Improving Multi-Agent Trajectory Prediction Using Traffic States on Interactive Driving Scenarios

Chalavadi Vishnu, Vineel Abhinav, Debaditya Roy, C. Krishna Mohan, *Senior Member, IEEE*, and Ch. Sobhan Babu

***Abstract*—Predicting trajectories of multiple agents in interactive driving scenarios such as intersections, and roundabouts are challenging due to the high density of agents, varying speeds, and environmental obstacles. Existing approaches use relative distance and semantic maps of intersections to improve trajectory prediction. However, drivers base their driving decision on the overall traffic state of the intersection and the surrounding vehicles. So, we propose to use traffic states that denote changing spatio-temporal interaction between neighboring vehicles, to improve trajectory prediction. An example of a traffic state is a clump state which denotes that the vehicles are moving close to each other, i.e., congestion is forming. We develop three prediction models with different architectures, namely, Transformer-based (TS-Transformer), Generative Adversarial Network-based (TS-GAN), and Conditional Variational Autoencoder-based (TS-CVAE). We show that traffic state-based models consistently predict better future trajectories than the vanilla models. TS-Transformer produces state-of-the-art results on two challenging interactive trajectory prediction datasets, namely, Eye-on-Traffic (EOT), and INTERACTION. Our qualitative analysis shows that traffic state-based models have better aligned trajectories to the ground truth.**

***Index Terms*—Trajectory prediction, generative adversarial networks, conditional variational autoencoder, transformers.**

I. Introduction

PREDICTING the future behavior of multiple agents (vehicles, pedestrians, cyclists) accurately and reliably is necessary for developing safe and reliable autonomous driving systems. The movement of traffic around agents plays an important role in deciding what trajectories they will take. Interaction between agents is represented using social pooling [1], global scene information [2], and spatio-temporal features of all agents [3], [4], [5]. These methods do not capture the changing relationship (distance) between the agents in dense lane-less traffic particularly at intersections.

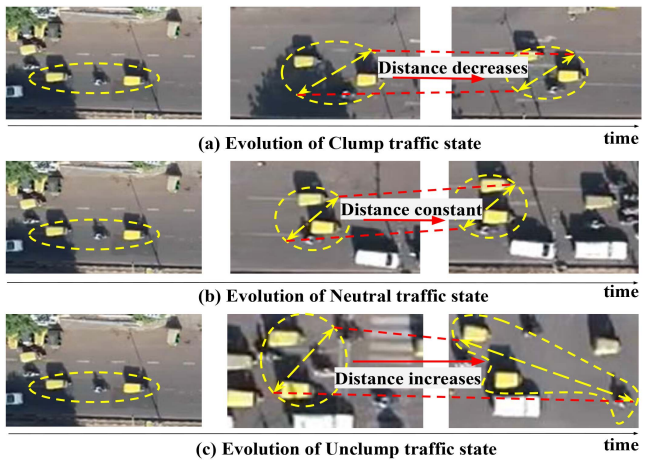


Fig. 1. Traffic agents showing different spatio-temporal behavior based on traffic state starting from similar situations. (a) Clump: agents move in closer over time. (b) Neutral: distance between agents remains same over time. (c) Unclump: agents move further apart over time.

We propose dynamic traffic states for multi-agent trajectory prediction in interactive driving scenarios such as roundabouts, un-signalized and signalized intersections, and merging and lane changing. Dynamic traffic states are defined based on the relative distance between a group of agents [6]. Traffic states depict changing interactions between vehicles over time and represent whether traffic congestion is forming, the congestion is dispersing, or traffic is flowing regularly. The traffic state variable informs the trajectory prediction framework whether future trajectories of nearby agents will be closer, further or the same distance as the past trajectories as illustrated for two agents with dotted yellow circles Fig. 1. Thus, the trajectory prediction model can predict different future trajectories for similar past trajectories using traffic state (as shown by dotted red lines in Fig. 1).

In this work, we develop three different trajectory prediction models to show the effectiveness of traffic states. These three models depict state-of-the-art architectures in multi-agent trajectory prediction. Our first model is a traffic state based Generative Adversarial Network (TS-GAN) which has been used for multimodal trajectory prediction in [7]. TS-GAN introduces traffic state into the pooling of agent trajectories to denote the changing interaction between agents through time. Our second

Manuscript received 23 October 2022; accepted 25 February 2023. Date of publication 17 March 2023; date of current version 28 March 2023. This letter was recommended for publication by Associate Editor C. Feng and Editor C. Cadena Lerma upon evaluation of the reviewers' comments. This work was supported in part by the DST-India as the part of the Design and Development of Machine Learning Algorithms for Traffic Analytics and part by the National Research Foundation Singapore through its AI Singapore Program under Grant AISG-RP-2019-010. *(Corresponding author: Chalavadi Vishnu.)*

Chalavadi Vishnu, Vineel Abhinav, C. Krishna Mohan, and Ch. Sobhan Babu are with the Department of Computer Science and Engineering, Indian Institute of Technology Hyderabad, Telangana 502285, India (e-mail: cs16m18p000001@iith.ac.in; cs20mtech01007@iith.ac.in; ckm@cse.iith.ac.in; sobhan@cse.iith.ac.in).

Debaditya Roy is with the Institute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A*STAR), Singapore 138632 (e-mail: debadityaroy5555@gmail.com).

Digital Object Identifier 10.1109/LRA.2023.3258685

model is a Conditional Variational Auto Encoder (CVAE) that are shown to be effective for conditional generation of future trajectories of multiple agents simultaneously in [2], [8]. We add traffic state as conditional supervision along with the past trajectories. The conditional latent distribution in the CVAE learns agent movements with respect to each traffic state that helps in more accurate generation of future trajectories.

Our third model is a transformer model that are demonstrated to be quite effective at trajectory prediction than LSTM or GRU based models [9]. Our transformer model uses traffic state in both the encoder along with past trajectories and the decoder while generating future trajectories. The multi-headed attention layer in the encoder combines the traffic states of various agents to produce a summary of the entire neighborhood of agents. The decoder uses the traffic state in its attention layer to generate appropriate trajectories for each agent.

We evaluate all three networks on two interactive and challenging datasets - EyeonTraffic (EoT) [10] that covers intersections with lane-less traffic and INTERACTION [11] that covers lane-based intersections and roundabouts. Traffic state based networks predict better future trajectories for both lane-less and lane-based traffic scenarios compared to non traffic-state based networks. On the INTERACTION dataset, traffic state network outperforms state-of-the-art approaches. Our contributions are as follows:

- We introduce traffic states to predict traffic specific future trajectories for multiple agents in interactive driving scenarios. We are the first to evaluate trajectory prediction in both lane-less (EoT dataset) and lane-based traffic (INTERACTION).
- Traffic state improves the performance of different trajectory prediction models, such as, TS-GAN, TS-CVAE, and TS-Transformer models. TS-Transformer outperforms state-of-the-art methods on the INTERACTION dataset.

II. Related Work

Recent trajectory prediction models predict multiple future trajectories to deal with variability in person and vehicle behavior. DESIRE [12] and SocialGAN [7] introduces conditional variational autoencoder (CVAE) and generative adversarial networks (GANs), respectively to generate multiple future trajectories. Another GAN-based framework, i.e, SoPhie [13] considers past trajectory history of all agents in a scene to effectively learn interactions to predict the future. However, the above mentioned approaches do not consider the destination of the agents which plays an important role in predicting their trajectories. To overcome this limitation, Zou et al. [14] utilize reinforcement learning to imitate agent history including the destination, also known as imitation learning with the help of GANs.

Vehicle Trajectory Prediction - Some works on trajectory prediction consider only the prior history of the road user agents [15], [16]. Due to this, the prediction errors are generally high in road scenarios with dense traffic. To overcome this limitation, additional context [17], [18] such as, local & global vehicle interactions needs to be considered to improve vehicle tracking and trajectory prediction. Recently, dynamic & convolutional graphs [19], [20], [21] have also been used as a cue for additional context, along with natural scenes [22]. Bayesian networks [23] were also explored to construct a Gaussian Mixture Model (GMM) for multiple trajectories combined with a Hidden Markov Model (HMM) for prediction of vehicle trajectories.

Multi-modal Trajectory Prediction - In [24], the authors account for the inherent multi-modal nature of vehicle trajectory prediction i.e., self-driving vehicles attached with various types of sensors which accounts for vehicles' past history that can lead to many different plausible future trajectories. The cross-modal embedding framework helps in learning a latent distribution of complementary features. The shared latent distribution is further optimized over various inputs in the objective functions. To improve interaction dynamics between agents, sparse graph convolution networks [25], [26], [27] can efficiently combine the direction of traffic, especially pedestrians in road traffic scenarios. Taking these snapshots of the traffic states is measured in traffic volume, density, or speed [6], [28], [29], [30] to learn global context information effectively. Generally, object detection [31], [32] and aggregation methods are considered to extract traffic state information. But, these methods are prone to high mis-classification errors when the traffic is dense and complex with various local & global interactions between road agents.

III. Proposed Method

In this section, we present our method for interactive multi-agent trajectory prediction using a traffic state variable.

A. Problem Formulation

The problem of multi-agent trajectory forecasting is to predict the future trajectories of all N road agents present in the scene by considering the traffic state. For the N agents, we are given input trajectories $X = X_1, X_2, \ldots, X_N$ and we need to predict the future trajectories $\hat{Y} = \hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_N$ of all the agents. For every agent i, the input trajectory comprises of its location for the past t_O time-steps given by $X_i = \{(x_i^1, y_i^1), \ldots, (x_i^{t_O}, y_i^{t_O})\}$. We need to predict the location of agent i for t_F time-steps in the future, i.e., $\hat{Y}_i = \{(\hat{x}_i^{t_O+1}, \hat{y}_i^{t_O+1}), \ldots, (\hat{x}_i^{t_O+t_F}, \hat{y}_i^{t_O+t_F})\}$. The actual ground truth future trajectory is given by $Y_i = \{(x_i^{t_O+1}, y_i^{t_O+1}), \ldots, (x_i^{t_O+t_F}, y_i^{t_O+t_F})\}$.

B. Traffic State

The traffic state variable represents the flow of traffic in a lane which informs the trajectory prediction model on what the future will be like. Multiple directions of traffic converge at roundabouts and intersections and it is difficult to predict the future only based on past trajectories. Traffic state provides spatio-temporal information about neighboring vehicles and what their movement will be in the future. Possible traffic states are *clump*, *unclump*, and *neutral* [6]. Clump indicates that the relative distance between vehicles decreases over a period of t seconds, unclump indicates that the relative distance between vehicles

increases over a period of t seconds, and neutral indicates that vehicles maintain the same relative distance over t seconds [6]. The group of vehicles which exhibit unclump, clump, or neutral traffic behavior is all allocated to that particular traffic state [6]. This means that trajectories of all vehicles are provided with a 3-dimensional one-hot vector S where 001 represents clump, 010 represents unclump, and 100 represents neutral. We now present our three trajectory prediction architectures.

C. Traffic State GAN

The GAN model consists of three components: an encoder-decoder LSTM which is the generator and another encoder LSTM which is the discriminator.

Generator: Each agent has an independent encoder LSTM that outputs the feature encoding of its past trajectory. Then, we encode the location of each agent at time t, (x_i^t, y_i^t) into an embedding e_i^t using embedding function ϕ_e. ϕ_e is an MLP with ReLU activation:

$$e_i^t = \phi_e(x_i^t, y_i^t). \tag{1}$$

The embedding e_i^t is of size 16 and given as input to LSTM to compute the hidden state for agent i at time t as follows

$$h_{ei}^t = LSTM_e(h_{ei}^{t-1}, e_i^t) \tag{2}$$

where $LSTM_e$ represents the encoder LSTM with a hidden layer of size 32.

We represent the interaction of agent i with all other agents by pooling their hidden states into a pooled representation P_i using a pooling module PM

$$P_i = PM(h_{e1}^{t-1}, \ldots, h_{eN}^t) \tag{3}$$

The pooled representation records interactions with neighboring agents. The traffic state variable quantifies the type of interaction between agents - coming together (clumping), going apart (unclumping), or maintaining the same distance (neutral). With traffic state, we add complementary information about the type of traffic movement shown by an entire group of interactive agents. We incorporate the traffic state S, the pooled representation P_i, and the encoder hidden state as input to the decoder as follows:

$$h_{di}^t = [\gamma_d([P_i, h_{ei}^t]), S, z] \tag{4}$$

where P_i is pooling output for agent i, S is traffic state, γ_d is an MLP with ReLU activation function, and $[\cdot, \cdot]$ represents concatenation. We sample z from a standard normal distribution to generate multiple plausible future trajectories. The reason we use S instead of S_i for agent i is that all interacting agents that interact with agent i are in the same direction of travel and hence have the same traffic state. For example, we can see this behavior when vehicles are coming to a stop at a traffic signal and all neighboring vehicles exhibit clumping behavior.

The decoder is initialized in (4). Now we predict agent-specific future trajectory $\hat{Y}_i^t$ as follows:

$$e_i^t = \phi_{ed}(x_i^{t-1}, y_i^{t-1}) \tag{5}$$

$$h_{di}^t = LSTM_d(\gamma_d(P_i, h_{di}^{t-1}), e_i^t) \tag{6}$$

$$\hat{Y}_i^t = \gamma_o(h_{di}^t) \tag{7}$$

where $LSTM_d$ is the decoder LSTM, and ϕ_{ed} and γ_o are MLPs with ReLU activation.

Discriminator: The discriminator is a separate encoder that takes real trajectory $[X_i, Y_i]$ or fake trajectory $[X_i, Y^i]$ and classifies them as real or fake. The encoder's last hidden state is used to obtain a classification score with the help of an MLP. The discriminator learns to classify unacceptable trajectories based on traffic state and interactions with neighboring agents as "fake" while the generator learns to produce more plausible trajectories. We train TS-GAN with adversarial loss [33] and variety loss [7].

D. Traffic State CVAE

Graph-based Conditional Variational AutoEncoder (CVAE) has been demonstrated to be effective for trajectory prediction [2], [8].

Agent Past Representation: We represent the entire scene using a graph with each agent denoted by a node similar to [2]. Each node is represented by an LSTM with 8 hidden dimensions that encodes the position information of the agent over the observed t_O time-steps into node history $h_{i,node}^t$. The interaction between agents is encoded using edges which are added only if the neighbors are within a certain distance of each other. Each edge is represented as an LSTM which outputs the edge history encoding for agent i as $h_{i,edge}^t$ following Structural RNN [34] Traffic state helps in understanding how many neighbors are coming close to the vehicle or going away from the vehicle. Adding traffic state helps in generating future trajectories that resemble the state of the traffic at that time. We concatenate the node history, edge history, and traffic state to obtain a node representation vector e_i for agent i as follows:

$$e_i = [h_{i,edge}^t, h_{i,node}^t, S]. \tag{8}$$

Agent Future Representation: In CVAE, the latent variable z learns the mean and variance of underlying input trajectory distribution [2] instead of a standard normal distribution as in GAN. This requires the agent's entire trajectory which includes the observed trajectory and the future trajectory. So, an agent's future trajectory Y_i is encoded using an bidirectional LSTM with 32 hidden units in encoder as node future encoding $h_{i,node}^{t+}$ as follows

$$h_{i,node}^{t+} = BiLSTM(Y_i). \tag{9}$$

The node future encoding is computed only during training. During testing, we only use the node representation to estimate the distribution of the latent variable.

Conditional Latent Distribution: We use the node representation vector and node future encoding along with the class of the agent C_i to estimate the distribution parameters of the latent variable z (16-dimensional) during training as follows:

$$f_i = \phi(e_i, h_{i,node}^{t+}), C_i) \tag{10}$$

$$z \sim q_\phi(z|X_i, Y_i; f_i) \tag{11}$$

where ϕ is a 1-dimensional CNN with 4 layers and with filter sizes {5,5,5,3} & strides {2,2,1,1}, in each layer, respectively. The distribution q_ϕ is a categorical distribution parameterized by

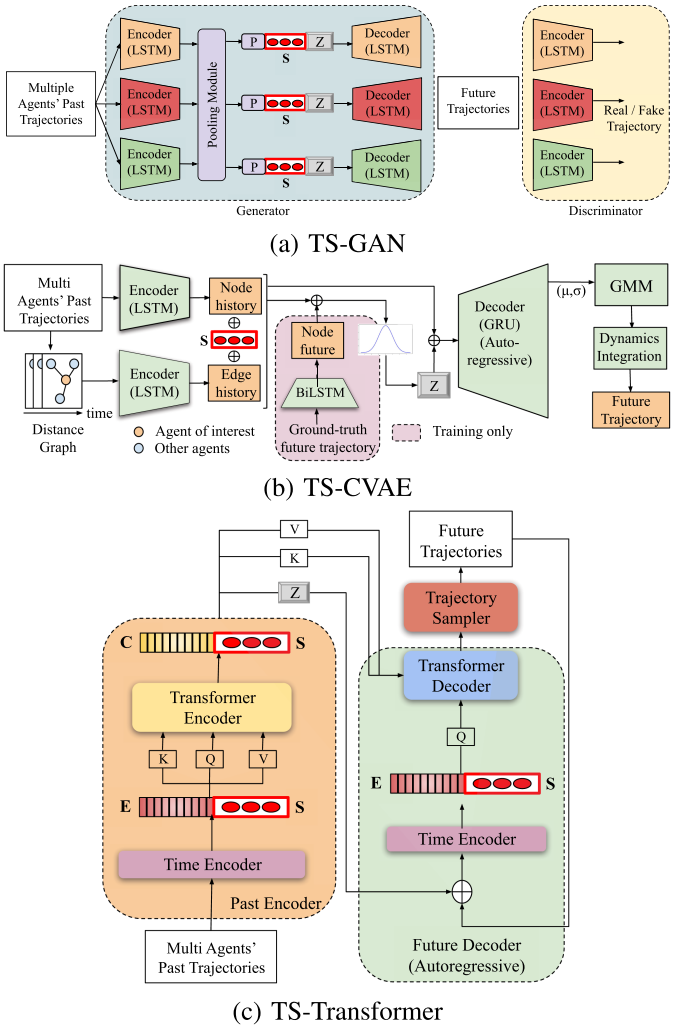


Fig. 2. Different traffic state based trajectory prediction architectures. (a) TS-GAN - traffic state S is introduced after pooling module. (b) TS-CVAE - traffic state is added to encoder output (LSTM). (c) TS-Transformer - traffic state is added to the output of both Time Encoder and Transformer Encoder. The location of traffic state insertion is empirically determined based on trajectory prediction performance in Section IV-C.

f_i and captures the patterns in the agent trajectory with respect to each traffic state.

Future Trajectory Prediction: After obtaining the latent variable, we employ a decoder which is a 128-dimensional Gated Recurrent Unit (GRU) to generate parameters of 16 components for a Gaussian Mixture Model (GMM). Then, we sample the future trajectory for agent i from the 16-component GMM as shown below:

$$\hat{Y}_i^t \sim GMM(GRU([e_i, z, S])). \tag{12}$$

The agent's dynamics are also integrated to obtain future trajectories [2]. In our work, we use linear dynamics for all types of vehicles as it is difficult to predict the exact dynamics of different vehicles in lane-less traffic. Fig. 2(b) shows the mechanism of introducing traffic states into the TS-CVAE architecture.

Training: We use the loss function described in InfoVAE [35] to learn the distribution parameters that generate z and subsequently the best future trajectory.

E. Traffic State Transformer

The third architecture that we consider is a transformer encoder-decoder which shows impressive trajectory prediction performance in [9]. Their model lacks in capturing information such as how distance changes between neighboring agents over time which is provided by traffic state. Our transformer architecture is inspired by [9] and consists of transformers acting as both encoder and decoder in a CVAE framework.

Time Encoder: To incorporate time information of every agent in a given trajectory sequence, the position encoder of the transformer is replaced with time encoder which encodes the timestamp of the agent. The time encoder outputs timestamped sequence E from observed trajectory X. We concatenate traffic state variable S with timestamped sequence E i.e., $E = [E, S]$. This helps the transformer encoder model attend specifically to the traffic state of the agent along with their trajectory information.

Past Encoder: Past encoder takes historical trajectories $E_1, \ldots, E_N$ of multiple agents as input which is fed into the time encoder to get a timestamped sequence. The timestamped sequences act as queries, keys, and values to the transformer encoder. The transformer encoder is a single attention layer with 8 attention heads that produce past feature sequence C using agent-aware attention [9].

$$Q = K = V = [E_1, E_2, \ldots, E_N], \tag{13}$$

$$\text{AgentAwareAttention}(Q, K, V) = \text{softmax}\left(\frac{A}{\sqrt{d_k}}\right) V \tag{14}$$

$$A = M \odot (Q_{self} K_{self}^T) + (1 - M) \odot (Q_{other} K_{other}^T) \tag{15}$$

$$Q_{self} = QW_{self}^Q, K_{self} = KW_{self}^K \tag{16}$$

$$Q_{other} = QW_{other}^Q, K_{other} = KW_{other}^K \tag{17}$$

where $E_1, \ldots, E_N$ denotes the past trajectories of N agents, $\odot$ denotes element-wise product and d_k is the dimension of K. W_{self}^Q, W_{self}^K and W_{other}^Q, W_{other}^K represent projection weights for the agent whose trajectory is being predicted, and other agents, respectively. These weights generate projected keys K_{self} and K_{other}, and queries Q_{self} and Q_{other}. M represents the mask that allow attention weights in A to be computed differently based on whether the i^{th} query and j^{th} key belong to same agent. The dimensions d_k of keys and queries are all set to 256 followed by feedforward layers consisting of 512 hidden dimensions. Agent-aware attention preserves identity of each agent by learning to attend past trajectories of same agent differently than the past trajectories of other agents.

We obtain past feature sequence for every agent by applying agent-aware attention to the past trajectory E of that agent.

$$C = \text{AgentAwareAttention}(Q, K, V) + [E_1, \ldots, E_N] \tag{18}$$

Past feature sequence C is concatenated with traffic state variable S i.e., $C = [C, S]$ to explicitly convey the traffic state of all the agents to the future decoder. The future decoder produces traffic state-specific future trajectories for all agents.

Future Decoder: Future decoder is autoregressive where at each time step, the generated trajectories are fed back into the

model to produce the trajectories at the next time step. We pass the concatenated encoded output $E = [E, S]$ as a query to the transformer decoder so that it attends specifically to the traffic state of the particular agent along with its trajectory information. The key and value are obtained from the output of the past encoder. Fig. 2(c) shows the architecture of the TS-Transformer and where the traffic state is incorporated.

Trajectory Sampler: After training the TS-Transformer, we sample multiple trajectories to produce diverse and plausible trajectories. The trajectory sampler generates K sets of latent codes $\{Z^{(1)}, \ldots, Z^{(K)}\}$ where each set $Z^{(k)} = z_1^{(k)}, \ldots, z_N^{(k)}\}$ comprises latent codes of 32 dimensions of all agents. We append the traffic state to the latent codes $Z^{(k)} = [z_1^{(k)}, S], \ldots, [z_N^{(k)}, S]\}$ to generate traffic state specific latent codes. The future decoder takes in the latent code to generate a multi-agent future trajectory sample $Y^{(k)}$.

Training: The TS-Transformer is trained using the trajectory sampler loss introduced in [9].

IV. Experimental Results

We have evaluated our methods on two publicly available datasets EyeonTraffic (EoT) [6], [10] and INTERACTION [11].

A. Datasets, Metrics, and Implementation

EyeonTraffic dataset [6] - consists of aerial videos of vehicles at intersections which estimate the traffic congestion state under lane-less behavior. This dataset consists diverse traffic conditions with rich vehicle to vehicle interactions. There are a total of 4,021 distinct vehicle tracks including 421 cars, 77 buses, 2,185 two-wheelers, and 973 auto-rickshaws. The traffic state annotation for each lane is provided in [6]. We sample the tracks at 2 frames per second.

INTERACTION dataset [11] - provides interactive driving scenarios at intersections and roundabouts. There are 4 different categories of interactions: roundabout (10479 vehicles), un-signalized intersection (14867 vehicles), signalized intersection (10933 vehicles), and merging and lane changing (3775 vehicles). For each recorded scenario, the dataset contains a high-definition (HD) map, recorded vehicle tracks, and recorded pedestrian tracks. We annotate the traffic states using the methodology in [6].

Evaluation Metrics. We evaluate the proposed method using the following standard error metrics, such as, *Average Displacement Error (ADE)* [2], *Final Displacement Error (FDE)* [2], *Kernel Density Estimate-based Negative Log Likelihood (KDE NLL)* [2], *minADE*$_k$ [32], *minFDE*$_k$ [32], and $KDENLL_k$ [2]. All the experiments are conducted by taking an observation length of 8 time steps (3.2 s) and a prediction horizon of 12 time steps (4.8 s) following [9]. This gives us a total time $t = 8^{\sim}s$ and we annotate each 8 s duration with clump, unclump, or neutral traffic state. We compare our method on three metrics $minADE_k$ [32], $minFDE_k$ [32], and $KDENLL_k$ where k refers to the number of samples.

Implementation Details. All three models - TS-GAN, TS-CVAE, and TS-Transformer are trained with a learning rate of 0.003 and batch size of 64 using Adam optimizer in PyTorch. TS-GAN, TS-CVAE, and TS-Transformer for 200, 150, and 300 epochs, respectively. The experiments are conducted on 2 $\times$ 12 GB NVIDIA Tesla K40c GPUs. For the EyeonTraffic dataset, we used the training and testing splits provided by the authors in [6]. For the INTERACTION dataset, train/validation split for each scene are followed as used in [11], [32].

B. Evaluating Effect of Traffic State

We compare the performance of models with traffic state and without traffic state by drawing different number of samples, i.e., $K = 1, 5, 10, 20$. In Table I, traffic state based models perform better than vanilla models on the EoT dataset when the most likely output is considered ($K = 1$). The TS-Transformer architecture performs the best among all traffic state models. Traffic state improve prediction performance across larger sample sizes also. This shows that not only the best future trajectory is improved by incorporating traffic state but also the overall quality of future trajectories. To further understand the quality of samples, we compare the generation ability of models with traffic state using kernel density estimator negative log-likelihood (KDE NLL) metric. KDE NLL is the mean NLL of the ground truth trajectory using the probability density function (PDF) of a distribution found by fitting a kernel density estimate on trajectory samples [2]. Table I also shows the $KDENLL$ metric values for all the architectures for $k = 20$ samples. We observe that both TS-CVAE and TS-Transformer have lower KDE NLL and better future trajectory generation capability.

In Table II, we do a scene-wise evaluation of INTERACTION dataset to analyze the effect of the traffic state following [32]. We sample six trajectories per agent for each architecture ($minADE_6/minFDE_6$) to compare with existing state-of-the-art approaches. We observe that traffic state scene with TS-Transformer performs the best on a majority of the scenes even outperforming state-of-the-art methods such as ITRA. These scenes comprise mainly of roundabouts and intersections that have multiple directions of traffic flow that converge onto or diverge from each other. Hence, using the traffic state to denote converging or diverging traffic greatly improves the prediction of future trajectories.

We follow the evaluation protocol on INTERACTION dataset and aggregate the scene-wise results to compare with state-of-the-art approaches in Table III. TS-Transformer significantly outperforms generative models such as ITRA [32] and TNT [30] on the $minFDE_6$ metric. We think that because the traffic state quantifies the spatio-temporal change in vehicle interactions in the future, it is able to better predict the final location of the future trajectory. Our approach also performs better in terms of $minADE_6$ Our results on both EoT and INTERACTION datasets validate the hypothesis regarding the efficacy of traffic state in highly interactive traffic.

C. Ablation Study - Best Traffic State Model

We present trajectory prediction performance of different architectures obtained by introducing traffic state in various modules of TS-GAN, TS-CVAE, and TS-Transformer. For all

TABLE I
EVALUATION ON EOT DATASET

Architecture Type	$minADE_1/minFDE_1$	$minADE_5/minFDE_5$	$minADE_{10}/minFDE_{10}$	$minADE_{20}/minFDE_{20}$	$KDENLL_{20}$
GAN	0.9688 / 1.4771	0.8366 / 1.2866	0.7652 / 1.2080	0.7598 / 1.1802	2.2217
TS-GAN	0.5315 / 0.8378	0.5174 / 0.7735	0.4632 / 0.7486	0.3590 / 0.6849	1.6965
CVAE	1.1858 / 1.9973	1.0145 / 1.7817	0.9864 / 1.6480	0.7261 / 1.6204	5.1323
TS-CVAE	0.8414 / 1.1297	0.7819 / 1.0618	0.6757 / 0.9188	0.6293 / 0.8644	4.7687
Transformer	0.7230 / 1.2734	0.6675 / 1.1046	0.5648 / 1.0581	0.5252 / 0.8548	0.7226
TS-Transformer	**0.3625 / 0.5111**	**0.2931 / 0.4786**	**0.2131 / 0.2850**	**0.1784 / 0.2461**	**0.5268**

Traffic state reduces MINADE, MINFDE, and KDE NLL for GAN, CVAE, and transformer models.

TABLE II
EVALUATING EFFECT OF TRAFFIC STATE SCENE-WISE ON INTERACTION DATASET

Scene	**Metric** ($minADE_6/minFDE_6$)						
	GAN	TS-GAN	CVAE	TS-CVAE	Transformer	TS-Transformer	ITRA [32]
CHN Merging ZS	1.4624 / 1.5229	1.3743 / 1.3872	1.2357 / 1.4015	1.2154 / 1.3462	0.4331 / 0.4647	0.1878 / **0.2133**	**0.127** / 0.356
CHN Roundabout LN	0.7644 / 0.9125	0.4783 / 0.6531	0.5434 / 0.7666	0.3811 / 0.5690	0.3517 / 0.5019	**0.1944** / **0.2438**	0.199 / 0.549
DEU Merging MT	0.6602 / 0.7016	0.5233 / 0.5939	0.4899 / 0.5201	0.2064 / **0.2516**	0.3148 / 0.3982	**0.1549** / 0.2617	0.226 / 0.678
DEU Roundabout OF	0.5098 / 0.6467	0.4476 / 0.5494	0.5179 / 0.5138	0.4321 / 0.4857	0.2408 / 0.2794	**0.1346** / **0.1741**	0.283 / 0.766
USA Intersection EP0	0.4668 / 0.4791	0.3720 / 0.4180	0.4397 / 0.5123	0.2076 / 0.2118	0.4036 / 0.4369	0.2250 / **0.2538**	**0.215** / 0.640
USA Intersection EP1	0.2334 / 0.2667	0.1344 / 0.1980	0.2204 / 0.2383	0.1734 / 0.2566	0.1947 / 0.1932	**0.1103** / **0.1120**	0.234 / 0.678
USA Intersection GL	0.2687 / 0.3721	0.1687 / 0.1721	0.1572 / 0.2368	0.1433 / 0.1697	0.1378 / 0.2224	**0.1362** / **0.1790**	0.203 / 0.611
USA Intersection MA	0.5282 / 0.5478	0.4459 / 0.5081	0.2788 / 0.4769	0.2516 / 0.3343	0.2392 / 0.2514	0.2174 / **0.2494**	**0.212** / 0.634
USA Roundabout EP	0.3434 / 0.3536	0.3237 / 0.2647	0.2597 / 0.2756	0.2192 / 0.2386	0.1793 / 0.2270	**0.1469** / **0.1816**	0.234 / 0.690
USA Roundabout FT	0.4309 / 0.5017	0.4143 / 0.4883	0.3801 / 0.4775	0.2055 / 0.2721	0.2109 / 0.2335	**0.1527** / **0.1679**	0.219 / 0.661
USA Roundabout SR	0.2312 / 0.2495	0.1746 / 0.2249	**0.1401** / 0.2812	0.1376 / 0.2366	0.1909 / 0.1818	0.1534 / **0.1537**	0.178 / 0.525

TABLE III
PERFORMANCE COMPARISON WITH STATE-OF-THE-ART METHODS ON INTERACTION DATASET

Method	$minADE_6/minFDE_6$
DESIRE [12]	0.32 / 0.88
MultiPath [16]	0.30 / 0.99
TNT [30]	0.21 / 0.67
ReCoG [27]	0.19 / 0.65
ITRA [32]	0.17 / 0.49
TS-Transformer (Ours)	**0.16 / 0.20**

For each scene, the trajectory with the smallest error is selected, independently for ADE and FDE following [32].

TABLE IV
TS-GAN ABLATION BY INCORPORATING TRAFFIC STATE AT DIFFERENT MODULES

Traffic State @			$minADE_1/$	$minADE_{20}/$
Pooling Module	Encoder	Decoder	$minFDE_1$	$minFDE_{20}$
✗	✗	✗	0.9748 / 1.3665	0.6286 / 0.9843
✓	✗	✗	0.8949 / 1.1163	0.5601 / 0.8279
✗	✗	✓	0.7640 / 1.2726	0.4362 / 0.7852
✓	✗	✓	**0.5656 / 0.8915**	**0.4178 / 0.7465**
✓	✓	✗	0.6239 / 0.9140	0.5356 / 0.7921

Results on EOT dataset.

TABLE V
TS-CVAE ABLATION BY INCORPORATING TRAFFIC STATE AT DIFFERENT MODULES

Traffic State @				$minADE_1/$	$minADE_{20}/$
Dynamics integration	Future Encoding	Encoder	Decoder	$minFDE_1$	$minFDE_{20}$
✓	✓	✗	✗	0.8693 / 1.2376	0.6532 / 0.8961
✗	✓	✗	✗	1.2076 / 1.3107	0.7977 / 0.9155
✓	✗	✗	✗	1.0482 / 1.3611	0.7229 / 0.9256
✓	✓	✓	✗	0.9538 / 1.4020	0.6785 / 0.8754
✗	✓	✓	✗	1.2294 / 1.4499	0.8409 / 0.8965
✓	✗	✓	✗	1.0196 / 1.2988	0.6462 / 0.9229
✓	✓	✗	✓	**0.8740 / 1.1313**	**0.6412 / 0.8712**
✗	✓	✗	✓	0.8846 / 1.4716	0.6477 / 0.8818
✓	✗	✗	✓	0.9671 / 1.4171	0.7062 / 0.9306
✓	✓	✓	✓	0.8751 / 1.2887	0.6983 / 0.9159
✗	✓	✓	✓	0.9044 / 1.2493	0.6538 / 0.8973
✓	✗	✓	✓	0.8358 / 1.3972	0.781 / 0.9799

Results on EOT dataset.

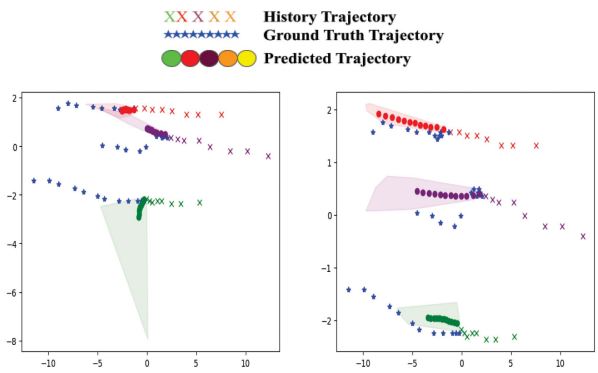
TABLE VI
TS-TRANSFORMER ABLATION BY INCORPORATING TRAFFIC STATE AT DIFFERENT MODULES

Traffic State @			$minADE_1/$	$minADE_{20}/$
Past Encoder	Traj Sampler	Future Decoder	$minFDE_1$	$minFDE_{20}$
✗	✗	✗	0.9929 / 1.2451	0.4013 / 0.4551
✓	✗	✗	0.8818 / 1.1813	0.2205 / 0.3496
✓	✓	✗	0.6001 / 1.0370	0.2506 / 0.2602
✓	✗	✓	0.6727 / 1.4552	0.3224 / 0.3414
✓	✓	✓	**0.4918 / 0.5762**	**0.2064 / 0.2570**

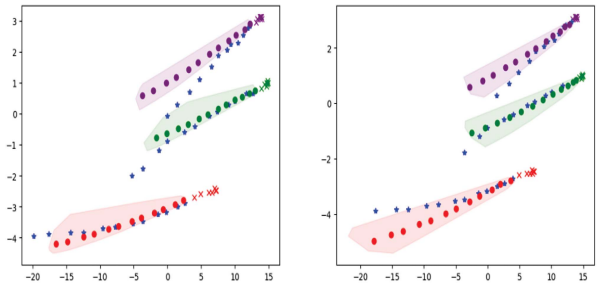
Results on EOT dataset.

comparisons, we use most likely output $minADE_1/minFDE_1$ and mean of 20 samples $minADE_{20}/minFDE_{20}$. For TS-GAN (Table IV), we observe the traffic state at the pooling module and the decoder yields the least error. So, traffic state contributes the most when it is used both during pooling past trajectory information from multiple agents and also during future trajectory prediction. For TS-CVAE (Table V), injecting traffic state into the decoder, dynamics integration, and future encoding (node future encoding) individually helps in drastically reducing the errors. Dynamics integration accounts for agent dynamics which are dependent on traffic state so it makes sense that traffic state contributes to better integration of the dynamics of multiple agents. Traffic state influences the future movement of vehicles and hence it leads to better trajectories when integrated with the future conditioned prediction module (Y-Future) of CVAE. Hence, applying traffic state at both dynamics integrator and Y-future leads to the best trajectories.

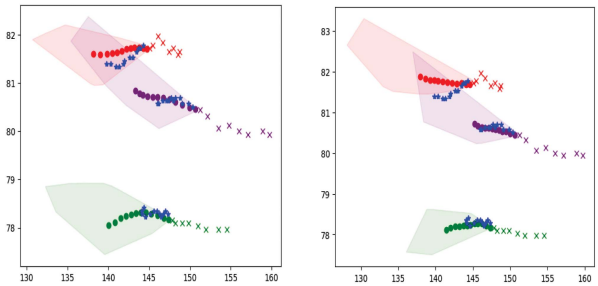
In the case of TS-Transformer (Table VI), traffic state produces better samples when introduced at all the modules, past encoder, trajectory sampler, and future decoder. In both past encoder and future decoder, traffic state and its interaction



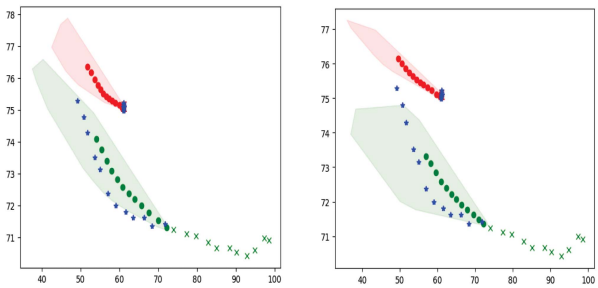
(a) **Left:** GAN vs. **Right:** TS-GAN



(b) **Left:** CVAE vs. **Right:** TS-CVAE



(c) **Left:** Transformer vs. **Right:** TS-Transformer



(d) **Left:** Transformer vs **Right:** TS-Transformer.

Fig. 3. Traffic State makes. (a) GAN. (b) CVAE. (c) Transformer predict more accurate mean trajectories that are closer to the ground truth. Lightly shaded areas represent the convex hull of 20 sampled trajectories. TS-based networks have tighter hulls compared to the non-TS networks which result in lower error ($minADE_k$) in prediction. (d) Failure case - Predictions are not closer to ground truth when traffic state is incorporated.

with agent trajectories computed through self-attention lead to improvement in trajectory prediction. In the trajectory sampler, the traffic state is used as a conditioning input to produce more plausible trajectories. Interaction with traffic state and conditioning based on traffic state contribute to better prediction.

The findings of Tables IV–VI provide the best possible architectures of TS-GAN, TS-CVAE, and TS-Transformer, respectively, as presented in Section III.

D. Qualitative Analysis of Future Trajectories

We compare the predicted trajectories of TS-GAN, TS-CVAE, and TS-Transformer against their vanilla counterparts. In Fig. 3, we sample 20 future trajectories for each agent and plot the convex hull. We observe tighter convex hulls with traffic state which means that traffic state-based trajectories are more aligned to the ground truth trajectories. This explains the improvement in $minADE_{20}$ scores of traffic state-based models over their vanilla counterparts as shown in Tables I and II. We also plot the mean of the 20 samples and observe the mean trajectory is much closer to the ground truth for TS-GAN, TS-CVAE, and TS-Transformer. This explains the improvement in $minADE_1$ scores of traffic state-based architectures.

We also present a case where traffic state is not able to improve future trajectory prediction. As observed in Fig. 3(d), predictions are not closer to ground truth when traffic state is incorporated. Perhaps this is due to the traffic state transition between the observed and predicted trajectories. However, for almost all the cases, traffic state improves prediction which is reflected in the overall prediction performance across EoT and INTERACTION datasets.

V. Conclusion

We propose a way to improve vehicle trajectory prediction in interactive scenarios using traffic states. We present three models - TS-GAN, TS-CVAE, and TS-Transformer to show how traffic state affects trajectory prediction. We evaluate our models on complex interactive datasets, namely, EoT and INTERACTION. We observe that traffic state consistently improves the performance of GAN, CVAE, and Transformer on both datasets. TS-Transformer significantly outperforms state-of-the-art approaches which show that traffic state can effectively predict the interactive vehicle trajectories. We also show examples of future trajectories being more aligned with ground truth due to the presence of traffic state. In the future, we plan to infer traffic states from the groups of trajectories and predict future trajectories simultaneously in order to remove the manual labeling process.

References

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

[2] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++ : Dynamically-feasible trajectory forecasting with heterogeneous data," in *Proc. Eur. Conf. Comput. Vis.*2020, pp. 683–700.

[3] Y. Zhu, D. Ren, M. Fan, D. Qian, X. Li, and H. Xia, "Robust trajectory forecasting for multiple intelligent agents in dynamic scene," 2020, *arXiv:2005.13133*.

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

[5] K. Mangalam, Y. An, H. Girase, and J. Malik, "From goals, waypoints & paths to long term human trajectory forecasting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 15233–15242.

[6] D. Roy, K. N. Kumar, and C. K. Mohan, "Defining traffic states using spatio-temporal traffic graphs," in *Proc. 23rd Int. Conf. Intell. Transp. Syst.*, 2020, pp. 1–6.
[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. Vis. Pattern Recognit.*, 2018, pp. 2255–2264.
[8] B. Ivanovic and M. Pavone, "The Trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2375–2384.
[9] Y. Yuan, X. Weng, Y. Ou, and K. M. Kitani, "AgentFormer: Agent-aware transformers for socio-temporal multi-agent forecasting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9813–9823.
[10] D. Roy, T. Ishizaka, C. K. Mohan, and A. Fukuda, "Detection of collision-prone vehicle behavior at intersections using siamese interaction LSTM," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 4, pp. 3137–3147, Apr. 2022.
[11] W. Zhan et al., "Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps," 2019, *arXiv:1910.03088*.
[12] 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. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 336–345.
[13] 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. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1349–1358.
[14] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, pp. 4565–4573, 2016.
[15] 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. Vis. pattern Recognit.*, 2016, pp. 961–971.
[16] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "MultiPath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," 2019, *arXiv:1910.05449*.
[17] B. T. Morris and M. M. Trivedi, "Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2287–2301, Nov. 2011.
[18] H. Song, H. Liang, H. Li, Z. Dai, and X. Yun, "Vision-based vehicle detection and counting system using deep learning in highway scenes," *Eur. Transport Res. Rev.*, vol. 11, no. 1, pp. 1–16, 2019.
[19] Y. Lu, W. Wang, X. Hu, P. Xu, S. Zhou, and M. Cai, "Vehicle trajectory prediction in connected environments via heterogeneous context-aware graph convolutional networks," *IEEE Trans. Intell. Transp. Syst.*, early access, May 24, 2022, doi: 10.1109/TITS.2022.3173944 .
[20] Z. Sheng, Y. Xu, S. Xue, and D. Li, "Graph-based spatial-temporal convolutional network for vehicle trajectory prediction in autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17654–17665, Oct. 2022.
[21] J. An, W. Liu, Q. Liu, L. Guo, P. Ren, and T. Li, "DGInet: Dynamic graph and interaction-aware convolutional network for vehicle trajectory prediction," *Neural Netw.*, vol. 151, pp. 336–348, 2022.
[22] M. Bahari et al., "Vehicle trajectory prediction works, but not everywhere," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 17123–17133.
[23] Y. Jiang, B. Zhu, S. Yang, J. Zhao, and W. Deng, "Vehicle trajectory prediction considering driver uncertainty and vehicle dynamics based on dynamic bayesian network," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 53, no. 2, pp. 689–703, Feb. 2023.
[24] C. Choi, J. H. Choi, J. Li, and S. Malla, "Shared cross-modal trajectory prediction for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 244–253.
[25] L. Shi et al., "SGCN: Sparse graph convolution network for pedestrian trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8994–9003.
[26] V. Kosaraju, A. Sadeghian, R. Martín-Martín, I. Reid, S. H. Rezatofighi, and S. Savarese, "Social-BiGAT: Multimodal trajectory forecasting using bicycle-GAN and graph attention networks," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 137–146.
[27] X. Mo, Y. Xing, and C. Lv, "ReCoG: A deep learning framework with heterogeneous graph for interaction-aware trajectory prediction," 2020, *arXiv:2012.05032*.
[28] P. Chakraborty, Y. O. Adu-Gyamfi, S. Poddar, V. Ahsani, A. Sharma, and S. Sarkar, "Traffic congestion detection from camera images using deep convolution neural networks," *Transp. Res. Rec.*, vol. 2672, no. 45, pp. 222–231, 2018.
[29] S. Zhang, G. Wu, J. P. Costeira, and J. M. Moura, "Understanding traffic density from large-scale web camera data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5898–5907.
[30] H. Zhao et al., "TNT: Target-driven trajectory prediction," in *Proc. Conf. Robot Learn.*, 2021, pp. 895–904.
[31] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149,Jun. 2017.
[32] A. Scibior, V. Lioutas, D. Reda, P. Bateni, and F. Wood, "Imagining the road ahead: Multi-agent trajectory prediction via differentiable simulation," in *Proc. 2021 IEEE Int. Intell. Transp. Syst. Conf.*, 2021, pp. 720–725 .
[33] I. Goodfellow et al., "Generative adversarial nets," in *Proc 27th Int. Conf. Adv. Neural Inf. Process. Syst.*, 2014, vol. 27, pp. 2672–2680.
[34] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-RNN: Deep learning on spatio-temporal graphs," in *Proc. ieee Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5308–5317.
[35] S. Zhao, J. Song, and S. Ermon, "InfoVAE: Balancing learning and inference in variational autoencoders," in *Proc. aaai Conf. Artif. Intell.*, 2019, vol. 33, pp. 5885–5892.