Implicit Latent Variable Model for Scene-Consistent Motion Forecasting

Sergio Casas[⋆1,2], Cole Gulino[⋆1], Simon Suo[⋆1,2],
Katie Luo[1], Renjie Liao[1,2], Raquel Urtasun[1,2]

Uber ATG[1], University of Toronto[2]
{sergio.casas, cgulino, suo, katie.luo, rjliao, urtasun}@uber.com

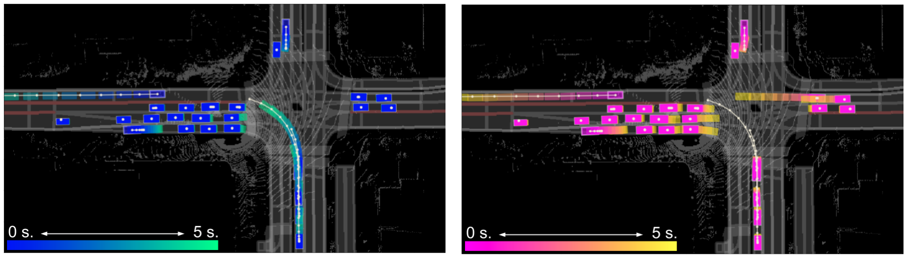
Abstract. In order to plan a safe maneuver an autonomous vehicle must accurately perceive its environment, and understand the interactions among traffic participants. In this paper, we aim to learn scene-consistent motion forecasts of complex urban traffic directly from sensor data. In particular, we propose to characterize the joint distribution over future trajectories via an implicit latent variable model. We model the scene as an interaction graph and employ powerful graph neural networks to learn a distributed latent representation of the scene. Coupled with a deterministic decoder, we obtain trajectory samples that are consistent across traffic participants, achieving state-of-the-art results in motion forecasting and interaction understanding. Last but not least, we demonstrate that our motion forecasts result in safer and more comfortable motion planning.

1 Introduction

Self driving vehicles (SDV) have the potential to make a broad impact in our society, providing a safer and more efficient solution to transportation. A critical component for autonomous driving is the ability to perceive the world and forecast all possible future instantiations of the scene. 3D perception algorithms have improved incredibly fast in recent years [29, 33, 42, 53, 54, 57], yielding very accurate object detections surrounding the SDV. However, producing multi-modal motion forecasts that precisely capture multiple plausible futures consistently for all actors in the scene remains a very open problem.

The complexity is immense: the future is inherently uncertain as actor behaviors are influenced not only by their own individual goals and intentions but also by the other actors' actions. For instance, an actor at an intersection may choose to turn right or go straight due to its own destination, and yield or go if the behavior of a nearby traffic participant is aggressive or conservative. Moreover, unobserved traffic rules such as the future traffic light states heavily affect the traffic (see Fig.1). It is clear that all these aspects cannot be directly observed and require complex reasoning about the scene as a whole, including its geometry, topology and the interaction between multiple agents.

[⋆] Denotes equal contribution



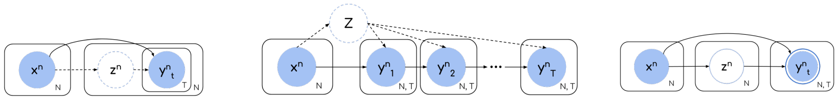
(a) Sample 1: protected left turn (b) Sample 2: horizontal traffic flow

Fig. 1: **Two scene-consistent future trajectory samples from our model**. Ground truth trajectories are shown as white polylines.

In an autonomy system, detections and motion forecasts for other actors in the scene are typically passed as obstacles to a motion-planner [43, 48] in order to plan a safe maneuver. Importantly, the distribution over future trajectories needs to cover the ground-truth for the plan to be safe, but also must exhibit low enough entropy such that a comfortable ride with reasonable progress is achieved. Thus in complex urban environments the SDV should reason about multiple futures separately [15, 20, 27], and plan proactively by understanding how its own actions might influence other actors' behaviors [16,39]. Furthermore, as self-driving vehicles get closer to full autonomy, closed-loop simulation is becoming increasingly critical not only for testing but also for training. In a self-driving simulator [6, 14, 38], smart-actor models [3, 4, 7, 52] are responsible for generating stochastic joint behaviors that are realistic at a scene-level, with actors obeying to underlying scene dynamics with complex interactions.

These applications require learning a joint distribution over actors' future trajectories that characterizes how the scene might unroll as a whole. Since this is generally intractable, many motion forecasting approaches [9, 11, 12, 18] assume marginal independence across actors' future trajectories, failing to get scene-consistent futures. Alternatively, auto-regressive formulations [45, 51] model interactions at the output level, but require sequential sampling which results in slow inference and compounding errors [47].

To overcome these challenges, we propose a novel way to characterize the joint distribution over motion forecasts via an implicit latent variable model (ILVM). We aim to recover a latent space that can summarize all the unobserved scene dynamics given input sensor data. This is challenging given that (i) modern roads present very complex geometries and topologies that make every intersection unique, (ii) the dynamic environment is only partially observed through sensor returns, and (iii) the number of actors in a scene is variable. To address these, we model the scene as an interaction graph [9, 22, 26, 32], where nodes are traffic participants. We then partition the scene latent space into a distributed representation among actors. We leverage graph neural networks (GNN) [2] both to encode the full scene into the latent space as well as to decode latent samples into socially consistent future trajectories. We frame the decoding of all actors'



(a) Independent output (b) Social auto-regressive (c) ILVM (Ours)

Fig. 2: **Graphical models of trajectory distribution**. Dashed arrows/circles denote that only some approaches within the group use those components. Double circle in (c) denotes that it is a deterministic mapping of its inputs.

trajectories as a deterministic mapping from the inputs and scene latent samples, making the latent variables capture all the stochasticity in our generative process. Furthermore, this allows us to perform efficient inference via parallel sampling.

We show that our ILVM significantly outperforms the motion forecasting state-of-the-art in ATG4D [54] and nuScenes [8]. We observe that our ILVM is able to generate scene-consistent samples (see Fig. 1) while producing less entropic joint distributions that also better cover the ground-truth. Moreover, when using our scene-consistent motion forecasts, a state-of-the-art motion planner [48] can plan safer and more comfortable trajectories.

2 Related Work

In this section, we review recent advances in motion forecasting, with a focus on realistic approaches that predict from sensor data, explicitly reason about the multi-modality of the output distribution, or model multi-agent interactions.

In traditional self-driving stacks, an object detection module is responsible for recognizing other traffic participants in the scene, followed by a motion forecasting module that predicts how the scene might unroll given the current state of each actor. However, the actor state is typically a very compact representation that includes pose, velocity, and acceleration. As a consequence, it is hard to incorporate uncertainty due to sensor noise or occlusion.

We follow the works of [10, 35, 55], which unified these two tasks by having a single fully convolutional backbone network predict both the current and future states for each pixel in a bird's eye view grid directly from a voxelized LiDAR point-cloud and semantic raster of an HD map. This approach naturally propagates uncertainty between the two tasks in the feature space, without the need of explicit intermediate representations. While these models reason about uncertainty in sensor observations, they neglect inherent uncertainty in the actors' future behavior. [9, 32] add agent-agent interaction reasoning to this framework. [9] introduces spatially-aware graph neural networks that aggregate features from neighboring actors in the scene to predict a single trajectory per actor with gaussian waypoints, assuming marginal independence across actors. This approach is still limited in expressivity since (i) a uni-modal characterization of the future is insufficient for downstream motion planning to make safe decisions, and (ii) modeling the marginal distribution per actor cannot provide trajectory samples that are consistent across actors.

Another research stream [1,13,19,26,30,37,44,45,51] has focused on the problem of multi-agent trajectory prediction from perfect perception, i.e., assuming that the ground-truth past trajectory of all actors' is given. Unfortunately, this is not realistic in self-driving vehicles, which rely on imperfect perception with noise that translates into failures such as false positive and false negative detections and id switches in tracking. Nonetheless, these methods have proposed output parameterizations that can predict multi-modal distributions over future trajectories, which are applicable to our end-to-end perception and prediction setting.

Various factorizations of the joint distribution over N actors' trajectories $p(Y|X) = p(y_1, \cdots, y_N|x_1, \cdots, x_N)$ with different levels of independence assumptions have been proposed to sidestep the intractability of $p(Y|X)$. The simplest approximation is to assume *independent futures* across actors and time steps $p(Y|X) = \prod_n \prod_t p(y_n^t|X)$, as shown in Fig. 2a. Some approaches directly regress the parameters of a mixture of Gaussians over time [11, 12, 34], which provides efficient sampling but can suffer from low expressivity and unstable optimization. Non-parametric approaches [23,24,41,46] have also been proposed to characterize the multi-modality of one actor's individual behavior. These approaches either score trajectory samples from a finite set [41, 56] with limited coverage or predict an occupancy grid at different future horizons [23, 24, 46], which is very memory consuming. [44] proposed to learn a one-step policy that predicts the next waypoint based on the previous history, avoiding the time independence assumption. Variational methods [18, 31] inspired by [25, 50] have also been proposed to learn an actor independent latent space to capture unobserved actor dynamics such as goals. Unfortunately, none of these methods can accurately characterize the joint distribution in interactive situations, since the generative process is independent per actor.

An alternative approach to better characterize the behavior of multiple actors jointly is *autoregressive generation* with social mechanisms [1, 45], which predict the distribution over the next trajectory waypoint of each actor conditioned on the previous states of all actors $p(Y|X) = \prod_n \prod_t p\left(y_n^t|Y^{0:t-1}, X\right)$. This approach has been enhanced by introducing latent variables [22, 26, 51], as in Fig. 2b. In particular, [26] introduces discrete latent variables to model pairwise relationships in an interaction graph, while in [22,51] they capture per-actor high-level actions. Autoregressive approaches, however, suffer from compounding errors [28,40,47]. During training, the model is fed the ground-truth $Y^{0:t-1}$, while during inference, the model must rely on approximate samples from the learned distribution. While scheduled sampling [5] has been proposed to mitigate this issue, the objective function underlying this method is improper [21] and pushes the conditional distributions $p(y_n^t|Y^{0:t-1})$ to model the marginal distributions $p(y_n^t)$ instead. Moreover, these methods require sequential sampling, which is not amenable to real-time applications such as self-driving.

In contrast to previous works, we propose to model interaction in a scene latent space that captures all sources of uncertainty, and use a deterministic decoder to characterize an implicit joint distribution over all actors' future trajectories without any independence assumptions at the output level, as shown in Fig. 2c. This design features efficient parallel sampling, high expressivity and yields trajectory samples that are substantially more consistent across actors.

3 Scene Level Reasoning for Motion Forecasting

In this section we introduce our approach to model **the joint distribution** $P(Y|X)$ **over** N **actors' future trajectories** $Y = \{y_1, y_2, \cdots, y_N\}$ given each actor's local context $X = \{x_1, x_2, \cdots, x_N\}$ extracted from sensor data and HD maps. An actor's trajectory y_n is composed of 2D waypoints over time y_n^t in the coordinate frame defined by the actor's current position and heading. In the following, we first explain our implicit latent variable model, then introduce our concrete architecture including the actor feature extraction from sensor data, and finally explain how to train our model in an end-to-end manner.

3.1 Implicit Latent Variable Model with Deterministic Decoder

We formulate the generative process of future trajectories over actors with a latent variable model:

$$P(Y|X) = \int_Z P(Y|X,Z)P(Z|X)dZ$$

where Z is a latent variable that captures unobserved scene dynamics such as actor goals and style, multi-agent interactions, or future traffic light states.

We propose to use a **deterministic mapping** $Y = f(X,Z)$ to implicitly characterize $P(Y|X,Z)$, instead of explicitly representing it in a parametric form. This approach allows us to avoid factorizing $P(Y|X,Z)$ (as in Fig. 2a or Fig. 2b) and sidestep the associated shortcomings discussed in Section 2. In this framework, generating scene-consistent future trajectories Y across actors is simple and highly efficient, since it only requires one stage of parallel sampling:

1. Draw latent scene samples from prior $Z \sim P(Z|X)$
2. Decode with the deterministic decoder $Y = f(X,Z)$

We emphasize that this modeling choice encourages the latent Z to capture *all* stochasticity in our generative process. To this end, we leverage a *continuous latent* Z for high expressivity. This stands in contrast to previous methods [22,26,51], where discrete latent Z are employed to model discrete high-level actions or pairwise interactions, and an explicit $P(Y|X,Z)$ to represent continuous uncertainty.

Producing a latent space that can capture all the uncertainties in any scenario is challenging: scenarios vary drastically in the number of actors N, the road topology as well as traffic rules. To mitigate this challenge, we propose to

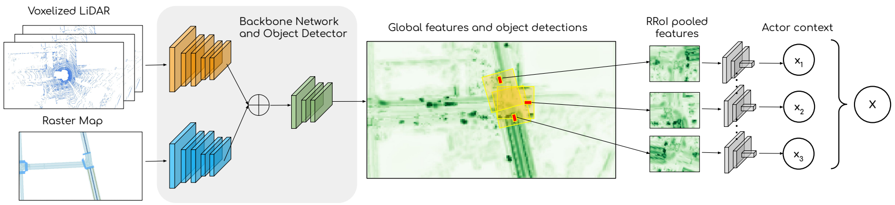


Fig. 3: **Actor Feature Extraction**. Given LiDAR and maps, our backbone CNN detects the actors in the scene, and individual feature vectors per actor are extracted via RRoI Align [36], followed by a CNN with spatial pooling.

partition the scene latent as $Z = \{z_1, z_2, \cdots, z_N\}$, obtaining a distributed representation where z_n is anchored to actor n in an interaction graph with traffic participants as nodes. The distributed representation has the benefit of naturally scaling the capacity of the latent space as the number of actors grow. Furthermore, the anchoring gives the model an inductive bias that eases the learning of a scene latent space. Intuitively, each latent z_n encodes unobserved dynamics most relevant to actor n, including interactions with neighboring actors and traffic rules that apply in its locality. We represent each z_n as a diagonal multivariate gaussian $z_n \sim \mathcal{N}\left(\left[\mu_n^1(X), \cdots, \mu_n^D(X)\right], \text{diag}\left(\left[\sigma_n^1(X), \cdots, \sigma_n^D(X)\right]\right)\right)$, as is common with variational models [25,50]. We emphasize that although factorized, the latent space is not marginally independent across actors since each z_n is conditioned on all $x_1, \cdots, x_N$ as shown in the graphical model in Fig. 2c.

Since integration over Z is intractable, we exploit amortized variational inference [25,50]. By introducing an encoder distribution $Q(Z|X,Y)$ to approximate the true posterior $P(Z|X,Y)$, the learning problem can be reformulated as a maximization of the Evidence Lower BOund (ELBO). Please visit the supplementary for a more thorough description of variational inference.

3.2 Joint Perception and Motion Forecasting Architecture

Our architecture consists of an actor feature extractor that detects objects in the scene and provides rich representations of each actor (Fig. 3), encoder/prior modules that infer a scene latent space at training/inference respectively, and a decoder that predicts the actors' future trajectories (Fig. 4). To implement the prior, encoder and decoder modules, we leverage a flexible scene interaction module (SIM) as our building block for relational reasoning (Alg. 1).

Actor Feature Extractor: Fig. 3 shows how we extract per actor features $X = \{x_1, x_2, \cdots, x_N\}$ from raw sensor data and HD maps in a differentiable manner, such that perception and motion forecasting can be trained jointly end-to-end. We use a CNN-based perception backbone network architecture inspired by [10,54] to extract rich geometrical and motion features about the whole scene from a past history of voxelized LiDAR point clouds and a raster map. We then detect [54] the traffic participants in the scene, and apply Rotated Region of Interest Align [36] to the backbone features around each object detection,

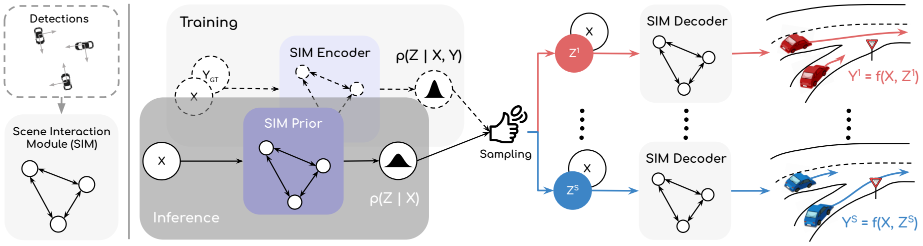


Fig. 4: **Our Implicit Latent Variable Model** encodes the scene into a latent space, from which it can efficiently sample multiple future realizations in parallel, each with socially consistent trajectories.

providing the local context for all actors, as proposed by [9]. As mentioned at the beginning of Section 3, this will be the input to our motion forecasting module. This contrasts with previous approaches (e.g., [11, 13, 45, 51]) that assume past trajectories for each actor are given. We refer the reader to the supplementary material for more details about our perception module, including the backbone architecture and detection parameterization.

Scene Interaction Module (SIM): This is a core building block of our encoder, prior, and decoder networks, as shown in Fig. 4. Once we have extracted individual actor features, we can frame the scene as a fully-connected interaction graph where each traffic participant is a node. Inspired by [9], we use a spatially-aware graph neural network to model multi-agent dynamics, as described in Alg. 1. Our SIM performs a single round of message passing to update the nodes' representation, taking into account spatiotemporal relationships.

Encoder: To approximate the true posterior latent distribution $P(Z|X,Y)$, we introduce an approximate posterior $q_\phi(Z|X,Y)$, implemented by our SIM and parameterized by ϕ. This network is also commonly known as recognition network, since it receives the target output variable Y as an input, and thus it can *recognize* the scene dynamics that are unobserved by the prior $p_\gamma(Z|X)$. Note that the encoder can only be used during training, since it requires access to the ground-truth future trajectories. We initialize the node representations as $h_n = \mathrm{MLP}(x_n \oplus \mathrm{GRU}(y_n))$, where $\oplus$ denotes concatenation along the feature dimension. After running one round of message passing, the scene interaction module predicts the distribution over scene latent variables $Z = \{z_1, z_2, \cdots, z_N\}$. We stress that despite anchoring each partition of the scene latent to an actor, each individual z_n contains information about the full scene, since each final node representation is dependent on the whole input X because of the message propagation in the fully-connected interaction graph.

Prior: The prior network $p_\gamma(Z|X)$ is responsible for approximating the prior distribution of the scene latent variable Z at inference time. Similar to the encoder, we model the scene-level latent space with our SIM, where the only difference is that the initial node representations in the graph propagation are $h_n = \mathrm{MLP}(x_n)$, since y_n is not available at inference time.

Algorithm 1 Scene Interaction Module (SIM)

```
Input: Initial hidden state for all of the actors in the scene $\{h_0, h_1, \cdots, h_N\}$. BEV coordinates (centroid and heading) of the detected bounding boxes $\{c_0, c_1, ..., c_N\}$.
Output: Feature vector per node $\{o_0, o_1, \cdots, o_N\}$.

Construct fully-connected interaction graph $G = (V, E)$ from detections
Compute pairwise coordinate transformations $\mathcal{T}(c_u, c_v)$, $\forall (u, v) \in E$
for $(u, v) \in E$ do        ▷ Compute message for every edge in the graph in parallel
    $m_{u \to v} = \text{MLP}\left(h_u, h_v, \mathcal{T}(c_u, c_v)\right)$
for $v \in V$ do        ▷ Update node states in parallel
    $a_v = \text{MaxPooling}\left(\{m_{u \to v} : u \in \mathbf{N}(v)\}\right)$        ▷ Aggregate messages from neighbors
    $h'_v = \text{GRU}\left(h_v, a_v\right)$        ▷ Update the hidden state
    $o_v = \text{MLP}\left(h'_v\right)$        ▷ Compute outputs
return $\{o_0, o_1, \cdots, o_N\}$
```

Deterministic Decoder: Recall that our scene latent has been partitioned into a distributed representation $Z = \{z_1, z_2, \cdots, z_N\}$. To leverage actor features and distributed latents from the whole scene, we parameterize the decoder with another SIM. We can then predict the s-th realization of the future at a scene level via message passing, where each actor trajectory y_n^s takes into account a sample from all the partitions of the scene latent $Z^s = \{z_1^s, \cdots, z_n^s\}$ as well as all actors' features X, enabling reasoning about multi-agent interactions such as car following, yielding, etc. More precisely, given each actor context x_n, we initialize its node representation for the decoder graph propagation as $h_n^s = \text{MLP}(x_n \oplus z_n^s)$. After a round of message passing in our SIM, $h_n^{\prime s}$ contains an updated representation of actor n that takes into account the underlying dynamics of the whole scene summarized in Z^s. Finally, the s-th trajectory sample for actor n is deterministically decoded $y_n^s = \text{MLP}(h_n^{\prime s})$ by the SIM output function, without additional sampling steps. The trajectory-level scene sample is simply the collection of all actor trajectories $Y^s = \{y_1^s, \ldots, y_N^s\}$. We can generate S possible futures for all actors in the scene in parallel by batching S scene latent samples.

In this fashion, our model implicitly characterizes the joint distribution over actors' trajectories, achieving superior scene-level consistency. In the experiments section we ablate the design choices in the encoder, prior and decoder, and show that although all of them are important, the deterministic decoder is the key contribution towards socially-consistent trajectories.

3.3 Learning

Our perception and prediction model can be trained end-to-end using stochastic gradient descent. In particular, we minimize a multi-task loss for detection and motion forecasting: $\mathcal{L} = \mathcal{L}_{\text{det}} + \lambda \cdot \mathcal{L}_{\text{forecast}}$

Detection: For the detection classification branch we employ a binary cross entropy loss with hard negative mining $\mathcal{L}_{\text{cla}}$. We select all positive examples from the ground-truth and 3 times as many negative examples. For box fitting, we apply a smooth ℓ_1 loss $\mathcal{L}_{\text{reg}}$ to each of the 5 parameters $(x_i, y_i, w_i, h_i, \phi_i)$ of the bounding boxes anchored to a positive example i. The overall detection loss is a linear combination $\mathcal{L}_{\text{det}} = \mathcal{L}_{\text{cla}} + \alpha \cdot \mathcal{L}_{\text{reg}}$.

Motion Forecasting: We adapt the variational learning objective of the CVAE framework [50] and optimize the evidence-based lower bound (ELBO) of the log-likelihood $\log P(Y|X)$. In our case, due to the deterministic decoder leading to an implicit distribution over Y, we use Huber loss ℓ_δ as the reconstruction loss, and reweight the KL term with β as proposed by [17]:

$$\mathcal{L}_{\text{forecast}} = \sum_{n}^{N} \sum_{t}^{T} \ell_\delta(y_n^t - y_{n,GT}^t) + \beta \cdot \text{KL}\left(q_\phi\left(Z|X, Y_{GT}\right) || p_\gamma\left(Z|X\right)\right)$$

where the first term minimizes the reconstruction error between all the trajectories in the scene $Y = \{y_n^t | \forall n, t\} = f_\theta(Z)$, $Z \sim q_\phi\left(Z|X, Y_{GT}\right)$ and their corresponding ground-truth Y_{GT}, and the second term brings the privileged *posterior* $q_\phi(Z|X, Y_{GT})$ and the approximate *prior* $p_\gamma(Z|X)$ distributions closer.

4 Experimental Evaluation

In this section, we first explain the metrics and baselines we use for evaluation. Next, we compare our model against state-of-the-art motion forecasting algorithms on predicting the future 5 second trajectories on two real-world datasets: ATG4D [54] and nuScenes [8] (see supplementary for details). Then, we measure the impact on motion planning. Finally, we carry out an ablation study to understand which part of our model contributes the most.

4.1 Scene Level Motion Forecasting Metrics

Previous methods use sample quality metrics at the actor level such as the popular minimum/mean average displacement error (minADE/meanADE). However, these metrics only evaluate the quality of the underlying marginal distribution per actor. For instance, minADE takes the trajectory sample that best fits the ground-truth of each actor independently, which does not measure the consistency between different actors' sample trajectories and can be easily cheated by predicting high entropy distributions that cover all the space but are not precise.

We propose scene-level sample quality metrics to evaluate how well the models capture the joint distribution over future outcomes. To this end, we define a scene-level counterpart of the popular minimum/mean average displacement error. We emphasize that in this context, each scene sample $s \in 1, ..., S$ is a

Type	Model	SCR$_{5s}$ (%)	min SFDE(m)	min SADE(m)	mean SFDE(m)	mean SADE(m)
Indep. Output	SpAGNN [9]	8.19	2.83	1.34	4.37	1.92
	RulesRoad [18]	6.66	2.71	1.32	4.21	1.84
	MTP [12]	3.98	1.91	0.95	3.11	1.37
	MultiPath [11]	4.41	1.97	0.95	3.14	1.36
	R2P2-MA [44]	4.63	2.13	1.09	3.27	1.49
Social Auto-regressive	SocialLSTM [1]	6.13	2.75	1.38	4.05	1.83
	NRI [26]	7.00	2.68	1.43	3.81	1.74
	ESP [45]	2.67	1.91	0.97	2.84	1.29
	MFP [51]	5.15	2.35	1.13	3.35	1.45
	ILVM	**0.70**	**1.53**	**0.76**	**2.27**	**1.02**

Table 1: **[ATG4D] Scene-level motion forecasting** ($S = 15$ samples)

collection of N future trajectories, one for each actor in the scene.

$$\text{minSADE} = \min_{s \in 1 \dots S} \frac{1}{NT} \sum_{n=1}^{N} \sum_{t=1}^{T} ||y_{n,GT}^{t} - y_{n,s}^{t}||^2$$

$$\text{meanSADE} = \frac{1}{NTS} \sum_{s=1}^{S} \sum_{n=1}^{N} \sum_{t=1}^{T} ||y_{n,GT}^{t} - y_{n,s}^{t}||^2$$

We also compute their final counterparts minSFDE and meanSFDE, which evaluate only the motion forecasts at the final timestep (i.e. at 5 seconds).

Furthermore, to evaluate the consistency of the motion forecasts we propose to measure the scene collision rate (SCR). It measures the percentage of trajectory samples that collide with any other trajectory in the same scene sample s. Two trajectory samples are considered in collision if the overlap between their future bounding boxes at any time step is higher than a small IOU threshold ε_{IOU}. To compute this, we first obtain the bounding boxes for future time steps $\{b_{i,s}^{t}\}$. The size of the bounding boxes are the same as their object detections and the future headings are extracted by finite differences on the trajectory samples.

$$\text{SCR}_T = \frac{1}{NS} \sum_{s=1}^{S} \sum_{i=1}^{N} \min \left(1, \sum_{j>i}^{N} \sum_{t=1}^{T} \mathbb{1} \left[IoU(b_{i,s}^{t}, b_{j,s}^{t}) > \varepsilon_{IOU} \right] \right)$$

Finally, to perform a fair comparison in motion forecasting metrics, which are evaluated on true positive detections, we follow [9] and operate the object detector at 90% recall point for all models in ATG4D and 80% in nuScenes.

4.2 Baselines

In this section, we discuss the state of the art motion forecasting models that we use as baselines. It is important to note that most baselines are designed for motion forecasting given perfect perception, i.e., ground-truth past trajectories. However, this is not realistic in self-driving vehicles, which rely on imperfect noisy

Type	Model	SCR_{5s} (%)	min SFDE(m)	min SADE(m)	mean SFDE(m)	mean SADE(m)
Indep. Output	SpAGNN [9]	7.54	2.07	1.00	3.85	1.82
	RulesRoad [18]	5.67	2.10	1.01	3.55	1.67
	MTP [12]	8.68	1.86	0.91	3.86	1.85
	MultiPath [11]	7.31	2.01	0.95	3.50	1.65
	R2P2-MA [44]	4.56	2.25	1.08	3.47	1.67
Social Auto-regressive	SocialLSTM [1]	6.45	2.71	1.33	4.20	2.05
	NRI [26]	5.98	2.54	1.28	3.91	1.88
	ESP [45]	5.09	2.16	1.07	3.46	1.67
	MFP [51]	4.94	2.74	1.30	4.11	1.95
	ILVM	**1.91**	**1.84**	**0.86**	**2.99**	**1.43**

Table 2: **[nuScenes] Scene-level motion forecasting** ($S = 15$ samples)

perception. Thus, we adapt them to the realistic setting by replacing their past trajectory encoders with our extracted actor features (see Fig. 3) and training end-to-end with our perception backbone (see supplementary for details).

Independent output: We benchmark against SpAGNN [9], MTP [12], MultiPath [11], RulesRoad [18], and R2P2-MA [44]. Since the trajectory sampling process from these models is independent per actor, we define a scene sample s by drawing one sample for each actor in the scene.

Social auto-regressive: We compare against SocialLSTM [1], ESP [45], MFP [51], and NRI [26]. It is worth sharing that for these baselines to achieve competitive results we had to perturb the ground-truth trajectories with white noise during training. This is because these models suffer from a distributional shift between training and inference, as explained in Section 2. We note that white noise was more effective than teacher forcing [28] or scheduled sampling [5].

4.3 Motion Forecasting Results

Experimental results for motion forecasting in the ATG4D dataset (with $S = 15$ samples) are shown in Table 1. Our ILVM outperforms the baselines across all metrics. Very notably, it **achieves a 75% reduction in collision rate** with respect to the strongest baseline in this metric (ESP [45]), thus highlighting the better characterization of the joint distribution across actors (which also translates into scene-consistent samples). Our model is also much more precise (20% reduction in meanSFDE) while exhibiting better coverage of the ground-truth data (19% reduction in minSFDE). We include an analysis of how the minSADE and minSFDE vary across different number of samples S in the supplementary.

Fig. 5 shows individual samples. We heuristically select the two most distinct samples for visualization to show diverse realizations of the future. The baseline models capture variations in individual actors' future, but do not capture the yielding interaction at the intersection, which our model does. In addition, Fig. 6

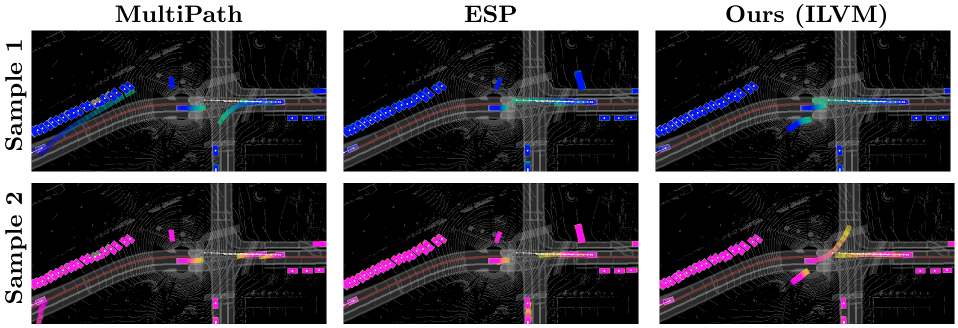


Fig. 5: **Scene-level samples**. Our latent variable model captures underlying scene dynamics at the intersection level (i.e. yield vs. go)

showcases the full distribution learned by the models. More concretely, this plot shows a Monte Carlo estimation of the marginal distribution per actor, where 50 samples are drawn from each model. Transparency in the plots illustrates the probability density at a given location. These examples support the same conclusion taken from the quantitative results and highlight the ability of our model to understand complex road geometries and the multi-modal behaviors they induce. This is particularly interesting since all models share the same representation of the environment and backbone architecture.

To show that our improvements generalize to a dataset with a different distribution of motions and road topologies, we validate our method on nuScenes. Table 2, shows that ILVM brings improvements over the baselines across all metrics. In particular, we observe significant gains in scene-consistency (SCR) and precision metrics (meanSADE and meanSFDE).

4.4 Motion Planning Results

To validate the system-level impact of different perception and prediction models, we use the state-of-the-art learnable motion planner of [49] to plan a trajectory for the SDV (τ_{SDV}):

$$\tau_{\mathrm{SDV}} = \arg\min_{\tau \in \mathrm{T}} \mathbb{E}_{p(Y|X)} \left[c(\tau, Y \setminus y_{\mathrm{SDV}}) \right] \approx \arg\min_{\tau \in \mathrm{T}} c(\tau, \left\{ Y^s \setminus y^s_{\mathrm{SDV}} : \forall s \in 1 \dots S \right\})$$

where $p(Y|X)$ is the distribution over future trajectories output by the perception and prediction model, T is a predefined set of SDV trajectories given the map and high-level route, and c is a costing function that measures safety and comfort taking into account the motion forecasts for the rest of the vehicles. More concretely, the motion planner receives a Monte Carlo estimate of the future trajectory distribution with $S = 50$ sample trajectories (see Fig.6) for every detected vehicle (excluding the SDV), which are considered obstacles in order to approximate the expected cost of plans $\tau \in \mathrm{T}$.

The experiments in Table 3 measure how different motion forecasts translate into the safety and comfort of the SDV trajectory (τ_{SDV}), an impact often

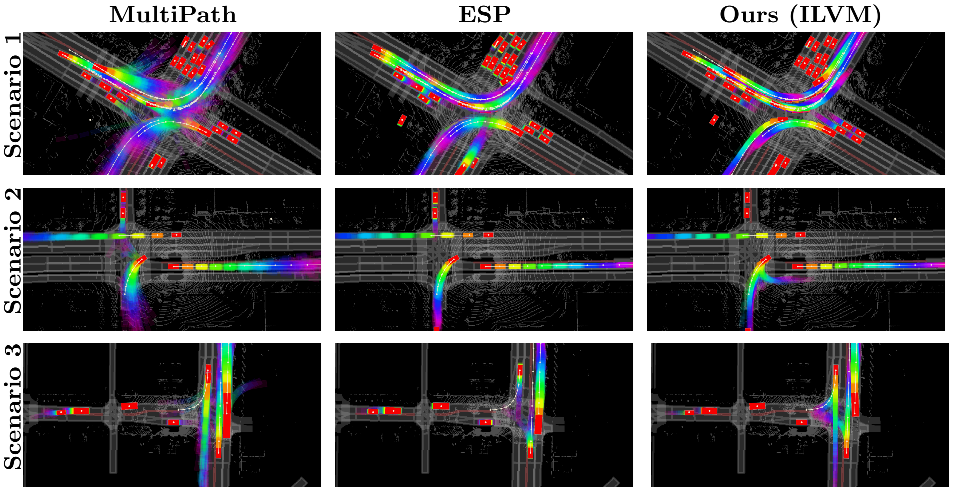


Fig. 6: **Motion forecasting visualizations of 50 samples**. Time is encoded in the rainbow color map ranging from red (0s) to pink (5s).

overlooked by previous works. Our motion forecasts (ILVM) enable the motion planner to execute significantly safer and more comfortable trajectories. We notice that the ego-motion plans make similar progress across models, but our approach produces the closest trajectories to the expert demonstrations (lowest ℓ_2 distance at 5 seconds into the future), while yielding much fewer collisions. We include planning qualitative results in our supplementary material.

4.5 Ablation Study

Implicit vs. Explicit Decoder: We ablate ILVM ($\mathcal{M}_0$ in Table 4) by replacing the proposed implicit decoder with an explicit decoder that produces a full covariance bi-variate Gaussian per waypoint, and the reconstruction loss with Negative Log Likelihood. This gives us $\mathcal{M}_1$, where ancestral sampling is used for inference: first sample latent, then sample output. Here, we can see that assuming conditional independence across actor at the output level significantly degrades all aspects of the motion forecasting performance. Most notably, the high scene collision rate shows that the samples are no longer socially consistent.

Learned vs. Fixed Prior: A comparison between $\mathcal{M}_0$ and $\mathcal{M}_2$ in Table 4 shows that using a learned prior network $P(Z|X)$ achieves a better precision diversity trade-off compared to using a fixed prior distribution of isotropic Gaussians.

ILVM architecture: In Table 4, $\mathcal{M}_3$ ablates the SIM encoder and prior networks by replacing them with MLPs that model $p(z_n|x_n)$ and $p(z_n|x_n, y_n)$ at the actor-level, respectively. $\mathcal{M}_4$ replaces the SIM decoder by an MLP per actor $y_n^s = \text{MLP}(X, z_n^s)$. Finally, $\mathcal{M}_5$ applies the changes in $\mathcal{M}_3$ and $\mathcal{M}_4$. These experiments show that both the graph based prior/encoder and decoder are important

Type	Model	Collision (% up to 5s)	L2 human (m @ 5s)	Lat. acc. (m/s^2)	Jerk (m/s^3)	Progress (m @ 5s)
Indep. Output	SpAGNN [9]	4.19	5.98	2.94	2.90	32.37
	RulesRoad [18]	4.04	5.83	2.84	2.76	32.50
	MTP [12]	3.10	5.67	2.83	2.66	33.14
	MultiPath [11]	3.30	5.58	2.73	2.57	32.99
	R2P2-MA [45]	3.71	5.65	2.84	2.53	33.90
Social Auto-regressive	SocialLSTM [1]	4.22	5.92	2.76	2.66	32.60
	NRI [26]	4.94	5.73	2.78	2.55	33.43
	ESP [45]	3.13	5.48	2.76	2.44	33.74
	MFP [51]	4.14	5.57	2.61	2.43	32.94
	ILVM	**2.64**	**5.33**	**2.59**	**2.30**	33.72

Table 3: **[ATG4D] System Level Performance (ego-motion planning)**

ID	Learned Prior	Implicit Output	SIM Encoder	SIM Decoder	SCR_{5s}	min SFDE	min SADE	mean SFDE	mean SADE
$\mathcal{M}_0$	✓	✓	✓	✓	**0.70**	**1.53**	**0.76**	**2.27**	**1.02**
$\mathcal{M}_1$	✓		✓	✓	8.46	2.66	1.31	4.17	1.80
$\mathcal{M}_2$		✓	✓	✓	1.10	1.53	0.76	2.43	1.08
$\mathcal{M}_3$	✓	✓		✓	1.03	1.57	0.78	2.42	1.08
$\mathcal{M}_4$	✓	✓	✓		1.52	1.67	0.81	2.44	1.09
$\mathcal{M}_5$	✓	✓			1.74	1.66	0.81	2.43	1.08

Table 4: **[ATG4D] Motion Forecasting Ablation Study** ($S = 15$ samples)

for our latent variable model. In particular, the large gap in scene level collision demonstrates that our proposed SIM encoder and decoder capture scene-level understanding that is not present in the ablations with independent assumptions at the latent or output level.

5 Conclusion and Future Work

We have proposed a latent variable model to obtain an implicit joint distribution over actor trajectories that characterizes the dependencies over their future behaviors. Our model achieves fast parallel sampling of the joint trajectory space and produces scene-consistent motion forecasts. We have demonstrated the effectiveness of our method on two challenging datasets by significantly improving over state-of-the-art motion forecasting models on scene-level sample quality metrics. Our method achieves much more precise predictions that are more socially consistent. We also show that our method produces significant improvements in motion planning, even though the planner does not make explicit use of the strong consistency of our scenes. We leave it to future work to design a motion planner to better utilize joint distributions over trajectories.

References

1. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social lstm: Human trajectory prediction in crowded spaces. In: Proceedings of the IEEE CVPR (2016)
2. Battaglia, P.W., Hamrick, J.B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gulcehre, C., Song, F., Ballard, A., Gilmer, J., Dahl, G., Vaswani, A., Allen, K., Nash, C., Langston, V., Dyer, C., Heess, N., Wierstra, D., Kohli, P., Botvinick, M., Vinyals, O., Li, Y., Pascanu, R.: Relational inductive biases, deep learning, and graph networks (2018)
3. Behbahani, F., Shiarlis, K., Chen, X., Kurin, V., Kasewa, S., Stirbu, C., Gomes, J., Paul, S., Oliehoek, F.A., Messias, J., et al.: Learning from demonstration in the wild. 2019 International Conference on Robotics and Automation (ICRA) (May 2019). https://doi.org/10.1109/icra.2019.8794412
4. Behrisch, M., Bieker, L., Erdmann, J., Krajzewicz, D.: Sumo–simulation of urban mobility: an overview. In: Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation. ThinkMind (2011)
5. Bengio, S., Vinyals, O., Jaitly, N., Shazeer, N.: Scheduled sampling for sequence prediction with recurrent neural networks. In: Advances in Neural Information Processing Systems. pp. 1171–1179 (2015)
6. Best, A., Narang, S., Pasqualin, L., Barber, D., Manocha, D.: Autonovi-sim: Autonomous vehicle simulation platform with weather, sensing, and traffic control. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 1161–11618 (2018)
7. Bhattacharyya, R.P., Phillips, D.J., Wulfe, B., Morton, J., Kuefler, A., Kochenderfer, M.J.: Multi-agent imitation learning for driving simulation. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (Oct 2018). https://doi.org/10.1109/iros.2018.8593758
8. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. arXiv preprint arXiv:1903.11027 (2019)
9. Casas, S., Gulino, C., Liao, R., Urtasun, R.: Spatially-aware graph neural networks for relational behavior forecasting from sensor data. arXiv preprint arXiv:1910.08233 (2019)
10. Casas, S., Luo, W., Urtasun, R.: Intentnet: Learning to predict intention from raw sensor data. In: Conference on Robot Learning (2018)
11. Chai, Y., Sapp, B., Bansal, M., Anguelov, D.: Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. arXiv preprint arXiv:1910.05449 (2019)
12. Cui, H., Radosavljevic, V., Chou, F.C., Lin, T.H., Nguyen, T., Huang, T.K., Schneider, J., Djuric, N.: Multimodal trajectory predictions for autonomous driving using deep convolutional networks. arXiv preprint arXiv:1809.10732 (2018)
13. Djuric, N., Radosavljevic, V., Cui, H., Nguyen, T., Chou, F.C., Lin, T.H., Schneider, J.: Motion prediction of traffic actors for autonomous driving using deep convolutional networks. arXiv preprint arXiv:1808.05819 (2018)
14. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: Proceedings of the 1st Annual Conference on Robot Learning. pp. 1–16 (2017)

15. Hardy, J., Campbell, M.: Contingency planning over probabilistic obstacle predictions for autonomous road vehicles. IEEE Transactions on Robotics (2013)
16. Henaff, M., Canziani, A., LeCun, Y.: Model-predictive policy learning with uncertainty regularization for driving in dense traffic. arXiv preprint arXiv:1901.02705 (2019)
17. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-vae: Learning basic visual concepts with a constrained variational framework.
18. Hong, J., Sapp, B., Philbin, J.: Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
19. Hoshen, Y.: Vain: Attentional multi-agent predictive modeling. In: Advances in Neural Information Processing Systems. pp. 2701–2711 (2017)
20. Hubmann, C., Schulz, J., Becker, M., Althoff, D., Stiller, C.: Automated driving in uncertain environments: Planning with interaction and uncertain maneuver prediction. IEEE Transactions on Intelligent Vehicles **3**(1), 5–17 (2018)
21. Huszár, F.: How (not) to train your generative model: Scheduled sampling, likelihood, adversary? arXiv preprint arXiv:1511.05101 (2015)
22. Ivanovic, B., Pavone, M.: The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2375–2384 (2019)
23. Jain, A., Casas, S., Liao, R., Xiong, Y., Feng, S., Segal, S., Urtasun, R.: Discrete residual flow for probabilistic pedestrian behavior prediction. arXiv preprint arXiv:1910.08041 (2019)
24. Kim, B., Kang, C.M., Kim, J., Lee, S.H., Chung, C.C., Choi, J.W.: Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). pp. 399–404. IEEE (2017)
25. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2013)
26. Kipf, T., Fetaya, E., Wang, K.C., Welling, M., Zemel, R.: Neural relational inference for interacting systems. arXiv preprint arXiv:1802.04687 (2018)
27. Klingelschmitt, S., Damerow, F., Eggert, J.: Managing the complexity of inner-city scenes: An efficient situation hypotheses selection scheme. In: 2015 IEEE intelligent vehicles symposium (IV). pp. 1232–1239. IEEE (2015)
28. Lamb, A.M., ALIAS PARTH GOYAL, A.G., Zhang, Y., Zhang, S., Courville, A.C., Bengio, Y.: Professor forcing: A new algorithm for training recurrent networks. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29. Curran Associates, Inc. (2016), `http://papers.nips.cc/paper/6099-professor-forcing-a-new-algorithm-for-training-recurrent-networks.pdf`
29. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 12697–12705 (2019)
30. Le, H.M., Yue, Y., Carr, P., Lucey, P.: Coordinated multi-agent imitation learning (2017)
31. Lee, N., Choi, W., Vernaza, P., Choy, C.B., Torr, P.H., Chandraker, M.: Desire: Distant future prediction in dynamic scenes with interacting agents. In: Proceedings of the IEEE CVPR (2017)

32. Li, L., Yang, B., Liang, M., Zeng, W., Ren, M., Segal, S., Urtasun, R.: End-to-end contextual perception and prediction with interaction transformer. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2020)
33. Liang, M., Yang, B., Chen, Y., Hu, R., Urtasun, R.: Multi-task multi-sensor fusion for 3d object detection. In: Proceedings of the IEEE CVPR (2019)
34. Liang, M., Yang, B., Hu, R., Chen, Y., Liao, R., Feng, S., Urtasun, R.: Learning lane graph representations for motion forecasting. In: ECCV (2020)
35. Luo, W., Yang, B., Urtasun, R.: Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In: Proceedings of the IEEE CVPR (2018)
36. Ma, J., Shao, W., Ye, H., Wang, L., Wang, H., Zheng, Y., Xue, X.: Arbitrary-oriented scene text detection via rotation proposals. IEEE Transactions on Multimedia (2018)
37. Ma, W.C., Huang, D.A., Lee, N., Kitani, K.M.: Forecasting interactive dynamics of pedestrians with fictitious play. In: Proceedings of the IEEE CVPR (2017)
38. Martinez, M., Sitawarin, C., Finch, K., Meincke, L., Yablonski, A., Kornhauser, A.: Beyond grand theft auto v for training, testing and enhancing deep learning in self driving cars (2017)
39. Okamoto, M., Perona, P., Khiat, A.: Ddt: Deep driving tree for proactive planning in interactive scenarios. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). pp. 656–661. IEEE (2018)
40. Osa, T., Pajarinen, J., Neumann, G., Bagnell, J.A., Abbeel, P., Peters, J., et al.: An algorithmic perspective on imitation learning. Foundations and Trends® in Robotics (2018)
41. Phan-Minh, T., Grigore, E.C., Boulton, F.A., Beijbom, O., Wolff, E.M.: Covernet: Multimodal behavior prediction using trajectory sets. arXiv preprint arXiv:1911.10298 (2019)
42. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE CVPR (2017)
43. Ratliff, N.D., Bagnell, J.A., Zinkevich, M.A.: Maximum margin planning. In: Proceedings of the 23rd international conference on Machine learning. pp. 729–736 (2006)
44. Rhinehart, N., Kitani, K.M., Vernaza, P.: R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 772–788 (2018)
45. Rhinehart, N., McAllister, R., Kitani, K., Levine, S.: PRECOG: PREdiction Conditioned On Goals in Visual Multi-Agent Settings. arXiv e-prints arXiv:1905.01296 (May 2019)
46. Ridel, D., Deo, N., Wolf, D., Trivedi, M.: Scene compliant trajectory forecast with agent-centric spatio-temporal grids. IEEE Robotics and Automation Letters (2020)
47. Ross, S., Gordon, G., Bagnell, D.: A reduction of imitation learning and structured prediction to no-regret online learning. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics. pp. 627–635 (2011)
48. Sadat, A., Ren, M., Pokrovsky, A., Lin, Y.C., Yumer, E., Urtasun, R.: Jointly learnable behavior and trajectory planning for self-driving vehicles. arXiv preprint arXiv:1910.04586 (2019)
49. Sadat, A., Ren, M., Pokrovsky, A., Lin, Y.C., Yumer, E., Urtasun, R.: Jointly learnable behavior and trajectory planning for self-driving vehicles. arXiv preprint arXiv:1910.04586 (2019)

50. Sohn, K., Lee, H., Yan, X.: Learning structured output representation using deep conditional generative models. In: Advances in neural information processing systems. pp. 3483–3491 (2015)
51. Tang, C., Salakhutdinov, R.R.: Multiple futures prediction. In: Advances in Neural Information Processing Systems. pp. 15398–15408 (2019)
52. Treiber, M., Hennecke, A., Helbing, D.: Congested traffic states in empirical observations and microscopic simulations. Physical review E (2000)
53. Yang, B., Guo, R., Liang, M., Sergio, C., Urtasun, R.: Exploiting radar for robust perception of dynamic objects. In: ECCV (2020)
54. Yang, B., Luo, W., Urtasun, R.: Pixor: Real-time 3d object detection from point clouds. In: Proceedings of the IEEE CVPR (2018)
55. Zeng, W., Luo, W., Suo, S., Sadat, A., Yang, B., Casas, S., Urtasun, R.: End-to-end interpretable neural motion planner. In: Proceedings of the IEEE CVPR (2019)
56. Zeng, W., Wang, S., Liao, R., Chen, Y., Yang, B., Urtasun, R.: Dsdnet: Deep structured self-driving network. In: ECCV (2020)
57. Zhou, Y., Sun, P., Zhang, Y., Anguelov, D., Gao, J., Ouyang, T., Guo, J., Ngiam, J., Vasudevan, V.: End-to-end multi-view fusion for 3d object detection in lidar point clouds. arXiv preprint arXiv:1910.06528 (2019)