Multimodal Trajectory Predictions for Autonomous Driving without a Detailed Prior Map

Atsushi Kawasaki Akihito Seki
Corporate R&D Center, Toshiba, Japan
{atsushi1.kawasaki | akihito.seki}@toshiba.co.jp

Abstract

Predicting the future trajectories of surrounding vehicles is a key competence for safe and efficient real-world autonomous driving systems. Previous works have presented deep neural network models for predictions using a detailed prior map which includes driving lanes and explicitly expresses the road rules like legal traffic directions and valid paths through intersections. Since it is unrealistic to assume the existence of the detailed prior maps for all areas, we use a map generated from only perceptual data (3D points measured by a LiDAR sensor). Such maps do not explicitly denote road rules, which makes prediction tasks more difficult. To overcome this problem, we propose a novel generative adversarial network (GAN) based framework. A discriminator in our framework can distinguish whether predicted trajectories follow road rules, and a generator can predict trajectories following it. Our framework implicitly extracts road rules by projecting trajectories onto the map via a differentiable function and training positional relations between trajectories and obstacles on the map. We also extend our framework to multimodal predictions so that various future trajectories are predicted. Experimental results show that our method outperforms other state-of-the-art methods in terms of trajectory errors and the ratio of trajectories that fall on drivable lanes.

1. Introduction

Predicting the future motion of surrounding vehicles is a crucial task for path planning by autonomous vehicles and for early detection of abnormal driving behavior. Safe autonomous driving requires robust trajectory predictions of surrounding vehicles in a wide variety of traffic scenarios.

For more accurate and long-term predictions, it is necessary to use maps of surrounding environments. Recent studies [6, 9, 17, 18] have tackled the task of predictions using a detailed prior map which includes lane centerlines or road boundaries, as shown in Figure 1-(b). Those methods

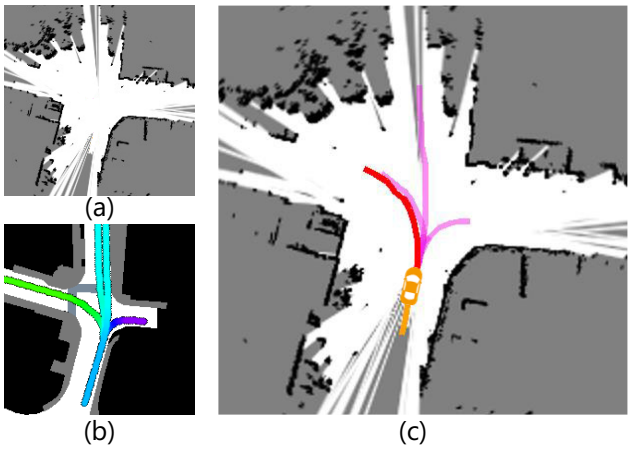

Figure 1. (a) A top-view image generated from perceptual data (LiDAR points) and (b) a detailed prior map. Map (b) includes lane information (with colors indicating lane directions) regarding road rules, but this information is not included in (a). We use only (a) as map information to predict multimodal trajectories following road rules. Subfigure (c) shows the result of our method. Orange, pink, and red lines respectively indicate the input, output, and ground-truth trajectories.

can predict accurate trajectories by using recurrent neural networks (RNN) or convolutional neural networks (CNN). However, it is unrealistic to assume availability of such maps for all areas, and maps may differ from the current situation due to construction work or traffic accidents.

In contrast, we predict trajectories using a map generated only from perceptual data. In our setting, we use an occupancy grid map (OGM) generated from 3D points measured by a LiDAR sensor. OGMs can represent the presence of obstacles in the area around an ego-vehicle, and they are commonly used for autonomous driving. A detailed prior map explicitly denotes road rules such as legal traffic directions and valid paths through an intersection. However, a perceptual-data-based map does not explicitly denote road rules, which makes prediction tasks even more difficult. In the example of Figure 1, it is necessary to learn from an

OGM road rules such as wide or tight turns at intersections.

To overcome this problem, we propose a novel GAN-based framework, which can distinguish whether predicted trajectories follow road rules and generate predicted trajectories that follow it. Positional relations between trajectories and obstacles on the map are important when extracting road rules from an OGM. For example, it is necessary to predict wide or tight turning trajectories by capturing an overall road shape from obstacles. However, since the representation method differs between trajectories (vector formats) and image maps (array formats), most conventional methods associate them by fully connected layers and impaired spatiality of the maps. We maintain spatiality by associating them through a differentiable function that projects predicted trajectories onto the map. We can implicitly extract road rules by introduce this function into the generator and the discriminator in our framework.

We also extend our framework to the multimodal prediction task. Prediction tasks generally involve high uncertainty due to diversity in behavioral characteristics and road structures. For example, when an oncoming vehicle approaches an intersection, we cannot uniquely determine its future trajectory. It is thus appropriate to represent prediction results as multimodal rather than unimodal distributions. If some predicted trajectories ignore road rules, false collision warnings may arise. We therefore introduce a novel adversarial loss function for multiple trajectories so that more predicted trajectories follow road rules.

Our method is evaluated using two datasets containing a large variety of real-world traffic scenarios. Experimental results show that the prediction performance of our method outperforms state-of-the-art methods. Furthermore, we quantitatively show that predicted trajectories by our method follow road rules more faithfully than do those by other methods.

The contributions of this paper are as follows. 1) We propose a framework for trajectory predictions following road rules without a detailed prior map. Our framework trains positional relations between obstacles and projected trajectories on the map, thereby implicitly extracting road rules. The discriminator can distinguish whether predicted trajectories follow road rules, and the generator can predict trajectories following it. 2) We extend our framework to multimodal predictions. By introducing a novel adversarial loss function for multiple trajectories, we can predict more trajectories that follow road rules. 3) Experiments on two datasets show that our method outperforms other state-of-the-art methods in terms of trajectory errors and the ratio of trajectories that fall on drivable lanes.

2. Related works

Vehicle trajectory prediction using deep models: Predictions of the future trajectories of surrounding vehicles have been actively studied. The many traditional approaches include constant velocity, Kalman filters [20, 21], Gaussian process regression models [33, 34], hidden Markov models [4, 13], and Bayesian networks [14, 27]. However, these approaches might not scale to cover prohibitively large variations in real-world traffic scenes.

In recent works, data-driven deep neural network models have resolved problems such as lack of expressiveness and complicated parameter tuning. In particular, RNNs and their variants, such as long short term memory (LSTM) [16] and gated recurrent units (GRU) [8], have been used for sequence prediction tasks. Most methods are built on the encoder-decoder framework [7] and feed sequential positional coordinates into the encoder and output predicted future positions from the decoder. Further developing this network structure, several studies have addressed the problem of modeling constraints from scene contexts [2, 3, 9, 12, 17, 26] or modeling social interactions among multiple agents [1, 15, 29, 31, 37]. In this paper, we focus on trajectory predictions based on the scene context.

Using map information: Most conventional methods use a 2D map which are expected to facilitate confident predictions of future trajectories in complicated scenarios. We can classify map formats into two types: prior maps and perceptual-data-based maps. Prior maps are generally highly-detailed-maps in which elements such as road areas, road markings, or lane centerlines are embedded. In [9, 17], spatial features are extracted from highly-detailed maps via a CNN to predict multimodal trajectories. By using vector maps of lanes, LAMP-Net [19] can handle any shapes and number of traffic lanes and predict both future trajectories along each lane and the probabilities of each lane being selected. Recently, Argoverse [6] and nuScenes [5], which are datasets for prediction tasks including highly-detailed maps, have been published. However, it is unrealistic to prepare such prior maps in everywhere.

A perceptual-data-based map is generated by projecting sensor data, such as those from cameras or LiDAR equipped on autonomous robots, onto a top-view image. Such maps are advantageous in that infrastructure upgrades are not necessary. Lee *et al*. [26] projected LiDAR points with semantic labels onto a top-view image used for predictions. In our problem setting, we use an OGM generated from LiDAR points. OGMs can directly represent drivable areas from sensor measurements. However, OGMs do not include road rules such as legal traffic directions and valid paths. We therefore present a method to predict trajectories following road rules by using adversarial training.

Adversarial training: Recent studies have applied adversarial training to prediction tasks. Social-GAN [15] proposed a new pooling method over all agents globally involved in a scene, thereby applying adversarial training to generation of a stochastic human behavior model. MATF-

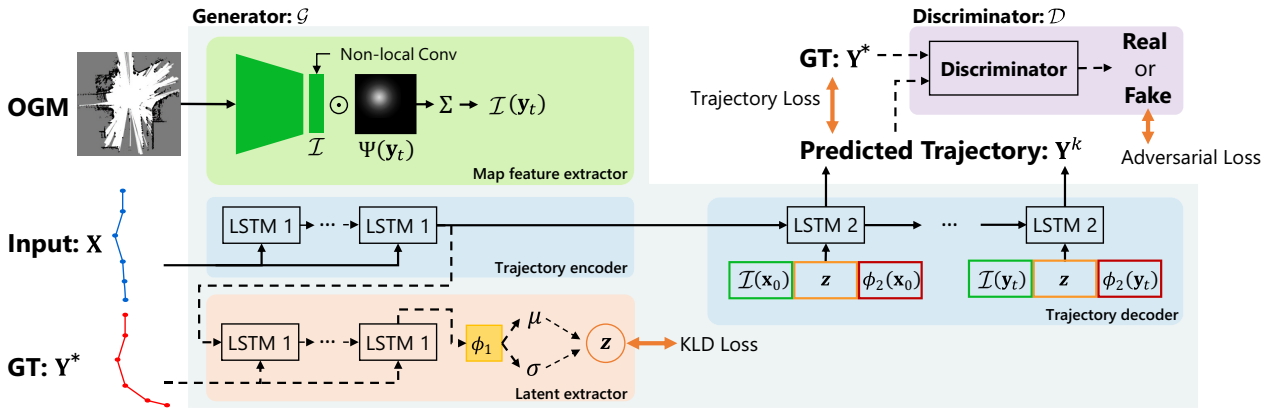

Figure 2. Overview of the proposed prediction framework, which consists of a generator $\mathcal{G}$ and a discriminator $\mathcal{D}$. Both solid and dotted arrows are used as forward propagation paths in the training phase, and only solid arrows are used in the inference phase.

GAN [37] used conditional generative adversarial training to capture comprehensive social and contextual information. GAIL-GRU [25] used generative adversarial imitation learning to learn a stochastic policy that reproduces human expert driving behaviors.

Inspired by these works, we introduce adversarial training for training of road rules. The discriminator in our framework can distinguish whether predicted trajectories follow road rules, and the generator can predict trajectories following it. It is important to understand spatial relations between trajectories and a map in order to extract road rules from them. However, conventional methods [15, 37] have associated them by fully connected layers and impaired map spatiality because their representation methods are different. We maintain map spatiality by associating them via a differentiable function that projects trajectories on the map. By introducing this function into the generator and the discriminator, we can implicitly extract road rules.

Multimodal predictions: Several works have addressed the problem of modeling multimodality in future motion. Deo *et al*. [10, 11] proposed single networks that predict different trajectories for each maneuver. However, these networks can only predict pre-defined maneuvers. To overcome this problem, previous studies [2, 3, 17, 18, 26] jointly modeled multiple future trajectories using a recurrent conditional variational auto encoder (CVAE) [23]. Our multimodal generative module is inspired by such CVAE-based methods. Additionally, in the task of multimodal predictions, we experimentally analyze the best loss function for enhancing the effectiveness of adversarial training.

3. Proposed method

In this section, we elaborate a novel algorithm for predicting multimodal trajectories that follow road rules. Figure 2 shows an overview of our method, which consists of a generator $\mathcal{G}$ and a discriminator $\mathcal{D}$. $\mathcal{G}$ predicts multiple trajectories using the observed trajectory and map information as input. $\mathcal{D}$ distinguishes whether the generated trajectories follow road rules. We describe the problem setting and the details of the main algorithm bellow.

3.1. Problem setting

We assume access to real-time perceptual data from sensors such as cameras or LiDAR equipped on an autonomous vehicle. We can obtain the vehicle state $\mathbf{x}_t = (u_t, v_t, \theta_t)$ at each time t, where u, v, θ are 2D positions and an angle in a Cartesian coordinate system, from vehicle detection and tracking systems using perceptual data. The current time is $t = 0$ and observed sequential states until the current time are defined as $\mathbf{X} = \{\mathbf{x}_{-T_{obs}}, ..., \mathbf{x}_0\}$. The output state at each time is defined as $\mathbf{y}_t = (u_t, v_t)$, and multiple sequential predicted states until $t = T_{pred}$ are defined as $\mathbf{Y}^k = \{\mathbf{y}_1^k, ..., \mathbf{y}_{T_{pred}}^k\}$, where k is a sampling ID. Ground truth (GT) is represented as $\mathbf{Y}^* = \{\mathbf{y}_1^*, ..., \mathbf{y}_{T_{pred}}^*\}$. The coordinate origin is set at (u_0, v_0), which is the last observed vehicle position.

Additionally, we can obtain an OGM $\mathcal{O}$ generated from LiDAR points. The ego-vehicle equipped with LiDAR is positioned at the center of the OGM. Each cell in the OGM represents occupancy probabilities $p_o \in [0, 1]$. The OGM in Fig. 1-(a) depicts obstacles, free space, and unobserved space in black, white, and gray, respectively.

3.2. Association of trajectory with map

Before explaining our network architecture, we describe the module which associates trajectories with a map. Since the representation method differs between trajectories (vector format) and maps (array format), conventional methods reduce the dimensions of arrays to match the vector format and associate them by fully connected layers. However, connections by fully connected layers cannot consider spatial structures of the map. We therefore effectively pass fea-

tures between them by converting trajectories into a probability map inspired by the soft-argmax layer [28].

We project vehicle positions onto the OGM using a probability map which has the same format as the map. Assuming knowledge of a projection equation $\pi(\cdot)$ from the trajectory coordinate system to OGM coordinate system, the position on the OGM is represented as $(\hat{u}_t, \hat{v}_t) = \pi(\mathbf{y}_t) = \pi(u_t, v_t)$. The probability map Ψ which has a maximum value at $(\hat{u}_t, \hat{v}_t)$ is calculated as follows:

$$\Psi(\mathbf{y}_t) = PDF(\hat{u}_t - \mathrm{I}_W) \odot PDF(\hat{v}_t - \mathrm{I}_H), \quad (1)$$

where I_W and I_H are respectively matrices whose elements are its indices along the x-axis and y-axis, $PDF(\cdot)$ is a probability density function of normal Gaussian distribution, and a hadamard operation shows element-wise multiplication. Figure 3 depicts a graphical representation of Eq. (1). This differentiable module maintains map spatiality because the formats of trajectories and map are uniformed. We introduce it into our $\mathcal{G}$ and $\mathcal{D}$.

3.3. Generator

Our generator $\mathcal{G}$, as shown in Figure 2, combines LSTMs with a CVAE encoder-decoder architecture following [3, 26]. Our goal is for CVAE to learn the distribution $p(\mathbf{Y}^k|\mathbf{X}, \mathcal{O})$ of multiple outputs $\mathbf{Y}^k$ conditioned on the input trajectory $\mathbf{X}$ and the OGM $\mathcal{O}$ by introducing latent variables z. In the training phase, various trajectories $\mathbf{Y}^*$ are encoded into latent variables z. In the inference phase, z are randomly sampled from the latent space and decoded through the decoder module to generate a prediction hypothesis. This framework can model stochastic multimodality in the prediction task.

In our implementation, encoder LSTM (LSTM 1 in Figure 2) reads an input state $\mathbf{X}$, and a latent vector based on $\mathbf{X}$ and $\mathbf{Y}^*$ is created using the same LSTM (LSTM 1). The output of LSTM 1 at T_{pred} passes through the embedding function (ϕ_1) to generate both mean μ_z and standard deviation σ_z over z. The distribution of z is fitted as a Gaussian distribution and is regularized by the KL divergence in the training phase. $\mathbf{Y}^k$ is reconstructed by randomly sampling the latent variable from a Gaussian distribution. Since back-propagation is not possible through random sampling, we adopt the re-parameterization trick [24] to make it differentiable.

Additionally, our framework can more effectively pass map features to the trajectories. Map features are extracted from $\mathcal{O}$ by VGG-16 [32] like model, and we add a non-local convolution layer [35] to the final layer in order to take into account the global characteristics. This feature map is defined as $\mathcal{I}$. Most conventional methods input the flattened map features to the decoder LSTM, but we recurrently input the map features corresponding to the trajectory position $\mathbf{y}_t$ to the decoder LSTM (LSTM 2 in Figure 2) in order

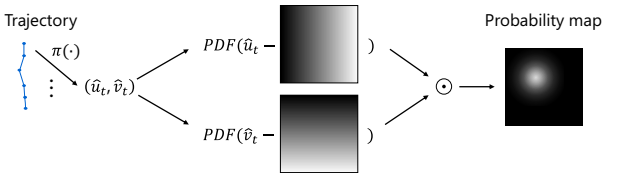

Figure 3. Graphical representation of Eq. (1). The input is a 2D coordinate vector and the output 2D array is a probability map with a peak at the input coordinate. The gray-scale gradient images show I_W and I_H in Eq. (1).

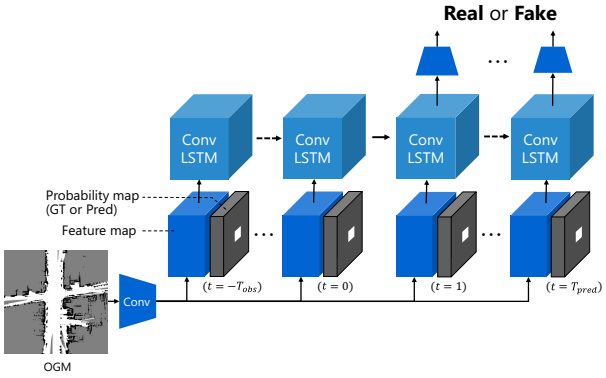

Figure 4. Network architecture of our proposed discriminator $\mathcal{D}$.

to maintain the spatial structure. The map features corresponding to the position can be extracted by multiplying $\mathcal{I}$ by the probability map $\Psi(\mathbf{y}_t)$ and calculating the sum of its elements. This process is synonymous with bilinear interpolation with a wide receptive field. Finally, LSTM 2 is recurrently conditioned on trajectory feature $\phi_2(\mathbf{y}_t)$, map feature $\mathcal{I}(\mathbf{y}_t)$, and sampled latent variables z. The decoder output at each time step is transformed into the predicted position via a common linear embedding function.

3.4. Discriminator

We introduce a novel discriminator $\mathcal{D}$ that distinguishes whether generated trajectories follow the road rules. Positional relations between trajectories and obstacles / free spaces on the OGM is key information to distinguish real or fake trajectories. Our discriminator can train their positional relations with maintaining spatiality by projecting trajectories onto the map through a probability map. Figure 4 shows the detailed architecture of $\mathcal{D}$. First, each position of the input, output, and GT trajectory is converted to the probability map. Each probability map from $t = -T_{obs}$ to T_{pred} and the feature map given by the OGM are concatenated in the channel and are input to Convolutional LSTM [36]. Each output of the Convolutional LSTM from $t = 0$ to T_{pred} is converted via weight-shared convolution layers into a probability that indicates a real or fake future trajectory. We train $\mathcal{D}$ such that $\mathcal{D}(\mathbf{X}, \mathbf{Y}^*) = 1$ if the input trajectory is real (GT trajectory), and $\mathcal{D}(\mathbf{X}, \mathbf{Y}^k) = 0$ if the input trajectory is fake (generated trajectory).

3.5. Training

$\mathcal{G}$ and $\mathcal{D}$ are simultaneously trained in a two-player min-max game framework which most GAN models employed. As more predicted trajectories are supposed to follow road rules, we formulate a loss function L_G for generator $\mathcal{G}$ as follows:

$$L_G = \lambda_1 L_{trj} + \lambda_2 L_{kld} + \lambda_3 L_{adv}, \tag{2}$$

$$L_{trj} = \min_k \left(\frac{1}{T_{pred}} \sum_{t=1}^{T_{pred}} \|\mathbf{y}_t^k - \mathbf{y}_t^*\|_2 \right), \tag{3}$$

$$L_{adv} = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{T_{pred}} (1 - \log \mathcal{D}(\mathbf{X}, \mathbf{Y}^k)), \tag{4}$$

where L_{trj}, L_{kld}, and L_{adv} are respectively the trajectory loss, the KL divergence loss, and the adversarial loss, and λ_1, λ_2, and λ_3 are weighting parameters. We use $\lambda_1 = 1.0$, $\lambda_2 = 3.0$, and $\lambda_3 = 1.0$ in our experiments. L_{trj} takes the L2 loss between the predicted states and GT at each time step. In L_{trj}, we use only the sample k with the smallest loss, which is called BMS-loss in [3]. BMS-loss allows predictions of diverse trajectories. On the other hand, L_{adv} is calculated using all samples because we expect all predicted trajectories that follow road rules.

When $\mathcal{D}$ is trained, the following function is maximized:

$$L_D = \max_{\mathcal{D}} \frac{1}{T_{pred}} \Big(\log \mathcal{D}(\mathbf{X}, \mathbf{Y}^*) + (1 - \log \mathcal{D}(\mathbf{X}, \mathbf{Y}^{k'})) \Big), \tag{5}$$

where k' is an index randomly selected from $\{1, ..., K\}$, and one of the generated multiple trajectories is used to calculate the discriminator loss. We use random sampling for two reasons. One reason is that this equalizes the balance between real and fake samples, which is generally sensitive for training on GAN. The other reason is that it is impossible to select a trajectory ignoring road rules from the loss magnitude. In order to make more accurate $\mathcal{D}$, fake samples that ignore it are necessary for training as well as true samples. However, a trajectory with the highest or lowest L_{trj} does not necessarily follow road rules. For example, in the lower right result in Figure 7, the loss magnitude of the trajectory that exists between straight and turning lanes is intermediate between ones of trajectories on these lanes. Random selection allows such samples to be used to train $\mathcal{D}$. Experiments described below quantitatively show that our calculation method of L_{adv} and L_D achieves the best performance.

3.6. Implementation

We train both $\mathcal{G}$ and $\mathcal{D}$ using Adam solver [22] with learning rate 0.0005 and momentum parameters $\beta_1 = 0.5$, $\beta_2 = 0.999$. The LSTMs 1, 2, Convolutional LSTM and the latent vector have 16 dimensions. The feature map $\mathcal{I}$ in $\mathcal{G}$ of size $40 \times 40 \times 16$ are extracted through convolutional layers with kernel size 3×3 through the ReLU activation and max pooling layers. The feature map in $\mathcal{D}$ are the same-size as $\mathcal{I}$ and extracted through convolutional layers with leaky ReLU activation and average pooling layers. The output of Convolutional LSTM is converted to a 16-dimensional vector via global average pooling and to a probability via fully connected layers with a sigmoid function. Our method is implemented using Tensorflow.

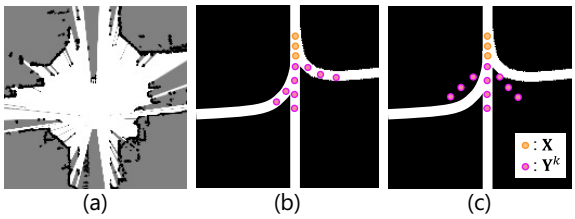

Figure 5. Illustrations of the PoP metric. The lane centerlines corresponding to OGM image (a) are drawn in (b) and (c). The PoP in (b) is higher than ones in (c) because predicted trajectories in (b) fall on the lanes.

4. Experiments

4.1. Datasets

We evaluate our method on our dataset and a public dataset including a wide variety of real-world traffic scenarios. Our dataset provides sensor data, including images and LiDAR point clouds, of driving scenes in a dense urban area of Tokyo. Inputs and GT trajectories are obtained by detecting and tracking 3D bounding boxes of vehicles and projecting center locations of bounding boxes into world coordinates. Input and output sequence lengths are the past 2 seconds and the future 4 seconds at 10 Hz. This dataset consists of 66 different intersection scenarios, and we trained and tested our model and the baseline methods using 55 and 11 scenarios, respectively. The number of sequences is 11,242 for training and 2,319 for testing.

The nuScenes dataset [5] is a public dataset for vehicle trajectory prediction tasks. It consists of raw sensor data, such as images, LiDAR, and GNSS, of driving scenes in Boston and Singapore, providing the position of all traffic agents. The input and output sequence are the past 2 seconds and future 6 seconds at 2 Hz. We extracted other vehicles within 64 m from an ego-vehicle, preparing 20,327 sequences from the training set and 5,399 sequences from the validation set.

For both datasets, we created an OGM of size 320 $\times$ 320 with a cell size of 0.4 m and max distance of 64 m. OGMs are composited from 2 seconds of previous scans. Both datasets provide the lane centerlines which we used

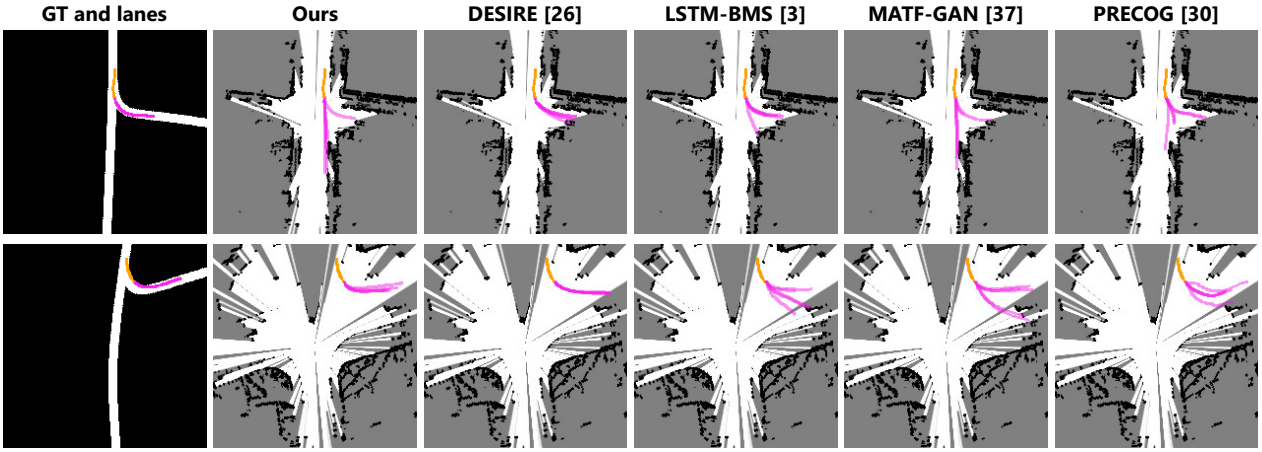

Figure 6. Prediction results by our method and four previous methods on our dataset. Input and output trajectories are shown in orange and pink, respectively. The first column shows GT trajectories and driving lanes (white areas).

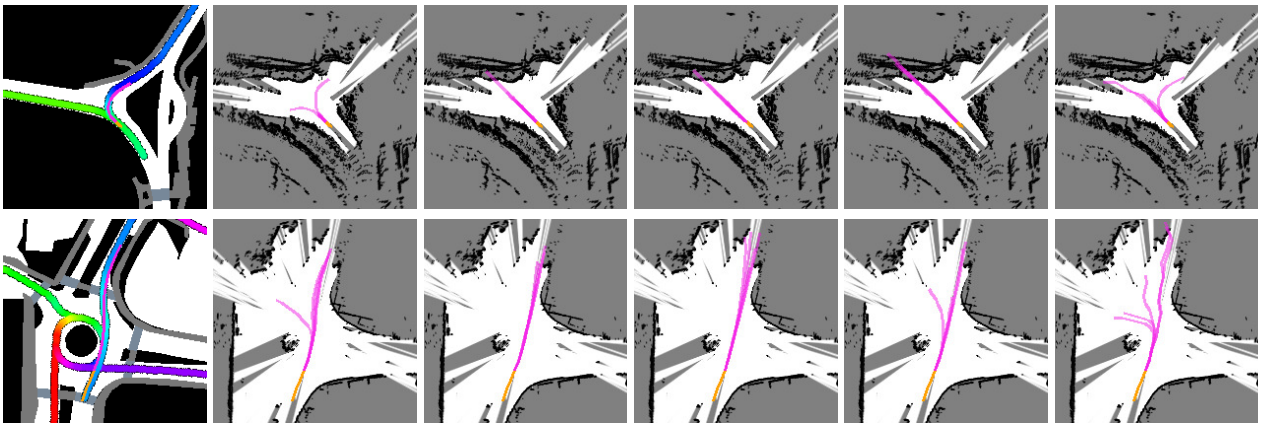

Figure 7. Prediction results by our method and previous methods on the nuScenes dataset [5], similar to Figure 6. In the first column, color paths show driving lanes and path colors show lane directions.

only for evaluations whether the predicted trajectories are on the lanes.

4.2. Metrics

We evaluate our methods by three metrics: the minimum of average displacement errors over K samples (mADE_K), the minimum of final displacement errors (mFDE_K), and the prediction on path (PoP).

mADE_K and mFDE_K: These metrics have been used in many prior works [6, 12, 15, 17, 26, 37] for multimodal trajectory predictions. The mADE_K is the minimum of average prediction performance along the trajectory over K samples, while the mFDE_K considers only the minimum prediction precision at the end points, as follows:

$$\text{mADE}_K = \min_{k\in\{1,\dots,K\}} \frac{1}{T_{pred}} \sum_{t=1}^{T_{pred}} \|\mathbf{y}_t^* - \mathbf{y}_t^k\|_2 \quad (6)$$

$$\text{mFDE}_K = \min_{k\in\{1,\dots,K\}} \|\mathbf{y}_{T_{pred}}^* - \mathbf{y}_{T_{pred}}^k\|_2. \quad (7)$$

It can be seen that some predicted trajectories are closer to GT as they have smaller error. However, these metrics do not penalize implausible predicted trajectories ignoring road rules. Thus, they are insufficient for evaluating the performance of multimodal predictive distributions.

PoP_K: Inspired by [12], we measure the percentage of all predicted trajectory location falling on drivable lanes. We create a mask of drivable lane areas from the lane centerline included in the datasets, shown in Figure 5. It can be seen that predicted trajectories follow road rules more faithfully as PoP_K is higher.

4.3. Baselines

We compare our methods with the following baselines:

DESIRE [26]: This is a CVAE-based multimodal trajectory prediction method. This method predicted trajectories using past trajectories and a top-view image onto which LiDAR points with semantic labels are projected. In our experiments, we replace the top-view image with the OGM. Each

Methods	Our dataset						The nuScenes dataset [5]					
	mFDE [m] ↓		mADE [m] ↓		PoP [%] ↑		mFDE [m] ↓		mADE [m] ↓		PoP [%] ↑	
	$K=3$	$K=5$	$K=3$	$K=5$	$K=3$	$K=5$	$K=3$	$K=5$	$K=3$	$K=5$	$K=3$	$K=5$
DESIRE [26]	4.06	4.00	1.95	1.92	76.4	76.3	6.95	5.93	3.00	2.59	68.3	68.4
LSTM-BMS [3]	3.70	2.93	1.82	1.52	70.7	71.0	6.94	5.42	3.06	2.45	63.1	63.2
MATF-GAN [37]	3.58	2.89	1.80	**1.48**	73.1	73.0	7.29	5.47	3.17	2.42	65.2	65.0
PRECOG [30]	3.72	2.97	1.82	1.53	72.5	72.7	6.92	5.21	3.16	2.45	69.7	69.9
PRECOG [30] w/ $\mathcal{D}$	3.59	2.88	1.80	1.49	74.8	74.9	6.80	5.09	3.03	2.36	70.2	70.3
Ours w/o $\mathcal{D}$	3.64	2.90	1.80	1.51	74.2	74.3	6.56	**4.92**	2.93	2.29	70.6	70.7
Ours	**3.54**	**2.85**	**1.78**	1.49	**77.2**	**77.2**	**6.45**	4.97	**2.86**	**2.28**	**74.4**	**74.4**

Table 1. Quantitative results of all methods on two datasets. The best results are shown in bold.

predicted trajectory tends to be similar because this method uses average trajectory losses over multiple samples.

LSTM-BMS [3]: This is a CVAE-based multimodal prediction method. This method introduces BMS loss, which uses the minimum loss over multiple samples, to provide diverse predictions. Our method also uses BMS loss. In LSTM-BMS, map features are flattened and concatenated with trajectory features.

MATF-GAN [37]: This is a conditioned GAN-based multimodal predictions method. In this generator and discriminator, map features are flattened and connected to trajectory features. This method used average trajectories loss for multiple samples as in DESIRE, but we replace its loss with BMS-loss for easy comparison with our method.

PRECOG [30]: This is a CVAE-based multimodal prediction method. This method extracts spatial features from the feature map by bilinear interpolation centered on the predicted position at each time. We evaluated the performance of this method. Additionally, in order to know our discriminator works on other network, we checked RORECOG with our discriminator.

4.4. Qualitative results

Figure 6 and 7 show prediction results by our method and baseline methods on our dataset and the nuScenes dataset [5], respectively. We can see that our method predicts diverse multimodal trajectories, such as going straight and turning left and right, depending on the intersection geometries. Additionally, our method predicts trajectories exactly on lanes more frequently than other methods do.

In Figure 6, since trajectories predicted by other methods are on the white areas (free spaces) on the OGM, these methods can also recognize that there are no obstacles in the white areas. However, some predicted trajectories would collide with obstacles (black areas) if their trajectories were naturally extended in a few seconds, suggesting that these methods cannot accurately extract road rules. MATF-GAN [37] can also predict trajectories following road rules at a typical intersection (the first row in Figure 6), but the predicted trajectories in the second rows are not on the lanes. These results show the effects of introducing a GAN architecture to prediction tasks, and our proposed discriminator further contributes to learning road rules by maintaining spatiality.

In the first row in Figure 7, predicted trajectories by our method and PRECOG [30] are in white areas, but predictions by other methods (DESIRE [26], LSTM-BMS [3], MATF-GAN [37]) collide with obstacles. These previous methods cannot handle rare scenes such as forked roads because they flatten map information and lose spatiality. In contrast, our method and PRECOG [30] recurrently give the decoder LSTM map information corresponding to predicted positions, so they can recognize such areas without obstacles. The second row in Figure 7 is a roundabout scenario. Our method can predict trajectories following road rules even in such a scenario.

Figure 8 shows failure examples in which some predicted trajectories by our method are not on lanes. Since most OGM cells around the target vehicles are "unobserved" (gray areas), the map information is insufficient. In such scenarios, the variability of predicted trajectories becomes large.

4.5. Quantitative results

Table 1 shows mADE_K, mFDE_K, and PoP_K by all methods. Our method outperforms the other methods for most metrics on the two datasets. In particular, we can see that our method has the highest PoP, which shows that the predicted trajectories by our method follow road rules more faithfully than do those by other methods.

There is an overall tendency for large differences in mFDEs (and mADEs) between the two datasets because our dataset and the nuScenes dataset [5] set different prediction horizons (4 and 6 seconds), and the former has primarily typical intersection scenarios while the latter has more varied traffic scenarios. When K is changed from 3 to 5, mFDE and mADE become smaller. That is because a larger number of candidates allow more diverse predictions. The variation range on DESIRE [26] is small because it does not use the loss that expresses diversity. On the other hand, chang-

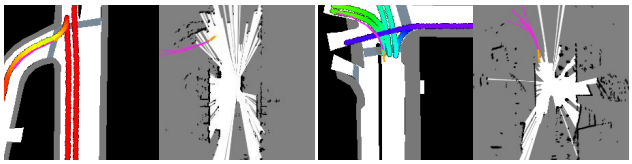

Figure 8. Examples of prediction failure by our method on the nuScenes dataset [5].

Methods	2 sec		4 sec		6 sec	
	mFDE	PoP	mFDE	PoP	mFDE	PoP
Ours w/o $\mathcal{D}$	0.94	91.6	2.59	81.8	4.92	70.7
Ours	0.93	92.1	2.61	84.0	4.97	74.4

Table 2. The mFDE and PoP of our method over future time steps on the nuScenes dataset [5] with $K = 5$.

Methods	K= 1		K= 5		K= 10	
	mFDE	PoP	mFDE	PoP	mFDE	PoP
Ours w/o $\mathcal{D}$	12.32	71.1	4.92	70.7	3.81	70.7
Ours	11.65	74.3	4.97	74.4	3.77	74.4

Table 3. The mFDE and PoP of our method on the nuScenes dataset [5] at 6 seconds for several K settings.

Methods	mFDE [m] ↓	PoP [%] ↑
$\mathcal{G}$: mean, $\mathcal{D}$: random (ours)	**4.97**	**74.4**
$\mathcal{G}$: mean, $\mathcal{D}$: mean	5.17	71.9
$\mathcal{G}$: mean, $\mathcal{D}$: min	5.09	69.9
$\mathcal{G}$: random,$\mathcal{D}$: random	5.01	72.0
$\mathcal{G}$: random, $\mathcal{D}$: mean	5.01	71.4
$\mathcal{G}$: random, $\mathcal{D}$: min	5.09	71.4
$\mathcal{G}$: min, $\mathcal{D}$: random	5.14	68.3
$\mathcal{G}$: min $\mathcal{D}$: mean	5.18	68.5
$\mathcal{G}$: min, $\mathcal{D}$: min	5.04	71.1

Table 4. Ablation studies on adversarial loss for multiple samples. "Mean" indicates that L_{adv} or L_D is calculated using the average of all samples, "random" indicates that losses are calculated using a sample randomly selected from all samples, and "min" indicates that losses are calculated using only a sample corresponding to ones with the smallest trajectory loss L_{trj}, as with BMS-loss [3]. These ablation studies were performed on the nuScenes dataset [5] with $K = 5$.

ing K results in nearly no change in PoPs for most methods. We believe this is because three in five are the same samples and are used to calculate the metrics. Additionally, trained models do not predict trajectories completely randomly, but rather with regularity based on the map.

We next compare our method with PRECOG [30]. Comparing both models without $\mathcal{D}$ (the 4th and 6th rows in Table 1), we can see that our method is superior under all metrics. Since our $\mathcal{G}$ uses non-local convolution [35] and a probability map, the receptive field in the map feature extractor of our $\mathcal{G}$ is wider than that in PRECOG. Our method can thus consider the entire map and obtain more accurate results. The 5th and 7th rows in Table 1 show the results of adding our $\mathcal{D}$ to these models. Prediction performances are improved in both models, so the effectiveness of $\mathcal{D}$ is clarified.

We conducted further experiments varying K and future time steps, as shown in Tables 2 and 3.

4.6. Ablation studies on the adversarial loss

In our adversarial loss for multiple samples, L_{adv} in Eq. (4) uses the mean of all samples $\{1, ..., K\}$ and L_D in Eq. (5) uses randomly selected k'. To perform ablation studies on the adversarial loss for multiple samples, we compare the combinations shown in Table 4.

Our proposed combination of losses provided the highest prediction performance (the first row in Table 4). Expecting all predicted trajectories to follow road rules, we chose the "mean" for L_{adv}, which allowed losses to back-propagate across all trajectories. However, performance did not be improved without using a randomly selected sample for L_D. Random selection provides two advantages: the real-fake quantitative balance is equalized because GANs learning is generally sensitive to the balance of the samples, and predicted trajectories ignoring road rules should be used for training. If L_D is calculated using only the sample with the smallest trajectory loss, only predicted trajectories close to GT are used and the remaining trajectories are ignored for $\mathcal{D}$. In order to create more accurate $\mathcal{D}$, fake samples that ignore road rules are necessary for training as well as true samples. Random selection allows such fake samples to be used to train $\mathcal{D}$. Our proposed combination of losses is therefore effective.

5. Conclusion

We tackled the problem of predicting multimodal vehicle trajectories using only perceptual data without a detailed prior map. Since perceptual-data-based maps do not explicitly denote road rules, the prediction task becomes more difficult. We therefore proposed a novel GAN-based framework. The discriminator in our framework can distinguish whether predicted trajectories follow road rules, and the generator can predict trajectories that follow it. We maintained spatiality by associating trajectories and the map through a probability map. By training positional relations trajectories and obstacles / free spaces on the map, our framework allowed implicitly extracting road rules. We also extended our framework to multimodal prediction tasks, introducing a novel adversarial loss function for multiple trajectories so that more predicted trajectories followed road rules. Experiments on two datasets showed that our method outperformed other state-of-the-art methods. We also quantitatively showed that predicted trajectories by our method followed road rules more faithfully than did those by other methods.

References

[1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social LSTM: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971, 2016. 2

[2] A. Bhattacharyya, M. Hanselmann, M. Fritz, B. Schiele, and C. N. Straehle. Conditional flow variational autoencoders for structured sequence prediction. *arXiv preprint arXiv:1908.09008*, 2019. 2, 3

[3] A. Bhattacharyya, B. Schiele, and M. Fritz. Accurate and diverse sampling of sequences based on a "best of many" sample objective. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8485–8493, 2018. 2, 3, 4, 5, 7, 8

[4] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 994–999, 1997. 2

[5] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11621–11631, 2020. 2, 5, 6, 7, 8

[6] M. T. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, et al. Argoverse: 3D tracking and forecasting with rich maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8748–8757, 2019. 1, 2, 6

[7] K. Cho, B. V. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 2

[8] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 2

[9] 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 *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2090–2096, 2019. 1, 2

[10] N. Deo and M. M. Trivedi. Convolutional social pooling for vehicle trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1468–1476, 2018. 3

[11] N. Deo and M. M. Trivedi. Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 1179–1184, 2018. 3

[12] N. Deo and M. M. Trivedi. Trajectory forecasts in unknown environments conditioned on grid-based plans. *arXiv preprint arXiv:2001.00735*, 2020. 2, 6

[13] J. Firl, H. Stübing, S. A. Huss, and C. Stiller. Predictive maneuver evaluation for enhancement of car-to-x mobility data. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 558–564, 2012. 2

[14] T. Gindele, S. Brechtel, and R. Dillmann. Learning driver behavior models from traffic observations for decision making and planning. *IEEE Intelligent Transportation Systems Magazine*, 7(1):69–79, 2015. 2

[15] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2255–2264, 2018. 2, 3, 6

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2

[17] J. Hong, B. Sapp, and J. Philbin. Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8454–8462, 2019. 1, 2, 3, 6

[18] Y. Hu, W. Zhan, L. Sun, and M. Tomizuka. Multi-modal probabilistic prediction of interactive behavior via an interpretable model. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 557–563, 2019. 1, 3

[19] A. Kawasaki and A. Seki. Multimodal trajectory predictions for urban environments using geometric relationships between a vehicle and lanes. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 9203–9209, 2020. 2

[20] A. Kawasaki and T. Tasaki. Trajectory prediction of turning vehicles based on intersection geometry and observed velocities. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 511–516, 2018. 2

[21] B. Kim and K. Yi. Probabilistic and holistic prediction of vehicle states using sensor fusion for application to integrated vehicle safety systems. *IEEE Transactions on Intelligent Transportation Systems*, 15(5):2178–2190, 2014. 2

[22] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[23] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Proceedings of advances in neural information processing systems (NIPS)*, pages 3581–3589, 2014. 3

[24] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 4

[25] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer. Imitating driver behavior with generative adversarial networks. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 204–211, 2017. 3

[26] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. S. Torr, and M. Chandraker. DESIRE: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 336–345, 2017. 2, 3, 4, 6, 7

[27] S. Lefèvre, C. Laugier, and J. Ibañez-Guzmán. Exploiting map information for driver intention estimation at road intersections. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 583–588, 2011. 2

[28] D. C. Luvizon, D. Picard, and H. Tabia. 2D/3D pose estimation and action recognition using multitask deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5137–5146, 2018. 4

[29] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel. Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14424–14432, 2020. 2

[30] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine. PRECOG: Prediction conditioned on goals in visual multi-agent settings. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2821–2830, 2019. 7, 8

[31] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1349–1358, 2019. 2

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4

[33] Q. Tran and J. Firl. Modelling of traffic situations at urban intersections with probabilistic non-parametric regression. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 334–339, 2013. 2

[34] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):283–298, 2007. 2

[35] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7794–7803, 2018. 4, 8

[36] S. H. I. Xingjian, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Proceedings of advances in neural information processing systems (NIPS)*, pages 802–810, 2015. 4

[37] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12126–12134, 2019. 2, 3, 6, 7