×32 is the ideal setting for Univ and Hotel, and 16×16 is the suitable choice for ETH, ZARA1, and ZARA2.

2) Vehicle Trajectory Dataset: Vehicle trajectories from NGSIM US-101 and NGSIM I-80 datasets are further employed to evaluate our model. Specifically, NGSIM records real-world vehicle driving trajectories on the highway, each of which is sampled at 10 Hz from a 45-minute video. These vehicle trajectories also involve different types of traffic conditions and social interactions. We utilize trajectories of the observed 3 seconds to forecast the next 5 seconds.

TABLE VI
ABLATION STUDY ON STANFORD DRONE DATASETS

Method	STS*	STS	SRM	ADE	FDE
STSF-Net				37.35	77.13
STSF-Net	✓			18.46	34.71
STSF-Net	✓		✓	16.80	33.36
STSF-Net			✓	16.14	31.53
STSF-Net		✓		15.89	31.47
STSF-Net		✓	✓	**14.81**	**28.03**

To verify the effectiveness of STSF-Net in fast-moving patterns, we also perform evaluations on NGSIM. In addition to comparisons against benchmark methods such as vanilla LSTM, S-LSTM, S-GAN, and MATF-GAN, we also include comparisons with some approaches specifically designed for vehicle trajectory prediction with domain-specific knowledge, *e.g.*, CV-GMM [17], GAIL-GRU [22], CS-LSTM [23], M-LSTM [37], and ST-LSTM [38]. We evaluate the performance using root squared mean error in meters as [24] for the future 5 seconds: $rmse = \sqrt{\frac{1}{n}\sum_{i=1}^{n}((x_i^t - \hat{x}_i^t)^2 + (y_i^t - \hat{y}_i^t)^2)}$. Quantitative results shown in Table II indicate that our method improves the performance over other benchmarks. Note that GAIL-GRU accesses the ground truth of adjacent vehicles when predicting a specific agent trajectory and CS-LSTM uses extra supervised signals from horizontal and vertical maneuver

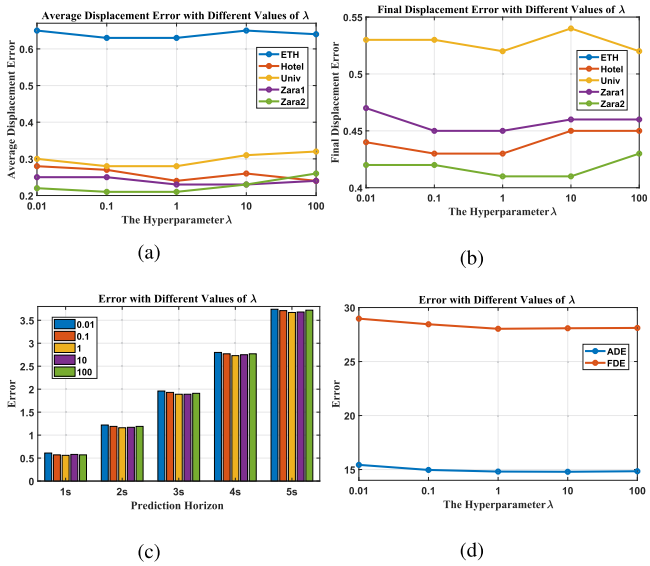

Fig. 4. A linear search is performed on the value of hyper-parameter λ to investigate its impact on model performance. (a) five scenarios of ETH, HOTEL, UNIV, ZARA1, and ZARA2 using ADE as the evaluation criteria; (b). five scenarios of ETH, HOTEL, UNIV, ZARA1, and ZARA2 using FDE as the evaluation criteria; (c). NGSIM US-101 and I-100 datasets using root squared mean error (d). Stanford Drone dataset using ADE and FDE.

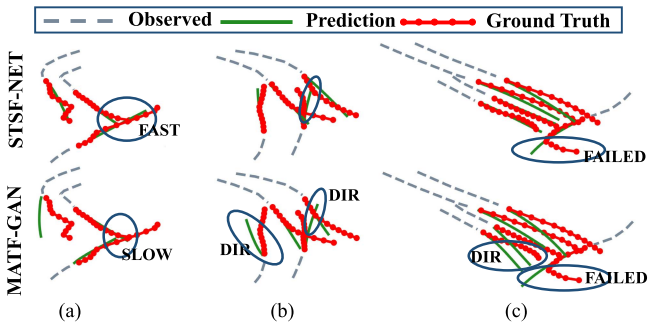

Fig. 5. Examples of predictions from our model and MATF-GAN on ETH and UCY. SLOW and FAST mean that agents change speed for collision avoidance. DIR stands for a change of direction and FAILED denotes the failed cases. The circle highlights some details.

classes, while our method does not. But our method is still better than theirs.

Besides, we also investigate the impacts of spatial range on predictions. Since NGSIM US-101 and NGSIM I-80 record trajectories of vehicles on the highway with designed lanes, current vehicles are generally only affected by the vehicles on adjacent lanes. Hence, we only consider two adjacent lanes in the horizontal direction followed [23], which are discretized into 3 cells. We study the effect of different resolutions on vertical direction, and results are presented in Fig. 3(c). It illustrates that 3×37 is the ideal choice.

3) Heterogeneous Agent Dataset: For universality, we evaluate our approach on a more challenging scenario with Stanford Drone Dataset. This dataset consists of more complex scenarios, where a variety of types of agents, including pedestrians, cars, and bicyclists will appear at the same time. We follow

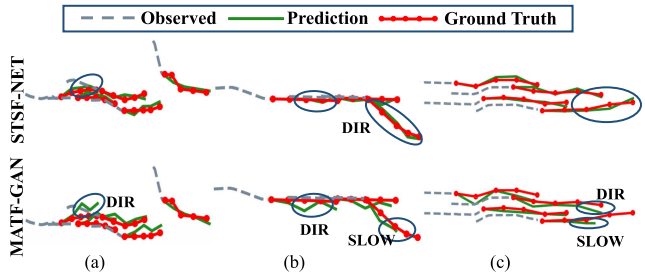

Fig. 6. Examples of predictions from our model and MATF-GAN on NGSIM US-101 and NGSIM I-80.

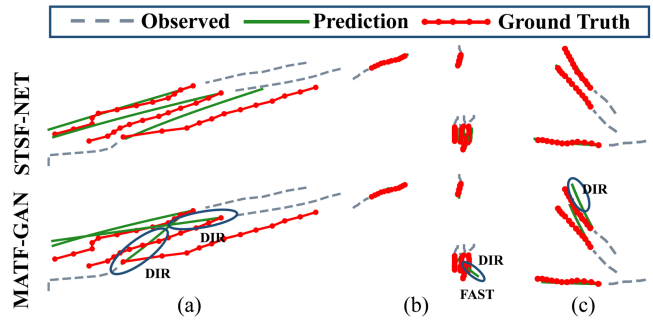

Fig. 7. Examples of predictions from our model and MATF-GAN on Stanford Drone Dataset.

prior work [12], [13], [18], [21], [24], [39] and use the same dataset splits and evaluation protocols (*i.e.* ADE and FDE). Table III demonstrates that STSF-NET is superior to all baselines for both ADE and FDE. Such results also demonstrate the model's ability to handle scenarios involving heterogeneous agents with different speeds and movement characteristics. Note that MATF-GAN, as an important baseline for our comparison, is far worse than our algorithm, and we will further analyze this in Section V-B.

In addition, we also set different spatial resolutions on Stanford Drone Dataset. The illustration in Fig. 3(d) shows a similar phenomenon with ETH and UCY, and 64×64 is found to be the ideal choice for both ADE and FDE.

B. Ablation Study

The major contributions of this work are to socially capture temporal dependencies across multiple trajectories and temporal correlations between interactions by introducing a spatio-temporal sequence (STS) and a social recurrent mechanism (SRM), respectively. To verify their benefits, we respectively removed STS and SRM from the complete STSF-Net and observed their performance on ETH, UCY, NGSIM US-101, NGSIM I-80, and Stanford Drone Datasets. We follow the same metrics as described in Section V-A for evaluation. As shown in Tables IV, V, and VI, the complete model gives the best results on both pedestrian and vehicle trajectories. Note that STS extends the single-frame tensor fusion mechanism in MATF-GAN

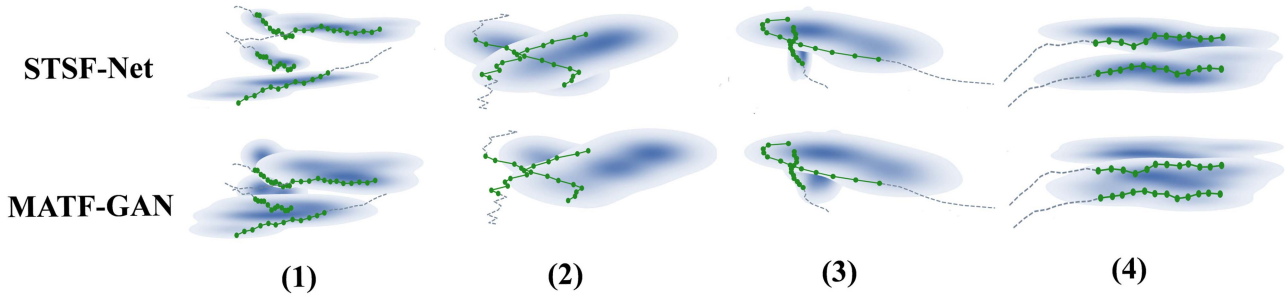

Fig. 8. Qualitative analysis about the generated trajectory distributions of STSF-Net and MATF-GAN on ETH and UCY datasets. For each scenario, the blue area stands for the generated trajectory distribution, the grey dotted line denotes observation, and the green solid line is the ground truth.

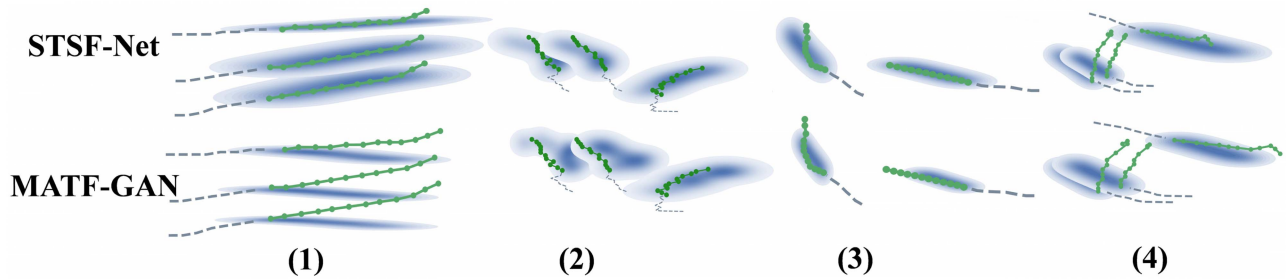

Fig. 9. Qualitative analysis about the generated trajectory distributions of STSF-Net and MATF-GAN on NGSIM US-101 and NGSIM I-100 datasets. For each scenario, the blue area stands for the generated trajectory distribution, the grey dotted line denotes observation, and the green solid line is the ground truth. (4) is a failed case.

to spatio-temporal sequences. In this way, STS can capture temporal dependencies across multiple trajectories while preserving the spatial distribution of agents throughout the motion, so that the performance of STS alone surpasses baselines in ADE and FDE. Since MATF-GAN performs pre-training on other datasets, while our model is directly trained from scratch. To this end, in order to reveal the advantages of introducing spatio-temporal structure over single-frame tensor fusion in MATF-GAN, we degenerate the STS by only considering the location structure at time-step t_{obs}, termed as STS*, which is equivalent to MATF-GAN without pre-training on other datasets. For both pedestrian and vehicle datasets, we observe that the performance of using STS is better than that of using STS*, and the combination of STS and SRM is also superior to the combination of STS* and SRM. Besides, the combination of STS* and SRM also wins over the use of STS* alone. Although STS and SRM have the same performance, their combination achieves a greater performance improvement. Note that this is not due to the effect of introducing more parameters, because SRM only performs element-wise addition for features at each time-step, without introducing more parameters.

In addition, a linear search strategy is performed on the value of hyper-parameter λ to investigate its impact on model performance. We set λ from 0.01 to 100 in steps of 10 times to observe changes in model performance for all datasets. The details are shown in Fig. 4. The observation demonstrates that setting λ to 1 is a reasonable and desirable choice for all datasets. In fact, we find that our model is not sensitive to the value of hyper-parameter λ on all datasets, which also proves the robustness of our model.

C. Qualitative Evaluation

Social trajectory prediction is a complex task, which depends on the ability to capture the spatio-temporal properties of agents involved in the scene. Agents are affected by the social interactions of others in the environment, showing different motion patterns, including forming groups, changing speed, adjusting directions for collision avoidance, etc. We show some scenes from ETH and UCY in Fig. 5, and from NGSIM US-101 and NGSIM I-80 in Fig. 6. Compared with MATF-GAN, our method can better capture social behaviors and take appropriate actions in time to avoid collisions. In Fig. 5, (a) When two pedestrians merge, SFSF-NET accurately captures interactions and takes appropriate acceleration and deceleration operations, while MATF misunderstands such interactions and produces the opposite operations; (b) When multiple pedestrians from different directions meet, our model generates reasonable inferences, while trajectories predicted by MATF-GAN produces unnecessary steering operations; (c) A failure example shows that when someone is crossing a crowd, the sudden large-angle turning behavior cannot be captured. In Fig. 6, (a) Motion behaviors of vehicles are correctly captured by our model in a relatively congested scene, while MATF-GAN attempts to avoid underlying collisions by changing movement directions; (b) Compared to MATF-GAN, behaviors of different vehicles at the fork is

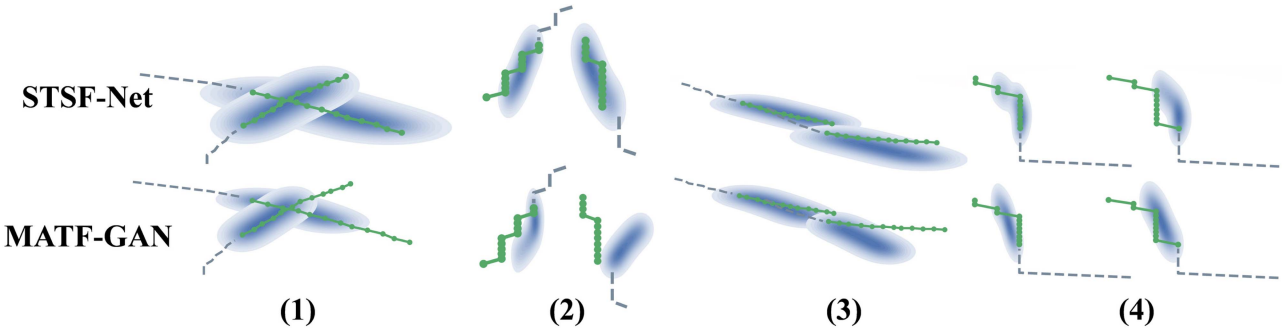

Fig. 10. Qualitative analysis about the generated trajectory distributions of STSF-Net and MATF-GAN on Stanford Drone dataset. For each scenario, the blue area stands for the generated trajectory distribution, the grey dotted line denotes observation, and the green solid line is the ground truth. (4) is a failed case.

reasonably inferred by our model; (c) A scene with multiple vehicles in parallel, our prediction is closer to the ground truth. For Stanford Drone datasets, as illustrated in Fig. 7, (a) Parallel agent trajectories are effectively inferred with STSF-NET, while the untimely steering is taken by MATF-GAN, resulting in unacceptable inferences; (b) Trajectories of slowly moving objects in crowded scenes are reasonably predicted with STSF-NET, instead of the blind steering and acceleration operations in MATF-GAN; (c) Compare with MATF-GAN, the trajectory is more accurately predicted by STSF-NET when moving in the form of a fork.

To evaluate the generated multi-modal distribution, we visualized results for ETH-UCY, NGSIM, and Stanford Drone datasets respectively in Figs. 8, 9, and 10 with kernel density estimation. The darker colors in the illustration, the greater the predicted probability. Specifically, STSF-NET generates convincing distributions that cover ground truth with a high probability than MATF-GAN for all examples in illustrations. Note that the case (4) in Fig. 9 and the case (4) in Fig. 10 provide failed cases. The model fails to reason about trajectories because agents turn sharply in Fig. 9 (4) and continuously changing direction in Fig. 10 (4). Nevertheless, we find that generated samples in Fig. 10 (4) with STSF-NET capture the trend of agent movement, while MATF-GAN does not.

VI. Conclusion

In this paper, some novel insights on trajectory prediction are presented. To address some concerns about modeling spatio-temporal properties of agents involved in the scene, we introduce a method of constructing a spatio-temporal trajectory sequence to model temporal dependencies across multiple trajectories, as well as a social recurrent mechanism to capture temporal correlations between interactions. The proposed approach is evaluated on pedestrian and vehicle trajectory datasets. Quantitative results and qualitative analysis show the superiority of our model.

References

[1] H. Luo *et al.*, "Real-time dense monocular SLAM with online adapted depth prediction network," *IEEE Trans. Multimedia*, vol. 21, no. 2, pp. 470–483, Feb. 2019.

[2] P.-J. Duh, Y.-C. Sung, L.-Y. F. Chiang, Y.-J. Chang, and K.-W. Chen, "V-Eye: A vision-based navigation system for the visually impaired," *IEEE Trans. Multimedia*, vol. 23, pp. 1567–1580, 2021, doi: 10.1109/TMM.2020.3001500.

[3] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 6015–6022.

[4] Y. Liu *et al.*, "Context-aware three-dimensional mean-shift with occlusion handling for robust object tracking in RGB-D videos," *IEEE Trans. Multimedia*, vol. 21, no. 3, pp. 664–677, Mar. 2019.

[5] S. Li and D.-Y. Yeung, "Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4140–4146.

[6] X. Dong and J. Shen, "Triplet loss in Siamese network for object tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 459–474.

[7] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. S. Torr, "Fast online object tracking and segmentation: A unifying approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1328–1338.

[8] S. Zhang *et al.*, "Person re-identification in aerial imagery," *IEEE Trans. Multimedia*, vol. 23, pp. 281–291, 2021, doi: 10.1109/TMM.2020.2977528.

[9] S. Gong *et al.*, "Faster person re-identification," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 275–292.

[10] L. Zheng, Y. Huang, H. Lu, and Y. Yang, "Pose-invariant embedding for deep person re-identification," *IEEE Trans. Image Process.*, vol. 28, no. 9, pp. 4500–4509, Sep. 2019.

[11] G. Wang *et al.*, "High-order information matters: Learning relation and topology for occluded person re-identification," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6449–6458.

[12] A. Alahi *et al.*, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 961–971.

[13] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2255–2264.

[14] W. Choi and S. Savarese, "Understanding collective activities of people from videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1242–1257, Jun. 2014.

[15] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg, "Who are you with and where are you going?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1345–1352.

[16] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert, "Activity forecasting," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 201–214.

[17] N. Deo, A. Rangesh, and M. M. Trivedi, "How would surround vehicles move? A unified framework for maneuver classification and motion prediction," *IEEE Trans. Intell. Veh.*, vol. 3, no. 2, pp. 129–140, Jun. 2018.

[18] A. Sadeghian *et al.*, "SoPhie: An attentive GAN for predicting paths compliant to social and physical constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1349–1358.

[19] 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 Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5725–5734.

[20] I. Hasan *et al.*, "Forecasting people trajectories and head poses by jointly reasoning on tracklets and vislets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 4, pp. 1267–1278, Apr. 2021.

[21] N. Lee *et al.*, "DESIRE: Distant future prediction in dynamic scenes with interacting agents," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 336–345.
[22] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," in *Proc. IEEE Intell. Veh. Symp.*, 2017, pp. 204–211.
[23] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 1468–1476.
[24] T. Zhao *et al.*, "Multi-agent tensor fusion for contextual trajectory prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12126–12134.
[25] Y. C. Tang and R. Salakhutdinov, "Multiple futures prediction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 15424–15434.
[26] M. Bansal, A. Krizhevsky, and A. Ogale, "ChauffeurNet: Learning to drive by imitating the best and synthesizing the worst," in *Proc. Robot.: Sci. Syst.*, 2019.
[27] Y. Huang, H. Bi, Z. Li, T. Mao, and Z. Wang, "STGAT: Modeling spatial-temporal interactions for human trajectory prediction," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 6272–6281.
[28] W. Choi and S. Savarese, "A unified framework for multi-target tracking and collective activity recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 215–230.
[29] J. Amirian, J.-B. Hayet, and J. Pettré, "Social ways: Learning multi-modal distributions of pedestrian trajectories with GANs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 2964–2972.
[30] V. Kosaraju *et al.*, "Social-BiGAT: Multimodal trajectory forecasting using bicycle-GAN and graph attention networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 137–146.
[31] P. Zhang, W. Ouyang, P. Zhang, J. Xue, and N. Zheng, "SR-LSTM: State refinement for LSTM towards pedestrian trajectory prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12085–12094.
[32] 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 Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 14424–14432.
[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.
[34] J. Sun, Q. Jiang, and C. Lu, "Recursive social behavior graph for trajectory prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 660–669.
[35] D. Zhao and J. Oh, "Noticing motion patterns: Temporal CNN with a novel convolution operator for human trajectory prediction," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 628–634, Apr. 2021.
[36] G. Habibi, N. Jaipuria, and J. P. How, "SILA: An incremental learning approach for pedestrian trajectory prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 1024–1025.
[37] N. Deo and M. M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based LSTMs," in *Proc. IEEE Intell. Veh. Symp.*, 2018, pp. 1179–1184.
[38] S. Dai, L. Li, and Z. Li, "Modeling vehicle interactions via modified LSTM models for trajectory prediction," *IEEE Access*, vol. 7, pp. 38287–38296, 2019, doi: 10.1109/ACCESS.2019.2907000.
[39] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, no. 5, pp. 4282–4286, 1995.
[40] J. Li, F. Yang, M. Tomizuka, and C. Choi, "EvolveGraph: Multi-agent trajectory prediction with dynamic relational reasoning," in *Proc. Neural Inf. Process. Syst.*, 2020, pp. 19783–19794.
[41] S. Pellegrini, A. Ess, and L. Gool, "Improving data association by joint modeling of pedestrian trajectories and groupings," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 452–465.
[42] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," in *Computer Graphics Forum*. Hoboken, NJ, USA: Wiley, 2007, pp. 655–664, vol. 26, no. 3.
[43] R. Alexandre, S. Amir, A. Alexandre, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 549–565.
[44] J. Colyar and J. Halkias, "U.S. highway 101 dataset," Federal Highway Administration (FHWA), Tech. Rep. FHWA-HRT-07-030, 2007.
[45] J. Colyar and J. Halkias, "U.S. highway I-80 dataset," Federal Highway Administration (FHWA), Tech. Rep. FHWA-HRT-07-030, 2007.