Stepwise Goal-Driven Networks for Trajectory Prediction

Chuhua Wang, Yuchen Wang, Mingze Xu, and David J. Crandall

***Abstract*—We propose to predict the future trajectories of observed agents (*e.g.*, pedestrians or vehicles) by estimating and using their goals at multiple time scales. We argue that the goal of a moving agent may change over time, and modeling goals continuously provides more accurate and detailed information for future trajectory estimation. To this end, we present a recurrent network for trajectory prediction, called Stepwise Goal-Driven Network (SGNet). Unlike prior work that models only a single, long-term goal, SGNet estimates and uses goals at multiple temporal scales. In particular, it incorporates an encoder that captures historical information, a stepwise goal estimator that predicts successive goals into the future, and a decoder that predicts future trajectory. We evaluate our model on three first-person traffic datasets (HEV-I, JAAD, and PIE) as well as on three bird's eye view datasets (NuScenes, ETH, and UCY), and show that our model achieves state-of-the-art results on all datasets. Code has been made available at: *https://github.com/ChuhuaW/SGNet.pytorch*.**

***Index Terms*—Autonomous vehicle navigation, autonomous agents, trajectory prediction.**

I. Introduction

PREDICTING the future behavior of other agents is crucial in the real world [1]: safe driving requires predicting future movements of other cars and pedestrians, for example, while effective social interactions require anticipating the actions of social partners. However, predicting another agent's future actions is challenging because such actions depend on numerous factors including the environment and the other agent's internal state (*e.g.*, its intentions and goals). Recent work [2]–[5] has explored goal-driven methods for trajectory prediction, which explicitly try to estimate the other agent's long-term goal to help predict its future behavior. While these models make a significant step forward, they adopt the simplistic assumption that an agent's intentions are exclusively represented by a single long-term goal.

However, work in psychology and cognitive science suggests that people base their actions not on a single long-term goal, but instead on a series of goals at different time scales. Some literature [6], [7] uses the term *intention* to refer to a representation of planned actions and the term *goal* to stress the end result of an action. This suggests that a series of goals may better represent an intention than using a single, long-term goal. Besides, people's intentions build up progressively, and each decision made in the past may have an impact on how the future is determined. For example, a pedestrian plans a trajectory before crossing the street; when he or she takes a step forward, the prior planned trajectory and the actual situation are both taken into account to update a new set of stepwise goals. We show that estimating stepwise goals at the initial time step and carrying them over to subsequent time steps results in a more precise model of intention and better guidance for future trajectory.

We present Stepwise Goal-Driven Network (SGNet) to address the trajectory prediction problem. SGNet consists of three main components: (1) A stepwise goal estimator (SGE) that predicts coarse future goals at multiple temporal scales to encode a comprehensive representation of the intention. To determine the significance of each stepwise goal, a lightweight module with an attention mechanism is used. (2) An encoder that records past data in conjunction with predicted stepwise goals, to incorporate a richer hidden representation that aids in predicting the future and in creating new stepwise goals for the next time step. (3) A decoder that takes advantage of stepwise goals to predict future trajectories. We show how stepwise goals evolve and aid in predicting future trajectory in Fig. 1.

We evaluate our model on multiple first- and third-person datasets, including both vehicles and pedestrians, and compare to an extensive range of existing work ranging from deterministic to stochastic approaches. We surpass or match the state-of-the-art performance on multiple benchmarks, including different viewpoints (i.e., first- and third-person) and agents (i.e., cars and pedestrians).

The contributions of this letter are three-fold. First, our work highlights a new direction for goal-driven trajectory prediction by modeling goals at multiple time scales. SGE is a versatile module that may be applied to a wide range of architectures. Second, we show how to effectively incorporate each series of stepwise goals into an encoder and decoder. By integrating stepwise objectives into the decoder, we can direct the trajectory prediction in the current step, and by embedding stepwise goals into encoder, we can generate more accurate future goals for the next time step. Finally, our goal aggregator employs an attention mechanism to adaptively learn the relative relevance of each stepwise goal, thereby improving performance even further.

Manuscript received September 9, 2021; accepted December 30, 2021. Date of publication January 25, 2022; date of current version February 1, 2022. This letter was recommended for publication by Associate Editor G. Gallego and Editor C. Cadena Lerma upon evaluation of the reviewers' comments. This work was supported in part by the U.S. Navy under Grant N00164-21-1-1002 and in part by the Indiana University Office of the Vice Provost for Research through the Emerging Areas of Research Project Learning: Brains, Machines, and Children. *(Chuhua Wang and Yuchen Wang contributed equally to this work.) (Corresponding author: Chuhua Wang.)*

Chuhua Wang, Yuchen Wang, and David J. Crandall are with the Luddy School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN 47408 USA (e-mail: cw234@iu.edu; wang617@indiana.edu; djcran@indiana.edu).

Mingze Xu was with the Luddy School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN 47408 USA. He is now with Amazon/AWS AI, Seattle, WA 98170 USA (e-mail: mx6@indiana.edu).

Digital Object Identifier 10.1109/LRA.2022.3145090

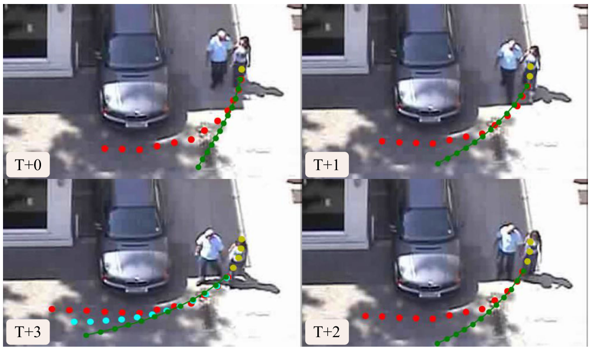

Fig. 1. Illustration of stepwise goals (clockwise from top left). Panes (T+0), (T+1), and (T+2) show the initial, second, and third time step, respectively, where yellow indicates observed trajectory, red indicates ground truth future trajectory, green indicates stepwise goals, and cyan indicates prediction. As the model accumulates historical data, goals build up progressively and become more accurate, creating better representations of intention. The goals help predict future trajectory in Pane (T+3). Goals are represented as feature vectors, but we map them to locations for visualization purposes.

II. Related Work

Trajectory prediction from first-person views jointly models the motion of observed objects and the ego-camera. Bhattacharyya *et al.* [8] propose the Bayesian LSTMs to model observation uncertainty and predict the distribution of future locations. Yagi *et al.* [9] use multi-modal data, such as human pose, scale, and ego-motion, as cues in a convolution-deconvolution (Conv1D) framework to predict future pedestrian locations. Yao *et al.* [10] introduce a multi-stream encoder-decoder that separately captures both object location and appearance. Makansi *et al.* [11] estimate a reachability prior for objects from the semantic map and propagate them into the future.

Trajectory prediction from a bird's eye view simplifies the problem by removing the ego-motion. Alahi *et al.* [12] propose Social-LSTM to model pedestrians' trajectories and interactions. Their social pooling module was improved by [13] to capture global context. SoPhie [14] applies generative models to model the uncertainty of future paths. Lee *et al.* [15] use RNNs with conditional variational autoencoders (CVAEs) to generate multi-modal predictions. Recent work [16]–[18] also proposes graph-based recurrent models, simultaneously predicting potential trajectories of multiple objects, while [19] exploits more dynamic and heterogeneous inputs. PLOP [20] and Argoverse [21] use the ego trajectory in a bird's eye view map. Simaug and SMARTS [22], [23] take advantage of simulation data to train the prediction model. Others [24]–[29] have explored multimodal inputs, such as Lidar [30]–[33], to aid in trajectory prediction.

Goal-driven trajectory prediction incorporates estimated future goals. Rhinehart *et al.* [2] anticipate multi-modal semantic actions as the goal and conduct conditional forecasting using imitative models. Deo *et al.* [34] estimate goal states and fuse the results with past trajectories using maximum entropy inverse reinforcement learning. PECNet [3] infers distant destinations to improve long-range trajectory prediction. TNT [4] decomposes the prediction task into three stages: predicting potential target states, generating trajectory state sequences, and estimating trajectory likelihoods. BiTraP [5] uses a bi-directional decoder on the predicted goal to improve long-term trajectory prediction.

In contrast to the above methods that only estimate and use the final goal (i.e., the destination), our model predicts goals at multiple temporal scales and incorporates them into our encoder-decoder framework using attentive mechanisms.

III. Stepwise Goal-Driven Network (SGNet)

At time step t, given an object's observed trajectory in the last ℓ_e steps, $\mathbf{X}_t = \{\mathbf{x}_{t-\ell_e+1}, \mathbf{x}_{t-\ell_e+2}, \ldots, \mathbf{x}_t\}$, where $\mathbf{x}_t$ includes its bounding box (i.e., centroid position and width and height in pixels) and motion (*e.g.*, optical flow, velocity, and acceleration), our goal is to predict its future positions $\mathbf{Y}_t = \{\mathbf{y}_{t+1}, \mathbf{y}_{t+2}, \ldots, \mathbf{y}_{t+\ell_d}\}$ in the next ℓ_d frames.

A. Overview

Because a person's intentions are a representation of planned actions, employing a single long-term goal to portray it is quite limited. To give a more comprehensive intention representation and improve the quality of trajectory prediction, we propose estimating numerous smaller goals along the way and explicitly including them at each decoder time step. Furthermore, people regularly adjust and optimize their intentions, and previous intentions can impact how they perceive the present, as well as aid in the development of new future plans. Thus historical goals can be treated as additional information in the encoder for embedding the present representation and to forecast future stepwise goals. One simple way to merge the stepwise goals is through average pooling as in [35]. However, we believe that at each time step, each individual goal has a different impact on the prediction. Average pooling smooths out the goal features and hence the important goal features may not be identified. Therefore, we use an aggregator that adaptively learns the importance of each stepwise goal with an attention mechanism. We propose a new recurrent encoder-decoder architecture, Stepwise Goal-Driven Network (SGNet), which predicts goals step-by-step to provide guidance during trajectory prediction as well as supplementary features to help predict new goals at the next time step.

Fig. 2 presents an overview of SGNet. In particular, the stepwise goal estimator (SGE) predicts an object's future locations from $t+1$ to $t+\ell_d$ as stepwise goals, and embeds them as input to the decoder in an incremental manner to ensure trajectory prediction is guided without receiving any redundant information. SGE also fuses and feeds all predicted goals into the encoder for the next time step, which, as our experiments will show, helps encode a better representation of the current information, and help predicting new stepwise goals for the next time step.

B. Encoder

The encoder captures an agent's movement behavior as a latent vector by embedding its historical trajectory $\mathbf{X}_t$ using a single fully-connected layer. If additional motion features (*e.g.*, optical flow) are available, they are also embedded using a separate fully-connected layer and concatenated with the trajectory representations. The input feature $\mathbf{x}_t^e$ is then concatenated with the aggregated goal information $\widetilde{\mathbf{x}}_t^e$ from the previous time step $t-1$, and then the new hidden state h_t^e is updated through a recurrent cell. Hidden state h_{t+1}^e and goals $\widetilde{\mathbf{x}}_{t+1}^e$ are both set to zero for the first time step. We next discuss how to obtain the aggregated goal input.

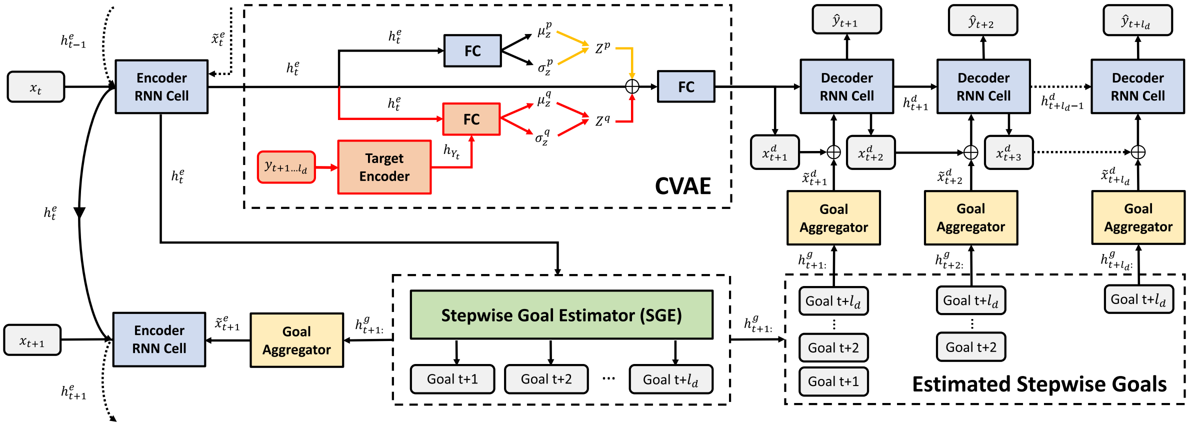

Fig. 2. Visualization of SGNet. Arrows in red, yellow, and black indicate connections during training, inference, and both training and inference, respectively. Encoder time evolves vertically, from time t to $t+1$, while decoder time flows horizontally, predicting the trajectory at $t+1$ to $t+l_d$. For deterministic results, we replace CVAE with a non-linear embedding.

C. Stepwise Goal Estimator (SGE)

The main idea of SGE is to generate coarse stepwise goals to assist trajectory prediction in a coarse-to-fine manner. These goals also help the network to create a new stepwise goal for the next step. Consequently, we design SGE to predict and convey the predicted coarse stepwise goals to both encoder and decoder. For encoder, at each time step t, a set of stepwise goals from $t-1$ is concatenated with the input to serve as supplementary features, helping the encoder learn a more discriminative representation of the sequence. Since inaccurate future goals may mislead the prediction, we use a goal aggregator that adaptively learns the importance of each stepwise goal by using an attention mechanism. Meanwhile, for decoder, at each time step $t+i$ $(i \in [1, \ell_d])$, a subset of stepwise goals $h_{t+i:}$ serves as coarse guidance to help trajectory prediction, and a goal aggregator again gathers the selected goals.

We define a generic SGE module f_{SGE} as,

$$h^g_{t+1:} = f_{SGE}(ReLU(\mathbf{W}^T_\gamma h^e_t + \mathbf{b}_\gamma)) \tag{1}$$

where $h^g_{t+1:}$ is a sequence of stepwise goals, and $h^g_{t+1:} = \{h^g_{t+1}, h^g_{t+2}, \ldots, h^g_{t+\ell_d}\}$. h^e_t is the encoder hidden state at time t. We found that SGE is open to different implementations including recurrent, convolution, and fully-connected layers, as will be described in Sec. IV-D. To regularize SGE to generate goals that contain precise future information, we regress the goal position $\widehat{\mathbf{Y}}^g_t$ using the same regressor defined in Sec. III-E to minimize the distance between goal position and the ground truth.

Goal Aggregator for Encoder and Decoder. We design goal aggregators for both the encoder and decoder, in order to combine and compress multiple goals into a single representation with attention. We define the goal aggregator,

$$w = \mathrm{Softmax}(\mathbf{W}^T \mathrm{Tanh}(h^g_{t+i:}) + \mathbf{b}) \tag{2}$$

$$\widetilde{\mathbf{x}}_{t+i} = f_{attn}(h^g_{t+i:}) = \sum_{s=t+i}^{\ell_d} w_s h^g_s, \tag{3}$$

where w is an attention vector corresponding to the probability distribution over a subset of estimated goals, and w_s is the weight for each individual goal h^g_s. The overall structure is shown in Fig. 3.

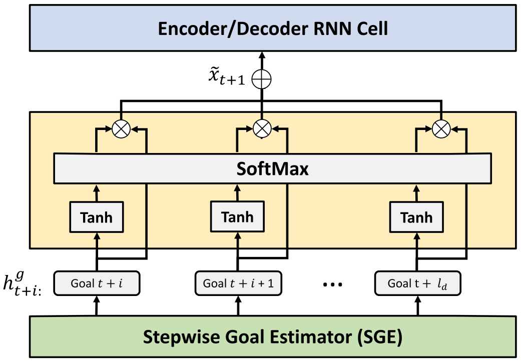

Fig. 3. Detailed structure of Goal Aggregator. It receives a set of stepwise goals as inputs and calculates the attention weights adaptively. The re-weighted goal features are fed into encoder or decoder.

For the encoder, we hypothesize that the anticipation differs at each time step, and people's anticipation in the past may influence their perception of the present and future. As a result, at time t, the encoder receives stepwise goals $h^g_{t+1:} = \{h^g_{t+1}, h^g_{t+2}, \ldots, h^g_{t+\ell_d}\}$. **For the decoder**, at time step $t+i$, the decoder receives goals $h^g_{t+i:} = \{h^g_{t+i}, h^g_{t+i+1}, \ldots, h^g_{t+\ell_d}\}$, and the goals before $t+i$ are ignored, because these are redundant and have been encoded in the historical information.

D. Conditional Variational Autoencoder (CVAE)

A Conditional Variational Autoencoder (CVAE) framework is applied to learn the distribution of future trajectory Y_t conditioned on the observed trajectory X_t by introducing a latent variable z. Following prior work [3], [5], [8], our CVAE consists of three components: recognition network $Q_\phi(z|X_t, Y_t)$, prior network $P_\nu(z|X_t)$, and generation network $P_\theta(h^d_t|X_t, z)$, where ϕ, ν, θ denote the parameters of these three networks, and h^d_t is the trajectory encoded by the generation network. Recognition, prior, and generation networks are implemented with fully-connected layers.

During training, the ground truth future trajectory Y_t is fed into the target encoder to output the hidden state h_{Y_t}. To capture the dependencies between observed and ground truth trajectories, the recognition network takes hidden states h_t^e and h_{Y_t} and predicts the distribution mean μ_z^q and standard deviation σ_z^q. The prior network takes h_t^e only and predicts μ_z^p and σ_z^p. We sample z from $N(\mu_z^q, \sigma_z^q)$ and concatenate it with h_t^e to generate h_t^d with the generation network. **During testing**, the ground truth future trajectory is not available. The generation network concatenates z, which is sampled from $N(\mu_z^p, \sigma_z^p)$ with h_t^e and produces h_t^d.

E. Decoder

Our recurrent decoder outputs the final trajectory $\mathbf{Y}_t = \{\mathbf{y}_{t+1}, \mathbf{y}_{t+2}, \ldots, \mathbf{y}_{t+\ell_d}\}$ with a trajectory regressor. Given h_t^d and estimated goal input $\widetilde{\mathbf{x}}_{t+1}^d$, it produces a new hidden state for the next time step through a recurrent cell. Our trajectory regressor is a single fully-connected layer that takes hidden states h_{t+i}^d and computes a trajectory $\widehat{\mathbf{y}}_{t+i}$ at each time step,

F. Loss Functions

We use the Root Mean Square Error (RMSE) as loss function to supervise trajectory prediction from our decoder. For our stochastic model using CAVE, we follow [5], [8] and adapt best-of-many (BoM) approach to minimize the distance between our best prediction $\widehat{\mathbf{Y}}_t$ and target $\mathbf{Y}_t$. This approach leads to more accurate and diverse predictions and encourages the model to capture the true variation in data. To ensure SGE to predict accurate goal states, we also optimize the prediction from SGE using RMSE between goal prediction $\widehat{\mathbf{Y}}_t^g$ and ground truth $\mathbf{Y}_t$. Finally, we add KL-divergence loss (KLD) to optimize the prior network in the CVAE. Thus, for each training sample, our final loss is summarized as follows,

$$\begin{aligned}\mathcal{L}_{total} = & \min_{\forall k \in K} \text{RMSE}(\widehat{\mathbf{Y}}_t^k, \mathbf{Y}_t) + \text{RMSE}(\widehat{\mathbf{Y}}_t^g, \mathbf{Y}_t) \\ & + \text{KLD}(Q_\phi(z|X_t, Y_t), P_\nu(z|X_t)),\end{aligned}$$

where $\widetilde{\mathbf{Y}}_t^k$ is the k-th trajectory hypothesis from CVAE, $\widetilde{\mathbf{Y}}_t^g$ is the predicted stepwise goal location, and $\mathbf{Y}_t$ is the object's ground truth location at time t.

IV. Experiments

A. Datasets

First-person datasets. JAAD [41] and PIE [37] have egocentric videos recorded at 30 frames per second (fps), and consist of 2,800 and 1,835 pedestrian trajectories, respectively. Following [37], we divided the datasets into train (50%), validation (10%), and test (40%) sets. HEV-I [10] includes 230 videos that are splitted into 40,000 train and 17,000 test samples. Following [10], the annotations are generated by using Mask-RCNN and Sort [42] with a Kalman filter. We use 1.6 seconds of observations to predict future trajectories of length 0.5, 1.0, and 1.5 seconds.

Third-person dataset. ETH [43] and UCY [44] include 1,536 pedestrians in 5 sets of data with 4 unique scenes. Following prior work [19], a leave-one-out strategy is used to split the train and test sets. We use 3.2 seconds of observations to predict 4.8 seconds future trajectories. NuScenes [45] is a dataset for autonomous driving, and we follow the their prediction challenge splits and settings. We use 2 seconds of observations to predict 6 seconds future trajectories.

B. Implementation Details

We use Gated Recurrent Units (GRUs) as the backbone for both encoder and decoder with 512 hidden size. The length of the observation ℓ_e is determined by the default setting of each benchmark: ℓ_e is 16 on HEV-I, 15 on JAAD, 15 on PIE, and 8 on ETH-UCY. Object bounding boxes are taken as inputs for JAAD and PIE, and following [10], optical flow is also included on HEV-I. We follow [19] to use object centroids, velocities, and accelerations as inputs for ETH and UCY. We use the Adam [46] optimizer with initial learning rate 5×10^{-4}, which is dynamically reduced based on the validation loss. Our models are optimized end-to-end with batch size 128 and the training is terminated after 50 epochs.

C. Evaluation Protocols

Our main evaluation metrics are **average displacement error (ADE)**, which measures accuracy along the whole trajectory, and **final displacement error (FDE)**, which measures accuracy only at the trajectory end point. For first-person datasets, we use upper-left and lower-right coordinates of bounding boxes to calculate ADE and FDE, except for HEV-I where we only use the upper-left coordinate to be consistent with prior work. In the bird's eye view ETH and UCY datasets, we follow prior work to use the coordinates of points. To compare to the state-of-the-art [36] in HEV-I, we also use **final intersection over union (FIOU)**, the overlap between the predicted bounding box and ground truth at the final step, which measures the model's ability to predict both the scale and location of bounding boxes in the long-term. We use **mean squared error (MSE)** to evaluate our performance on JAAD and PIE, calculated according to the upper-left and lower-right coordinates of the bounding box. **Center mean squared error (C$_{MSE}$)** and **center final mean squared error (CF$_{MSE}$)** are similar to ADE and FED except they are computed based on the bounding box centroids. All results metrics used for HEV-I, JAAD, and PIE dataset are in pixels, while for ETH and UCY we compute the ADE and FDE in Euclidean space.

D. Exploration Study

We begin with experiments for design trade-offs and training strategies of our architecture with JAAD and PIE.

How to implement SGE? We considered three instantiations of f_{attn} in the SGE module. **First,** we implement SGE with GRUs (Table I, row 9). The hidden size is set to 128, since we only need a lightweight module to produce a coarse goal prediction. h_{t+i}^g is defined to be the hidden state at time $t+i$, which is initialized by using the encoder hidden state h_t^e after a linear transformation. The GRU input $\mathbf{x}_{t+1}^g$ is initialized with a zero vector and updated using the auto-regressive strategy, and the sequence of the output hidden state at each time step is used as the stepwise goals. **Second,** we try SGE with a multilayer perceptron (MLP) (Table I, row 10) that takes the encoder hidden

TABLE I

EXPLORATION STUDY OF OUR MODEL ON JAAD AND PIE. ↓ DENOTES LOWER IS BETTER. THE LAST ROW IS OUR BEST MODEL

SGE	Encoder	Decoder	#Goals	Attention	JAAD			PIE		
					MSE ↓ (0.5s / 1.0s / 1.5s)	C_{MSE} ↓ (1.5s)	CF_{MSE} ↓ (1.5s)	MSE ↓ (0.5s / 1.0s / 1.5s)	C_{MSE} ↓ (1.5s)	CF_{MSE} ↓ (1.5s)
GRU	✓		45	✓	90 / 372 / 1176	1117	4497	40 / 154 / 496	464	1939
GRU		✓	45	✓	87 / 350 / 1121	1065	4355	37 / 148 / 478	450	1891
GRU	✓	✓	45		85 / 341 / 1102	1044	4269	37 / 143 / 476	445	1905
GRU	✓	✓	1	✓	94 / 372 / 1162	1077	4369	41 / 150 / 499	452	1919
GRU	✓	✓	2	✓	87 / 355 / 1140	1074	4394	40 / 148 / 482	449	1934
GRU	✓	✓	4	✓	87 / 345 / 1100	1038	4248	38 / 144 / 479	448	1925
GRU	✓	✓	8	✓	86 / 345 / 1088	1028	4179	38 / 145 / 476	446	1890
GRU	✓	✓	16	✓	85 / 339 / 1082	1015	4173	38 / 145 / 473	443	1866
GRU	✓	✓	32	✓	85 / 333 / 1064	1011	4146	37 / 142 / 467	438	1868
MLP	✓	✓	45	✓	84 / 338 / 1098	1032	4376	39 / 153 / 496	464	1967
CNN	✓	✓	45	✓	88 / 344 / 1079	1016	4170	37 / 145 / 470	441	1859
GRU	✓	✓	45	✓	**82 / 328 / 1049**	**996**	**4076**	**34 / 133 / 442**	**413**	**1761**

states h_t^e as input, and directly outputs ℓ_d goals of size 128. We call this SGNet-ED-MLP. **Third,** we follow [9] and implement a convolution-deconvolution framework for SGE to predict the stepwise goals (Table I, row 11).

As shown in Table I, all of the above variants achieve the state-of-the-art results, indicating that the SGE module is not sensitive to the choice of f_{attn}. The results also suggest that using temporal models, such as GRU, as SGE is more effective and robust in future trajectory prediction. Thus, we use this version in the remaining experiments (SGNet-ED).

What if the decoder or the encoder does not receive predicted goals? To evaluate the importance of using predicted goals in the decoder and encoder, we implemented two baselines that remove the connection between SGE and the decoder (first row) or the encoder (second row). Table I demonstrates that our best model (last row) outperforms both baselines significantly, highlighting the importance of incorporating predicted goals in both the decoder and encoder for achieving more accurate results.

Does SGE need supervision? To investigate whether supervised stepwise goals offer more accurate information for trajectory prediction, we train our network without the goal module loss, RMSE($\widehat{\mathbf{Y}}_t^g, \mathbf{Y}_t$). This change decreases the accuracy of our best model by 13/68/208 and 15/57/181 on JAAD/PIE for 0.5 s, 1.0 s, and 1.5 s MSE, respectively. This suggests supervision on SGE leads to better goal representation to help predicting trajectory, and thus using loss to optimize SGE is important in our model.

What if we exclude goal aggregator? For each time step, each individual goal has a different impact on the prediction. Thus we develop goal aggregators that use an attention mechanism to understand the relative importance of different subsets of future goals. The results in Table I show that excluding this attention mechanism (row 3) reduces significantly reduces results compared to our full model.

Are stepwise goals better than fewer goals? To further illustrate that using stepwise goals is superior to fewer or only long-term goals, we replace stepwise goals with different number of goals in the encoder and decoder. We show the results in Table I (Row 4–9). For each setting, we always include the last goal, and add more goals incrementally. The results indicate that as goals are added, they offer extra information and can significantly enhance trajectory prediction performance. Performance is optimal when goals are predicted at each time step.

Our full model (SGNet-ED) thus uses GRU as the backbone for SGE, and the output stepwise goals are fed into both encoder and decoder. Our final loss term includes goal loss, BoM trajectory loss, and KLD loss.

How to handle multi-modal data? Handling multi-modal input such as social cues/interactions or ego-motion is a classic problem in trajectory prediction. For demonstration, we conducted an experiment on NuScenes prediction challenge split in the Table VI to show the ability to integrate map and interaction information: we followed CoverNet [49] to rasterize the location of each agent in the scene and overlay it on the map. We use a four layer CNN to extract features from the overplayed map and combine it with the initial decoder hidden state. We submitted our result to NuScenes and ranked third place in NuScenes prediction challenge at the 6th AI Driving Olympics [50]. The results suggest that social cues and map information can be implicitly learned and integrated into our model, considerably improving the final results. Note our proposed module is a simple and efficient temporal unit, and we only take historical trajectory as an input in the following experiments.

E. Comparison With the State-of-The-Art

In this section, we compare our best model under two different settings: deterministic, in which the model returns a single trajectory, and stochastic, in which we report the best-performing sample among K trajectories.

1) Deterministic Results on First-Person Benchmarks: We start by evaluating our model's performance on predicting pedestrian trajectory in two first-person datasets. As shown in Table II, our model (SGNet-ED) significantly outperforms the state-of-the-art on the first-person pedestrian detection datasets. Compared to [5] on JAAD, our model reduces the MSE error by 13%, 15%, and 15% for 0.5 s, 1.0 s, and 1.5 s prediction, respectively. On PIE, our model reduces MSE by 3%, 9%, and 11% for 0.5 s, 1.0 s, and 1.5s prediction. As the prediction time range increases (and thus the problem becomes harder), our model decreases MSE more significantly, suggesting that SGE is especially helpful for long-term prediction. We obtain similar

TABLE II
Deterministic Results on JAAD and PIE in Terms of MSE/C$_{MSE}$/CF$_{MSE}$. ↓ Denotes Lower is Better

Method	JAAD			PIE		
	MSE ↓ (0.5s / 1.0s / 1.5s)	C$_{MSE}$ ↓ (1.5s)	CF$_{MSE}$ ↓ (1.5s)	MSE ↓ (0.5s / 1.0s / 1.5s)	C$_{MSE}$ ↓ (1.5s)	CF$_{MSE}$ ↓ (1.5s)
Bayesian-LSTM [8]	159 / 539 / 1535	1447	5615	101 / 296 / 855	811	3259
FOL-X [36]	147 / 484 / 1374	1290	4924	47 / 183 / 584	546	2303
PIE$_{traj}$ [37]	110 / 399 / 1280	1183	4780	58 / 200 / 636	596	2477
BiTraP-D [5]	93 / 378 / 1206	1105	4565	41 / 161 / 511	481	1949
SGNet-ED	**82 / 328 / 1049**	**996**	**4076**	**34 / 133 / 442**	**413**	**1761**

TABLE III
Deterministic Results on ETH and UCY in Terms of ADE/FDE. ↓ Denotes the Lower the Better

Method	ADE (4.8s) ↓ / FDE (4.8s) ↓					
	ETH	HOTEL	UNIV	ZARA1	ZARA2	Avg
Social-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
Social-GAN [13]	1.13 / 2.21	1.01 / 2.18	0.60 / 1.28	0.42 / 0.91	0.52 / 1.11	0.74 / 1.54
MATF [38]	1.33 / 2.49	0.51 / 0.95	0.56 / 1.19	0.44 / 0.93	0.34 / 0.73	0.64 / 1.26
FvTraj [39]	0.62 / 1.23	0.53 / 1.10	0.57 / 1.19	0.42 / 0.89	0.38 / 0.79	0.50 / 1.04
STAR-D [40]	**0.56 / 1.11**	0.26 / 0.50	0.52 / 1.15	0.41 / 0.90	0.31 / 0.71	0.41 / 0.87
Trajectron++ [19]	0.71 / 1.68	**0.22 / 0.46**	0.41 / 1.07	0.30 / 0.77	0.23 / 0.59	0.37 / 0.91
SGNet-ED	0.63 / 1.38	0.27 / 0.63	**0.40 / 0.96**	**0.26 / 0.64**	**0.21 / 0.53**	**0.35 / 0.83**

TABLE IV
Stochastic Results on JAAD and PIE in Terms of MSE/C$_{MSE}$/CF$_{MSE}$. ↓ Denotes Lower is Better

Method (Best of 20)	JAAD			PIE		
	MSE ↓ (0.5s / 1.0s / 1.5s)	C$_{MSE}$ ↓ (1.5s)	CF$_{MSE}$ ↓ (1.5s)	MSE ↓ (0.5s / 1.0s / 1.5s)	C$_{MSE}$ ↓ (1.5s)	CF$_{MSE}$ ↓ (1.5s)
BiTrap-GMM [5]	153 / 250 / 585	501	998	38 / 90 / 209	171	368
BiTrap-NP [5]	38 / 94 / 222	177	565	23 / 48 / 102	81	261
SGNet-ED	**37 / 86 / 197**	**146**	**443**	**16 / 39 / 88**	**66**	**206**

results from other evaluation metrics, as shown in Table II. In addition, we perform experiments on the HEV-I first-person vehicle dataset. Our SGNet-ED yields 6.28, 11.35 and 18.27 for 0.5 s, 1.0 s, and 1.5 s ADE, 39.86 for FDE, and 0.63 for FIOU. Our results improve by an average of 10% over [36].

2) Deterministic Results on Third-Person Benchmarks: Table III shows that our model outperforms the state-of-the-art by more than 10% in terms of ADE and FDE on average.

3) Stochastic Results on First-Person Benchmarks: To fairly compare with [5], we generate $K = 20$ proposals and report the best-performing sample. For the first-person datasets, our method outperforms the state-of-the-art by an average of 14% on JAAD and 25% on PIE (Table IV).

4) Stochastic Results on Third-Person Benchmarks: For ETH-UCY, we follow the leave-one-out evaluation protocol with $K = 20$ by following the prior work in Table V. SGNet-ED outperforms the current state-of-the-art stochastic model [5] by 5% on average on ETH, ZARA1 and ZARA2, and achieves comparable results on HOTEL and UNIV.

As with the first-person datasets, our model leads to a larger improvement as the prediction length increases, implying that estimated stepwise goals provide better temporal information for accurately predicting the location and magnitude of objects. For the third-person dataset, our model does not explicitly model interaction, which may explain why our model is better on less complex scenes (ETH, ZARA1 and ZARA2). Including interaction or scene maps may help our model to improve on crowd scenes.

F. *Qualitative Results*

The first row of Fig. 4 shows four examples of our best model's deterministic predictions on JAAD, HEV-I, ETH, and UCY, respectively. In (a), the pedestrian intends to cross the street as the ego-vehicle approaches. The historical trajectory was determined by the pedestrian's movement and the vehicle's ego-motion. The prediction shows the intention of the pedestrian to cross ahead of the ego-vehicle, avoiding a collision. In (b), the target vehicle makes a right turn ahead of the ego-vehicle, and the ego-vehicle waits for it after the turn. In (c), the pedestrian follows a curved route, and (d) demonstrates our ability to notice the pedestrian's immediate change and make an accurate prediction.

The second row of Fig. 4 shows four examples of our best model's stochastic predictions. We show all 20 stochastic trajectories generated by our best model. Images (e) and (f) illustrate the results of JAAD and PIE, respectively, which both anticipate the presence of a pedestrian crossing the street. Images

TABLE V
STOCHASTIC RESULTS ON ETH AND UCY IN TERMS OF ADE/FDE. ↓ DENOTES LOWER IS BETTER

Method (Best of 20)	ADE (4.8s) ↓ / FDE (4.8s) ↓					
	ETH	HOTEL	UNIV	ZARA1	ZARA2	Avg
Social-GAN [13]	0.81 / 1.52	0.72 / 1.61	0.60 / 1.26	0.34 / 0.69	0.42 / 0.84	0.58 / 1.18
Sophie [14]	0.70 / 1.43	0.76 / 1.67	0.54 / 1.24	0.30 / 0.63	0.38 / 0.78	0.54 / 1.15
CGNS [47]	0.62 / 1.40	0.70 / 0.93	0.48 / 1.22	0.32 / 0.59	0.35 / 0.71	0.49 / 0.97
MATF GAN [38]	1.01 / 1.75	0.43 / 0.80	0.44 / 0.91	0.26 / 0.45	0.26 / 0.57	0.48 / 0.90
FvTraj [39]	0.56 / 1.14	0.28 / 0.55	0.52 / 1.12	0.37 / 0.78	0.32 / 0.68	0.41 / 0.85
DSCMP [48]	0.66 / 1.21	0.27 / 0.46	0.50 / 1.07	0.33 / 0.68	0.28 / 0.60	0.41 / 0.80
PECNet [3]	0.54 / 0.87	0.18 / 0.24	0.35 / 0.60	0.22 / 0.39	0.17 / 0.30	0.29 / 0.48
STAR [40]	0.36 / **0.65**	0.17 / 0.36	0.31 / 0.62	0.26 / 0.55	0.22 / 0.46	0.26 / 0.53
Trajectron++ [19]	0.43 / 0.86	**0.12** / **0.19**	0.22 / 0.43	0.17 / 0.32	0.12 / 0.25	0.21 / 0.41
BiTrap-GMM [5]	0.40 / 0.74	0.13 / 0.22	0.19 / 0.40	0.14 / 0.28	0.11 / 0.22	0.19 / 0.37
BiTrap-NP [5]	0.37 / 0.69	**0.12** / 0.21	**0.17** / **0.37**	0.13 / 0.29	**0.10** / **0.21**	**0.18** / **0.35**
SGNet-ED	**0.35** / **0.65**	**0.12** / 0.24	0.20 / 0.42	**0.12** / **0.24**	**0.10** / **0.21**	**0.18** / **0.35**

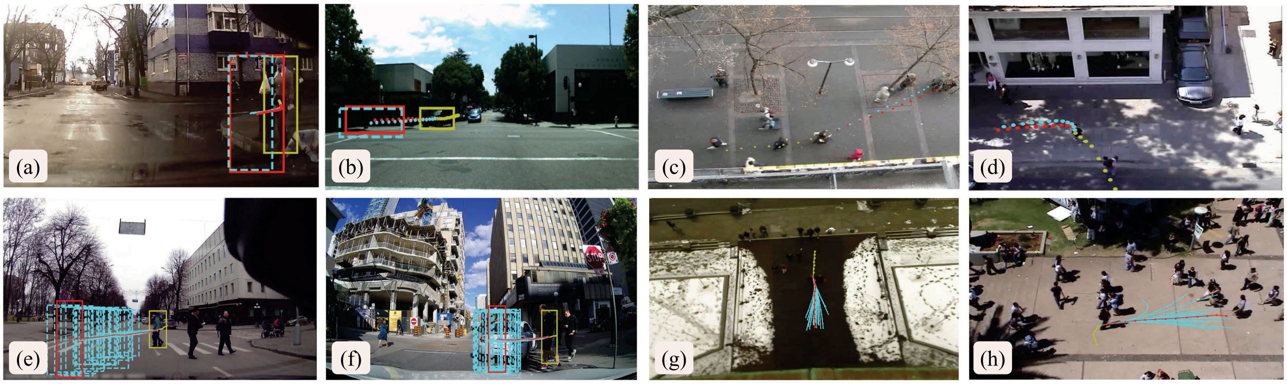

Fig. 4. Qualitative results. The yellow color indicates the observed trajectory, the red color indicates the ground truth future trajectory, and the cyan color indicates the predictions from our SGDNet-ED model (better in color).

TABLE VI
RESULTS ON NUSCENES WITHOUT (FIRST ROW) AND WITH (SECOND ROW) MAP AND INTERACTION (MI)

Method	NuScenes (K=5)	NuScenes (K=10)
	ADE / FDE (6s) ↓	ADE / FDE (6s) ↓
SGDNet-ED	2.1 / 4.65	1.67 / 3.53
SGDNet-ED (MI)	**1.85** / **3.87**	**1.32** / **2.50**

(g) and (h) show the scene in ETH and UCY. Most of the predictions are close to the ground truth trajectory and bounding box, indicating the stability of our stochastic prediction model.

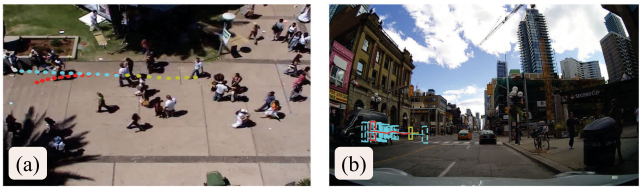

Fig. 5. Failure cases of deterministic (a) and stochastic (b) predictions. The yellow color indicates the observed trajectory, the red color indicates the ground truth future trajectory, and cyan color indicates the predictions from our SGNet-ED model (better in color).

G. Failure Cases

Fig. 5 studies two failure cases of our model. In case (a), the marked person is descending the stairs in the near future, but our model fails to predict the correct trajectory due to the lack of context information and interaction between pedestrians. In case (b), most of the stochastic predictions show the pedestrian walking along the road, but one prediction indicates the possibility that the pedestrian may cross the road along the zebra crossing. However, this error may be a "blessing in disguise," for example helping an autonomous driving system to prepare for a potential future risk.

V. CONCLUSION

We presented SGNet to tackle the trajectory prediction problem. Unlike most existing goal-driven models that only estimates final destination or distant goals, SGNet predicts both long- and short-term goals and explicitly incorporates these estimated future states. We showed that these goals can help to predict the trajectory. We conducted extensive experiments on three first-person and three bird's eye view datasets to evaluate the proposed approach. Experimental results showed the effectiveness and robustness of our models against the state-of-the-art methods.

References

[1] 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.

[2] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine, "Precog: Prediction conditioned on goals in visual multi-agent settings," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 2821–2830.

[3] K. Mangalam *et al.*, "It is not the journey but the destination: Endpoint conditioned trajectory prediction," in *Proc. Euro Conf. Comput. Vis.*, 2020, pp. 759–776.

[4] H. Zhao *et al.*, "TNT: Target-driven trajectory prediction," *CoRL*, vol. 155, pp. 895–904, 2020.

[5] Y. Yao, E. Atkins, M. Johnson-Roberson, R. Vasudevan, and X. Du, "Bitrap: Bi-directional pedestrian trajectory prediction with multi-modal goal estimation," *Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1463–1470, 2021.

[6] E. A. Locke and G. P. Latham, *A Theory of Goal Setting & Task Performance*. Hoboken, NJ, USA: Prentice-Hall, 1990.

[7] M. E. Tubbs and S. E. Ekeberg, "The role of intentions in work motivation: Implications for goal-setting theory and research," *Acad. Manage. Rev.*, vol. 16, no. 1, pp. 180–199, 1991.

[8] A. Bhattacharyya, M. Fritz, and B. Schiele, "Long-term on-board prediction of people in traffic scenes under uncertainty," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4194–4202.

[9] T. Yagi, K. Mangalam, R. Yonetani, and Y. Sato, "Future person localization in first-person videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7593–7602.

[10] Y. Yao, M. Xu, C. Choi, D. J. Crandall, E. M. Atkins, and B. Dariush, "Egocentric vision-based future vehicle localization for intelligent driving assistance systems," in *Proc. Int. Conf. Robot. Autom.*, 2019, pp. 9711–9717.

[11] O. Makansi, O. Cicek, K. Buchicchio, and T. Brox, "Multimodal future localization and emergence prediction for objects in egocentric view with a reachability prior," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4354–4363.

[12] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. 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. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2255–2264.

[14] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, "Sophie: An attentive gan for predicting paths compliant to social and physical constraints," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1349–1358.

[15] 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 *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 336–345.

[16] B. Ivanovic and M. Pavone, "The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 2375–2384.

[17] Y. C. Tang and R. Salakhutdinov, "Multiple futures prediction," *Advances in Neural Inf. Process. Syst.*, vol. 32, pp. 15424–15434, 2019.

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

[19] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron : Multi-agent generative trajectory forecasting with heterogeneous data for control," in *Proc. Euro. Conf. Comput. Vis.*, 2020, pp. 683–700.

[20] T. Buhet, E. Wirbel, and X. Perrotton, "Plop: Probabilistic polynomial objects trajectory planning for autonomous driving," in *Conf. Robot Learning*, vol. 155, pp. 329–338, 2020.

[21] M.-F. Chang *et al.*, "Argoverse: 3D tracking and forecasting with rich maps," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8748–8757.

[22] J. Liang, L. Jiang, and A. Hauptmann, "Simaug: Learning robust representations from simulation for trajectory prediction," in *Proc. Euro. Conf. Comput. Vis.*, 2020, pp. 275–292.

[23] M. Zhou *et al.*, "Smarts: Scalable multi-agent reinforcement learning training school for autonomous driving," in *Conf. Robot Learning*, vol. 155, pp. 264–285, 2020.

[24] P. Zhang, W. Ouyang, P. Zhang, J. Xue, and N. Zheng, "Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12085–12094.

[25] N. Deo and M. M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms," in *Proc. IEEE Intell. Vehicles Symp.*, 2018, pp. 1179–1184.

[26] B. Brito, H. Zhu, W. Pan, and J. Alonso-Mora, "Social-vrnn: One-shot multi-modal trajectory prediction for interacting pedestrians," in *Conf. Robot Learning*, vol. 155, pp. 862–872, 2020.

[27] X. Li, G. Rosman, I. Gilitschenski, J. DeCastro, C.-I. Vasile, and S. K. Daniela Rus, "Differentiable logic layer for rule guided trajectory prediction," in *Conf. Robot Learning*, vol. 155, pp. 2178–2194, 2020.

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

[29] C. Choi, J. H. Choi, J. Li, and S. Malla, "Shared cross-modal trajectory prediction for autonomous driving," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 244–253.

[30] N. Rhinehart, K. M. Kitani, and P. Vernaza, "R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting," in *Proc. Euro. Conf. Comput. Vis.*, 2018, pp. 772–788.

[31] C. Choi, S. Malla, A. Patil, and J. H. Choi, "Drogon: A. trajectory prediction model based on intention-conditioned behavior reasoning," in *Conf. Robot Learning*, vol. 155, pp. 49–63, 2020.

[32] M. Shah *et al.*, "Liranet: End-to-end trajectory prediction using spatio-temporal radar fusion," in *Conf. Robot Learning*, vol. 155, pp. 31–48, 2020.

[33] W. Zeng *et al.*, "End-to-end interpretable neural motion planner," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8660–8669.

[34] N. Deo and M. M. Trivedi, "Trajectory forecasts in unknown environments conditioned on grid-based plans," 2020, *arXiv:2001.00735*.

[35] M. Xu, M. Gao, Y.-T. Chen, L. S. Davis, and D. J. Crandall, "Temporal recurrent networks for online action detection," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 5532–5541.

[36] Y. Yao, M. Xu, Y. Wang, D. J. Crandall, and E. M. Atkins, "Unsupervised traffic accident detection in first-person videos," in *Proc. Int. Conf. Intell. Robots Syst.*, 2019, pp. 273–280.

[37] A. Rasouli, I. Kotseruba, T. Kunic, and J. K. Tsotsos, "PIE: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 6262–6271.

[38] T. Zhao *et al.*, "Multi-agent tensor fusion for contextual trajectory prediction," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12126–12134.

[39] H. Bi, R. Zhang, T. Mao, Z. Deng, and Z. Wang, "How can i see my future? fvtraj: Using first-person view for pedestrian trajectory prediction," in *Proc. Euro. Conf. Comput. Vis.*, 2020, pp. 576–593.

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

[41] I. Kotseruba, A. Rasouli, and J. K. Tsotsos, "Joint attention in autonomous driving (JAAD)," in *IEEE Intell. Vehicles Symp.*, 2017, pp. 264–269.

[42] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. int. Conf. Image Process.*, 2016, pp. 3464–3468.

[43] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Proc. Int. Conf. Comput. Vis.*, 2009, pp. 261–268.

[44] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," in *Comput. Graph. Forum*, vol. 26, no. 3, pp. 655–664, 2007.

[45] H. Caesar *et al.*, "nuscenes: A. multimodal dataset for autonomous driving," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11621–11631.

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, *arXiv:1412.6980*.

[47] J. Li, H. Ma, and M. Tomizuka, "Conditional generative neural system for probabilistic trajectory prediction," in *Proc. Int. Conf. Intell. Robots Syst.*, 2019, pp. 6150–6156.

[48] C. Tao, Q. Jiang, L. Duan, and P. Luo, "Dynamic and static context-aware lstm for multi-agent motion prediction," in *Proc. Euro. Conf. Comput. Vis.*, 2020, pp. 547–563.

[49] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff, "Covernet: Multimodal behavior prediction using trajectory sets," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 14074–14083.

[50] "NuScenes prediction challenge at the 6th AI Driving Olympics," 2021. Accessed: Feb. 22, 2022. [Online]. Available: https://www.nuscenes.org/prediction