LaPred: Lane-Aware Prediction of Multi-Modal Future Trajectories of Dynamic Agents

Anonymous CVPR 2021 submission

Paper ID 8473

Abstract

In this paper, we address the problem of predicting the future motion of a dynamic agent (called target agent) given its present and past states and the information on its environment. It is paramount to develop a prediction model that can exploit the contextual information in both static and dynamic environments around the target agent and output diverse trajectory samples meaningful in a traffic context. We propose a novel prediction model, referred to as the lane-aware prediction (LaPred) network, which uses the instance-level lane entities extracted from a semantic map to predict the multi-modal future trajectories. For each lane candidate found in the neighborhood of the target agent, LaPred extracts the joint features relating the lane and the trajectories of the neighboring agents. Then, the features for all lane candidates are fused with the attention weights learned through a self-supervised learning task that identifies the lane candidate likely to be followed by the target agent. Using the instance-level lane information, LaPred can produce the trajectories compliant with the surroundings better than 2D raster image-based methods and generate the diverse future trajectories given multiple lane candidates. The experiments conducted on the public nuScenes dataset and Argoverse dataset demonstrate that the proposed LaPred method significantly outperforms the existing prediction models, achieving state-of-the-art performance in the benchmarks.

1. Introduction

Predicting the future motion of a dynamic agent given its past trajectory is crucial for self-driving robots and vehicles to conduct path planning and collision avoidance. However, predicting motion in realistic environments is challenging because the motion of the dynamic agent is determined by various indirectly observed factors, including the agent's intention, the static environment around the agent, and the interaction with other agents. Such uncertainties in prediction

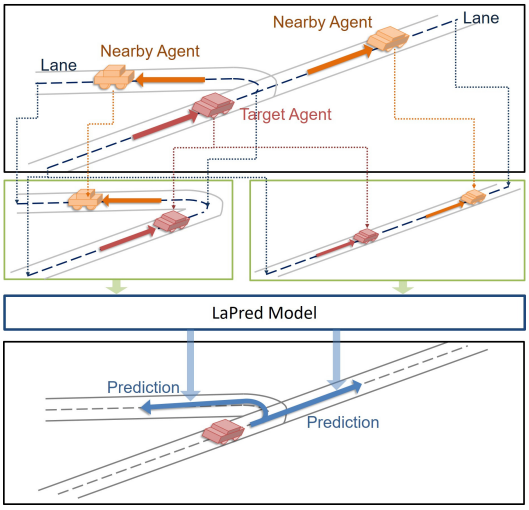

Figure 1: **Lane-Aware Trajectory Prediction:** Depending on which lane is conditioned, the trajectories predicted by the proposed method differ.

tasks entail multiple plausible trajectories that an agent can take to reach its intended goals. Specifically, the distribution of an agent's future trajectory will be multi-modal in that the agent can exhibit different maneuvers (e.g., right turn, left turn, or straight) for given specific scenes (e.g., four-way crossroad), or change the lanes and adjust its speed interacting with other agents. Therefore, in order to comply with the distribution, a prediction model should suggest more than one plausible trajectory samples for the given situation.

To predict such a diverse set of trajectories, the model must understand the environmental context consisting of social patterns in temporal motion, as well as observations for static scene environment via sensors or semantic maps. Therefore, the model's ability to extract and meaningfully represent such multiple cues is crucial. Deep neural networks (DNN) suit this task well, owing to their large capacity and capability of end-to-end learning representation. Previous art has contributed sophisticated architectures for interaction modeling [1, 6, 8, 10, 12, 14,

15, 19, 20, 23, 25, 26, 27, 29], static-scene processing [3, 5, 15, 21, 22, 23, 24, 29], and multi-modal trajectory modeling [3, 5, 10, 11, 15, 19, 21, 22, 23]. However, intermediate logic in DNN models is often not interpretable, and thus human designers have limited space to intervene with a prediction instance. For example, most prediction models do not allow to explicitly condition on a particular set of lanes on the road, while such decisions implicitly stem from random input noise. This is a practical drawback when the models are to be used in self-driving robots, causing them to sample inefficiently many trajectories to cover all plausible modes in the future.

In this paper, we propose a new trajectory prediction method, referred to as the *lane-aware prediction* (LaPred) model, which uses the instance-level lane information extracted from semantic maps. (see Fig. 1 for motivation.) LaPred aims to predict the future trajectory of a target agent using a novel *per-lane joint feature*, which explicitly captures the complicated relation between a lane, a target agent, and a nearby agent at their instance level. Our per-lane joint feature explicitly aggregates the physical environment and the agent interaction at the instance-level, and it is useful for generating diverse trajectory samples from the distinct modes of distribution of future trajectories. We also propose an auxiliary model that determines which lane should be attended to predict the most plausible future trajectory by imposing a *self-supervised learning loss* [13]. This auxiliary model guides LaPred to attend to the lanes that the target agent desires to follow, further enhancing the prediction accuracy.

We evaluate the performance of LaPred on the public *Argoverse* dataset [4] and *nuScenes* dataset [2], which provide the semantic map along with the trajectory data. Our experiments demonstrate that LaPred produces reasonable prediction results for various complex traffic scenarios and achieves a significant performance gain over existing methods. The code will be publicly available.

Our contributions are summarized as follows;

- We propose a trajectory prediction network that uses the instance-level lane entities to represent the complicated relation between the lane and agent trajectories.
- We show that our per-lane joint feature is capable of finding better representation in both static and dynamic environments than other raster image-based methods, consequently producing the trajectories reflecting the lane constraints well.
- We propose an auxiliary model that learns to attend the lane candidates critical for trajectory prediction via a self-supervised classification task. This allows our model to generate high-quality trajectories reflecting important modes associated with lanes.
- We achieve state-of-the-art performance for some categories of Argoverse and nuScene benchmarks.

2. Related Works

2.1. Interaction-Aware Prediction

Numerous methods have been proposed to predict the future trajectories of dynamic agents accounting for interaction with the neighboring agents. Grid-based pooling methods [1, 6, 15, 20, 27, 29] arrange the embedding vectors obtained by encoding the other agents' past motions to a regular tensor structure. Social LSTM [1] applies a pre-defined pooling window to distinct tensors centered at each individual agent. Desire [15] employs a custom log-polar coordinate system and a fully-connected layer to learn a pooling mechanism. CSP-LSTM [6] utilizes convolutional layers for pooling. MATF [29] combines both embedding vectors and visual context to achieve a joint representation. While grid-based pooling methods provide well-visualized ways to model social interaction, they require hand-craft designs for the coordinate systems and tensor resolution. Also, calculations are often inefficient due to a high sparsity in the grid tensor.

Global pooling methods [8, 10, 19, 23, 26] treat embedding vectors without any spatial arrangements. Instead, they directly construct the embedding vectors through global pooling [10] or neural attention architectures [19, 23, 26]. Since there are no spatial constraints imposed by agent layout, these methods can process the interaction among an arbitrary number of agents without the race condition. Recently, spatiotemporal graph-based methods [12, 14, 25] were employed to extend global pooling methods. Trajectron [12, 25] uses this graph structure to encode the dynamic influence of agents over time, and Bigot [14] applies the attention mechanism to the graph to focus on the important parts of the interactions.

We propose a new interaction modeling that directly ties each agent with its corresponding lane instance in order to ease the model design, as well as imposing a relation between agents and the static environment.

2.2. Scene Context-Aware Prediction

The majority of schemes generate 2D scene images using camera or LiDAR sensors. CAR-Net [24] and SoPhie [23] apply spatial attention to focus the salient regions relevant to trajectory prediction. Desire [15] combines a flattened scene feature with agent embedding vectors to refine trajectories. MATF [29] constructs a tensor structure capturing both interactions between agents and scene contexts while retaining the spatial relationships. R2P2 [22] learns a policy model obtained by minimizing the symmetric cross-entropy, given the scene constraints.

The scene context can also be obtained from the infor-

mation embedded in semantic maps. In [3, 5, 7, 17, 21, 25], 2D scene images were constructed by rasterizing the semantic map. Different types of information in the map can be encoded in each channel of an image. However, since trajectory data tend to be represented by the sequence of points in spatial coordinates, it is not trivial to reason the relationship between the trajectories and scene context (e.g., lanes) drawn on a 2D image. Hence, it can be useful to represent scene context in the instance-level, represented in the same coordinate domain as the trajectory. For example, LA network [18] aims to model real-time interactions between agents and lanes using an attention-guided graph, and MTPLA [16] estimates the most correlated lane using instance-level lane information.

While our method shares a similar idea to model instance-level information, it differs from previous methods in treating each lane and their associated agents in a single joint representation, leading to much simpler model architecture while achieving empirical gains in the prediction performance. Furthermore, our method uses the auxiliary reference lane identification task to attend lanes that the target agent tries to follow, which has not been considered in [16, 18].

2.3. Generating diverse trajectory samples

In practical applications, the models are often required to generate multi-modal trajectories with their likelihoods. Generative models, including GAN and VAE, have been used to generate realistic trajectory samples [10, 15, 23]. These methods require large sample sets to achieve acceptable performance because the sample set should cover all trajectories with low probability but high importance in a traffic context. As remedies, DSF [28] proposes a diversity sampling function based on a determinantal point process. Diversity [11] uses a latent space that controls the generation of semantically discrete trajectories, Multi-Path [3] uses a fixed set of trajectory anchors that capture a driver's intentions. CoverNet [21] formulates trajectory prediction as classification over the set of possible physically feasible trajectories. MTP [5] proposes a multi-task loss to generate multiple hypotheses.

We conjecture that instance-level lanes are closely related to the modes of the trajectory distribution such that the lane-aware prediction eases the difficulty of finding the meaningful modes. We find that lane information offers a strong prior on the semantic behavior of a driver, and guides finding physically feasible modes in a traffic environment.

3. Proposed Lane-Aware Multi-Modal Trajectory Prediction Method

In this section, we present the details of the proposed LaPred method.

3.1. Problem Formulation

Suppose that the N lane instances (called *lane candidates*) are found in the neighborhood of the target agent from the map where N can vary with time. We aim to predict the future trajectory sequence of the target agent based on its past trajectory sequence, the lane candidates, and the nearby agents. Each lane candidate is represented by the sequence of coordinates that are equally spaced and have the same length. For each lane candidate, the trajectory of a nearby agent that would interact the most with the target agent is identified. Here are the notations used for our derivation;

- $\mathbf{V}^{(p)} = \left\{V_{l-\tau}^{(p)}, V_{l-\tau+1}^{(p)}, ..., V_{l}^{(p)}\right\}$: the sequence of the past trajectory of the target agent over τ time steps where l denotes the time index for the present, and $V_{l-i}^{(p)}$ is the coordinate of the target agent's centroid relative to $V_{l}^{(p)} = (0, 0)$.
- $\mathbf{V}^{(f)} = \left\{V_{l+1}^{(f)}, V_{l+2}^{(f)}, ..., V_{l+h}^{(f)}\right\}$: the sequence of the future trajectory of the target agent over h time steps.
- $\mathbf{L}^{n} = \{L_{1}^{n}, L_{2}^{n}, ..., L_{M}^{n}\}$: the sequence of M coordinate points on the center of the nth lane candidate.
- $\mathbf{V}^{n} = \left\{V_{l-\tau}^{n}, V_{l-\tau+1}^{n}, ..., V_{l}^{n}\right\}$: the sequence of the past trajectory of a single nearby agent selected for the nth lane candidate.

We call the lane candidate that the target agent tries to follow a *reference lane*. We assume that the reference lane always exists among the set of N lane candidates $\mathbf{L}^1, ..., \mathbf{L}^N$.

The trajectory prediction task can be formulated as finding the posterior distribution of the future trajectory given all available observations, $p\left(\mathbf{V}^{(f)}|\mathbf{V}^{(p)}, \mathbf{L}^{1:N}, \mathbf{V}^{1:N}\right)$ where $\mathbf{L}^{1:N} = \left\{\mathbf{L}^1, ..., \mathbf{L}^N\right\}$ and $\mathbf{V}^{1:N} = \left\{\mathbf{V}^1, ..., \mathbf{V}^N\right\}$. The trajectory prediction can be performed through the encoding and decoding steps. The encoding step seeks to find $p(\xi|\mathbf{V}^{(p)}, \mathbf{L}^{1:N}, \mathbf{V}^{1:N})$, where ξ represents the abstract representation of the given observations. Then, based on the representation ξ, the decoding step produces multiple trajectory samples from $p(\mathbf{V}^{(f)}|\xi)$.

Let E_i be the event that the ith lane candidate becomes the reference lane, then the conditional distribution $p(\xi|\mathbf{V}^{(p)}, \mathbf{L}^{1:N}, \mathbf{V}^{1:N})$ can be expressed as

$$p\left(\xi|\mathbf{V}^{(p)}, \mathbf{L}^{1:N}, \mathbf{V}^{1:N}\right) = \sum_{i=1}^{N} p\left(\xi, E_i\middle|\mathbf{V}^{(p)}, \mathbf{L}^{1:N}, \mathbf{V}^{1:N}\right)$$
$$= \sum_{i=1}^{N} p\left(\xi\middle|E_i, \mathbf{V}^{(p)}, \mathbf{L}^{1:N}, \mathbf{V}^{1:N}\right) p\left(E_i\middle|\mathbf{V}^{(p)}, \mathbf{L}^{1:N}, \mathbf{V}^{1:N}\right). \quad (1)$$

In (1), we assume that $p(E_i \cap E_j) = 0$ and $p(\cup_{i=1}^{N} E_i) = 1$. The first term in (1) indicates the representation ξ of $\mathbf{V}^{(p)}, \mathbf{L}^{1:N}, \mathbf{V}^{1:N}$ under the condition that the ith lane candidate is identified as a reference lane. We also assume that conditioned on E_i, the representation ξ depends solely on the ith lane and the nearby agent on the ith lane, i.e.,

$$p\left(\xi \middle| E_i, \mathbf{V}^{(p)}, \mathbf{L}^{1:N}, \mathbf{V}^{1:N}\right) \approx p\left(\xi \middle| \mathbf{V}^{(p)}, \mathbf{L}^{i}, \mathbf{V}^{i}\right). \quad (2)$$

The second term in (1) represents the probability of E_i given all observations available. Let ξ^i be the representation of $\{\mathbf{V}^{(p)}, \mathbf{L}^i, \mathbf{V}^i\}$, then we have

$$p\left(E_i \middle| \mathbf{V}^{(p)}, \mathbf{L}^{1:N}, \mathbf{V}^{1:N}\right) = p\left(E_i \middle| \xi^{1:N}\right), \quad (3)$$

where $\xi^{1:N} = \{\xi^1, ..., \xi^N\}$. From (1), (2), and (3), the conditional distribution is given by

$$\begin{aligned} &p\left(\xi \middle| \mathbf{V}^{(p)}, \mathbf{L}^{1:N}, \mathbf{V}^{1:N}\right) \\ &= \sum_{i=1}^{N} p\left(\xi \middle| \mathbf{V}^{(p)}, \mathbf{L}^{i}, \mathbf{V}^{i}\right) p\left(E_i \middle| \xi^{1:N}\right). \end{aligned} \quad (4)$$

Note that equation (4) suggests the actual procedure of the encoding step;

- **Trajectory-lane feature extraction:** find the term $p\left(\xi \middle| \mathbf{V}^{(p)}, \mathbf{L}^{i}, \mathbf{V}^{i}\right)$. This is implemented by extracting the trajectory-lane features ξ^i from $\mathbf{V}^{(p)}, \mathbf{L}^i, \mathbf{V}^i$.
- **Reference lane identification:** find the term $p\left(E_i \middle| \xi^{1:N}\right)$. We identify the reference lane based on the set of the trajectory-lane features $\xi^{1:N}$. The probability $p\left(E_i \middle| \xi^{1:N}\right)$ is considered as the attention given to the ith lane candidate.
- **Weighted feature aggregation:** compute $\sum_{i=1}^{N} p\left(\xi \middle| \mathbf{V}^{(p)}, \mathbf{L}^{i}, \mathbf{V}^{i}\right) p\left(E_i \middle| \xi^{1:N}\right)$. The joint representation ξ is constructed by fusing the trajectory-lane features $\xi^{1:N}$ using the attention weight $p\left(E_i \middle| \xi^{1:N}\right)$.

The joint representation ξ obtained in the encoding step is passed to the decoding step to learn the predictive distribution $p\left(\mathbf{V}^f \middle| \xi\right)$. The multiple trajectory samples are generated based on the distribution $p\left(\mathbf{V}^f \middle| \xi\right)$. Inspired by the aforementioned problem formulation, we build the proposed trajectory prediction algorithm, which will be presented in detail in the next section.

3.2. Structure of LaPred Network

In this section, we describe the detailed structure of the proposed LaPred network.

3.2.1 Overall System

Fig. 2 depicts the overall structure of LaPred network. First, the preprocessing block extracts the lane candidates $\mathbf{L}^{1:N}$ from the map. The trajectory sequences are collected for the N neighboring agents $\mathbf{V}^{1:N}$ selected for each lane candidate. The trajectory-lane feature extraction (TFE) block extracts the features ξ^i from $\mathbf{V}^{(p)}, \mathbf{L}^i, \mathbf{V}^i$. Note that the parameters of the TFE block are shared over the lane candidates. Based on N trajectory-lane features $\xi^{1:N}$, the lane attention (LA) block produces the attention weights given by the probability that the ith candidate lane is identified as a reference lane. The trajectory-lane features $\xi^{1:N}$ are weighted by the attention weights and combined to produce the joint representation ξ. Finally, the multi-modal trajectory prediction (MTP) block produces K multi-modal trajectories based on the aggregated features ξ.

3.2.2 Preprocessing

In the preprocessing step, the N lane candidates are identified based on the distance from the current position of the target agent. First, we search for the lane segments (a set of coordinates comprising the middle of a lane segment) within the search radius (e.g., 10 meters) from the centroid of the target agent. Then, we extend the lane segments by attaching the preceding and succeeding lane segments based on lane connectivity information in the map until the length of the extended lane reaches a predefined value. When there are more than N connected lane segments, only N lane instances are selected based on Euclidean distance from the current location of the target agent. The selected N lane instances become *lane candidates*. The set of coordinates for these N lane candidates are resampled such that any two adjacent coordinate points have equal distance.

The lane candidate closest to the future trajectory of the target agent is labeled as a reference lane. The distance between the lane candidate and the trajectory is measured by

$$\mathcal{D}\left(\mathbf{V}^{(f)}, \mathbf{L}^n\right) = \sum_{i=1}^{h} \eta(i) \min_{m \in \{1,...,M\}} \left\| V_{l+i}^{(f)} - L_m^n \right\|, \quad (5)$$

where $\eta(i)$ is the scaling weight applied for different time steps. The reference lane is decided such that higher weight is applied for the lane points on farther horizons e.g., $\eta(i) = i$. Note that the lane candidates are autonomously labeled using the aforementioned criterion without manual labor.

The preprocessing step also searches for the nearby agents, which are like to have the most influence on the trajectory of the target agent. It selects only one nearby agent for each lane candidate. Specifically, all nearby agents are identified within the fixed range from the center point of each lane candidate, and the nearest one in front of the target agent is selected as the most influential agent.

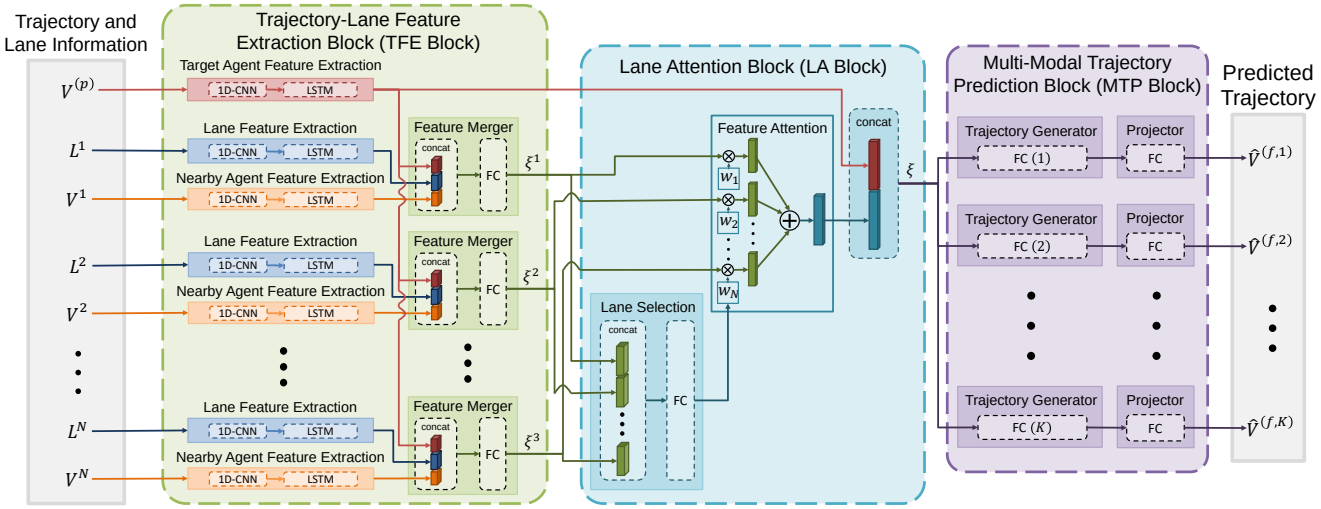

Figure 2: **Overall structure of LaPred Network:** The past trajectories of the target and nearby agents and the lane candidates are fed into the TFE block. The TFE block generates the trajectory-lane features for each lane candidate. The LA block produces the joint representation of the observations via weighted aggregation of the trajectory-lane features. Finally, the K multiple trajectory samples are generated in MTP block.

3.2.3 Trajectory-Lane Feature Extraction

The TFE block extracts the joint trajectory-lane features ξ^i from the observation $\{\mathbf{V}^{(p)}, \mathbf{L}^i, \mathbf{V}^i\}$ for the ith lane candidate. The observations $\mathbf{V}^{(p)}$, $\mathbf{L}^i$, and $\mathbf{V}^i$ are separately encoded by the one-dimensional (1D)-CNN followed by the long short term memory (LSTM) model

$$\xi_{V_p} = \text{LSTM}(\text{1D-CNN}(\mathbf{V}^{(p)})) \quad (6)$$

$$\xi_{L_i} = \text{LSTM}(\text{1D-CNN}(\mathbf{L}^i)) \quad (7)$$

$$\xi_{V_i} = \text{LSTM}(\text{1D-CNN}(\mathbf{V}^i)). \quad (8)$$

These CNN-LSTM encoding networks have different weights for each of $\mathbf{V}^{(p)}, \mathbf{L}^i$ and $\mathbf{V}^i$. The weights of the entire TFE block are shared over the lane candidates. The features ξ_{V_p}, ξ_{L_i} and ξ_{V_i} are concatenated and fed through the fully connected (Fc) layers to produce the joint features ξ^i for the ith lane candidate.

3.2.4 Lane Attention

The LA block produces the joint representation ξ of the given conditions via a weighted combination of the N trajectory-lane features $\xi^{1:N}$. As mentioned, the attention weight w_i for the ith lane candidate is obtained from $p\left(E_i|\xi^{1:N}\right)$. The attention weight w_i is obtained by concatenating the N trajectory-lane features $\xi^{1:N}$ and applying the Fc layers followed by the soft-max function. We consider the fixed number of lane candidates N, but sometimes, the number of lane candidates found in the preprocessing block can be less than N. In this case, zero vectors are fed into the model as an input. To allow the model to attend the lane candidates important for trajectory prediction better, we impose the additional reference lane identification task on top of the prediction task. The LA model performs the task of selecting the reference lane among N lane candidates supervised by the auto-labeled data. This enables the regularization of our model through the auxiliary self-supervised learning task [13]. Finally, the trajectory-lane features $\xi^{1:N}$ are combined using the attention weight w_i as $\xi = \sum_{i=1}^{N} w_i \xi^i$. As an approximation to our weighted feature aggregation, we can also consider the hard selection, which forces $w_i = 1$ for the lane candidate with the highest attention weight and sets $w_i = 0$ for the rest. In the experiment section, we evaluate the benefit of the soft selection over the hard selection.

3.2.5 Multi-Modal Trajectory Prediction

We first construct the complete joint representation by concatenating the features ξ_{V_p} to the output ξ of the LA block through the skip connection. The MTP block generates K hypotheses of the future trajectory of the target agent based on the joint representation $\{\xi, \xi_{V_p}\}$. To generate the multiple trajectory samples, we employ the multi-modal generator model proposed in [5]. The K trajectory samples $\hat{\mathbf{V}}^{(f,1)}, ..., \hat{\mathbf{V}}^{(f,K)}$ are generated using the K sample generator models. Each generator model is divided into two parts; 1) the Fc layers not shared across the K models and 2) the subsequent Fc layers shared. We use the shared Fc layers to alleviate the over-fitting problem that arises by using only a partition of the training data for each generator model. While only one of the K non-shared Fc layers is updated for the given input, the shared Fc layers are all up-

dated for the input.

3.3. Training Details

The loss function L_{total} used to train the proposed LaPred model is given by

$$L_{total} = \alpha L_{pred} + (1-\alpha) L_{cls}, \tag{9}$$

where L_{pred} denotes the mean absolute error loss, and L_{cls} denotes the cross-entropy loss for selecting the reference lane from the lane candidates. Since the MTP block produces the K outputs simultaneously, L_{pred} is given by

$$L_{pred} = \sum_{t \in Batch} \min_{k \in \{1,\cdots,K\}} L_{pred}^{t,k}, \tag{10}$$

where $L_{pred}^{t,k}$ denotes the loss function for the kth generator model evaluated for the tth training sample. The loss $L_{pred}^{t,k}$ is expressed as

$$L_{pred}^{t,k} = \beta L_{pos}^{t,k} + (1-\beta) L_{\text{lane-off}}^{t,k}, \tag{11}$$

where $L_{pos}^{t,k}$ is a smooth L_1 loss between $\hat{\mathbf{V}}^{(f,k)}$ and $\mathbf{V}^{(f)}$. To reflect the tendency of the target agent stick close to the reference lane in the future, we devise a new loss function $L_{\text{lane-off}}$, which is expressed as

$$L_{\text{lane-off}}^{t,k} = \frac{1}{h} \sum_{i=1}^{h} l\left(\hat{V}_{l+i}^{(f,k)}, V_{l+i}^{(f,k)}, \mathbf{L}^{(ref)}\right), \tag{12}$$

where $\mathbf{L}^{(ref)}$ is the lane instance selected as a reference lane and

$$l\left(\hat{V}, V, \mathbf{L}\right) = \begin{cases} \delta(\hat{V}, \mathbf{L}) & \text{if } \delta(\hat{V}, \mathbf{L}) > \delta\left(V, \mathbf{L}\right) \\ 0 & \text{otherwise,} \end{cases}$$

where $\delta\left(V, \mathbf{L}\right)$ denotes the distance from the point V to the lane $\mathbf{L}$. This loss function encourages the model to reduce the distance from the lane whenever the prediction deviates from a lane farther than the ground truth.

4. Experiments

In this section, we evaluate the performance of the proposed LaPred method.

4.1. Dataset

Two public datasets, nuScenes and Argoverse, are used to evaluate the performance of LaPred. These are the only datasets that provide the high definition (HD) map associated with the trajectory data. nuScenes dataset [2] provides the log of the ego vehicle's state, the annotations of nearby agents' location, and HD-map data. The dataset provides 245,414 trajectory instances in 1,000 different scenes. The trajectory instances consist of the sequence of two dimensional (2D) coordinates for 8 seconds duration sampled at $2Hz$. The trajectory prediction task defined in nuScenes benchmark is to predict a 6-second future trajectory based on a 2-second past trajectory for each target agent.

Argoverse forecasting dataset [4] is the dataset specialized for trajectory prediction tasks. The dataset contains the trajectories of the target agent, those of nearby agents, and HD-map data. A total of 324,557 scenarios are included, each of which is 5 seconds. The 2D coordinates comprising trajectories are sampled at $10Hz$. The trajectory prediction task is defined as predicting a 3-second future trajectory based on a 2-second past trajectory.

4.2. Experiment Results

In this section, we evaluate the performance of LaPred.

4.2.1 Evaluation Metric

We employ two popularly used evaluation metrics, *average displacement error* (ADE) and *final displacement error* (FDE), for measuring the mean absolute error between the ground truth $V_{l+i}^{(f)}$ and the kth hypothesis $\hat{V}_{l+i}^{(f,k)}$. Let $e_i^k = \|V_{l+i}^{(f)} - \hat{V}_{l+i}^{(f,k)}\|_2$, then the two evaluation metrics ADE_K and FDE_K are defined as

$$ADE_K = \mathbb{E}_p \min_{k \in \{1,\ldots,K\}} \left(\frac{1}{h} \sum_{i=1}^{h} e_i^k\right) \tag{13}$$

$$FDE_K = \mathbb{E}_p \min_{k \in \{1,\ldots,K\}} e_h^k. \tag{14}$$

4.2.2 Ablation Study

Table 1 presents the contributions of each idea to the performance gain achieved by LaPred over the baseline. Method A is the baseline, which predicts the future trajectory based on only the past trajectory of the target agent using the CNN-LSTM. Method B takes lane information as additional input and uses hard selection for feature aggregation. In Method C, the loss function $L_{\text{lane-off}}$ is used to train the model. Method D uses the past trajectories of the nearby agents on top of Method C. Method E uses the soft selection for the weighted feature aggregation.

We evaluate four performance metrics for the methods. As we add additional ideas to the baseline model, we see that the prediction accuracy of LaPred improves. Note that using lane information in Method B offers the largest performance improvement. When the trajectories of the nearby agents are used along with lane information, further performance improvement is observed, which shows that the proposed LaPred successfully extracts the representation of surrounding environments to enhance the prediction accuracy. We also note that soft feature combining used in

Methods	A	B	C	D	E
Lane Information		√	√	√	√
$L_{\text{lane-off}}$			√	√	√
Nearby Agents				√	√
Soft selected ξ_{agg}					√
ADE_1	4.77	$4.04_{\downarrow 0.73}$	$4.00_{\downarrow 0.04}$	$3.82_{\downarrow 0.18}$	$3.67_{\downarrow 0.15}$
FDE_1	11.10	$9.30_{\downarrow 1.80}$	$9.23_{\downarrow 0.07}$	$8.90_{\downarrow 0.33}$	$8.49_{\downarrow 0.41}$
ADE_5	2.48	$2.06_{\downarrow 0.42}$	$1.77_{\downarrow 0.29}$	$1.72_{\downarrow 0.05}$	$1.53_{\downarrow 0.19}$
FDE_5	5.33	$4.37_{\downarrow 0.96}$	$3.62_{\downarrow 0.75}$	$3.54_{\downarrow 0.08}$	$3.37_{\downarrow 0.17}$

Table 1: Ablation study conducted on nuScenes validation set

Network	ADE_1	FDE_1	ADE_5	FDE_5	ADE_{10}	FDE_{10}	ADE_{15}	FDE_{15}
MTP [5]	4.42	10.36	2.22	4.83	1.74	3.54	1.55	3.05
MultiPath [3]	4.43	10.16	1.78	3.62	1.55	2.93	1.52	2.89
CoverNet [21]	3.87	9.26	1.96	-	1.48	-	-	-
Trajectron++ [25]	-	9.52	1.88	-	1.51	-	-	-
MHA-JAM [17]	3.69	8.57	1.81	3.72	1.24	**2.21**	**1.03**	**1.7**
Ours	**3.67**	**8.49**	**1.53**	**3.37**	**1.21**	2.61	1.11	2.35

Table 2: Performance of several prediction methods evaluated on nuScenes validation set

Method E enables non-negligible performance gain over hard feature combining. This shows that consideration of diverse lane candidates is more helpful than using a single best lane candidate in predicting the future trajectory. We also observe that the loss function $L_{\text{lane-off}}$ used in the method C contributes to the performance achieved by the proposed LaPred.

4.2.3 Quantitative Results

We evaluate the performance of the proposed model on nuScenes and Argoverse datasets. For each dataset, we compare the proposed method with the existing methods that have already published performance results in the paper. Table 2 presents the results on the nuScenes validation set. We tried $K = 1, 5, 10$ and 15 for performance evaluation. The proposed LaPred model outperforms other prediction methods on most of metrics considered. For the metrics FDE_{10}, ADE_{15} and FDE_{15}, the proposed method is slightly worse than MHA-JAM [17], but the performance gap from the rest is significant. This result shows that the instance-level lane information offers the advantage in capturing the scene context and interaction with the nearby agents over 2D raster images used in the other methods under comparison.

Table 3 presents the results on the Argoverse validation set with $K = 1, 5, 6$ and 12. The proposed method achieves the performance gain over the competitors in ADE_5, FDE_5, ADE_6 and ADE_{12} metrics. In ADE_1, FDE_1, FDE_6 and FDE_{12} metrics, the performance of LaPred is comparable to that of the top-performing methods. In particular, the proposed method achieves remarkable performance gain when $K = 5$ and 6. We deduce that the effect of lane information for finding the meaningful modes from the trajectory distribution is maximized in these setups. MTPLA is the algorithm to use instance-level lane information and apply hard decision process, whereas the proposed method applies attention weighted soft selection process to consider to generate multi-modal trajectories. Note that the proposed method demonstrates its advantage over MTPLA in FDE_5 and FDE_6.

4.2.4 Prediction Examples

Fig. 3 illustrates the samples of prediction produced by LaPred and the baseline for particular scenarios selected from the nuScenes dataset. Note that the baseline employs the encoder, which extracts the features only from the past trajectory of the target agent without using lane information. The rest of the structure is the same. Fig. 3 (a), (b), (c) and (d) present the trajectory samples generated with $K = 1$ and the remaining figures show those with $K = 5$.

We see from Fig. 3 (a), (b), (c) and (d) that LaPred presents the predicted trajectory following one of the provided lanes well. On the contrary, the prediction from the baseline does not comply with the lane structure provided by the map. From the results, we confirm that the lane information provides important cues for improving the prediction accuracy, especially in lateral direction.

In Fig. (e), (f), (g) and (h), both methods produce the five trajectory hypotheses to reflect multi-modal trajectory distribution. We observe that the LaPred successfully gen-

Network	ADE_1	FDE_1	ADE_5	FDE_5	ADE_6	FDE_6	ADE_{12}	FDE_{12}
DESIRE[15]*	2.38	4.64	1.17	2.06	1.09	1.89	0.90	1.45
R2P2[22]*	3.02	5.41	1.49	2.54	1.40	2.35	1.11	1.77
VectorNet [9]	1.66	3.67	-	-	-	-	-	-
DiversityGAN [11]	-	-	1.33	2.72	-	-	-	-
MFP [26]	-	-	-	-	1.40	-	-	-
DATF[19]*	2.04	3.69	0.98	1.65	0.92	**1.52**	0.73	**1.12**
MTPLA [16]	**1.46**	**3.27**	-	-	1.05	2.06	-	-
Ours	1.59	3.56	**0.79**	**1.63**	**0.75**	1.53	**0.62**	1.20

$*$ indicates our own implementation.

Table 3: Performance of several prediction methods evaluated on Argoverse validation set

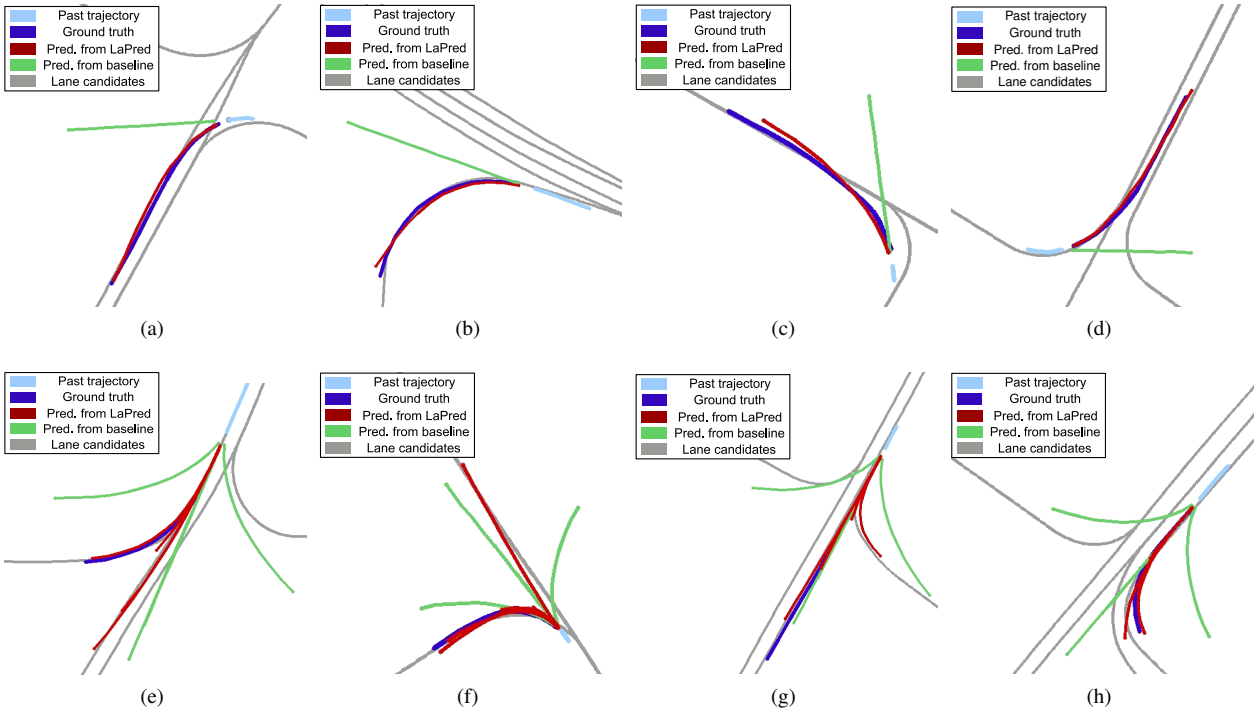

Figure 3: Examples of the predicted trajectories obtained with LaPred and baseline on nuScenes dataset: (a)-(d) $K = 1$ and (e)-(h) $K = 5$.

erates multiple trajectories that are associated with different lane candidates. We note that lane information provides the target agent with a choice of admissible routes and LaPred exploits it to produce the diverse trajectories meaningful in a traffic context. Without lane information, the baseline produces inadequate prediction results in traffic conditions. To overcome this issue, the baseline is required to generate more trajectory samples.

5. Conclusions

In this paper, we proposed a new multi-modal trajectory prediction method that uses instance-level lane information. The proposed LaPred captures the relation between the lanes and the trajectories of the agents using the network which fuses the trajectory-lane features with appropriate weights over all lane candidates extracted from the map. Based on the joint representation of the surrounding environment found using the lane instances, the LaPred generates multi-modal future trajectories compliant with the lane structure. In addition, we trained our model with a self-supervised learning loss, which guides our model to identify the reference lane among N lane candidates. The experiments conducted on nuScenes and Argoverse datasets demonstrated that the LaPred achieved state-of-the-art performance in some metrics and produced the reasonable prediction results in some challenging traffic scenarios.

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 *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971, June 2016. 2

[2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 2, 6

[3] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *Conference on Robot Learning*, pages 86–99, 2020. 2, 3, 7

[4] M. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays. Argoverse: 3d tracking and forecasting with rich maps. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8740–8749, June 2019. 2, 6

[5] H. Cui, V. Radosavljevic, F. Chou, T. Lin, T. Nguyen, T. Huang, J. Schneider, and N. Djuric. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In *Proc. International Conference on Robotics and Automation (ICRA)*, pages 2090–2096, May 2019. 2, 3, 5, 7

[6] Nachiket Deo and Mohan M. Trivedi. Convolutional social pooling for vehicle trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018. 2

[7] Liangji Fang, Qinhong Jiang, Jianping Shi, and Bolei Zhou. Tpnet: Trajectory proposal network for motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6797–6806, 2020. 3

[8] Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. Soft + hardwired attention: An lstm framework for human trajectory prediction and abnormal event detection. *Neural Networks*, 108:466 – 478, 2018. 2

[9] Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11525–11533, 2020. 8

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

[11] Xin Huang, Stephen G McGill, Jonathan A DeCastro, Luke Fletcher, John J Leonard, Brian C Williams, and Guy Rosman. Diversitygan: Diversity-aware vehicle motion prediction via latent semantic sampling. *IEEE Robotics and Automation Letters*, 5(4):5089–5096, 2020. 2, 3, 8

[12] B. Ivanovic and M. Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2375–2384, 2019. 2

[13] L. Jing and Y. Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020. 2, 5

[14] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, Hamid Rezatofighi, and Silvio Savarese. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 137–146. Curran Associates, Inc., 2019. 2

[15] 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 *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2165–2174, July 2017. 2, 3, 8

[16] Chenxu Luo, Lin Sun, Dariush Dabiri, and Alan Yuille. Probabilistic multi-modal trajectory prediction with lane attention for autonomous vehicles. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2370–2376. IEEE, 2020. 3, 8

[17] Kaouther Messaoud, Nachiket Deo, Mohan M. Trivedi, and Fawzi Nashashibi. Trajectory prediction for autonomous driving based on multi-head attention with joint agent-map representation, 2020. 3, 7

[18] Jiacheng Pan, Hongyi Sun, Kecheng Xu, Yifei Jiang, Xiangquan Xiao, Jiangtao Hu, and Jinghao Miao. Lane attention: Predicting vehicles' moving trajectories by learning their attention over lanes. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7949–7956. IEEE, 2020. 3

[19] Seong Hyeon Park, Gyubok Lee, Jimin Seo, Manoj Bhat, Minseok Kang, Jonathan Francis, Ashwin R Jadhav, Paul Pu Liang, and Louis-Philippe Morency. Diverse and admissible trajectory forecasting through multimodal context understanding. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 2, 8

[20] M. Pfeiffer, G. Paolo, H. Sommer, J. Nieto, R. Siegwart, and C. Cadena. A data-driven model for interaction-aware pedestrian motion prediction in object cluttered environments. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5921–5928, 2018. 2

[21] Tung Phan-Minh, Elena Corina Grigore, Freddy A Boulton, Oscar Beijbom, and Eric M Wolff. Covernet: Multimodal behavior prediction using trajectory sets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14074–14083, 2020. 2, 3, 7

[22] N. Rhinehart, K. M. Kitani, and P. Vernaza. R2P2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In *Proc. European Conference on Computer Vision (ECCV)*, Sept. 2018. 2, 8

[23] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese. SoPhie: An attentive GAN for predicting paths compliant to social and physical constraints. In *Proc. IEEE/CVF Conference on Computer Vision and Pat-*

tern Recognition (CVPR), pages 1349–1358, June 2019. 2, 3

[24] Amir Sadeghian, Ferdinand Legros, Maxime Voisin, Ricky Vesel, Alexandre Alahi, and Silvio Savarese. Car-net: Clairvoyant attentive recurrent network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 2

[25] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Multi-agent generative trajectory forecasting with heterogeneous data for control. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2, 3, 7

[26] Charlie Tang and Russ R Salakhutdinov. Multiple futures prediction. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 15424–15434. Curran Associates, Inc., 2019. 2, 8

[27] H. Xue, D. Q. Huynh, and M. Reynolds. Ss-lstm: A hierarchical lstm model for pedestrian trajectory prediction. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1186–1194, 2018. 2

[28] Ye Yuan and Kris M Kitani. Diverse trajectory forecasting with determinantal point processes. In *International Conference on Learning Representations*, 2019. 3

[29] 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 *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12118–12126, June 2019. 2