Congestion-aware Multi-agent Trajectory Prediction for Collision Avoidance

Xu Xie[1] Chi Zhang[1] Yixin Zhu[1] Ying Nian Wu[1] Song-Chun Zhu[1]

Abstract— **Predicting agents' future trajectories plays a crucial role in modern AI systems, yet it is challenging due to intricate interactions exhibited in multi-agent systems, especially when it comes to collision avoidance. To address this challenge, we propose to learn congestion patterns as contextual cues explicitly and devise a novel "Sense–Learn–Reason–Predict" framework by exploiting advantages of three different doctrines of thought, which yields the following desirable benefits: (i) Representing congestion as contextual cues via latent factors subsumes the concept of social force commonly used in physics-based approaches and implicitly encodes the distance as a cost, similar to the way a planning-based method models the environment. (ii) By decomposing the learning phases into two stages, a "student" can learn contextual cues from a "teacher" while generating collision-free trajectories. To make the framework computationally tractable, we formulate it as an optimization problem and derive an upper bound by leveraging the variational parametrization. In experiments, we demonstrate that the proposed model is able to generate collision-free trajectory predictions in a synthetic dataset designed for collision avoidance evaluation and remains competitive on the commonly used NGSIM US-101 highway dataset. Source code and dataset tools can be accessed via Github.**

I. Introduction

Since its inception, perceiving [1] and understanding [2] motions has become a key indicator for an intelligent system to interact with other agents in the environment felicitously. Unlike other topics (*e.g.*, action understanding, or activity analysis), trajectory prediction is unique and proves to be challenging as it requires inference about *multiple* agents in the *future* yet to be observed. In literature, trajectory prediction can be roughly categorized into three directions [3].

Modern modeling approaches first adopt a **physics-based** fashion [4] in a "Sense–Predict" framework [3] by directly forward simulating pre-defined and explicit dynamic models based on Newton's laws of motion. Although physics-based cues are robust prior knowledge, this family of models tends to be too brittle to handle noisy real-world data, especially in multi-agent scenarios where different agents may possess various types of dynamic models. Recent multi-model approaches [5] attempt to alleviate these difficulties.

In parallel, **pattern-based** approaches [6] tackle the trajectory prediction problem by learning different function approximators directly from data, following a "Sense–Learn–Predict" paradigm [3]. The essence of this stream of work is to leverage the power of data to provide a data-driven account of the solution, which has received increasing attention over the past few years due to readily available large datasets. However, such methods naturally suffer from interpretability issues and tend to overfit with a large space of parameters.

By taking a teleological stand and assuming rational agents, **planning-based** approaches [7] model the trajectory prediction problem by minimizing various costs, either by forward planning or inverse optimal control, following a "Sense–Reason–Act" principle [3]. However, such a *normative* perspective, which models what agents *ought* to do, may differ from real-world scenarios as the decision-making process often deviates from extreme rationality [8, 9].

Although different doctrines of thought have mostly been developing independently in literature, we argue that they do not conflict with each other and seek to answer how we can possibly fuse them and take advantage of these approaches to construct a new "Sense–Learn–Reason–Predict" framework. To give a desirable solution to this question, in this work, we start by answering the following three questions: (i) By reducing the parameter space, can a proper intermediate representation help to inject a better inductive bias for **pattern-based** approaches? Can such a representation be more generic and easy to, either explicitly or implicitly, incorporate the rational agent assumption in **planning-based** approaches and the physical constraints in **physics-based** approaches? (ii) Instead of using a single-stage learning process, will a well-designed multi-stage learning process improve the performance? (iii) Can such a design help emerge some crucial characteristics in multi-agent trajectory prediction, *e.g.*, collision avoidance?

Specifically, we address the challenging problems in the task of *collision-free multi-agent* trajectory prediction. In literature, first-order pattern-based approaches directly regress the trajectory based on the training data by fitting the position-based local transition patterns by either discrete cell [10–14], continuous position [15–17], or graph-based representations [18–20], without any semantics-based intermediate representation (except human body motions [21, 22]). Although higher-order pattern-based approaches incorporate some sorts of context, mostly in terms of relations between objects [23–33], they possess limited capability to emerge collision-free trajectories or verify whether the learning process or the learned model can do so. In contrast, we propose a learning method that incorporates high-level context cues of *congestion*, aiming at emerging collision-free trajectories and qualitatively verifying them.

The proposed method offers three unique advantages over prior methods. First, we represent the contextual cues by congestion via graph-based generative learning, wherein the node of the graph is the agent, and the edge of the graph is a measurement of the distance between two agents. Such a representation subsumes the Social Force (SF) commonly used in physics-based approaches; it not only models pair-wise SF

UCLA Center for Vision, Cognition, Learning, and Autonomy (VCLA) at Statistics Department. Emails: {xiexu, chi.zhang, yixin.zhu}@ucla.edu, {ywu, sczhu}@stat.ucla.edu.

The work reported herein was supported by ONR N00014-19-1-2153, ONR MURI N00014-16-1-2007, and DARPA XAI N66001-17-2-4029.

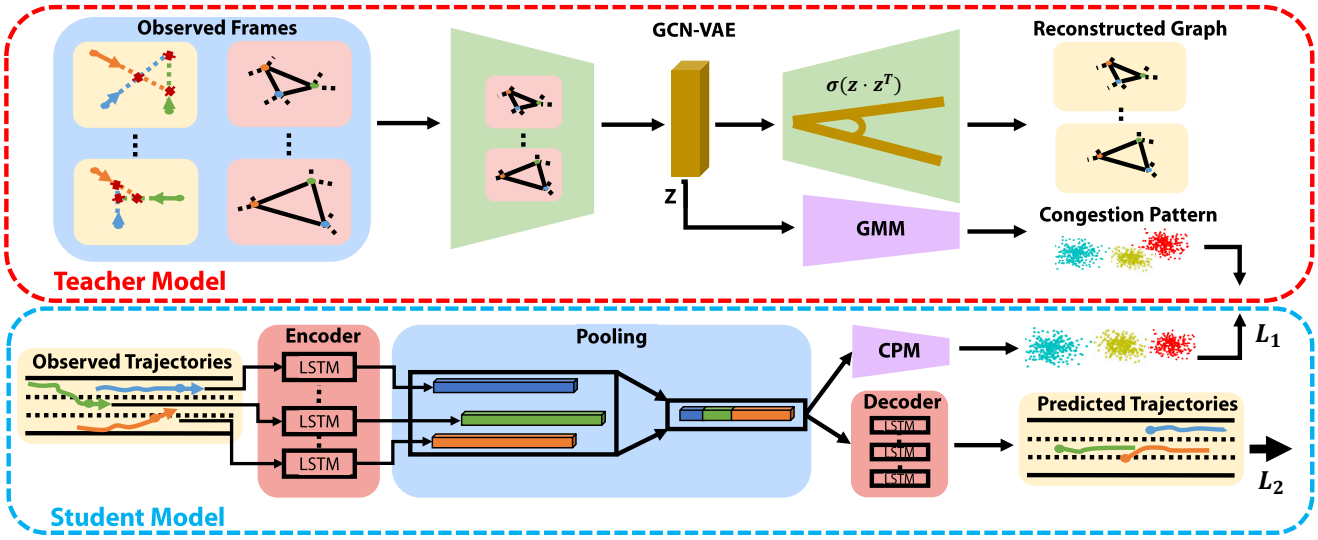


Fig. 1: **The proposed architecture for congestion-aware multi-agent trajectory prediction.** The teacher model (top) is composed of the frame-wise graph construction module, and the Graph Convolution Network (GCN)-VAE graph encoder and decoder. The learned latents are passed to a GMM and used to unsupervisedly learn the multi-modal congestion patterns. The student model (bottom) makes prediction based on the observed trajectories. It follows the encoder-pooling-decoder design and uses the CPM module to match the teacher's congestion patterns. The loss terms L_1 and L_2 are defined in Eq. (6) and Eq. (9), respectively.

but also provides a holistic view through graph modeling. Moreover, it implicitly encodes the (relative) distance as the cost, similar to how a planning-based method models the environment. Specifically, we use a Gaussian Mixture Model (GMM) to summarize the congestion patterns to (i) properly accounted for various modes in the congestion patterns, and (ii) reduce parameter space for the training process while maintaining a high task performance.

Second, we decouple the "Sense–Learn–Reason–Predict" framework into two processes: (i) A teacher "senses" and "learns" the contextual patterns (knowledge) as in a pure pattern-based approach. (ii) Instead of directly learning from observational data, a student reconstructs and "reasons" about the knowledge by minimizing a cost compared to what the teacher "learns", similar to a planning-based approach, while simultaneously "predicting" the future trajectory. Note that although such a design may look similar to GAN-based models for pedestrian trajectory predictions [34–36], they differ fundamentally. In GAN models, both the discriminator and the generator only focus on the predicted trajectory without explicit context modeling. In contrast, here in the proposed method, the student generates a trajectory supervised by explicitly learned contextual cues provided by the teacher. Such a design enables the student to learn from an inductive bias provided by the teacher, resulting in a faster training process and additional contextual constraints on the generated trajectories instead of random sampling.

Third, we formulate an optimization problem to bridge the connection between the congestion patterns and the learning objective of collision-free trajectory prediction, wherein the trajectories generated by the model are constrained by a learned congestion pattern distribution. By leveraging the variational parametrization, we derive an upper bound to make this optimization problem computationally tractable.

The final model recruits an encoder–pooling–decoder network design and is therefore compatible with many existing trajectory prediction architectures (*e.g.*, [28, 37]). In the experiments, we show the superiority of the proposed method in collision-free trajectory prediction in a new synthetic dataset designed to evaluate collision avoidance. Furthermore, the model remains competitive on the typical benchmark of the NGSIM US-101 highway dataset [38].

II. Related Works

Contextual Cues of Moving Agents: Agents' decisions on their motions depend on other agents' behaviors and interactions. In literature, such contextual cues are traditionally modeled by SF [39–47] or integrated into motion policies or cost functions [48–50]. Recent data-driven approaches, if ever, only implicitly learn the contextual cues by training on large datasets [23–31, 33, 51–53]. In contrast, we propose to explicitly use congestion as the contextual cues, which accounts for SF and is compatible with modern learning methods in pattern-based approaches.

Congestion Detection and Congestion Pattern: The notion of congestion has proven to be useful in various applications [54–58], especially on cooperative vehicular systems, such as vehicle-to-vehicle communications. Congestion patterns are introduced to quantify the congestion, either defined or learned in terms of discretized representations [59–63]. In this paper, we exploit the congestion patterns as contextual cues for multi-agent trajectory prediction.

Graph Neural Networks (GNNs) for Trajectory Prediction: With a powerful ability in representation learning, GNNs [64–66] saliently improve the performance on diverse tasks. Recently, researchers start to adopt GNNs for trajectory prediction; both Graph Attention Network (GAT) [36, 67] and GCN [68] have been used to aggregate information from all or neighboring agents. These works treat each agent as a graph node with various ways to construct weights of graph edges [69–72]. In comparison, the proposed method constructs a graph where each edge explicitly encodes an SF-based distance among all agents and recruits a GCN to learn congestion.

III. METHODS

This section describes the proposed method of congestion-aware multi-agent trajectory prediction; see Fig. 1 for an overview. We start with a brief problem definition of multi-agent trajectory prediction in Section III-A, following the conventions in [37]. Next, we introduce the learning of congestion patterns in multi-agent scenarios and describe how a teacher learns the contextual cues in Section III-B. In Section III-C, we formulate the optimization problem of congestion pattern matching so that a student can reason about congestion patterns taught by the teacher. We jointly solve this optimization problem with trajectory prediction by proposing a generic encoder-pooling-decoder model that predicts future collision-free trajectories in Section III-D.

A. Problem Definition

The goal of multi-agent trajectory prediction is to predict the future trajectories of all on-scene agents given their trajectory histories. We denote $\{\zeta^m, m=1,...,n\}$ as the set of trajectories for all n agents. At time step t, the position of the mth agent is represented by its local 2D coordinates $\zeta_t^m = (x_t^m, y_t^m)$. Given a time span $t = 1:T_h$, the observation is denoted as history trajectories $\zeta_h = \{\zeta_{t=1:T_h}^m\}$, and the future trajectories up to the time step T_p is denoted as $\zeta_p = \{\zeta_{t=T_h+1:T_p}^m\}$. In short, the multi-agent trajectory prediction is formulated as estimating the probability $P(\zeta_p|\zeta_h)$.

B. Learning Congestion Patterns

Intuitively, congestion patterns among agents in multi-agent navigation scenarios provide crucial contextual cues for trajectory prediction; they not only describe the position and intention of the agents, but also present safety-critical information about collisions. In literature, congestion patterns [61, 63] are mostly described by empirical equations about the relations of vehicle positions or clustering algorithms that group the observations into pre-defined categories. Such definitions have obvious shortcomings; for instance, it is sensitive with respect to the number of agents, vehicle moving velocities, *etc*. In fact, it is non-trivial to quantify the congestion patterns by an explicit set of rules or formulae. Instead, given the trajectory history as observation $o = \zeta_h$, we propose to learn the congestion patterns unsupervisedly: We use graph-based generative learning to derive the hidden congestion patterns and build a probabilistic GMM to account for various modes.

Graph Representation: To capture the congestion patterns embedded in physics constraints, we build the graph in the following way. Given two agents $u, v \in \mathcal{V}_t$, the graph $A_t = (\mathcal{V}_t, \mathcal{E}_t)$ at each frame $t \in \{1,..,T_h\}$ is constructed by their 2D locations (x_t, y_t) and velocities $(\dot{x}_t, \dot{y}_t)$. The graph adjacency matrix $\mathcal{E}_t = \{\mathcal{E}_t^{uv}\}, u, v = 1,..,n$ is defined as:

$$\mathcal{E}_t^{uv} = \mathcal{E}_t^{vu} = \begin{cases} 1/t_c^{uv}, t_c^{uv} > 0 \\ 0, t_c^{uv} = 0 \end{cases}, \tag{1}$$

where the estimated collision time is $t_c^{uv} = \max(-\frac{\Delta^{uv}x \times \Delta^{uv}\dot{x} + \Delta^{uv}y \times \Delta^{uv}\dot{y}}{\Delta^{uv}\dot{x}^2 + \Delta^{uv}\dot{y}^2}, 0)$, and $\Delta^{uv}(\cdot)$ denotes the quantity difference between agents u and v. Intuitively, a larger weight reflects a higher chance of collision, and the matrix describes the scene and congestion conditions.

Generative Learning: We leverage Variational Auto-Encoder (VAE) [73] to unsupervisedly learn the latent congestion pattern z in the graphs. Specifically, we follow the graph VAE approach proposed in GCN [74] where both the encoder and decoder are instantiated as graph convolutional layers. The objective is to optimize the reconstructed graph representation A_t while regularizing the latent distribution.

Gaussian Mixture Model (GMM): As the congestion patterns are naturally multi-modal, we further use a Gaussian Mixture Model to account for various modes. Specifically, treating the latent congestion pattern z as a random variable [75–77], we build the GMM as:

$$\mathcal{Q}(z) = \sum_i^{M_\mathcal{Q}} \lambda_i q_i(z), \tag{2}$$

where each mixture component $q_i(z)$ is a Gaussian distribution, λ_i is the mixture weight, and $M_\mathcal{Q}$ is the hyperparameter specifying the total number of mixtures. The mixture model can be learned by the stochastic EM [78] algorithm. As the hidden congestion pattern z is extracted from the observation o, we denote the mixture model as $\mathcal{Q}(o)$ henceforth.

C. Matching Congestion Patterns

While generating the trajectories, we hope that the student model can simultaneously match the congestion patterns taught by the teacher, such that the predicted trajectories are collision-free. We will first describe how to match the student's congestion model and that of the teacher's and defer the implementation details to the next section. Denoting the student's congestion pattern model as $\mathcal{P}(o)$, congestion pattern matching can be formulated as the KL-divergence between two pattern distributions:

$$\min \mathbb{D}_{\mathrm{KL}}(\mathcal{P}(o) \| \mathcal{Q}(o)). \tag{3}$$

Considering the mixture nature of $\mathcal{Q}(o)$, we model $\mathcal{P}(o)$ also as a Gaussian mixture, *i.e.*, $\mathcal{P}(o) = \sum_j^{M_\mathcal{P}} \omega_j p_j(o)$, where the total number of mixture $M_\mathcal{P}$ could be different from $M_\mathcal{Q}$.

Since there is no analytical solution for Eq. (3), we solve it by optimizing a variational upper bound. Similar to [79, 80], we propose a variational parametrization approach to solve the optimization problem. By decomposing the mixture weights $\omega_j = \sum_i^{M_\mathcal{Q}} \alpha_{ij}$ and $\lambda_i = \sum_j^{M_\mathcal{P}} \beta_{ij}$, the objective in Eq. (3) can be rewritten as:

$$\begin{aligned} \mathbb{D}_{\mathrm{KL}}(\mathcal{P}(o) \| \mathcal{Q}(o)) &= -\int \mathcal{P}(o) \log \frac{1}{\mathcal{P}(o)} \left(\sum_{i,j} \beta_{ij} q_i(o) \right) \\ &= -\int \mathcal{P}(o) \log \frac{1}{\mathcal{P}(o)} \left(\sum_{i,j} \frac{\beta_{ij} q_i(o) \alpha_{ij} p_j(o)}{\alpha_{ij} p_j(o)} \right). \end{aligned} \tag{4}$$

Using Jensen's inequality, Eq. (4) can be transformed to:

$$\begin{aligned} \mathbb{D}_{\mathrm{KL}}(\mathcal{P}(o) \| \mathcal{Q}(o)) &\leqslant -\int \mathcal{P}(o) \sum_{i,j} \frac{\alpha_{ij} p_j(o)}{\mathcal{P}(o)} \log \frac{\beta_{ij} q_i(o)}{\alpha_{ij} p_j(o)} \\ &= \sum_{i,j} \alpha_{ij} \mathbb{D}_{\mathrm{KL}}(p_j(o) \| q_i(o)) + D_{\mathrm{KL}}(\alpha \| \beta). \end{aligned} \tag{5}$$

We optimize Eq. (3) by minimizing its upper bound:

$$\min_{\{p_j\},\alpha,\beta} L_1 = \sum_{i,j} \alpha_{ij} \mathbb{D}_{\mathrm{KL}}(p_j(o) \| q_i(o)) + \mathbb{D}_{\mathrm{KL}}(\alpha \| \beta). \quad (6)$$

Note that the convergence of the optimization problem has been guaranteed as discussed in [79].

To solve Eq. (6), we iteratively optimize $\{p_j\}$, α, and β. Assuming fixed α and β,

$$\begin{aligned} &\min_{\{p_j\}} \sum_{i,j} \alpha_{ij} \mathbb{D}_{\mathrm{KL}}(p_j(o) \| q_i(o)) \\ &= \sum_{i,j} \alpha_{ij} \left(\mathbb{E}_{p_j(o)}[-\log q_i(o)] - \mathbb{H}[p_j(o)] \right). \end{aligned} \quad (7)$$

With $\{p_j\}$ learned, α and β can be updated by the closed-form solutions:

$$\alpha_{ij} = \frac{\omega_j \beta_{ij} \exp^{-\mathbb{D}_{\mathrm{KL}}(p_j(o) \| q_i(o))}}{\sum_{i'} \beta_{i'j} \exp^{-\mathbb{D}_{\mathrm{KL}}(p_j(o) \| q_{i'}(o))}}, \quad \beta_{ij} = \frac{\lambda_i \alpha_{ij}}{\sum_{j'} \alpha_{ij'}}. \quad (8)$$

The overall algorithm for the above congestion pattern matching (CPM) process is summarized in Algorithm 1. Please refer to supplementary video for a full derivation.

Algorithm 1: Congestion Pattern Matching (CPM)

1: Input: the learned congestion patterns $\mathcal{Q}(o)$
2: Initialize α_{ij} and β_{ij}
3: **while** not converged **do**
4: Fix α_{ij} and β_{ij} and optimize $\{p_j\}$ using Eq. (7)
5: Fix $\{p_j\}$ and update α_{ij} and β_{ij} using Eq. (8)
6: **end while**

D. Collision-free Trajectory Prediction

We make the student jointly predict trajectories and match the teacher's congestion patterns. As illustrated in Fig. 1, the trajectory prediction in the student model comprises an encoder module that encodes observed trajectories, a pooling module that models the spatial relations among agents, and a decoder module that recursively generates the future trajectories. The output of the social features at the pooling module is taken to match (distribution matching; see Eq. (6)) the teacher model's congestion pattern.

Our proposed student model can be trained end-to-end by iteratively minimizing the congestion pattern matching loss in Eq. (6) and the trajectory prediction loss, defined as

$$L_2 = -\frac{1}{m} \sum_{m} \sum_{t=T_h+1:T_p} \log P(\zeta_{p_t}^m | \zeta_h^m), \quad (9)$$

where ζ_h^m and ζ_p^m are the observed and predicted trajectories.

Implementation Details: The teacher model is composed of the GCN-VAE architecture with a latent dimension of 64. The GMM is a deep learned from the latent. We use fully connected layers to represent the congestion pattern. The teacher model is trained using Adam [81] with a learning rate of 1×10^{-4}. For the student model, it is compatible with prevalent encoder-pooling-decoder architectures [28, 37]. The pooling module is implemented following the social convolution pooling [28]. The encoder and decoder modules are created using LSTMs with a fixed hidden dimension of size 128. The $\mathrm{CPM}(\cdot)$ module is implemented as another deep GMM, which outputs the parameters of each mixture component. The number of components is a tunable hyper-parameter; see Section IV-D. The student model is learned using Adam [81] with a learning rate of 3×10^{-3}. Both models are implemented in PyTorch [82].

IV. Experiments

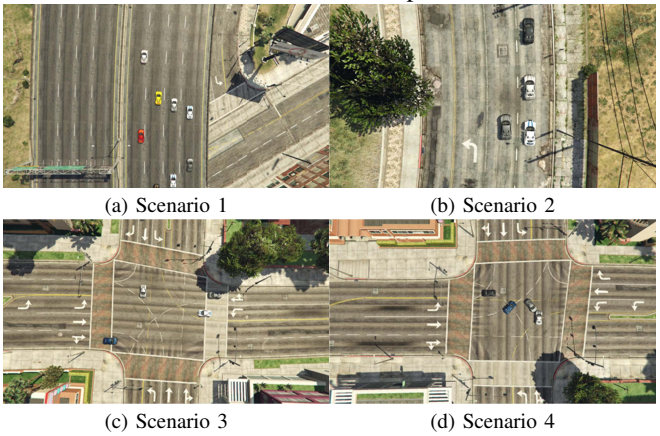
A. Datasets

GTA Dataset: To evaluate collision avoidance, we create a novel dataset based on the popular game platform of Grand Theft Auto (GTA). Compared to other platforms [83–85] that supports multi-agent simulations, the GTA models realistic urban-scale traffic commuting system; see Table I for other dataset statistics. This dataset focuses on trajectory prediction under safety-critical scenarios with rich vehicle interactions. By developing modding scripts, we create four types of safety-critical scenarios (see Fig. 2): (i) highway vehicle driving (mainly vehicle following), (ii) local vehicle driving (overtaking can frequently happen), (iii) driving in intersections (no traffic rules and crowded driving scenarios), and (iv) aggressive behaviors (almost lead to collisions). We use the four types of driving scenarios to study collision-free trajectory prediction. In experiments, we split the entire dataset into 3 folds for training and 1 fold for testing. All trajectories contain 3s of observations and 5s of predictions, and a model is tasked to predict agents' future paths.

TABLE I: GTA dataset statistics.

Total Clips	Vehicle Trajectories	Highway Trajectories	Local Trajectories
3300	27813	18229	9584
Following Events	Overtaking Events	Collision Events	
7055	2300	890	

NGSIM Dataset: We also evaluate the accuracy of trajectory prediction on the commonly used NGSIM US-101 [38] dataset to show the competitiveness of the proposed method. The dataset contains real highway traffic data that is captured over a time span of 45 minutes. Similar to the GTA dataset, We split the trajectories into 8s segments where 3s are used for observations and 5s for predictions.



(a) Scenario 1 (b) Scenario 2

(c) Scenario 3 (d) Scenario 4

Fig. 2: Sample top-views of the four scenarios in our GTA dataset.

B. Baselines and Evaluation Metrics

We compare our model with several well-established baselines [28, 31, 32, 34, 37, 86–90] and report the following metrics results:

Collision Rate evaluates the performance of collision avoidance for the predicted trajectories. In the GTA dataset, we calculate the collision rate by counting the collision events among trajectories and divide it by the total number of trajectories for all trials. The ground-truth collision event is obtained from the game simulator.

Root Mean Squared Error (RMSE) evaluates the accuracy of the predicted trajectories, calculated across trajectories given different time horizons (1s-5s). Note that for baseline models that generate trajectories using GANs [32, 34], we sample k output predictions for each trial and choose the "best" prediction in the sense of L_2 norm for evaluation.

C. Quantitative Results

Results on GTA Dataset: As shown in Tables II and III, the proposed **CF-LSTM** achieves the best performance on the collision rate and RMSE in the GTA dataset, proving its strength in collision-free trajectory prediction. Specifically, for the collision rate, CF-LSTM demonstrates the lowest error rates across all scenarios. Of the four scenarios, we note that Scenario 4 is the most challenging in the sense that there is no assumption of rational driving. Moreover, we notice a higher chance of collision in crowded space, *e.g.*, intersections where vehicles meet with each other. For RMSE, CF-LSTM attains the best result as presented in Table III, with the minimum average RMSE compared to other baseline methods. As shown in the table, the RMSE metric alone does not tell the difference between Scenario 3 and Scenario 4 for all the methods, which indicates that the accuracy of trajectory prediction does not perfectly reflect driving safety. We argue that there should be more effective metrics, such as the collision rate, to measure it.

TABLE II: Collision rate (%) on GTA

Methods	V-LSTM	CS-LSTM [28]	S-GAN [34]	CF-LSTM (Ours)
Scenario 1	4.219	3.086	3.372	**2.909**
Scenario 2	5.830	4.345	**4.015**	4.170
Scenario 3	8.331	6.997	5.805	**5.397**
Scenario 4	11.676	9.500	8.923	**8.766**
Avg	7.514	5.982	5.529	**5.310**

TABLE III: RMSE on GTA.

Methods	V-LSTM	CS-LSTM [28]	S-GAN [34]	CF-LSTM (Ours)
Scenario 1	1.88	1.25	1.40	**1.11**
Scenario 2	1.91	1.84	**1.74**	1.76
Scenario 3	2.98	2.55	2.67	**2.42**
Scenario 4	3.02	2.89	2.96	**2.76**
Avg	2.45	2.13	2.19	**2.01**

Results on NGSIM Dataset: Table IV shows performance of various models on the NGSIM dataset. The proposed **CF-LSTM** significantly outperforms the deterministic physics-based models of CV and C-VGMM+VIM [87] and surpasses planning-based models, such as GAIL-GRU [88] and PS-GAIL [89, 90], and pattern-based models, such as V-LSTM, S-LSTM [37], and CS-LSTM [28]. CF-LSTM improves the previous state-of-the-art in three settings compared to MFP [31]. However, the latter needs additional scene semantics for prediction in every time step while CF-LSTM does not. Our method also fares better than MATF GAN [32]. These results verify the competitiveness of CF-LSTM.

D. Qualitative Results

Congestion Patterns: We qualitatively examine our learned congestion patterns on the GTA dataset. In Fig. 3, we show the distribution λ_i ($M_{\mathcal{Q}} = 4$) of the learned GMM $\mathcal{Q}(o)$ on two different driving scenarios. The top row shows a series of driving behaviors involving lane changing and overtaking. When the overtaking occurs, the significance of the second component becomes rather evident, compared to the relatively uniform distributions at the start and end of the series. Such a distributional shift is reflected in the bottom two rows as well. In this scenario, multiple vehicles are driving into the intersection and must yield to each other to avoid the collision. As agents are getting closer to each other, making collisions more likely, some mixture weights are firing compared to the start and end frames. Taken together, these observational results verify that the learned congestion patterns can indeed reflect the contextual semantics of congestion. See ablation study for discussions on selecting the component numbers $M_{\mathcal{Q}}$ and $M_{\mathcal{P}}$.

Trajectory Predictions: We show a set of qualitative results on trajectory prediction in Fig. 4. Specifically, we compare the proposed CF-LSTM with CS-LSTM [28] and S-GAN [34] on four types of driving scenarios in the GTA dataset. As shown in the first row, our method generates more accurate trajectory points for the time interval than other baselines. For the local driving scenario with overtaking shown in the second row, our method successfully captures the overtaking vehicle's behavior. The predicted trajectory keeps a relatively safe distance from other vehicles during the lane change. Other methods either fail to keep a safe driving distance or diverge from the planned trajectory. For the intersection driving scenario in the third row, our method shows the vehicles' tendency to yield to avoid the collision when they get closer, demonstrating safety awareness from the contextual cues of congestion with more reasonable vehicle driving behaviors. The last row presents the case where one vehicle is aggressively driving through the intersection and eventually crashes into another. CS-LSTM and S-GAN are unaware of this dangerous situation, while our method shows the evident deceleration and yielding behavior. These qualitative analyses indicate the efficacy of the proposed model on safety-critical driving scenarios.

Ablation Study: We also conduct an ablation study to verify the efficiency of the proposed approach on collision-free trajectory prediction. Specifically, we show that our approach is compatible with other encoder-pooling-decoder architectures. By swapping the current architecture design to S-LSTM, we further improve S-LSTM's performance on both datasets. We compare whether directly enforcing the student model to match the latent congestion features could be better than distributional modeling and find that building a multi-modal distribution on congestion patterns can significantly improve performance. We hypothesize that this is because the GMM can account for the various modes in congestion patterns. Finally, we search for the hyperparameter on the number of mixtures. We notice that, coherent to the assumption on congestion pattern learning, the model achieves the best performance when the hidden mixture number equals that of the ground-truth. Please refer to the supplementary video for details of the ablation study.

TABLE IV: RMSE on NGSIM.

Times(s)	CV	C-VGMM+VIM [87]	V-LSTM	S-LSTM [37]	CS-LSTM [28]	MFP [31]	MATF GAN [32]	VAE	GAIL-GRU [88]	PS-GAIL [89, 90]	CF-LSTM (Ours)
1s	0.73	0.66	0.66	0.65	0.61	**0.54**	0.66	0.68	0.69	0.60	0.55
2s	1.78	1.56	1.64	1.31	1.27	1.16	1.34	1.72	1.51	1.83	**1.10**
3s	3.13	2.75	2.94	2.16	2.09	1.89	2.08	2.77	2.55	3.14	**1.78**
4s	4.78	4.24	4.59	3.25	3.10	2.75	2.97	3.94	3.65	4.56	**2.73**
5s	6.68	5.99	6.60	4.55	4.37	**3.78**	4.13	5.21	4.71	6.48	3.82

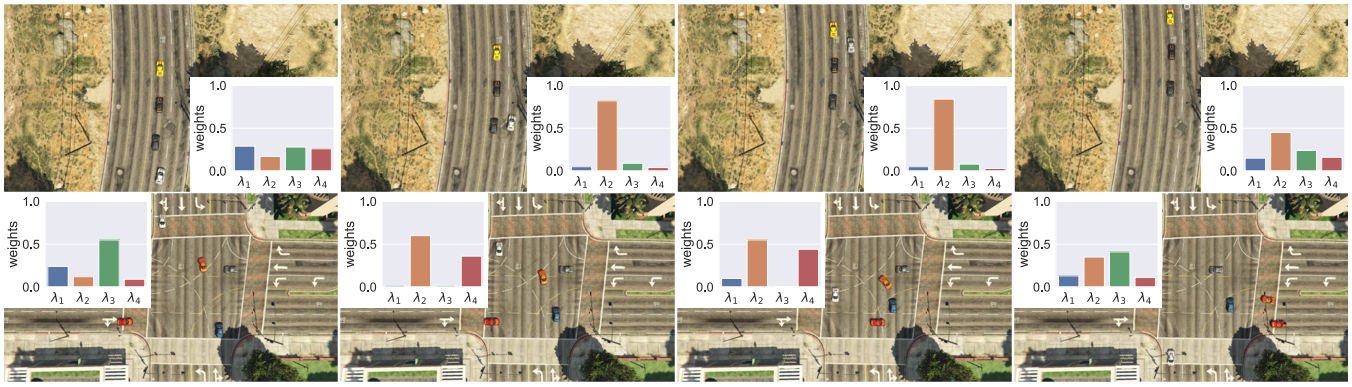
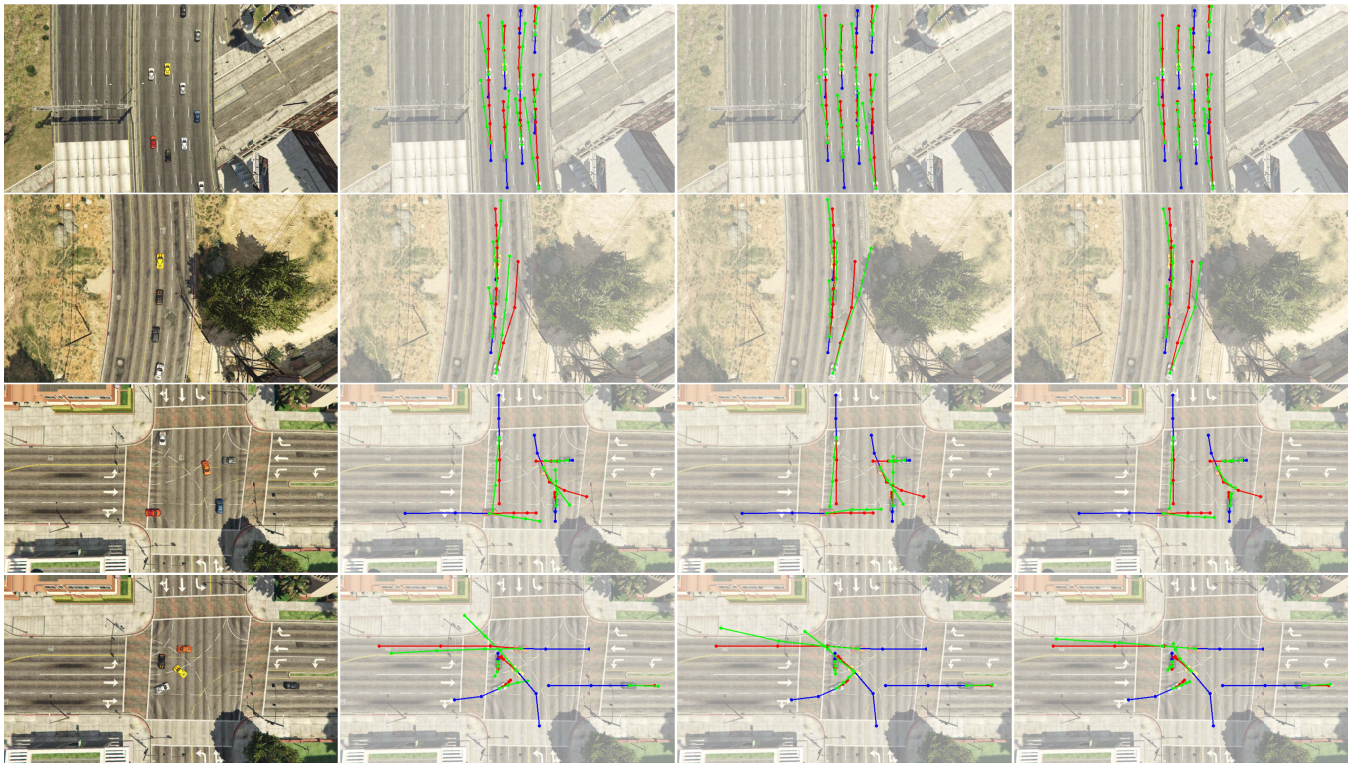


Fig. 3: **Mixture weights of congestion patterns (GMM) learned in the driving scenarios.** Top: the mixture weight distribution during overtaking. Bottom: the mixture weight distribution in a crowded intersection. Note that the vehicle number n does not have to match the number of Gaussian components $M_{\mathcal{Q}}$.



(a) Scenarios (b) S-GAN (c) CS-LSTM (d) CF-LSTM

Fig. 4: **Qualitative results of trajectory prediction in four types of scenarios from the proposed CF-LSTM and two baselines.** Blue: observed trajectories. Red: ground-truth future trajectories. Green: predicted future trajectories.

V. Conclusions

In this work, we study the problem of multi-agent trajectory prediction. We propose to explicitly learn congestion patterns as contextual cues and decouple the "Sense-Learn-Reason-Predict" framework into a teacher-student process. We formulate an optimization problem to bridge the connection between the congestion patterns and the learning objective of collision-free trajectory prediction. In experiments, we show that the proposed model is able to achieve the best performance on collision-free trajectory prediction on a synthetic dataset designed for collision avoidance evaluation while remaining competitive on regular trajectory prediction on the NGSIM US-101 highway dataset.

REFERENCES

[1] F. Heider and M. Simmel, "An experimental study of apparent behavior," *The American journal of psychology*, vol. 57, no. 2, pp. 243–259, 1944.

[2] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Perception & psychophysics*, vol. 14, no. 2, pp. 201–211, 1973.

[3] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *arXiv preprint arXiv:1905.06113*, 2019.

[4] Q. Zhu, "Hidden markov model for dynamic obstacle avoidance of mobile robot navigation," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 390–397, 1991.

[5] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. part v. multiple-model methods," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1255–1321, 2005.

[6] S. Tadokoro, Y. Ishikawa, T. Takebe, and T. Takamori, "Stochastic prediction of human motion and control of robots in the service of human," in *IEEE Systems Man and Cybernetics Conference*, 1993.

[7] A. Bruce and G. Gordon, "Better motion prediction for people-tracking," in *ICRA*, 2004.

[8] E. Stein, *Without good reason: The rationality debate in philosophy and cognitive science*. Clarendon Press, 1996.

[9] D. Kahneman and A. Tversky, "On the psychology of prediction.," *Psychological review*, vol. 80, no. 4, p. 237, 1973.

[10] E. Kruse and F. M. Wahl, "Camera-based observation of obstacle motions to derive statistical data for mobile robot motion planning," in *ICRA*, 1998.

[11] S. Thompson, T. Horiuchi, and S. Kagami, "A probabilistic model of human motion and navigation intent for mobile robot path planning," in *International Conference on Autonomous Robots and Agents*, 2009.

[12] T. Kucner, J. Saarinen, M. Magnusson, and A. J. Lilienthal, "Conditional transition maps: Learning motion patterns in dynamic environments," in *IROS*, 2013.

[13] L. Ballan, F. Castaldo, A. Alahi, F. Palmieri, and S. Savarese, "Knowledge transfer for scene-specific motion prediction," in *ECCV*, 2016.

[14] S. Molina, G. Cielniak, T. Krajník, and T. Duckett, "Modelling and predicting rhythmic flow patterns in dynamic environments," in *Annual Conference Towards Autonomous Robotic Systems*, 2018.

[15] J. Joseph, F. Doshi-Velez, A. S. Huang, and N. Roy, "A bayesian nonparametric approach to modeling motion patterns," *Autonomous Robots*, vol. 31, no. 4, p. 383, 2011.

[16] S. Ferguson, B. Luders, R. C. Grande, and J. P. How, "Real-time predictive modeling and robust avoidance of pedestrians with uncertain, changing intentions," in *Algorithmic Foundations of Robotics XI*, 2015.

[17] T. P. Kucner, M. Magnusson, E. Schaffernicht, V. H. Bennetts, and A. J. Lilienthal, "Enabling flow awareness for mobile robots in partially observable environments," *RA-L*, vol. 2, no. 2, pp. 1093–1100, 2017.

[18] L. Liao, D. Fox, J. Hightower, H. Kautz, and D. Schulz, "Voronoi tracking: Location estimation using sparse and noisy sensor data," in *IROS*, 2003.

[19] D. Vasquez, T. Fraichard, and C. Laugier, "Incremental learning of statistical motion patterns with growing hidden markov models," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 403–416, 2009.

[20] Y. F. Chen, M. Liu, and J. P. How, "Augmented dictionary learning for motion prediction," in *ICRA*, 2016.

[21] R. Quintero, J. Almeida, D. F. Llorca, and M. Sotelo, "Pedestrian path prediction using body language traits," in *IEEE Intelligent Vehicles Symposium Proceedings*, 2014.

[22] R. Q. Mínguez, I. P. Alonso, D. Fernández-Llorca, and M. Á. Sotelo, "Pedestrian path, pose, and intention prediction through gaussian process dynamical models and pedestrian activity recognition," *Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1803–1814, 2018.

[23] B. Kim, C. M. Kang, J. Kim, S. H. Lee, C. C. Chung, and J. W. Choi, "Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network," in *IEEE International Conference on Intelligent Transportation Systems*, 2017.

[24] F. Altché and A. de La Fortelle, "An lstm network for highway trajectory prediction," in *IEEE International Conference on Intelligent Transportation Systems*, 2017.

[25] S. H. Park, B. Kim, C. M. Kang, C. C. Chung, and J. W. Choi, "Sequence-to-sequence prediction of vehicle trajectory via lstm encoder-decoder architecture," in *IEEE Intelligent Vehicles Symposium*, 2018.

[26] W. Ding, J. Chen, and S. Shen, "Predicting vehicle behaviors over an extended horizon using behavior interaction network," in *ICRA*, 2019.

[27] N. Deo and M. M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms," in *IEEE Intelligent Vehicles Symposium*, 2018.

[28] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Workshops of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[29] S. Dai, L. Li, and Z. Li, "Modeling vehicle interactions via modified lstm models for trajectory prediction," *IEEE Access*, vol. 7, pp. 38287–38296, 2019.

[30] J. Li, H. Ma, W. Zhan, and M. Tomizuka, "Coordination and trajectory prediction for vehicle interactions via bayesian generative modeling," in *IEEE Intelligent Vehicles Symposium*, 2019.

[31] C. Tang and R. R. Salakhutdinov, "Multiple futures prediction," in *NeurIPS*, 2019.

[32] T. Zhao, Y. Xu, M. Monfort, W. Choi, C. Baker, Y. Zhao, Y. Wang, and Y. N. Wu, "Multi-agent tensor fusion for contextual trajectory prediction," in *CVPR*, 2019.

[33] S. Srikanth, J. A. Ansari, S. Sharma, *et al.*, "Infer: Intermediate representations for future prediction," *arXiv preprint arXiv:1903.10641*, 2019.

[34] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *CVPR*, 2018.

[35] 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 *CVPR*, 2019.

[36] 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 *NeurIPS*, 2019.

[37] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *CVPR*, 2016.

[38] J. Colyar and J. Halkias, "Us highway 101 dataset," *Federal Highway Administration (FHWA), Tech. Rep. FHWA-HRT-07-030*, 2007.

[39] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[40] M. Luber, J. A. Stork, G. D. Tipaldi, and K. O. Arras, "People tracking with human motion predictions from social forces," in *ICRA*, 2010.

[41] J. Elfring, R. Van De Molengraft, and M. Steinbuch, "Learning intentions for improved human motion prediction," *Robotics and Autonomous Systems*, vol. 62, no. 4, pp. 591–602, 2014.

[42] G. Ferrer and A. Sanfeliu, "Behavior estimation for a complete framework for human motion prediction in crowded environments," in *ICRA*, 2014.

[43] H. Kretzschmar, M. Kuderer, and W. Burgard, "Learning to predict trajectories of cooperatively navigating agents," in *ICRA*, 2014.

[44] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *ECCV*, 2016.

[45] V. Karasev, A. Ayvaci, B. Heisele, and S. Soatto, "Intent-aware long-term prediction of pedestrian motion," in *ICRA*, 2016.

[46] D. Xie, T. Shu, S. Todorovic, and S.-C. Zhu, "Learning and inferring "dark matter" and predicting human intents and trajectories in videos," *T-PAMI*, vol. 40, no. 7, pp. 1639–1652, 2017.

[47] Y. Zhu, T. Gao, L. Fan, S. Huang, M. Edmonds, H. Liu, F. Gao, C. Zhang, S. Qi, Y. N. Wu, *et al.*, "Dark, beyond deep: A paradigm shift to cognitive ai with humanlike common sense," *Engineering*, vol. 6, no. 3, pp. 310–345, 2020.

[48] J. Wu, J. Ruenz, and M. Althoff, "Probabilistic map-based pedestrian motion prediction taking traffic participants into consideration," in *IEEE Intelligent Vehicles Symposium*, 2018.

[49] P. Zechel, R. Streiter, K. Bogenberger, and U. Göhner, "Pedestrian occupancy prediction for autonomous vehicles," in *IEEE International Conference on Robotic Computing*, 2019.

[50] C. Muench and D. M. Gavrila, "Composable q-functions for pedestrian car interactions," in *IEEE Intelligent Vehicles Symposium*, 2019.

[51] F. Kuhnt, J. Schulz, T. Schamm, and J. M. Zöllner, "Understanding interactions between traffic participants based on learned behaviors," in *IEEE Intelligent Vehicles Symposium*, 2016.

[52] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," in *ICRA*, 2019.

[53] J. Hong, B. Sapp, and J. Philbin, "Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions," in *CVPR*, 2019.

[54] R. Bauza, J. Gozalvez, and J. Sanchez-Soriano, "Road traffic congestion detection through cooperative vehicle-to-vehicle communications," in *IEEE Local Computer Network Conference*, 2010.

[55] L. Lin and T. Osafune, "Road congestion detection by distributed vehicle-to-vehicle communication systems," 2011. US Patent 7,877,196.

[56] M. Milojevic and V. Rakocevic, "Distributed road traffic congestion quantification using cooperative vanets," in *Annual Mediterranean Ad Hoc Networking Workshop*, 2014.

[57] R. Bauza and J. Gozálvez, "Traffic congestion detection in large-scale scenarios using vehicle-to-vehicle communications," *Journal of Network and Computer Applications*, vol. 36, no. 5, pp. 1295–1307, 2013.

[58] R. Sundar, S. Hebbar, and V. Golla, "Implementing intelligent traffic control system for congestion control, ambulance clearance, and stolen vehicle detection," *Sensors*, vol. 15, no. 2, pp. 1109–1113, 2014.

[59] R. Horiguchi and K. Wada, "Effective probe data transmittal with detection of congestion pattern," in *Proceedings of World Congress on Intelligent Transport Systems*, 2004.

[60] K. Zhang, S. Batterman, and F. Dion, "Vehicle emissions in congestion: Comparison of work zone, rush hour and free-flow conditions," *Atmospheric Environment*, vol. 45, no. 11, pp. 1929–1939, 2011.

[61] L. Xu, Y. Yue, and Q. Li, "Identifying urban traffic congestion pattern from historical floating car data," *Procedia-Social and Behavioral Sciences*, vol. 96, pp. 2084–2095, 2013.

[62] K. Zhang, D. Sun, S. Shen, and Y. Zhu, "Analyzing spatiotemporal congestion pattern on urban roads based on taxi gps data," *Journal of Transport and Land Use*, vol. 10, no. 1, pp. 675–694, 2017.

[63] H. Xiong, A. Vahedian, X. Zhou, Y. Li, and J. Luo, "Predicting traffic congestion propagation patterns: a propagation graph approach," in *Proceedings of the ACM SIGSPATIAL International Workshop on Computational Transportation Science*, 2018.

[64] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, 2008.

[65] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," *arXiv preprint arXiv:1511.05493*, 2015.

[66] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[67] Y. Huang, H. Bi, Z. Li, T. Mao, and Z. Wang, "Stgat: Modeling spatial-temporal interactions for human trajectory prediction," in *ICCV*, 2019.

[68] J. Sun, Q. Jiang, and C. Lu, "Recursive social behavior graph for trajectory prediction," *arXiv preprint arXiv:2004.10402*, 2020.

[69] A. Vemula, K. Muelling, and J. Oh, "Social attention: Modeling attention in human crowds," in *ICRA*, 2018.

[70] S. Eiffert and S. Sukkarieh, "Predicting responses to a robot's future motion using generative recurrent neural networks," *arXiv preprint arXiv:1909.13486*, 2019.

[71] C. Choi, A. Patil, and S. Malla, "Drogon: A causal reasoning framework for future trajectory forecast," *arXiv preprint arXiv:1908.00024*, 2019.

[72] B. Ivanovic and M. Pavone, "The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs," in *ICCV*, 2019.

[73] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[74] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[75] N. Dilokthanakul, P. A. Mediano, M. Garnelo, M. C. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan, "Deep unsupervised clustering with gaussian mixture variational autoencoders," *arXiv preprint arXiv:1611.02648*, 2016.

[76] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, "Variational deep embedding: An unsupervised and generative approach to clustering," *arXiv preprint arXiv:1611.05148*, 2016.

[77] L. Yang, N.-M. Cheung, J. Li, and J. Fang, "Deep clustering by gaussian mixture variational autoencoders with graph embedding," in *CVPR*, 2019.

[78] G. Celeux, "The sem algorithm: a probabilistic teacher algorithm derived from the em algorithm for the mixture problem," *Computational Statistics Quarterly*, vol. 2, pp. 73–82, 1985.

[79] J. R. Hershey and P. A. Olsen, "Approximating the kullback leibler divergence between gaussian mixture models," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2007.

[80] J. Xie, R. Gao, E. Nijkamp, S.-C. Zhu, and Y. N. Wu, "Representation learning: A statistical perspective," *Annual Review of Statistics and Its Application*, vol. 7, 2019.

[81] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[82] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019.

[83] X. Xie, H. Liu, Z. Zhang, Y. Qiu, F. Gao, S. Qi, Y. Zhu, and S.-C. Zhu, "Vrgym: A virtual testbed for physical and interactive ai," in *Proceedings of the ACM Turing Celebration Conference-China*, 2019.

[84] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, 2017.

[85] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017.

[86] E. Schmerling, K. Leung, W. Vollprecht, and M. Pavone, "Multimodal probabilistic model-based planning for human-robot interaction," in *ICRA*, 2018.

[87] N. Deo, A. Rangesh, and M. M. Trivedi, "How would surround vehicles move? a unified framework for maneuver classification and motion prediction," *IEEE Transactions on Intelligent Vehicles*, 2018.

[88] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *NeurIPS*, 2016.

[89] R. P. Bhattacharyya, D. J. Phillips, B. Wulfe, J. Morton, A. Kuefler, and M. J. Kochenderfer, "Multi-agent imitation learning for driving simulation," in *IROS*, 2018.

[90] R. P. Bhattacharyya, D. J. Phillips, C. Liu, J. K. Gupta, K. Driggs-Campbell, and M. J. Kochenderfer, "Simulating emergent properties of human driving behavior using multi-agent reward augmented imitation learning," in *ICRA*, 2019.