2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

SparseGTN: Human Trajectory Forecasting with Sparsely Represented Scene and Incomplete Trajectories

Jianbang Liu*[1], Guangyang Li*[2], Xinyu Mao[1], Fei Meng[1], Jie Mei[2] and Max Q.-H. Meng†[3], Fellow, IEEE

***Abstract*— In recent years, great progress has been made in forecasting human motion in crowded scenes. However, current methods are far from practical applications due to the unbearable high computation costs, especially for encoding scene context. In addition, neglecting the partially detected trajectories makes the predicted outcome deviate from the real trajectory distribution. To handle the aforementioned concerns, we propose to represent the scene context and partially observed trajectories with sparse graphs. Customized for this special data structure, we design a hierarchical Graph Transformer Network model SparseGTN to predict multiple possible future trajectories of the target pedestrian by digesting the sparsely represented inputs. Our approach exhibits superiority over the state-of-the-art (SOTA) methods, utilizing a mere** 3.42% **of the number of floating point operations (FLOPs) and** 0.53% **of the number of model parameters. The code will be available online*.**

I. INTRODUCTION

It is crucial for mobile robots to predict the possible future trajectories of other dynamic agents, especially in crowded dynamic scenes. It has been demonstrated that the graph convolutional network (GCN) exhibits excellent performance in modeling the spatial-temporal interaction between pedestrians in such scenes since [1], [2]. Driven by GCN's outstanding capability in capturing interaction-related factors, many researchers represent trajectories as graphs and encode the motion dynamics with GCN-based modules for trajectory forecasting in crowded dynamic scenes [3]–[10].

Nevertheless, we need to point out two concerns which have been neglected in previous work. Firstly, the modeling of environmental influence is either missing or computationally demanding in recent trajectory prediction methods for crowded dynamic scenes. Most GCN-based methods put little effort into capturing the influence of scene context. It is difficult to model the environmental influence with GCN because GCN can only handle graph data, and the scene context usually has a dense image representation, such as a bird's-eye-view (BEV) image, a BEV semantic map or a high-definition (HD) map. On the other side, simply ignoring scene context may make predicted trajectories violate the physical law, e.g., pass through static obstacles, which is dangerous. Others (GCN-based [2] and non-GCN-based [11]–[15]) employ convolutional neural networks (CNN) to encode the densely represented scene context and investigate the correlation between the scene context and agent dynamics by a CNN-based module or attention mechanism. These methods are usually computationally expensive and infeasible for mobile robots with limited computational resources. Hence, how to efficiently encode scene context in sparse representation remains unsolved. Secondly, most methods assume each observed trajectory is complete, requiring all the pedestrians to be tracked at every time step during the observation. But, in practical scenarios, some pedestrians can be partially detected in practical cases due to sudden appearance or camera occlusion. This issue was first investigated in [8] and rarely addressed by existing methods. Ignoring this realistic aspect of real-world data makes these SOTA methods less applicable and their performance less convincing.

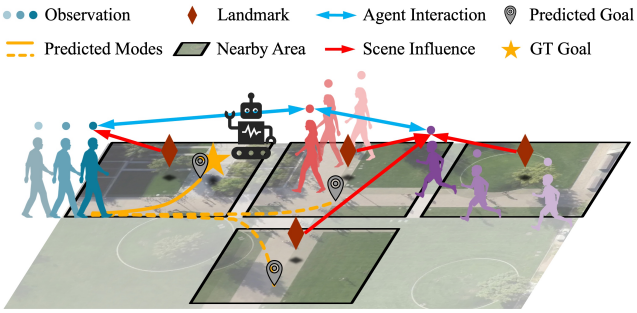

Fig. 1: Illustration of how to model the influence of scene context with sparse graph and predict the future trajectory of the target agent. Locations sampled within the scene are treated as landmarks. Each landmark covers the nearby area around it

In [16], Gao et al. prove that dense representation is redundant, while sparse representation is sufficiently informative for modeling the interaction with scene context. Inspired by their work, we propose our scene graph construction (SGC) to transform the scene context from a dense representation (image) to a sparse one (graph). Not every piece of the map affects the motion of pedestrians to the same extent. People tend to avoid obstacles (trees, walls, etc.) and walk

*Equal contribution. †The corresponding author.
[1]Jianbang Liu, Fei Meng and Xinyu Mao are with the Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong. ({henryliu, maoxinyu, feimeng}@link.cuhk.edu.hk)
[2]Guangyang Li and Jie Mei are with the Department of Mechanical Engineering and Automation, Harbin Institute of Technology, Shenzhen. (ligy0802@gmail.com and jmei@hit.edu.cn)
[3]Max Q.-H. Meng is with Shenzhen Key Laboratory of Robotics Perception and Intelligence and the Department of Electronic and Electrical Engineering at Southern University of Science and Technology in Shenzhen, China. He is a Professor Emeritus in the Department of Electronic Engineering at The Chinese University of Hong Kong in Hong Kong and was a Professor in the Department of Electrical and Computer Engineering at the University of Alberta in Canada. (max.meng@ieee.org.)
*Code available at https://github.com/Henry1iu/SparseGCN.

within free space by instinct. The SGC samples locations on the scene map and treats them as landmarks. The dense representation of scene context can be approximated by a sparse graph, in which landmarks are regarded as nodes and connected to nearby nodes by edges. To address the partial detection issue, we build frames of graphs with agent nodes and dummy nodes denoting the detected and unseen pedestrians, respectively. The pedestrian motion dynamics feature is extracted by encoding the sparse trajectory graphs spatially and temporally.

In this work, we build a hierarchical Graph Transformer Network model (called SparseGTN) composed of Transformer-GCN blocks to incorporate the sparse trajectory and scene context inputs. The details of our proposed model will be discussed in IV, and the experiments on public datasets show our approach can exceed the performance of recent GCN-based methods by a large margin and achieve comparable results to other SOTA methods, which digest densely represented input, with reduced model size and computational complexity. Overall, our contributions are threefold:

- We introduce a novel sparse scene graph representation for describing the scene context and demonstrate how to incorporate this sparse representation along with agent dynamics to model the environmental influence.
- We propose our hierarchical GCN-based model to predict the future of pedestrians with the sparse scene graph and the partially detected trajectories.
- We evaluate the proposed model on public datasets, and our model reaches a new SOTA with a smaller model size and reduced computation complexity.

II. Related Work

A. Trajectory prediction with Graph Convolutional Network

It is insufficient to express the relationship between pedestrians or the interaction between pedestrians and the surroundings with the fixed structure data (e.g., image) [17]. Convolutional neural networks (CNN) are limited to processing data with fixed structures such as images or text, where graphs can effectively represent non-Euclidean structures [18]. Conversely, GCN excels at expressing relationships between nodes through connections known as edges, making them highly suitable for addressing vehicles' and pedestrians' trajectory prediction problems involving interactive relations [4], [19].

GCN is first introduced to predict socially acceptable trajectories by Huang et al. [1]. They employ Graph Attention (GAT) scheme to capture the spatial correlation between pedestrians involved in the scene. Shi et al. [7] propose a sparse graph convolution network (SGCN) for predicting pedestrian trajectories by utilizing sparse directed spatial interactions. The AVGCN model proposed by Liu et al. [6] incorporates an attention mechanism to estimate the attention weights of nearby pedestrians related to a target pedestrian and integrates them with GCN-based prediction networks for future trajectory forecasting. The STAR, Spatio-Temporal graph transformer framework, proposed by Yu et al. [4], utilizes TGConv, a novel graph convolution mechanism based on the transformer model, to simulate crowd interaction within the graph. Bae and Jeon design a graph-based trajectory estimation and refinement network in [10], which constructs a multi-relational pedestrian graph and aggregates the spatial embedding using GCN.

However, in most works, the pedestrian trajectory is assumed to be complete. The incomplete trajectory issue has received limited attention in the existing literature. It is a common practice to filter out incomplete trajectories during data preparation [10]. Consequently, available samples are reduced, and an increased disparity exists between the learned and actual trajectory distribution. We utilize the mask to indicate whether there is a missing pedestrian position at each timestamp rather than completely discarding it, thereby enhancing the accuracy of our prediction.

B. Modeling the influence of Scene Context

Two decades ago, the Social Force model [20] proposed to consider the impact of the environment on pedestrians, which is abstracted as F_{env}. The F_{env} is determined by considering the pedestrian position as the reference point and taking into account the obstacles within its visual range. Drawing inspiration from this concept, researchers have proposed numerous specialized designs to effectively capture the impact of scene context on pedestrian motion.

Constrained by the densely represented scene context, several studies [11], [12], [14], [21] suggest encoding the scene context using a CNN, while the surrounding CNN feature of the pedestrian position will be extracted to serve as the environmental clue. However, the scene context that counts to pedestrians is not a simple surrounding local map. Social-BiGAT [2] employs a VGG backbone for scene encoding and applies soft attention between the agent's motion embedding and the entire scene embedding to extract the environmental clue. MATF-GAN [13] concatenates the agents' encoding with the scene encoding according to their last observed position on the map. Y-Net [15] proposed to represent the trajectory with heatmaps and forecast the possible goal, waypoints, and future trajectory distribution with a U-Net architecture.

An autonomous driving scene can be transformed from an HD map to polylines and vectors for representing the scene context more compactly, as proposed by researchers from Waymo in VectorNet [16]. VectorNet demonstrates superior performance over CNN-based methods [22] in terms of evaluation metrics and computation efficiency. The following studies built upon the sparse representation in VectorNet and propose novel approaches for predicting future trajectories in the context of autonomous driving [23]–[31]. These approaches leverage GCN-based or attention-based modules, continuously pushing the boundaries of trajectory prediction methods to new heights.

Although the sparse representation achieves greater success and becomes a dominant trend in trajectory prediction for autonomous driving, it is still absent in pedestrian trajectory prediction for crowded dynamic scenes. For autonomous

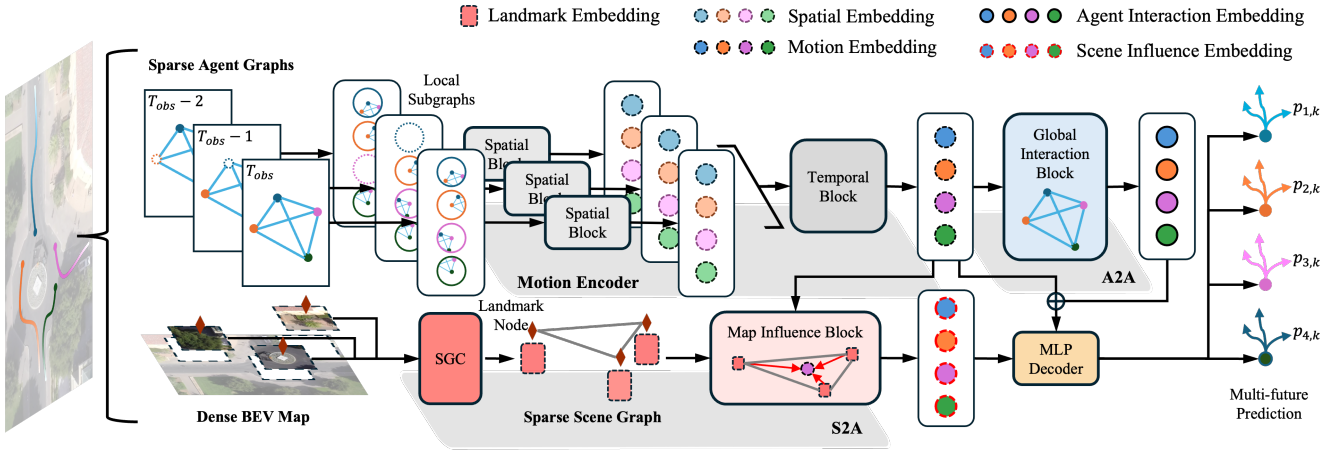

Fig. 2: The proposed architecture of SparseGTN. The sparse agent graphs are constructed from the given historical states $\mathscr{X}$. The sparse scene graph is constructed using our proposed scene graph construction. It takes the sparse agent graphs and the sparse scene graph and forecasts the multi-modal future of the target.

driving on structured roads, describing the scene with vectors and polylines is straightforward, as many line-following behaviours exist, but this is not the case for crowded dynamic scenes. In contrast to common practices, we adopt a novel approach of partitioning the map into discrete blocks and treating each block as a distinct node in a graph structure. This enables us to comprehensively represent the spatial relationship between pedestrians and their environment rather than just concerning local information.

III. Problem Formulation

Given the historical states $\mathscr{X} = \{\mathbf{S}_i^{1:T_{obs}} | i = 1,..,N\}$ of N pedestrians $a_i \in \{a_i | i = 1,..,N\}$ who appear in the scene within the observation horizon T_{obs}, the objective of our predictive approach is to forecast K future trajectories $\mathscr{Y} = \{\mathbf{S}_i^{T_{obs}+1:T_{obs}+T_{pred}} | i = i,...,N\}$ for each target pedestrian within the prediction horizon T_{pred} and the corresponding probability $p_{i,k} \in \{p_{i,k} | i = 1,...,N; k = 1,...,K\}$ for these K modes. Each pedestrian's state $\mathbf{s}_i^t \in \mathbf{S}_i$ can include its geometric attributes in the world coordinate at each timestamp $t \in \{1,...,T_{obs}\}$, such as position $x_i^t \in \mathbb{R}^2$, speed $v_i^t \in \mathbb{R}^2$, and heading angle θ_i^t. Additionally, the pedestrian may be partially detected during the observation period, which leads to the states missing at certain timestamps. The scene's static structure, represented by a BEV semantic map $\mathscr{M}$, is assumed to be available. By leveraging a known homography transformation H, positions on the BEV map can be associated with positions in the world coordinate.

IV. Methods

The structure of the proposed model is illustrated in Fig. 2. The hierarchical structure consists of local encoding, interaction modeling, and a prediction header. The history trajectories are transformed into a sequence of agent graphs and encoded spatially and temporally. The local motion features are fed into two Transformer-GCN blocks, which model the agent-to-agent interaction and map-to-agent influence. The map-to-agent influence block takes the sparse scene graph as an additional input and aggregates scene-related information to the agent node. With the local motion, agent interaction and map influence features, a prediction header simultaneously predicts multi-modal futures for all pedestrians.

In the following subsections, we first present how the inputs are formulated. Then, we will describe how motion information is aggregated locally and how interaction information is propagated globally.

A. Representing Inputs

In order to characterize the influence of scene semantic information using GCN, we propose to transform the scene map from a dense image to a sparse graph. The sparse scene graph is an undirected graph $\mathbf{M} = (\mathbf{V}_m, \mathbf{E}_m)$. As shown in Fig. 3, The set of nodes corresponds to the landmarks sampled on the semantic map. The landmark attributes include the position in world coordinates and the semantic image patch centered at the node. Each landmark node is connected to its nearest neighbour(s). The set of edges corresponds to the relative position from one landmark node to its connected node to preserve the spatial relationship. Our sparse graph preserves effective details of the scene context and reduces redundancy. As a result of reducing redundancy, the computation cost of encoding the scene context is also decreased.

Considering the partial detection issue, the sparse historical states of detected pedestrians are represented by a set of directed graphs similar to [8], which are called *agent graphs* $\mathscr{A} = \{\mathbf{A}^t = (\mathbf{V}^t, \mathbf{E}^t, \mathbf{I}^t) | t = 1,...,T_{obs}\}$ in this work. The set of nodes $\mathbf{V}^t = \{\mathbf{s}_i^t | i = 1,...,N\}$ corresponds to the states of pedestrians. An edge $\mathbf{e_{ij}} \in \mathbf{E}^t = \{e_{i,j}^t | i,j = 1,...,N; i \neq j\}$ depicts the spatial relationship between the target node i and its neighbouring node j using the relative translation vector. The node mask $I_i^t \in \mathbf{I^t}$ indicates the availability of node i's state at the timestamp t. If the node state is not available, it will be represented by a dummy node with all its state attributes filled with 0. Initially, the agent graphs are

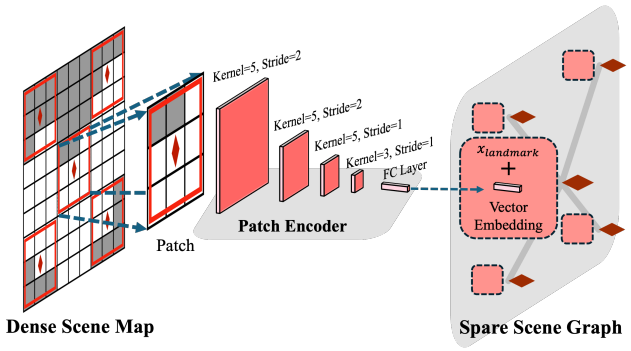

Fig. 3: Illustration of the proposed SGC. $x_{landmark}$ denotes the position of the sampled landmark in the same coordinate as the trajectory data. The node feature of each landmark includes both spatial and semantic attributes

constructed as complete directed graphs, in which every pair of distinct nodes is connected by a pair of unique edges.

B. Agent Graph Encoding

Intuitively, pedestrians tend to pay more attention to the individuals around them while walking, subconsciously avoiding collisions based on relative motion. Inspired by this behaviour preference, we deploy two Transformer-GCN blocks, spatial and temporal blocks, to learn the local interaction hidden in the historical states.

In the spatial GCN block, the target node i is taken as the center to select local neighbours by a distance threshold and availability $\mathbf{I^t}$. The geometric attributes of all selected neighbours are rotated according to the heading angle θ_i^t of the target node.

$$\mathbf{z}_i^t = \phi_{ctr}([\mathbf{R}_{\theta_i^t}^T x_i^t, \mathbf{R}_{\theta_i^t}^T v_i^t]), \tag{1}$$

$$\begin{aligned}\mathbf{z}_{j,i}^t =& \phi_{nbr}([\mathbf{R}_{\theta_i^t}^T x_j^t, \mathbf{R}_{\theta_i^t}^T v_j^t]) \\ &+ \phi_{rel}([\mathbf{R}_{\theta_i^t}^T e_{i,j}^t, cos(\triangle\theta_{j,i}), cos(\triangle\theta_{j,i})]),\end{aligned} \tag{2}$$

where ϕ_{ctr}, ϕ_{nbr}, ϕ_{rel} are Multi-Layer Perceptrons (MLPs), wei is the rotation parameterized by θ_i^t, and $\triangle\theta_{j,i} = \theta_j - \theta_i$. The state embedding $\mathbf{z}_i^t$ of node i is updated by aggregating the neighbours' embedding with its hidden embedding as follows:

$$\mathbf{q}_i^t = \mathbf{W}_Q^{spat}\mathbf{z}_i^t, \quad \mathbf{k}_{ij}^t = \mathbf{W}_K^{spat}\mathbf{z}_{j,i}^t, \quad \mathbf{v}_{ij}^t = \mathbf{W}_V^{spat}\mathbf{z}_{j,i}^t, \tag{3}$$

$$\alpha_i^t = softmax(\frac{\mathbf{q}_i^{tT}}{\sqrt{d_k}} \cdot \{\mathbf{k}_{ij}^t | j \in \mathcal{N}_i\}), \tag{4}$$

$$\mathbf{g}_i^t = sigmoid(\mathbf{W}^{gate}[\mathbf{z}_i^t, \sum_{j\in\mathcal{N}_i} \alpha_i^t \mathbf{v}_{ij}^t]) \tag{5}$$

$$\hat{\mathbf{z}_i^t} = \mathbf{g}_i^t \circ \mathbf{W}^{self}\mathbf{z}_i^t + (1 - \mathbf{g}_i^t) \circ \sum_{j\in\mathcal{N}_i} \alpha_i^t \mathbf{v}_{ij}^t \tag{6}$$

where $\mathbf{W}_Q^{spat}$, $\mathbf{W}_K^{spat}$, $\mathbf{W}_V^{spat}$, $\mathbf{W}^{gate}$, and $\mathbf{W}^{self}$ are learnable weights, d_k is the dimension for linear projection, $\mathcal{N}_i$ denotes the set of node i's neighbours, $\circ$ denotes element-wise product.

Algorithm 1: Landmark Sampling

Input : Semantic Map $\mathcal{M}$, Sample Step a, Path Size b, and Homography Matrix H
Output: Landmark Nodes $\mathbf{V}_m$

```
H, W, _ = shape(a);
i_s, j_s = a/2, a/2;
V_m = [ ];
while i_s < H do
    while j_s < W do
        M_patch = crop(M, (i_s, j_s), b);
        if isContainBorder(M_patch) then
            x_landmark = homoProj(H, (i_s, j_s));
            V_m.append([x_landmark, M_patch]);
        j_s += a;
    i_s += a;
return V_m;
```

The temporal block takes a similar structure as temporal encoding in [32]. The state embeddings of target node i at different time steps are stacked into a matrix $\mathbf{Z}_i$. $\mathbf{Z}_i$ is fed into the temporal attention module to obtain the motion embedding of target agent i:

$$\mathbf{Q}_i = \mathbf{Z}_i\mathbf{W}_Q^{temp}, \ \mathbf{K}_i = \mathbf{Z}_i\mathbf{W}_K^{temp}, \ \mathbf{V}_i = \mathbf{Z}_i\mathbf{W}_V^{temp}, \tag{7}$$

$$\hat{\mathbf{Z}}_i = softmax(\frac{\mathbf{Q}_i{\mathbf{K}_i}^T}{\sqrt{d_k}} + I^{temp})\mathbf{V}_i \tag{8}$$

$$I_{i_1,i_2}^{temp} = \begin{cases} -\infty & \text{if } i_1 < i_2; \\ 0 & \text{otherwise;} \end{cases} \tag{9}$$

where $\mathbf{W}_Q^{temp}$, $\mathbf{W}_K^{temp}$, $\mathbf{W}_V^{temp}$ are learnable weights, and I_{i_1,i_2}^{temp} denotes the temporal availability.

C. Interaction Modeling

Scene Influence Modeling (S2A): We build a heterogeneous graph to model the influence of static structure on human motion. This graph combines the agent nodes presented at the prediction time step and the sparse scene graph. All the edges between landmark nodes are preserved, while edges connecting from landmark nodes to their nearby agent nodes are added. An MLP and a CNN-based block encode the landmark node's geometric (position) and semantic (patched map image) attributes separately. The CNN block contains 4 CNN layers with kernel sizes of 5, 5, 5, 3 and strides of 2, 2, 1, 1 respectively. The output feature maps are flattened and then transformed by a linear layer into embeddings that have the same dimensions as the motion embedding. Another MLP layer merges the geometric and semantic embedding to obtain the landmark embedding. By fusing the embeddings of landmark and agent nodes in a similar way as the agent graph encoding in Section IV-B, we simulate how semantic information affects the motion of agents and embed the map influence.

Global Agent Interaction Modeling (A2A): We model the future interaction of agents with the agent graph at the

TABLE I: Quantitative Performance of our SparseGTN with other SOTA methods on ETH/UCY dataset. The performance of each methods is minADE / minFDE in meters with $K = 20$. The **bold** text denotes best results. The "−" denotes not reported.

Model	Social-STGCNN	Trajectron++	PECNet	Y-Net	AgentFormer	SGCN	STT	Graph-TERN	SparseGTN
Year	2020	2020	2020	2021	2021	2021	2022	2023	-
ETH	0.64 / 1.11	0.61 / 1.03	0.65 / 1.13	**0.28** / **0.33**	0.45 / 0.75	0.63 / 1.03	0.54 / 1.10	0.42 / 0.58	0.37 / 0.41
HOTEL	0.49 / 0.85	0.20 / 0.28	0.22 / 0.38	**0.10** / 0.14	0.14 / 0.22	0.32 / 0.55	0.24 / 0.46	0.14 / 0.23	0.11 / **0.13**
UNIV	0.44 / 0.79	0.30 / 0.55	0.35 / 0.57	0.24 / 0.41	0.25 / 0.45	0.37 / 0.70	0.57 / 1.15	0.26 / 0.45	**0.22** / **0.35**
ZARA1	0.34 / 0.53	0.24 / 0.41	0.25 / 0.45	0.17 / 0.27	0.18 / 0.30	0.29 / 0.53	0.45 / 0.94	0.21 / 0.37	**0.15** / **0.26**
ZARA2	0.30 / 0.48	0.18 / 0.32	0.18 / 0.31	0.13 / 0.22	0.14 / 0.24	0.25 / 0.45	0.36 / 0.77	0.17 / 0.38	**0.11** / **0.17**
AVG	0.44 / 0.75	0.31 / 0.52	0.33 / 0.57	0.18 / 0.27	0.23 / 0.39	0.37 / 0.65	0.43 / 0.88	0.24 / 0.38	0.19 / **0.26**

TABLE II: Quantitative Performance of our SparseGTN with other SOTA methods on SDD dataset. The performance of each methods is minADE / minFDE in meters. The **bold** text denotes best results. The "−" denotes not reported.

Model	S-GAN	DESIRE	TNT	Trajectorn++	PECNet	Y-Net	LB-EBM	Graph-TERN	SparseGTN
$K = 5$	-	19.25/34.05	12.23/21.16	-	12.79/29.58	11.49/**20.23**	13.58/26.57	12.35/23.32	**11.43**/20.34
$K = 20$	27.23/41.44	-	-	11.40/20.12	9.96/15.88	7.85/11.85	9.03/15.97	8.43/14.26	**6.70/11.39**

prediction time step. The edges of the fully connected graph are all preserved to capture the information flows between agents. We use another Transformer-GCN block described previously to fuse the pairwise embeddings and compute the global agent interaction embedding.

D. Prediction Header and Loss

In this work, we implement a simple MLP prediction header. It takes the local embedding, the map influence embedding, and the global interaction embedding of the target agent i to predict the K distribution sequences of relative displacement $\{\hat{\mu}_{i,k}^t = \triangle\hat{x}_{i,k}^t = \hat{x}_{i,k}^t - \hat{x}_{i,k}^{t-1}, \hat{\sigma}_{i,k}^t | t = T_{obs} + 1, ..., T_{obs} + T_{pred},\ k = 1, ..., K\}$ and corresponding probability $\hat{\mathbf{p}}_i = \{\hat{p}_{i,k} | k = 1, .., K\}$.

A multi-task training objective is adopted to optimize the proposed model:

$$\mathscr{L} = \mathscr{L}_{reg} + \mathscr{L}_{cls}, \tag{10}$$

where $\mathscr{L}_{reg}$ is the best-of-K loss [23], [33], and $\mathscr{L}_{cls}$ is the classification loss. The best-of-K loss is formulated as:

$$\mathscr{L}_{reg} = -\frac{1}{NT_{pred}} \sum_{i=1}^{N} \sum_{T_{obs}+1}^{T_{obs}+T_{pred}} log\, \mathbb{L}(\triangle x_i^t | \hat{\mu}_{i,k^*}^t, \hat{\sigma}_{i,k^*}^t) \tag{11}$$

where $\mathbb{L}(\cdot|\cdot)$ denotes Laplace distribution, k^* denotes the best mode, which has the final position closest to the ground truth, $p_{i,k}$ equals to 1 if $k = k^*$ and 0 otherwise. The classification loss is formulated as:

$$\mathscr{L}_{cls} = CrossEntropy(\hat{\mathbf{p}}_i, \mathbf{p}_i), \tag{12}$$

where $\hat{\mathbf{p}}_i$ and $\mathbf{p}_i$ denotes the estimated and ground truth probability, respectively.

TABLE III: Comparison of floating point operations (FLOPs), number of parameters, and input representation between models. The data is collected using the official release of Y-Net and AgentFormer.

Model	Params (M)	GFLOPs	Input Representation
Y-Net	53.16	812.01	Traj. Heatmap
SparseGTN	0.24	27.77	Sparse Graph

V. Experiments

A. Experimental Setup

Datasets:

1) ETH/UCY Dataset: The ETH [34]/UCY [35] dataset group comprises of the datasets ETH, HOTEL, UNIV, ZARA1 and ZARA2. The first two datasets are sampled from the ETH dataset, while the last three are obtained from the UCY dataset. This comprehensive dataset encompasses four distinct scenes (ZARA1 and ZARA2 being identical), each featuring various obstacles or no passing areas. The pedestrians' positions within these scenes are represented by coordinates measured in meters in real-world space. The commonly used leaving-one-out strategy [36] [7] [10], known as bench-marking, is employed for evaluation. Specifically, training is conducted with trajectories in four scenes while testing is performed on the fifth scene.

2) Stanford Drone Dataset (SDD): The SDD [37] dataset comprises twenty scenes collected in proximity to universities. It adopts a top-down perspective to observe the trajectories of pedestrians, cyclists, and motor vehicles. The dataset encompasses diverse passable areas across different scenes, thereby encompassing various pedestrian trajectory modes encountered in real-world scenarios. The division of the training and evaluation datasets follows the same methodology as described in [36] [10].

Metrics: The evaluation method employed for pedestrian trajectory prediction adheres to the established standards in previous research [38] [39]. The accuracy of trajectory prediction is measured using Average Displacement Error

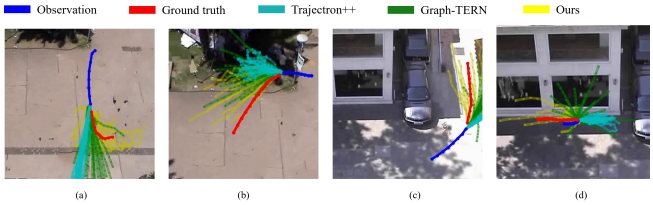

Fig. 4: Visualization of multi-modal predictions with 20 samples on UCY dataset. (a) When encountering a sharp turn, Trajecteron++ is unable to respond to it and Graph-TERN can only respond a little; (b)(c) Our method performs better than the others; (d) When a pedestrian suddenly turns back, Trajecteron++ predicts the wrong direction and Graph-TERN can not predict, while the predictive efficacy of our method still remains.

(ADE) and Final Displacement Error (FDE) metrics. ADE is the average L_2 distance between the ground truth and the prediction in the prediction horizon T_{pred}, FDE is the L_2 distance between the ground truth and the prediction at the final time point within the T_{pred}. Mathematically,

$$ADE = \frac{\sum_{t=T_{obs}+1}^{T_{obs}+T_{pred}} \|\hat{x}^t - x^t\|_2}{T_{pred}} \tag{13}$$

$$FDE = \left\|\hat{x}^{T_{obs}+T_{pred}} - x^{T_{obs}+T_{pred}}\right\|_2 \tag{14}$$

where $\hat{x}^t, x^t \in \mathbb{R}^2$ are the position of ground truth and the estimation at the time t. The best result among K modes is reported as prior works.

Implementation Details: The embedding dimension is 32 for both nodes and edges. The AdamW [40] optimizer is employed for the training, with an initial learning rate of 5×10^{-4}, a weight decay of 1×10^{-4}, and a dropout rate of 0.1. The semantic image, in which each pixel denotes its semantic label, is assumed to be accessible. For simplicity, the landmark nodes are sampled vertically and horizontally at the same distance on one semantic map in this work as shown in Algorithm 1. Meanwhile, applying algorithm [41] or other key point selection algorithms on the seismic map to extract landmarks is also feasible.

B. Quantitative Comparison

We compare our proposed SparseGTN with some SOTA methods: S-GAN [42], DESIRE [43], Social-STGCNN [3], Trajectron++ [21], PECNet [36], Y-Net [15], TNT [25], AgentFormer [14], SGCN [7], STT [44], LB-EBM [45], and Graph-TERN [10].

In Table I, our SparseGTN performs slightly better than the second-best method, Y-Net, on HOTEL, UNIV, ZARA1 and ZARA2. As shown in Table III, SparseGTN has a much smaller size (about 200 times smaller) and a lower computation complexity (about 30 times lower) compared with Y-Net. Interestingly, our method can significantly outperform methods utilizing only the observed trajectories in terms of both minADE and minFDE. The performance gaps between our method and Graph-TERN are large on ETH(∼ 12.7%), HOTEL(∼ 43.5%), UNIV(∼ 22.2%), ZARA1(∼ 33.3%), ZARA2(∼ 55.3%). Compared with methods digesting the map image, our method shows a better result on average with less computation complexity. The results on the SDD dataset are reported in Table II. Our method performs best, indicating our model can better capture the interaction between agents and the scene. It's worth noting that our model employs a simple MLP prediction header. The performance may be further enhanced by incorporating the way-point prediction proposed in [10], [15]

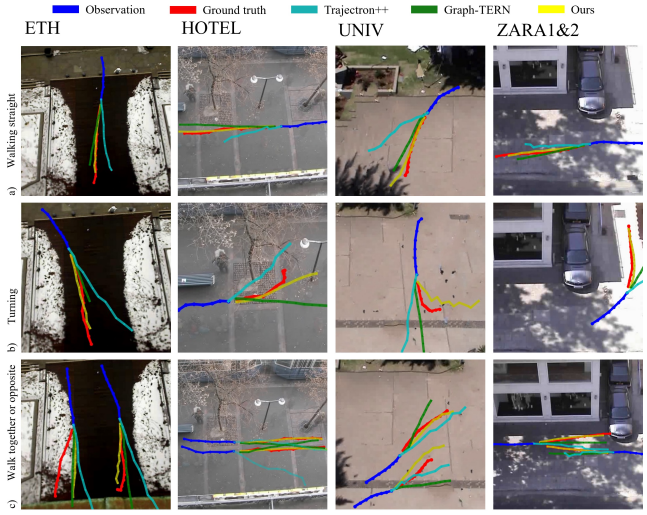

Fig. 5: Visualization of prediction results on ETH/UCY dataset. (a) When pedestrians go straight, our method is similar to Graph-TERN; (b) When turning, our method shows obvious superiority; (c) The performance of our method excels in extracting relations in pedestrians; on HOTEL dataset, the trajectory predicted by Graph-TERN crosses unpredictably in the future, indicating that the method does not correctly extract the relationship between the two pedestrians.

C. Qualitative Comparison

Fig. 6 illustrates the improvement of trajectory prediction results with map information. The most recent outstanding model has been selected as the reference for comparison. The GCN-based Graph-TERN [10] algorithm and Trajectron++ [21] based on CNN and RNN were selected as the two algorithms for visual comparison. Fig. 4 shows the difference in multi-modal predictions. Fig. 5 shows the results on ETH/UCY dataset, to aid visualization, trajectories with the best ADE on 20 samples are displayed.

D. Ablation Study

We conduct extensive ablation studies on the ETH/UCY dataset and SDD dataset.

Global Interaction Modeling: We alternately remove one of the interaction modeling blocks to demonstrate each module's contribution to the prediction performance. It's clearly shown in Fig. IV each interaction modeling block contributes to predicting an accurate result to a certain degree. The minADE drops 50% on average if the agent interaction

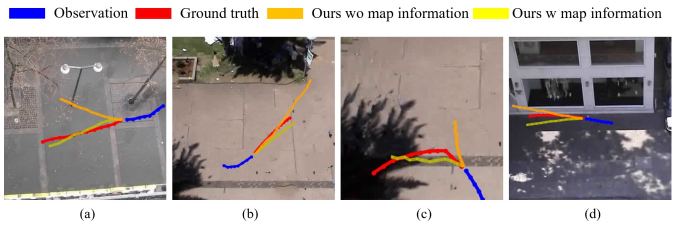

Fig. 6: After the fusion of map information, the accuracy of trajectory prediction is improved. (a)(c) After adding the map, the direction of the forecast will be more accurate;(b) Improvement in prediction distance; (d) The joining of the map will trigger a specific response to the surrounding areas that are impassable.

block is removed. This result indicates that encoding the motion individually or locally is not sufficient to predict a promising future for pedestrians in crowded scenes. It's necessary to model the interaction between agents explicitly. It's also observed that the improvement caused by appending the map influence modeling is not as notable as the agent interaction modeling. We think it's because the tested scenes are spacious and plain. They impose minor impacts on the pedestrians' motion as pedestrians would walk in a straight line unless they were not blocked by the scene or others.

Local Perception Radius: To investigate how the radius used to construct local subgraphs in spatial block affects the performance, we alter the radius of local encoding to examine the performance of SparseGTN. Table V shows the minFDEs reach a minimum at $r = 1.0$ and $r = 1.5$, which indicates including surrounding pedestrians in the local spatial encoding can facilitate forecasting the future. However, increasing the local radius degenerates the performance and introduces extra computations.

VI. CONCLUSIONS

We introduce representing the scene context and pedestrian dynamics with sparse graph representations to the context of pedestrian trajectory prediction in crowded dynamic scenes. We designed a novel GCN model where the first level locally encodes pedestrian dynamics, and the second level learns the higher-order interaction between agents and the scene context. Experiments on publicly available datasets show that, by compensating the missing scene context with the sparse scene graph, the proposed SparseGTN outperforms prior works both quantitatively and qualitatively. Our lightweight model can be effectively deployed on mobile robot platforms with limited computational resources, enabling accurate trajectory prediction in crowded scenarios. This may facilitate more stable collision avoidance for the robots. In the future, we plan to replace the naive MLP prediction header or employ the mixture-of-experts technique [46] to achieve better performance across different scenes.

TABLE IV: Ablation on the effectiveness of the agent interaction modeling(A2A) and the map influence modeling(M2A). During this comparison, the A2A and M2A are switched on and off to show the significance of each module. The percentage shows the reduction in minADE($K = 20$) compared with the base version.

A2A	M2A	ETH & UCY					
		ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
×	×	0.67	0.29	0.52	0.44	0.29	0.44
√	×	0.39 -41.8%	0.15 -48.3%	0.25 -51.9%	0.19 -56.8%	0.16 -44.8%	0.23 -47.7%
×	√	0.38 -43.3%	0.12 -58.6%	0.25 -51.9%	0.20 -54.5%	0.13 -55.2%	0.22 -50.0%
√	√	0.37 -44.8%	0.11 -62.1%	0.22 -57.7%	0.15 -65.9%	0.11 -62.1%	0.19 -56.8%

TABLE V: Ablation on the radius of local spatial encoding. The radius is adjusted for each split. The minFDE($K = 20$) is reported for comparison. For a better focus on the agent interaction, the map influence modeling is excluded during this comparison. "$r = 0.0$" means no neighbour agent is connected by an edge in local spatial encoding.

Local Radius (m)	ETH & UCY					
	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
$r = 0.0$	0.495	0.151	0.403	0.318	0.207	0.315
$r = 0.5$	0.491	0.154	0.406	**0.306**	0.202	0.312
$r = 1.0$	**0.486**	**0.144**	**0.386**	0.312	0.206	**0.307**
$r = 1.5$	0.492	0.146	0.404	0.314	**0.201**	0.311
$r = 2.0$	0.490	0.151	0.390	0.319	0.206	0.311
$r = 3.0$	0.491	0.153	0.396	0.313	0.209	0.312

REFERENCES

[1] Y. Huang, H. Bi, Z. Li, T. Mao, and Z. Wang, "Stgat: Modeling spatial-temporal interactions for human trajectory prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

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

[3] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 14 424–14 432.

[4] C. Yu, X. Ma, J. Ren, H. Zhao, and S. Yi, "Spatio-temporal graph transformer networks for pedestrian trajectory prediction," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*. Springer, 2020, pp. 507–523.

[5] C. Wang, S. Cai, and G. Tan, "Graphtcn: Spatio-temporal interaction modeling for human trajectory prediction," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2021, pp. 3450–3459.

[6] C. Liu, Y. Chen, M. Liu, and B. E. Shi, "Avgcn: Trajectory prediction using graph convolutional networks guided by human attention," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 14 234–14 240.

[7] L. Shi, L. Wang, C. Long, S. Zhou, M. Zhou, Z. Niu, and G. Hua, "Sgcn: Sparse graph convolution network for pedestrian trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8994–9003.

[8] Z. Huang, R. Li, K. Shin, and K. Driggs-Campbell, "Learning sparse interaction graphs of partially detected pedestrians for trajectory prediction," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1198–1205, 2021.

[9] Y. Xu, L. Wang, Y. Wang, and Y. Fu, "Adaptive trajectory prediction via transferable gnn," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6520–6531.

[10] I. Bae and H.-G. Jeon, "A set of control points conditioned pedestrian trajectory prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 5, 2023, pp. 6155–6165.

[11] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "Desire: Distant future prediction in dynamic scenes with interacting agents," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 336–345.

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

[13] T. Zhao, Y. Xu, M. Monfort, W. Choi, C. Baker, Y. Zhao, Y. Wang, and Y. N. Wu, "Multi-agent tensor fusion for contextual trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[14] Y. Yuan, X. Weng, Y. Ou, and K. Kitani, "Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9793–9803.

[15] K. Mangalam, Y. An, H. Girase, and J. Malik, "From goals, waypoints & paths to long term human trajectory forecasting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 15 233–15 242.

[16] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "Vectornet: Encoding hd maps and agent dynamics from vectorized representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 525–11 533.

[17] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen, "A survey on trajectory-prediction methods for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 652–674, 2022.

[18] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI open*, vol. 1, pp. 57–81, 2020.

[19] J. Liu, X. Mao, Y. Fang, D. Zhu, and M. Q.-H. Meng, "A survey on deep-learning approaches for vehicle trajectory prediction in autonomous driving," in *2021 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2021, pp. 978–985.

[20] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, pp. 4282–4286, May 1995.

[21] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*. Springer, 2020, pp. 683–700.

[22] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff, "Covernet: Multimodal behavior prediction using trajectory sets," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 14 062–14 071.

[23] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun, "Learning lane graph representations for motion forecasting," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 541–556.

[24] W. Zeng, M. Liang, R. Liao, and R. Urtasun, "Lanercnn: Distributed representations for graph-centric motion forecasting," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 532–539.

[25] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid, *et al.*, "Tnt: Target-driven trajectory prediction," in *Conference on Robot Learning*. PMLR, 2021, pp. 895–904.

[26] J. Gu, C. Sun, and H. Zhao, "Densetnt: End-to-end trajectory prediction from dense goal sets," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 15 303–15 312.

[27] L. Zhang, P. Li, J. Chen, and S. Shen, "Trajectory prediction with graph-based dual-scale context fusion," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 11 374–11 381.

[28] M. Wang, X. Zhu, C. Yu, W. Li, Y. Ma, R. Jin, X. Ren, D. Ren, M. Wang, and W. Yang, "Ganet: Goal area network for motion forecasting," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1609–1615.

[29] H. Chen, J. Wang, K. Shao, F. Liu, J. Hao, C. Guan, G. Chen, and P.-A. Heng, "Traj-mae: Masked autoencoders for trajectory prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 8351–8362.

[30] X. Mo, Y. Xing, H. Liu, and C. Lv, "Map-adaptive multimodal trajectory prediction using hierarchical graph neural networks," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3685–3692, 2023.

[31] X. Gao, X. Jia, Y. Li, and H. Xiong, "Dynamic scenario representation learning for motion forecasting with heterogeneous graph convolutional recurrent networks," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2946–2953, 2023.

[32] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[33] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2090–2096.

[34] S. Pellegrini, A. Ess, and L. Van Gool, "Improving data association by joint modeling of pedestrian trajectories and groupings," in *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I 11*. Springer, 2010, pp. 452–465.

[35] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," in *Computer graphics forum*, vol. 26, no. 3. Wiley Online Library, 2007, pp. 655–664.

[36] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik, and A. Gaidon, "It is not the journey but the destination: Endpoint conditioned trajectory prediction," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 759–776.

[37] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*. Springer, 2016, pp. 549–565.

[38] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[39] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2255–2264.

[40] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[41] B. Lau, C. Sprunk, and W. Burgard, "Efficient grid-based spatial representations for robot navigation in dynamic environments," *Robotics and Autonomous Systems*, vol. 61, no. 10, pp. 1116–1130, 2013.

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

[43] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. S. Torr, and M. Chandraker, "Desire: Distant future prediction in dynamic scenes with interacting agents," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2165–2174.

[44] A. Monti, A. Porrello, S. Calderara, P. Coscia, L. Ballan, and R. Cucchiara, "How many observations are enough? knowledge distillation for trajectory forecasting," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 6543–6552.

[45] B. Pang, T. Zhao, X. Xie, and Y. N. Wu, "Trajectory prediction with latent belief energy-based model," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 11 809–11 819.

[46] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen, "Gshard: Scaling giant models with conditional computation and automatic sharding," *arXiv preprint arXiv:2006.16668*, 2020.