

# Learning Sparse Interaction Graphs of Partially Detected Pedestrians for Trajectory Prediction

Zhe Huang<sup>1</sup>, Ruohua Li<sup>2</sup>, Kazuki Shin<sup>1</sup>, and Katherine Driggs-Campbell<sup>1</sup>

**Abstract**—Multi-pedestrian trajectory prediction is an indispensable element of autonomous systems that safely interact with crowds in unstructured environments. Many recent efforts in trajectory prediction algorithms have focused on understanding social norms behind pedestrian motions. Yet we observe these works usually hold two assumptions, which prevent them from being smoothly applied to robot applications: (1) positions of all pedestrians are consistently tracked, and (2) the target agent pays attention to all pedestrians in the scene. The first assumption leads to biased interaction modeling with incomplete pedestrian data. The second assumption introduces aggregation of redundant surrounding information, and the target agent may be affected by unimportant neighbors or present overly conservative motion. Thus, we propose Gumbel Social Transformer, in which an Edge Gumbel Selector samples a sparse interaction graph of partially detected pedestrians at each time step. A Node Transformer Encoder and a Masked LSTM encode pedestrian features with sampled sparse graphs to predict trajectories. We demonstrate that our model overcomes potential problems caused by the aforementioned assumptions, and our approach outperforms related works in trajectory prediction benchmarks. Code is available at <https://github.com/tedhuang96/gst>.

**Index Terms**—Human-Centered Robotics, Modeling and Simulating Humans.

## I. INTRODUCTION

**A**UTONOMOUS mobile robots must comprehensively understand dynamic human environments to safely and smoothly enter our daily lives [1], [2]. A human-centered robot should effectively encode motion patterns of surrounding pedestrians from observation, accurately predict their future trajectories, and efficiently plan its own paths for safe and rapid task execution [3], [4]. Significant progress has been made in understanding human-human interaction and predicting trajectories of multiple pedestrians [5]–[8], which inspired new contributions in crowd navigation [9], [10].

Despite the fruitful results in building socially aware architectures for multi-pedestrian trajectory prediction, previous works usually hold two assumptions which may burden their robotic applications. The first assumption is that positions

of all pedestrians are successfully tracked at all times. The second assumption is the target agent (pedestrian or robot) pays attention to all pedestrians in the public scene [7].

The first assumption defines *fully detected pedestrians* as people who are tracked at every time step during the considered observation and prediction period. This assumption implies that only the fully detected pedestrians are considered for modeling social interaction and predicting trajectories. In contrast, *partially detected pedestrians* comprise both fully detected pedestrians, and people whose positions are tracked for only a proportion of the considered period. Thus, pedestrians who enter the scene later than the beginning of the considered period and who exit earlier than the end are included in partially detected pedestrians. Partially detected pedestrians provide complete pedestrian data, while considering only the fully detected pedestrians results in 40.7% pedestrians ignored in benchmark datasets. The incomplete pedestrian data caused by the first assumption leads to biased modeling in social interactions. The second assumption requires *full connection* among all pedestrians in the scene, and causes pedestrians that are clearly non-influential to affect motion of the target agent. A workaround for the second assumption would be to restrict the target agent to pay attention to pedestrians in a pre-defined neighborhood and neglect the distant ones [5]. However, the joint influence from too many neighbors in close proximity would still potentially impair feature encoding of the target agent. With redundant concerns on the insignificant surrounding factors, excessively conservative agent behavior may be much like the notorious freezing robot problem [11].

We propose Gumbel Social Transformer (GST), which is composed of Edge Gumbel Selector, Node Transformer Encoder, and Masked LSTM. Each component is designed to be capable of processing features of partially detected pedestrians. As for the attention-to-all assumption, we formulate a directed interaction graph to represent the relationship of partially detected pedestrians at each time step. In the interaction graph, a node represents a pedestrian, and a directed edge represents a connection that the node at its tail pays attention to the node at its head. The graph is initialized with full connection which is equivalent as attention to all pedestrians. We apply the Edge Gumbel Selector to prune the edges by following an important constraint: *The target agent can pay attention to at most  $n$  pedestrians*. The hyperparameter  $n$  is to control the graph sparsity. With the most important relationships preserved between each agent and its  $n$  neighbors at each time step, the sparse interaction graphs inferred by the Edge Gumbel Selector are stacked in sequence. The sequence is then fed into the Node

Manuscript received: September 9, 2021; Revised November 25, 2021; Accepted December 14, 2021.

This paper was recommended for publication by Editor Jee-Hwan Ryu upon evaluation of the Associate Editor and Reviewers' comments. (*Corresponding author: Zhe Huang.*)

<sup>1</sup> Z. Huang, K. Shin, and K. Driggs-Campbell are with the Department of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: zheh4@illinois.edu; kazukis2@illinois.edu; krdc@illinois.edu).

<sup>2</sup> R. Li is with the Department of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor, MI 48109 USA (e-mail: ruohuali@umich.edu).

Digital Object Identifier (DOI): see top of this page.

Transformer Encoder and the Masked LSTM to spatially and temporally encode features of partially detected pedestrians, and recursively predict their trajectories.

Our contributions are fourfold: (1) We present a novel architecture to predict trajectories of partially detected pedestrians; (2) We introduce an Edge Gumbel Selector to sample dynamic and sparse interaction graphs of partially detected pedestrians; (3) We demonstrate in multi-agent simulation that our model mitigates the freezing robot problem and diminishes the disturbance from non-influential neighbors on the target agent; and (4) Our model outperforms state-of-the-art approaches on public human trajectory datasets.

## II. RELATED WORK

**Pedestrian Trajectory Prediction.** Early works have exhaustively investigated hand-engineered features of pedestrian motion [12], [13]. These works perform well in certain cases, but have non-negligible limitations like fixed motion patterns across all pedestrians [14]. Substantial contributions are made to resolve the problems by the integration of socially-aware structures and deep learning methods, including Social LSTM [5], Generative Adversarial Networks [7], Self-Attention Mechanism [6], [8], Graph Neural Networks [15], [16], and Transformer [17]. However, the assumptions of fully detected pedestrians and global attention over the scene are enforced in many previous works [6], [7], [15], [16]. For instance, the binary attention mask in Transformer-based Graph Convolution represents connection between pedestrians, and is set as a fully connected all-one square matrix with the dimension as the amount of fully detected pedestrians in the whole public scene [17]. Other works constrain the target agent to pay attention within a small neighborhood region, or do not clarify how motion prediction on fully detected pedestrians would be affected by considering partially detected pedestrians [5], [8]. In contrast to these works, our work infers a sparse interaction graph among pedestrians in an unsupervised way, and we explicitly study the influence of partially detected pedestrians on trajectory prediction.

**Graph Structure Learning.** Graph generation has a wide range of applications including causal discovery [18], neural architecture search [19], molecule design [20], and physical interaction inference [21]. Traditional approaches are typically hand-crafted for a specific family of graphs [22], whereas deep learning has recently been harnessed to learn graphs with suitable properties from observation data. One direction for graph generation is to perform sequential prediction on the next node or edge to be added to the graph [23]. If the number of graph nodes is fixed, another direction is to generate the adjacency matrix in one shot [24]. Dropout on the adjacency matrix (i.e., dropout on edges) is often used to alleviate overfitting and over-smoothing [25], [26]. Besides regularization, graph sparsity is emphasized in many domains where sparse graph representations are necessary to efficiently learn model parameters [19], [21]. Probability distribution of edges are usually assumed independent Bernoulli variables (existence of a single edge) [27] or independent categorical variables (type of a single edge) [21], [28]. In our work, we consider the

categorical distribution *over* edges which connect neighbor pedestrian nodes to the same target node. The inferred graph changes dynamically, which is consistent with the dynamic property of pedestrian interactions.

## III. METHOD

### A. Problem Formulation

Consider  $N$  partially detected pedestrians who appear in a scene during an observation period  $t \in \{1, \dots, T_{obs}\}$ . Their 2D positions are denoted by  $x_i^t$ ,  $i \in \{1, \dots, N\}$ . The task is to jointly predict their trajectories  $x_i^t$  during a following prediction period  $t \in \{T_{obs} + 1, \dots, T_{obs} + T_{pred}\}$ . These partially detected pedestrians enter the scene at or later than  $t=1$ . They leave the scene at or earlier than  $t=T_{obs} + T_{pred}$ .

We introduce *interaction graphs* to represent motion of partially detected pedestrians. A directed interaction graph  $G^t = (V^t, E^t, M^t, A^t)$  describes pedestrian motion at a time step  $t$ . The set of nodes  $V^t = \{v_i^t\}_{i=1:N}$  corresponds to pedestrian displacement (i.e., velocity). The set of edges  $E^t = \{e_{ij}^t\}_{i,j=1:N}$  corresponds to the relative position from a target pedestrian  $i$  to a neighbor  $j$ . The node masks  $M^t = \{m_i^t\}_{i=1:N}$  indicate whether the  $i$ th pedestrian's position is recorded at both  $t-1$  and  $t$ , and the binary-valued adjacency matrix  $A^t = \{a_{ij}^t\}_{i,j=1:N}$  specifies the validity of edges as in Equation 1, where the edge  $e_{ij}^t$  is nonexistent whenever either  $v_i^t$  or  $v_j^t$  is invalid. This setting guarantees the full connectivity of an initialized interaction graph  $G^t$  by removing the node of a pedestrian, who has not shown up yet or has left the scene, along with all relevant edges. The linear embedding layers and the masks for nodes and edges are applied to respective attributes to obtain high dimensional features, which are still denoted by  $v_i^t$  and  $e_{ij}^t$  as in Equation 1.

$$\begin{aligned} m_i^t &= \mathbb{1} \{x_i^{t-1} \text{ and } x_i^t \text{ are valid}\}, & a_{ij}^t &= m_i^t m_j^t, \\ v_i^t &= m_i^t \phi_v(x_i^t - x_i^{t-1}), & e_{ij}^t &= a_{ij}^t \phi_e(x_j^t - x_i^t). \end{aligned} \quad (1)$$

### B. Gumbel Social Transformer

The architecture of Gumbel Social Transformer is illustrated in Fig. 1. An Edge Gumbel Selector takes as input a combination of node and edge representations from interaction graphs, and samples a sparse interaction graph at each observation time step. A Node Transformer Encoder spatially aggregates node representations of the sampled sparse interaction graphs. The spatially encoded node features are sequentially fed into a Masked LSTM, from which hidden states are used to predict pedestrian positions at the next step. The recursion of feature embedding, edge sampling, node encoding, and node decoding is repeated until the end of the prediction period.

**Edge Gumbel Selector.** Though the initialized interaction graph  $G^t$  includes complete details of the pedestrian motion at time  $t$ , full connectivity could be redundant, and could even adversely affect the modeling of a target pedestrian  $i$ 's behavior. We impose the  $n$ -neighbor sparsity constraint on the interaction graph, which leads to a sparse interaction graph  $\tilde{G}^t = (\tilde{V}^t, \tilde{E}^t, \tilde{M}^t, \tilde{A}^t)$ . While  $\tilde{V}^t, \tilde{E}^t, \tilde{M}^t$  are identical to the counterparts in  $G^t$ , the weighted adjacency matrix

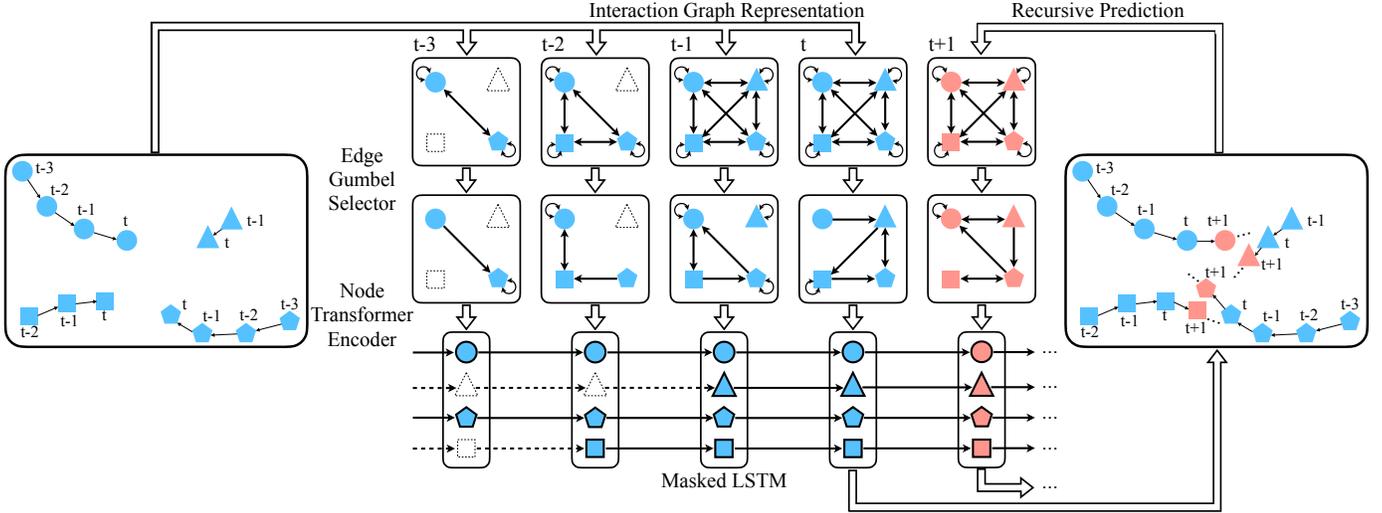


Fig. 1: Overview of Gumbel Social Transformer (GST). Observed trajectories (blue) are processed into interaction graph representations at each time step, which are fully connected except for pedestrians not observed at that moment. Under the  $n$ -neighbor sparsity constraint, Edge Gumbel Selector samples sparse interaction graphs, which are then encoded by Node Transformer Encoder and Masked LSTM. The encoded pedestrian features are used to recursively predict future trajectories (red).

$\tilde{A}^t = \{\tilde{a}_{ij}^t\}_{i,j=1:N} \in [0, 1]^{N \times N}$  becomes a sparse float-valued matrix to be inferred.

The inference of the adjacency matrix  $\tilde{A}^t$  is formulated as the problem to find the  $n$  neighbors who have the most influence on a target pedestrian  $i$  at time  $t$ . We first concatenate neighbor node features, target node features, and edge features to obtain augmented edge features  $\hat{e}_{ij}^t$ , which represents the pairwise interaction between target  $i$  and a neighbor  $j$ . A neighbor may draw the attention from the target that was originally paid to another neighbor. The relationship of these pairwise interactions  $\hat{e}_{ij}^t$  themselves is captured by a multi-head attention (MHA) at the edge level. The number of heads is set as  $n$ , where each head can be interpreted as one type of the interaction relationship:

$$\begin{aligned} \hat{e}_{ij}^t &= [v_j^t \| v_i^t \| e_{ij}^t], \\ \{\hat{e}_{ij}^{t,k}\}_{j=1:N}^{k=1:n} &= \text{MHA}\left(\{\hat{e}_{ij}^t\}_{j=1:N}, \text{mask} = \{a_{ij}^t\}_{j=1:N}\right). \end{aligned}$$

A multi-layer perceptron (MLP) maps the aggregated augmented edge features  $\hat{e}_{ij}^{t,k}$  to log probabilities  $\alpha_{ij}^{t,k}$  of a  $N$ -dimensional categorical distribution corresponding to the  $k$ th head. A reparameterization trick named Gumbel Softmax [29] is used to sample the most important neighbor to the  $i$ th target from these categorical distributions at each head while preserving differentiability. The samples are drawn from the concrete distribution approximation [30] as presented in Equation 2, where  $\mathbf{g} \in \mathbb{R}^N$  is a vector with elements sampled from independent and identically distributed random variables with Gumbel(0, 1) distribution.

$$\begin{aligned} \alpha_{ij}^{t,k} &= \text{MLP}\left(\hat{e}_{ij}^{t,k}\right), \\ \tilde{a}_{ij}^{t,k} &= \text{softmax}_j\left(\left(\alpha_{ij}^{t,k} + \mathbf{g}\right) / \tau\right), \\ \tilde{a}_{ij}^t &= \frac{1}{n} \sum_k \tilde{a}_{ij}^{t,k}. \end{aligned} \quad (2)$$

The temperature hyperparameter  $\tau$  in Equation 2 is annealed to near zero during training, so the approximate samples

gradually converge to one-hot samples from the categorical distributions. The entries of the sampled weighted adjacency matrix  $\tilde{a}_{ij}^t$ 's are the mean of generated samples across the heads. Note the sampling process assures that the set of invalid edges in  $\tilde{E}^t$  is a subset of the set of the removed edges in  $\tilde{A}^t$ .

**Node Transformer Encoder.** Given the sparse interaction graph  $\tilde{G}^t$ , the node features are spatially aggregated using an encoder inspired by Transformer-based Graph Convolution (TGConv) [17]. The Node Transformer Encoder in our case takes the weighted adjacency matrix  $\tilde{A}^t$  as the attention mask in TGConv, which is a sparse float matrix, and the inputs of source and target of Transformer are the representations of partially detected pedestrians  $\tilde{V}^t$ .

$$\{\hat{v}_i^t\}_{i=1:N} = \text{TGConv}\left(\text{target} = \tilde{V}^t, \text{source} = \tilde{V}^t, \text{mask} = \tilde{A}^t\right)$$

**Masked LSTM.** Pedestrian motion during the observation period is sliced into a stack of interaction graphs  $\{G^t\}_{t=1:T_{obs}}$ , which are processed through the Edge Gumbel Selector and the Node Transformer Encoder to obtain spatially encoded node features  $\{\hat{v}_i^t\}_{i=1:N}^{t=1:T_{obs}}$ . The node features are sequentially fed into a Masked LSTM to propagate hidden features  $h_i^t$ , which are used to predict pedestrian positions through a linear layer. The node masks  $m_i^t$ 's are set as one through the prediction period for all pedestrians except for who disappear before or at  $T_{obs}$ , as we assume they will never come back into the scene. The recurrence is introduced by generating the interaction graph at the next step with the predicted positions.

$$\begin{aligned} h_i^{t+1} &= (1 - m_i^t) h_i^t + m_i^t \text{LSTM}(\hat{v}_i^t, h_i^t), \\ \hat{x}_i^{t+1} &= \hat{x}_i^t + \phi_h(h_i^{t+1}). \end{aligned} \quad (3)$$

#### IV. EXPERIMENTS

We use two publicly available trajectory datasets for benchmarking: ETH [31] and UCY [32]. ETH is composed of two scenes ETH and HOTEL, and UCY is composed of three scenes UNIV, ZARA1, and ZARA2. The frame rate is 2.5 frames per second across the scenes. The task is to predict

TABLE I: Quantitative performance of all approaches on benchmark datasets. Three metrics Average Offset Error (AOE, unit:  $m$ ), Final Offset Error (FOE, unit:  $m$ ), and Negative Log Likelihood (NLL, no unit) on fully detected pedestrians are reported. GST (D) is Gumbel Social Transformer with a deterministic function  $\phi_h$  in Equation 3, while GST (P) is with a probabilistic function  $\phi_h$  that outputs Gaussian displacements. GST would be referred to as GST (D) if no additional clarification is given in this work. N/A presented in NLL is due to completely deterministic outputs, on which kernel density estimation cannot be performed.

Metric	Method	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
AOE ↓	SLSTM (D)	2.45±0.00	0.81±0.00	1.24±0.00	2.48±0.00	1.07±0.00	1.61±0.00
	STGCN (P)	2.93±0.59	1.07±0.21	0.76±0.05	0.95±0.23	0.87±0.14	1.32±0.25
	STGCN (D)	2.67±0.00	0.74±0.00	0.64±0.00	0.68±0.00	0.59±0.00	1.06±0.00
	SGAN (P)	1.78±0.55	0.32±0.08	0.62±0.03	0.59±0.11	0.43±0.10	0.75±0.17
	STGAT (P)	1.51±0.68	0.26±0.09	0.57±0.08	0.52±0.19	0.45±0.16	0.66±0.24
	Trajectron++ (P)	1.40±0.56	0.27±0.17	<b>0.50±0.23</b>	0.50±0.23	0.33±0.16	0.60±0.27
	GST (P)	1.10±0.14	0.22±0.02	0.64±0.04	0.59±0.03	0.46±0.03	0.60±0.05
	GST (D)	<b>0.96±0.20</b>	<b>0.21±0.02</b>	<b>0.50±0.01</b>	<b>0.40±0.00</b>	<b>0.32±0.02</b>	<b>0.48±0.05</b>
FOE ↓	SLSTM (D)	4.20±0.00	1.46±0.00	2.20±0.00	4.49±0.00	1.93±0.00	2.86±0.00
	STGCN (P)	4.98±0.89	1.54±0.29	1.39±0.09	1.48±0.30	1.31±0.20	2.14±0.35
	STGCN (D)	4.83±0.00	1.23±0.00	1.26±0.00	1.28±0.00	1.06±0.00	1.93±0.00
	SGAN (P)	3.60±1.33	0.60±0.19	1.31±0.06	1.28±0.26	0.94±0.21	1.55±0.41
	STGAT (P)	3.01±1.41	0.48±0.21	1.23±0.18	1.15±0.46	1.02±0.42	1.38±0.54
	Trajectron++ (P)	2.96±1.27	0.53±0.37	1.27±0.61	1.14±0.58	0.80±0.43	1.34±0.65
	GST (P)	2.22±0.33	<b>0.37±0.04</b>	1.32±0.09	1.15±0.06	0.95±0.07	1.20±0.12
	GST (D)	<b>2.09±0.47</b>	0.38±0.04	<b>1.08±0.02</b>	<b>0.86±0.00</b>	<b>0.70±0.04</b>	<b>1.02±0.11</b>
NLL ↓	SLSTM (D)	N/A	N/A	N/A	N/A	N/A	N/A
	STGCN (P)	10.07	3.57	3.06	4.20	2.13	4.61
	STGCN (D)	N/A	N/A	N/A	N/A	N/A	N/A
	SGAN (P)	8.11	11.99	13.56	9.20	2.61	9.09
	STGAT (P)	3.82	1.84	4.68	2.15	-0.16	2.47
	Trajectron++ (P)	1.76	-1.29	-0.75	-0.41	-1.83	-0.50
	GST (P)	<b>-0.44</b>	<b>-1.79</b>	<b>-1.98</b>	<b>-2.08</b>	<b>-3.61</b>	<b>-1.98</b>
	GST (D)	8.02	9.29	16.00	N/A	9.10	N/A

the trajectories in the next 4.8 seconds ( $T_{pred}=12$ ) given the positions tracked in the last 3.2 seconds ( $T_{obs}=8$ ). There are in average 40.7% of all detected pedestrians who are partially but not fully detected among these datasets. The trajectory samples in each scene are split into the training set (80%) and the test set (20%), and models are independently trained for each scene. Additionally, we conduct comparative study on our models with various configurations. A multi-agent simulation is performed with models trained in the comparative study, to evaluate their capabilities to generate trustworthy future trajectories in nontrivial social interaction scenarios.

#### A. Implementation Details

The embedding dimension of nodes is 32 and of edges is 64. The dimension of hidden states in LSTM is 32. The temperature  $\tau$  of the Edge Gumbel Selector is annealed linearly from 0.5 to 0.03 through training. We empirically find a higher initial temperature is likely to result in numerical instability during training. The Node Transformer Encoder has 3 Transformer Encoder layers with 8 attention heads, and a feed-forward dimension of 128. The Adam optimizer [33] is used to minimize the mean square error loss of prediction on trajectories of partially detected pedestrians, with an initial learning rate of 0.001. The model is trained for 200 epochs. Trajectory samples are randomly rotated during the training process for data augmentation. A *ghost agent* with zero-valued features is added in sparse configurations for promoting sparsity, which is inspired by the ghost link in [28].

#### B. Benchmark Evaluation

**Baselines.** Our model is compared against these existing methods: (1) Social LSTM (SLSTM) is a LSTM integrated

with a social pooling layer that outputs deterministic trajectories [5]; (2) Social STGCN (STGCN) is a spatial graph convolution network concatenated with a temporal convolution network that generates probabilistic outputs [16]; (3) The variant of STGCN which generates deterministic prediction; (4) Social GAN (SGAN) has a LSTM-based encoder-decoder architecture with a socially aware global pooling layer, and is trained using Generative Adversarial Networks for multi-modal trajectory prediction [7]; (5) Spatial-Temporal GAT (STGAT) applies graph attention networks to model crowd interaction, and uses different LSTMs for temporal encoding of single pedestrians and of spatially encoded interaction [15]; (6) Trajectron++ is a conditional variational autoencoder which encodes agent interaction through attention mechanism and semantic map through convolutional neural networks [34].

**Metrics.** Offset Error is defined as the distance between the target pedestrian’s ground truth position and the position predicted by the model at one time step [5], [35]. Average Offset Error (AOE) is the average of Offset Errors throughout the prediction period, and Final Offset Error (FOE) is the Offset Error at the last prediction step  $T_{obs} + T_{pred}$ . For evaluation of probabilistic methods, 20 trajectories are predicted, over which the mean and the standard deviation of AOE and FOE are reported. To evaluate trajectory sample distribution, we calculate Negative Log Likelihood (NLL) of the ground truth trajectory under the distribution from kernel density estimation [34].

**Results.** The quantitative results are presented in Table I. Note that our model GST is trained with partially detected pedestrians, while other baselines have to be trained with only fully detected pedestrians. Nevertheless, all metrics are

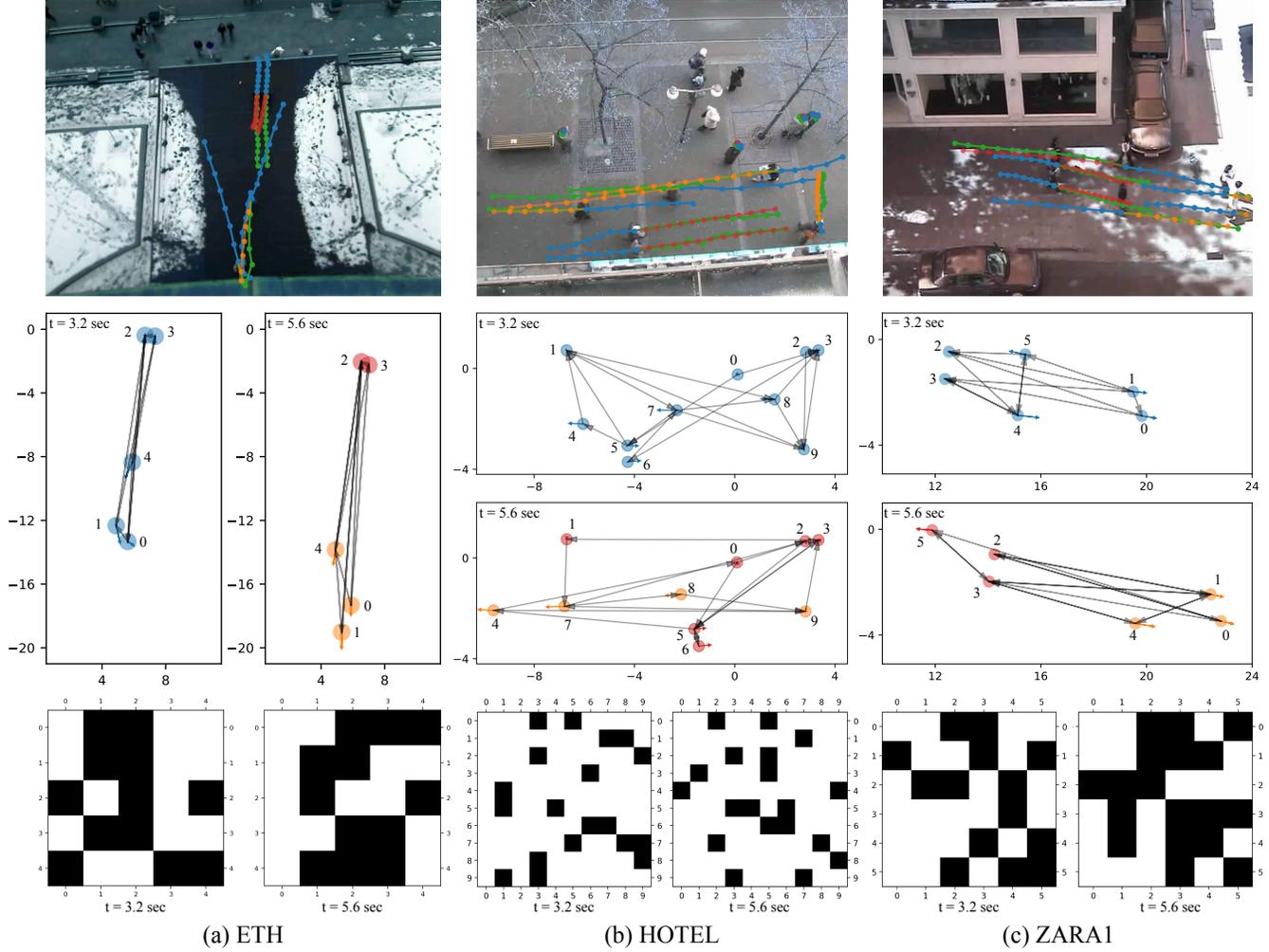


Fig. 2: Top row visualizes trajectory prediction on benchmark datasets. Blue denotes observation, green denotes ground truth, red denotes prediction on fully detected pedestrians, and orange denotes prediction on the other partially detected pedestrians. Middle row demonstrates the sparse interaction graph inference at the last observed time step, and at the middle of the prediction period. Colored arrows indicate pedestrian velocities. Gray arrows represent directed edges of the interaction graph, where the target pedestrian node at the tail pays attention to the neighbor node at the head. Bottom row shows the adjacency matrices corresponding to the inferred graphs in the middle row. The black entry at the  $i$ th row, the  $j$ th column indicates the  $i$ th target pays attention to the  $j$ th neighbor.

only reported on fully detected pedestrians for fair comparison across the methods. Our model GST (D), which is GST with a deterministic output function  $\phi_h$  in Equation 3, exceeds mean AOE/FOE performance of other state-of-the-art approaches on most datasets. However, large NLL of GST (D) indicates relative uni-modality of the prediction samples in contrast to probabilistic baselines, because the stochasticity of GST (D) is only from sampling of the sparse interaction graphs. A more diverse trajectory sample distribution can be achieved by applying a Gaussian displacement output function, which is presented as GST (P) in Table I.

**Visualization.** Trajectory prediction and inferred sparse interaction graphs in different scenes are visualized in Fig. 2. Our model is able to predict trajectories of all partially detected pedestrians, and also model their interaction by inferring sparse interaction graphs. We visualize these graphs at  $t=3.2$  sec and  $t=5.6$  sec in each scene, where the former is the last observed time step, and the latter is at the middle of the prediction period. The graph structure varies at different times, indicating

the evolution of the relationship between pedestrians due to the change of their positions and velocities. For example, in the HOTEL scene, the 5th and the 6th pedestrians were considered not interactive at  $t=3.2$  sec, while the mutual attention is added later at  $t=5.6$  sec during the recursive prediction. In contrast, there were initially bi-directed edges between the 4th and the 5th pedestrians in the ZARA1 scene at  $t=3.2$  sec. However, they walked away from each other, and both edges are removed at  $t=5.6$  sec.

C. Comparative Study

The effect that each component of Gumbel Social Transformer has on the performance of trajectory prediction is assessed by extensive comparative experiments. Besides ETH and UCY, crowd datasets CFF, LCAS, WILDTRACK and SYNTH are adopted from Trajnet++ to investigate how the performance of different configurations varies across datasets with different crowd densities [36]–[39]. Moreover, the trained models are directly applied in multi-agent simulation for gen-

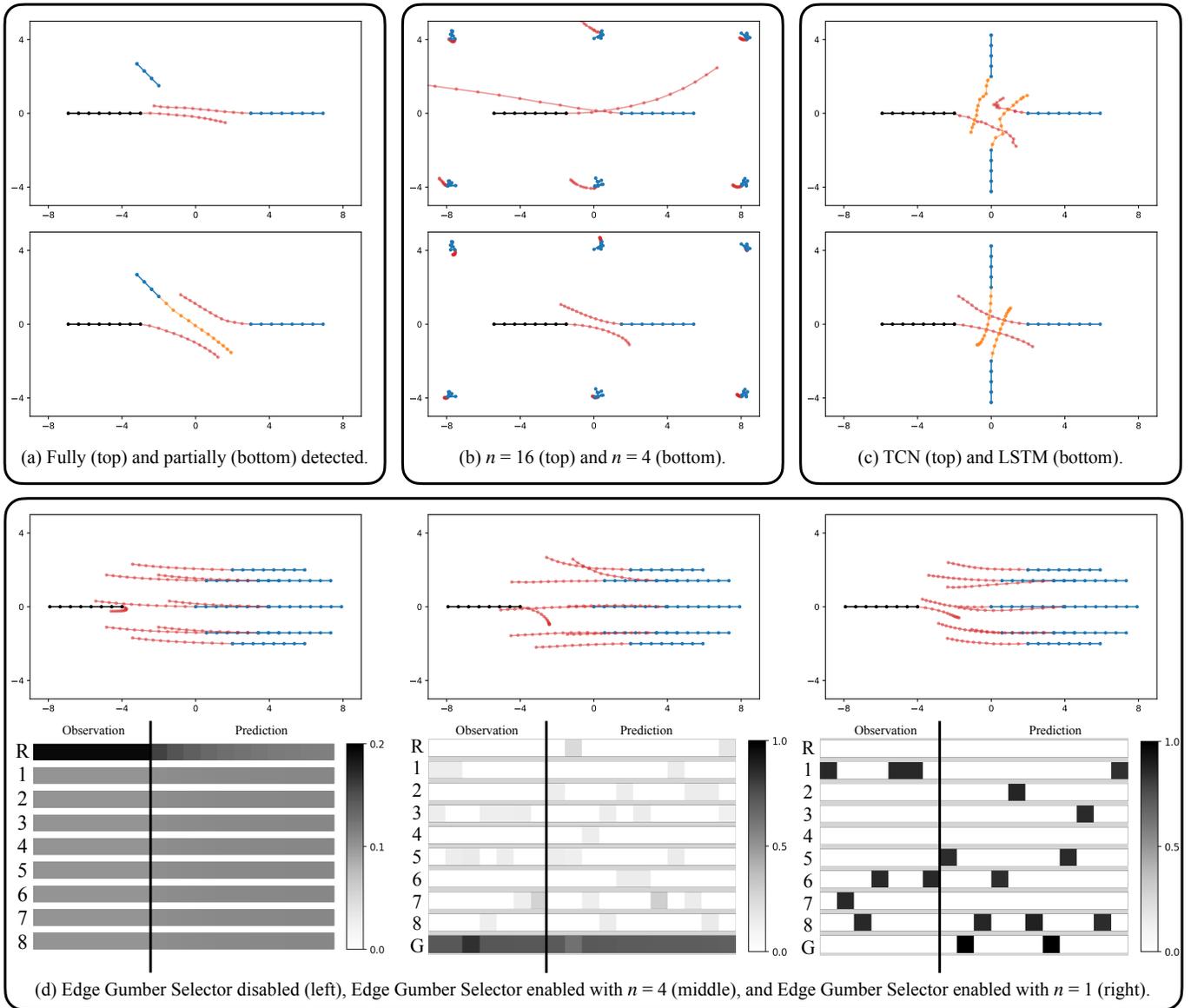


Fig. 3: Comparison of multi-agent simulation results in different human-robot interaction scenarios: (a) Human agents enter the robot agent’s field of view at different times; (b) Some human agents walk aimlessly in the scene; (c) robot meets humans at intersection; (d) One robot agent encounters a group of human agents. Black and blue represents the observation on robot and human agents. Red indicates prediction on the robot and fully detected humans, and orange indicates prediction on the other partially detected humans. In time-series robot attention plots of (d), R denotes robot, 1-8 denote humans, and G denotes a ghost agent with zero-valued features for encouraging sparsity.

erating future motion of robot and human agents. The quality of the generated motions is examined in common yet nontrivial human-robot interaction scenarios. The simulation is similar to the setup in [9], [10], where holonomic kinematics are used for both robot and human agents, and the displacements of the generated trajectories are action inputs to each agent.

**Partially or fully detected Pedestrians.** Fig. 4 reports that under most sparsity configurations, trajectory prediction is improved among all datasets by considering partially detected pedestrians. We reason that trajectories of partially detected pedestrians create a complete picture of pedestrian interaction during the observation period, and thus provides an unbiased input for encoding socially aware pedestrian features. The importance of partially detected pedestrians is illustrated in Fig. 3 (a), where a robot agent and a fully detected human

agent walk against each other. The second human agent appears 1.6 seconds (4 time steps) later and then starts approaching other agents. The top of Fig. 3 (a) shows the case when both the robot and the fully detected human ignore the partially detected one. In contrast, the bottom of Fig. 3 (a) illustrates that both agents deviate from the original path to dodge the partially detected human.

**Sparse or Fully Connected Interaction Graph.** The improvement of trajectory prediction by replacing the fully connected interaction graph with the best sparse configuration is demonstrated for each dataset in Fig. 5. We observe a trend that when the average number of partially detected pedestrians is larger in a scene, the improvement is less significant. This phenomenon may be attributed to the fact as presented in Fig. 6, where the best hyperparameter  $n$  is likely to be larger

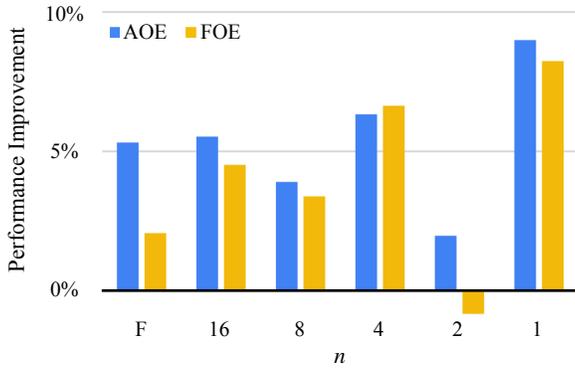


Fig. 4: Improvement of trajectory prediction performance by changing from fully detected pedestrians to partially detected pedestrians for different sparse configurations. F denotes full connection.

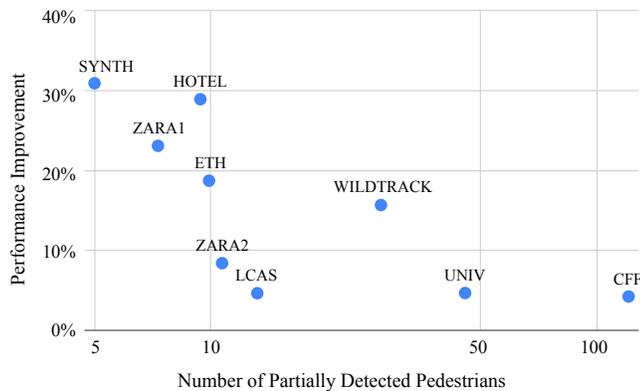


Fig. 5: Improvement of trajectory prediction performance by changing from fully connected configuration to the best sparse configuration across datasets.

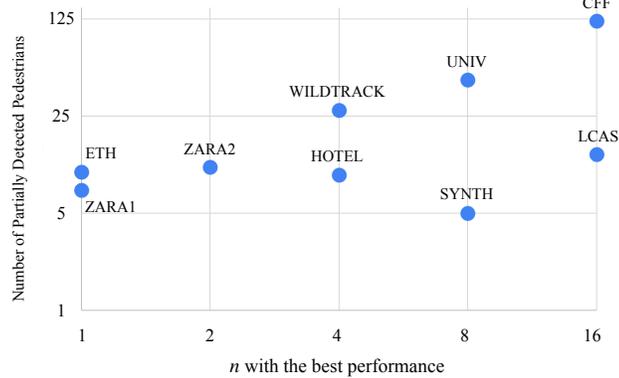


Fig. 6: The sparsity hyperparameter  $n$  with the best prediction performance across datasets.

if the considered dataset has a higher crowd density. A larger  $n$  indicates more connectivity between a target and neighbors, and thus a narrower gap between the corresponding sparse configuration and the fully connected counterpart.

Sparsity is critical to address the freezing agent problem caused by over-smoothing. As shown in Fig. 3 (d), a robot agent is moving right and encounters a human crowd of eight moving left. The left of Fig. 3 (d) presents the results when Edge Gumbel Selector is disabled. The robot agent turned around in order to dodge the crowd. The prediction on robot motion is too conservative to match pedestrian social norms.

The prediction of crowd motion which responds to the conservative robot motion would also be erroneous, and can affect the down-stream motion planning pipeline. The time-series attention visualization in Fig. 3 (d) left indicates the robot paid more attention to itself during the observation period, but then the attention is gradually distributed to human neighbors. We reason that the features of the crowd overwhelmed the robot features in the weighted sum step of self-attention. This smoothing effect is passed with the hidden states and leads to the U-turn behavior. This unnatural turnaround motion is effectively alleviated by enabling Edge Gumbel selector to introduce sparsity, as shown in middle and right of Fig. 3 (d). Note in Fig. 3 (d) left, the same attention across human agents and their identical predicted trajectories imply symmetry of human agents in the fully connected interaction graph.

As shown in Fig. 6, the sparsity constraint needs to be tradeoff conditioned on the crowd density of different scenes. It is important to control sparsity level and help target agent concentrate on interaction with critical neighbors. Fig. 3 (b) illustrates a scene where six human agents are moving randomly near the boundaries, and an important human agent is running into the robot. The top of Fig. 3 (b) shows the results with  $n = 16$ , where the robot and the interactive human exhibit exaggerated motion. However, the robot agent naturally avoids collision with the close human neighbor at the bottom of Fig. 3 (b), where  $n$  is set as 4. This indicates when the interaction graph is almost fully connected, the target agent is easy to be affected by the connected neighbors, even the ones who are clearly non-influential.

**Recursive or Readout Prediction.** We analyze the effect of recursive trajectory prediction by comparison between Masked LSTM and temporal convolution network (TCN), which has been applied to encode temporal relationship and make sequential prediction [16], [40]. While quantitative performances are found comparable, we see in Fig. 3 (c) the trajectories generated from TCN are less smooth than those from Masked LSTM. The intuition is that TCN implements a multi-layer perceptron to expand temporally encoded features from the sequence length of  $T_{obs}$  to  $T_{pred}$ , and generates distinctive and discontinuous features for prediction of consecutive displacements. In contrast, Masked LSTM encodes the features in time sequence with shared weights, so the temporally encoded features which will be mapped to the predicted displacements keep the continuity, and produce smoother and more natural future trajectories.

## V. CONCLUSIONS AND FUTURE WORK

We identify two common assumptions of existing pedestrian trajectory prediction approaches: pedestrian positions are always successfully tracked, and the target agent pays attention to all pedestrians in the detected range. These assumptions can cause issues of the deployment of trajectory prediction algorithms in real world robot applications. We present Gumbel Social Transformer to overcome these issues. Our model architecture is designed to encode features of partially detected pedestrians, and thus provides a complete input for unbiased modeling on pedestrian interaction. We

propose Edge Gumbel Selector, which is an unsupervised method that infers a sequence of sparse interaction graphs to summarize the evolving relationship among pedestrians. We demonstrate the the introduction of sparsity to modeling multi-agent interaction effectively alleviates the freezing robot problem, and minimizes the influence on generating target agent's motion from unimportant neighbors.

However, we also observe that the performance of a sparsity configuration is dependent on scene properties such as average number of partially detected pedestrians. A fixed sparse hyperparameter would constrain generalization across the scenes. As future work, we will extend our approach with learnable sparsity to handle scenes with varying crowd densities. To finetune our trajectory prediction model for navigation applications, a trajectory dataset of human-robot interaction will be collected with a bird's eye view camera similar to [3]. As for deployment on a generic camera-on-robot setup [10], we will explore occlusion inference approaches [41] to attempt to minimize the influence of long occlusion periods on the trajectory prediction performance.

#### REFERENCES

- [1] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2017, pp. 1343–1350.
- [2] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *Int. J. Robot. Res.*, vol. 39, no. 8, pp. 895–935, 2020.
- [3] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa, "Planning-based prediction for pedestrians," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2009, pp. 3931–3936.
- [4] P. Du, Z. Huang, T. Liu, K. Xu, Q. Gao, H. Sibai, K. Driggs-Campbell, and S. Mitra, "Online monitoring for safe pedestrian-vehicle interactions," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2020.
- [5] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 961–971.
- [6] A. Vemula, K. Muelling, and J. Oh, "Social attention: Modeling attention in human crowds," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 4601–4607.
- [7] 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. Vision Pattern Recognit.*, 2018, pp. 2255–2264.
- [8] 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. Vision Pattern Recognit.*, 2019, pp. 12 085–12 094.
- [9] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019, pp. 6015–6022.
- [10] S. Liu, P. Chang, W. Liang, N. Chakraborty, and K. Driggs-Campbell, "Decentralized structural-rnn for robot crowd navigation with deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021.
- [11] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2010, pp. 797–803.
- [12] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, no. 5, p. 4282, 1995.
- [13] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 1928–1935.
- [14] G. Ferrer and A. Sanfeliu, "Behavior estimation for a complete framework for human motion prediction in crowded environments," in *Proc. IEEE Int. Conf. Robot. Autom.* IEEE, 2014, pp. 5940–5945.
- [15] 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. Vision*, 2019, pp. 6272–6281.
- [16] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "Social-stgcn: A social spatio-temporal graph convolutional neural network for human trajectory prediction," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2020.
- [17] C. Yu, X. Ma, J. Ren, H. Zhao, and S. Yi, "Spatio-temporal graph transformer networks for pedestrian trajectory prediction," in *Proc. Eur. Conf. Comput. Vision*. Springer, 2020, pp. 507–523.
- [18] S. Zhu, I. Ng, and Z. Chen, "Causal discovery with reinforcement learning," in *Proc. Int. Conf. Learn. Rep.*, 2019.
- [19] S. Xie, H. Zheng, C. Liu, and L. Lin, "Snas: stochastic neural architecture search," in *Proc. Int. Conf. Learn. Rep.*, 2019.
- [20] W. Jin, R. Barzilay, and T. Jaakkola, "Junction tree variational autoencoder for molecular graph generation," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2018, pp. 2323–2332.
- [21] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel, "Neural relational inference for interacting systems," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2018, pp. 2688–2697.
- [22] P. Erdős and A. Rényi, "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci.*, vol. 5, no. 1, pp. 17–60, 1960.
- [23] J. You, R. Ying, X. Ren, W. Hamilton, and J. Leskovec, "Graphrnn: Generating realistic graphs with deep auto-regressive models," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2018, pp. 5708–5717.
- [24] N. Anand and P.-S. Huang, "Generative modeling for protein structures," in *Proc. Int. Conf. Neural Info. Process. Sys.*, 2018, pp. 7505–7516.
- [25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Int. Conf. Neural Info. Process. Sys.*, 2017, pp. 6000–6010.
- [26] Y. Rong, W. Huang, T. Xu, and J. Huang, "Dropedge: Towards deep graph convolutional networks on node classification," in *Proc. Int. Conf. Learn. Rep.*, 2019.
- [27] L. Franceschi, M. Niepert, M. Pontil, and X. He, "Learning discrete structures for graph neural networks," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2019, pp. 1972–1982.
- [28] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 3595–3603.
- [29] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *Proc. Int. Conf. Learn. Rep.*, 2017.
- [30] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," in *Proc. Int. Conf. Learn. Rep.*, 2017.
- [31] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Proc. IEEE Int. Conf. Comput. Vision*, 2009, pp. 261–268.
- [32] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," in *Comput. Graph. Forum*, vol. 26, no. 3. Wiley Online Library, 2007, pp. 655–664.
- [33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Rep.*, vol. 1412, 2015.
- [34] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Multi-agent generative trajectory forecasting with heterogeneous data for control," *Proc. Eur. Conf. Comput. Vision*, 2020.
- [35] Z. Huang, A. Hasan, K. Shin, R. Li, and K. Driggs-Campbell, "Long-term pedestrian trajectory prediction using mutable intention filter and warp lstm," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 542–549, 2021.
- [36] A. Alahi, V. Ramanathan, and L. Fei-Fei, "Socially-aware large-scale crowd forecasting," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2014, pp. 2203–2210.
- [37] L. Sun, Z. Yan, S. M. Mellado, M. Hanheide, and T. Duckett, "3dof pedestrian trajectory prediction learned from long-term autonomous mobile robot deployment data," in *Proc. IEEE Int. Conf. Robot. Autom.* IEEE, 2018, pp. 5942–5948.
- [38] T. Chavdarova, P. Baqué, S. Bouquet, A. Maksai, C. Jose, T. Bagautdinov, L. Letry, P. Fua, L. Van Gool, and F. Fleuret, "Wildtrack: A multi-camera hd dataset for dense unscripted pedestrian detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 5030–5039.
- [39] P. Kothari, S. Kreiss, and A. Alahi, "Human trajectory forecasting in crowds: A deep learning perspective," *IEEE Tran. Intell. Transp. Syst.*, 2021.
- [40] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018.
- [41] M. Itkina, Y.-J. Mun, K. Driggs-Campbell, and M. J. Kochenderfer, "Multi-agent variational occlusion inference using people as sensors," 2021, *arXiv:2109.02173*.