MultiLane: Lane Intention Prediction and Sensible Lane-Oriented Trajectory Forecasting on Centerline Graphs

David Sierra González, Anshul Paigwar, Özgür Erkent, Christian Laugier

MultiLane: Lane Intention Prediction and Sensible Lane-Oriented Trajectory Forecasting on Centerline Graphs

David Sierra-Gonzalez[1], Anshul Paigwar[1], Ozgur Erkent[1,2] and Christian Laugier[1]

***Abstract*— Forecasting the motion of surrounding traffic is one of the key challenges in the quest to achieve safe autonomous driving technology. Current state-of-the-art deep forecasting architectures are capable of producing impressive results. However, in many cases, they also output completely unreasonable trajectories, making them unsuitable for deployment. In this work, we present a deep forecasting architecture that leverages the map lane centerlines available in recent datasets to predict sensible trajectories; that is, trajectories that conform to the road layout, agree with the observed dynamics of the target, and react to the presence of surrounding agents. To model such sensible behavior, the proposed architecture first predicts the lane or lanes that the target agent is likely to follow. Then, a navigational goal along each candidate lane is predicted, allowing the regression of the final trajectory in a lane- and goal-oriented manner. Our experiments in the Argoverse dataset show that our architecture achieves performance on-par with lane-oriented state-of-the-art forecasting approaches and not far behind goal-oriented approaches, while consistently producing sensible trajectories.**

I. INTRODUCTION

The importance of motion forecasting in the architecture of an autonomous vehicle is perhaps undervalued, at least when compared to motion planning. The truth is that without performant motion forecasting no safe planning can be achieved. If the motion forecasts of surrounding dynamic obstacles extend all over the drivable area, this could lead to excessively defensive plans or even to a frozen robot situation. In contrast, if the forecasts fail to cover a dangerous motion modality that ends up taking place, the result could be an accident. Motion forecasting approaches must thus balance the need for a high recall of probable motion modalities, with the practical requirement of having a low false-positive ratio of predicting dangerous situations.

With the relatively recent introduction of motion forecasting datasets with vectorized map information [1], [2], new forecasting approaches with increasingly good performance results have been proposed [3], [4]. However, despite the impressive results, the low false-positive requirement continues to be an obstacle for the deployment of these models.

In this paper, we propose a forecasting architecture that aims to predict sensible vehicle behavior, i.e. we predict trajectories that conform to the road layout, agree with the observed dynamics of the target, and react to the presence of surrounding traffic. An example of such sensible behavior is illustrated in Fig. 1. In this example, the target vehicle is approaching an intersection, where it can turn right or continue straight. Our model first predicts which lanes the target is likely to follow, assigning roughly equal probability to both options. Then, a distribution over the trajectory endpoint within is each lane is predicted and illustrated as a heatmap. We see how this distribution takes into account the inter-vehicle interactions; for the going straight option, the most probable endpoints are located before a stopped vehicle blocking the road (black traces). Finally, trajectories are regressed in a lane- and goal-oriented manner, which makes them conform to the road layout and react to the presence of other vehicles.

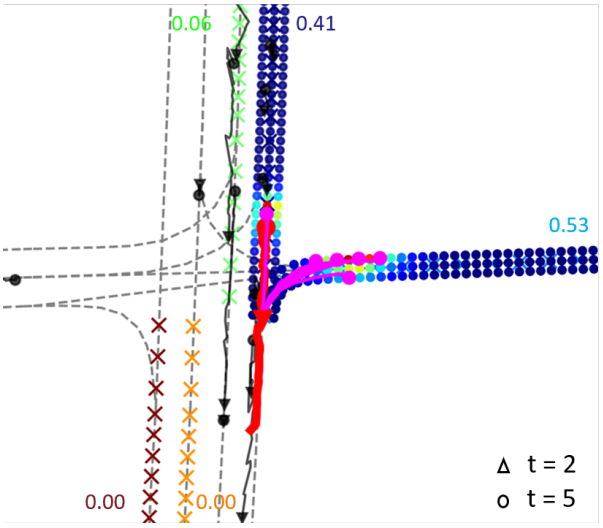

Fig. 1: Prediction results on validation sample 38 of the Argoverse motion forecasting dataset. The crosses represent nodes along the candidate lanes that the target could choose to follow. Our approach predicts the intent probability for the different lanes and, for each of them, a discrete distribution over potential endpoints of the target trajectory (shown in the figure as a heatmap). Multi-modal trajectory forecasts are produced by the final stage of our network, which takes in sampled lanes and endpoints, and produces sensible, lane-oriented and interaction-aware trajectories. The ground truth trajectory is shown in red. The trajectories of other agents in the scene are shown in black. The same color scheme is used throughout the paper.

II. RELATED WORK

Traditional methods for motion forecasting relied on hand-crafted features to model inter-vehicle interactions and intro-

This work was supported by Toyota Motor Europe.

[1]The authors are with Inria, Univ. Grenoble-Alpes, Grenoble, France {david.sierra-gonzalez, anshul.paigwar, ozgur.erkent, christian.laugier}@inria.fr

[2] Ozgur Erkent is with Hacettepe University.

duced constraints to make the forecasting problem tractable [5], [6], [7], [8].

The introduction of large-scale datasets [1], [2] led to a new wave of learning-based forecasting models. Initially, these models did not leverage map information, opting typically for encoder-decoder architectures [9], [10], soon to be expanded with attention mechanisms to balance the different factors affecting the forecasts [11], [12].

Map information was first leveraged through the use of convolutions, where different map and traffic scene information were rendered as images [13], [14]. Then, with the publication of vectorized map information, approaches turned to using graph convolutional neural networks to model the vehicle-vehicle and vehicle-road interactions [3], [4]. While these approaches managed to achieve impressive benchmarking results, they also consistently output unreasonable modalities not explained by the data (e.g. vehicles going completely out of the road bounds or counterflow driving).

To counter this problem some approaches proposed loss functions targeting those failure modes [15], [16], while others proposed to do attention over the render of the drivable area [17]. More recently, some authors recovered the idea of goal-oriented forecasting [18], [19], [20], successfully limiting the number of completely erroneous forecasts. The key idea is to first predict discrete goal locations, and then regress the trajectory that takes the target to the predicted location. The drawback of these approaches is that their forecasts tend to be unimodal, and are thus unsuitable for longer prediction horizons. Additionally, failure of the goal prediction task can again result in unreasonable forecasts, although this problem does not seem to be prevalent.

In contrast to goal-oriented forecasting, other approaches aim to first identify the lane that the target vehicle will follow and then proceed to regress the trajectory forecast [21], [22], [23], [24]. Alternatively, Deo et al propose instead to learn a centerline graph-traversal policy and use path samples from the policy to guide the trajectory regression task [25]. This type of approaches is more suitable for long-term predictions, although it appears to consistently rank lower on forecasting benchmarks. The techniques used to predict the lane intention of the target include heuristics [22], dot-product attention [21], and convolutions on rasters along the centerline [24].

In this paper, we first show how to leverage the LaneGCN graph-encoding architecture for lane intent prediction. While we also rely on dot-product attention [26], the lane and target representations and the architecture used to apply attention differ significantly from those presented by Luo et al [21]. We highlight here the importance of predicting lane intent in terms of scene understanding and utility for downstream tasks such as contingency-based planning [27].

As a next step, we demonstrate how to predict a conditional probability distribution over trajectory endpoints given a lane candidate. Note that this task differs over the unconditional goal prediction of TNT [18], Dense-TNT [19] or HOME [20]. In essence, we predict how far along a given lane the target vehicle might travel. This subtle difference enables us to change the trajectory regression formulation, posing it as an offset regression task from a set of anchors deployed along the predicted lane centerline. This idea was first explored in MultiPath [14], where anchors were placed along candidate trajectories that had been obtained from the dataset via clustering. The proposed combination of ideas aims to enable sensible, multi-modal, long-term trajectory forecasting.

III. PROPOSED APPROACH

A. Formulation

We consider the trajectory forecasting problem in a traffic scene with N interacting agents. Let $\mathbf{x}_t^n \in \mathbb{R}^2$ define the planar location of the nth agent at time step t. We are given: 1) the past location history of all agents in the scene, where $\mathbf{x}_{1:t}^n = [\mathbf{x}_1^n, \mathbf{x}_2^n, \dots, \mathbf{x}_t^n]$ represents the past trajectory of the nth agent; and 2) map contextual information $\mathcal{M}$. Following the same notation, our goal is to find a distribution over the future trajectory of the nth agent: $P(\mathbf{x}_{t+1:T}^n | \mathbf{x}_{1:t}^{1:N}, \mathcal{M})$, where $\mathbf{x}_{1:t}^{1:N}$ denotes the joint past trajectories of all N agents and T represents the prediction horizon. In this work, the map contextual information $\mathcal{M}$ consists in a graph of lane centerlines, providing the connectivity of the road.

B. Approach overview

Fig. 2 shows the main steps of the proposed forecasting model. As a first step, we run a graph encoder that takes in the agents' histories and the map centerlines, and returns a transformed representation of both. The graph encoder that we use is the one proposed in LaneGCN [4], and we provide a brief description of its characteristics in subsection III-C.

To find $P(\mathbf{x}_{t+1:T}^n | \mathbf{x}_{1:t}^{1:N}, \mathcal{M})$, we decompose the problem in several steps, as shown in Fig. 2. We first find out which lanes the target agent is likely to follow, then predict how far along each lane it might travel, and finally forecast its future locations along the lanes. Formally, the decomposition is as follows:

$$P(\mathbf{x}_{t+1:T}^n | \mathbf{x}_{1:t}^{1:N}, \mathcal{M}) = \sum_l \sum_{\mathbf{g}} P(\mathbf{x}_{t+1:T}^n | \mathbf{g}, l, \mathbf{x}_{1:t}^{1:N}) P(\mathbf{g} | l, \mathbf{x}_{1:t}^{1:N}) P(l | \mathbf{x}_{1:t}^{1:N}, \mathcal{M}) \quad (1)$$

where $l \in \mathcal{F}_m(\mathcal{M}, \mathbf{x}_t^n)$, $\mathbf{g} \in \mathcal{F}_g(l)$, $\mathcal{F}_m$ is a heuristic function that proposes potential lane centerlines around $\mathbf{x}_t^n$, and $\mathcal{F}_g(l)$ is a function that samples a grid of endpoints around a lane centerline. Note that $\mathcal{F}_g(l)$ is simplified in Fig. 2, where only endpoints on the centerline are sampled.

The term $P(l | \mathbf{x}_{1:t}^{1:N}, \mathcal{M})$ corresponds to the lane intention prediction problem and we discuss our solution in subsection III-D. The term $P(\mathbf{g} | l, \mathbf{x}_{1:t}^{1:N})$ refers to the conditional prediction of the endpoint of a trajectory given a lane, and is discussed further in subsection III-E. Finally, $P(\mathbf{x}_{t+1:T}^n | \mathbf{g}, l, \mathbf{x}_{1:t}^{1:N})$ represents a distribution over the future locations of the target agent given a lane and trajectory endpoint. In practice, for autonomous driving applications, we settle for finding discrete realizations of this distribution, i.e. trajectory forecasts. We discuss our solution in subsection III-F.

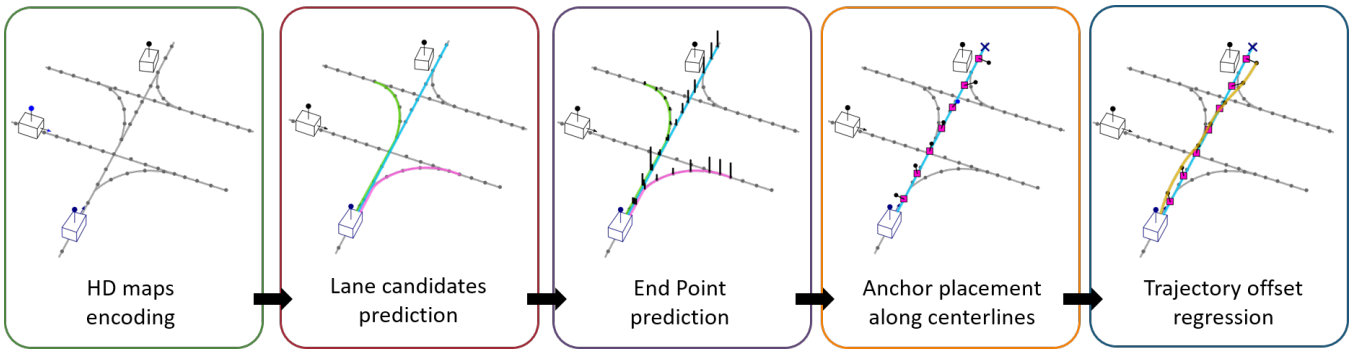

Fig. 2: Steps of the proposed forecasting model.

C. Graph encoder: LaneGCN

In the LaneGCN architecture, a graph encoder is proposed to model the interactions between agents navigating in a graph of road centerlines [4]. In short, the dynamic information summarizing the past motion of each agent is transferred to the centerline graph nodes using an attention mechanism, and propagated along the graph using graph convolutions. The updated information in the graph nodes is then transferred back to the nearby agent nodes using attention. In the end, intuitively, we can think that the agent nodes contain information about the road structure and the presence of other agents; and the centerline nodes contain information about the road structure and the present and future presence of agents in their neighborhood. For further details, we refer the reader to the original publication.

For the purpose of this work, we can think of the graph encoder as the following transformation:

$$\mathbf{h}_t^{1:N}, \mathcal{M}' = \mathcal{F}_{\text{GCN}}(\mathbf{x}_{1:t}^{1:N}, \mathcal{M}) \tag{2}$$

where $\mathbf{h}_t^n$ is a Q-dimensional feature vector summarizing the dynamics and interactions of agent n with surrounding traffic and the road structure, and $\mathcal{M}'$ is an updated graph of centerlines, where each node consists as well in a Q-dimensional feature vector representing the information intuitively described above. In practice, we use $Q = 128$.

D. Lane intention prediction

We define a lane i as a sequence of contiguous graph nodes $l^i = [\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{m^i}]$. Note that the number of nodes is not necessarily the same on the different candidate lanes, as the centerlines' sampling is coarser on straight roads than on turns.

1) Graph centerline preprocessing: We describe here the heuristic algorithm corresponding to function $\mathcal{F}_m$, that proposes potential lane centerlines around any given location in the map. An example of this process is illustrated in Fig. 3. The ground truth trajectory of the target is displayed as red crosses, ending on a red circular marker.

The process starts by using Argoverse API to poll all graph nodes in the proximity of the location of interest (in our case, the location of the target at time t). Depth-first search graph traversal is then performed up to a given distance, yielding a large number of candidate centerlines. This is illustrated in Fig. 3 (left).

Next, we project the last observed location of the target (time step t) to each centerline, and cull any nodes that fall behind the projected location. For computational reasons, we also need to limit the length of each candidate centerline. We do this by applying a constant acceleration model along each centerline, with accelerations randomly drawn from the range $[6, 12]^{\text{m}}/_{\text{s}^2}$. Far-away candidates are removed. The result of this operation is illustrated in Fig. 3 (center).

The final steps consist in removing duplicate or excessively overlapping lane candidates, and annotating the ground truth lane centerlines. Centerlines with less than 30% of their nodes different from those of other centerlines, and in which the totality of the differing nodes are located at the beginning or at the end of the sequence, are removed. To find the ground truth lanes, we first locate all centerlines within a threshold distance of the ground-truth trajectory endpoint, and select those with the lower average distance to the locations $\mathbf{x}_{t+1:T}^n$. This final step is shown in Fig. 3 (right), where the selected ground truth lanes are displayed with black crosses.

2) Attentional lane intention prediction: To obtain features relevant for the lane intention prediction task, we perform dot-product attention [26] between the target feature vector and the nodes in the centerline of the considered lane:

$$\mathbf{f}_{\text{ci}} = \text{softmax}\left(\frac{\boldsymbol{W}^q \mathbf{h}_t^n \odot \boldsymbol{W}^k l^i}{\sqrt{d^k}}\right) \boldsymbol{W}^v l^i \tag{3}$$

where the matrices $\boldsymbol{W}^\bullet$ are the learnable parameters that are used to obtain the query, key and value terms of the attention mechanism, and d^k is the dimensionality of the key and used as a scaling factor. This operation results in a lane contextual feature vector $\mathbf{f}_{\text{ci}}$ that represents the agreement between the target features and those of the centerline. We concatenate this feature vector with the target features and use a fully connected layer to predict a score for each candidate lane, as illustrated in the top architecture of Fig. 4. To transform the scores to probabilities we use a softmax layer over the alternative lane candidates.

We use binary cross-entropy loss to train this stage of the architecture:

$$\mathcal{L}_{\text{lane}} = \mathcal{L}_{\text{CE}}(S_L, S_L^T) \tag{4}$$

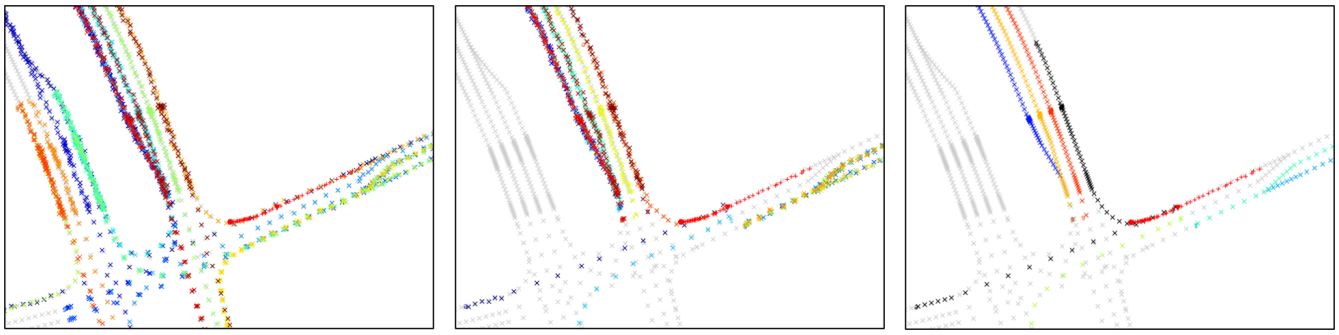

Fig. 3: Example of candidate lane centerline preprocessing for a sample in the Argoverse motion forecasting dataset. The sequences of black nodes in the right-most figure represent the lane centerlines chosen as ground truths.

where S_L represents the predicted lane probabilities and S_L^T are the target probabilities. To avoid penalizing excessively the candidate lanes that travel closer to the ground-truth, we perform smoothing, distributing 0.8 among the ground-truth lanes and the rest among the remaining lanes, where the assigned probability is inversely proportional to their distance from the ground-truth centerlines.

E. Lane-conditioned trajectory endpoint prediction

The endpoint prediction stage is presented at the bottom of Fig. 4. Having obtained the distribution over candidate lanes in the previous stage, we sample as many centerlines as trajectories will be forecast. For each of them, function $\mathcal{F}_g$ is used to select a grid of potential trajectory endpoints.

To obtain the distribution over endpoints for each lane, we first perform attention between the endpoint features and all node features in the sampled lane. This results in a contextual feature vector for each endpoint. This endpoint contextual vector is then concatenated with the lane contextual vector and the target features and passed through a MLP to obtain the score for that endpoint. A softmax operation over the alternative endpoints yields the final probability distribution.

In this stage, we also use a binary cross-entropy loss:

$$\mathcal{L}_{\text{EP}} = \mathcal{L}_{\text{CE}}(S_{EP}, S_{EP}^T) \tag{5}$$

F. Lane- end endpoint-conditioned trajectory forecasting

For each sampled lane candidate l', we now sample its corresponding endpoint $\mathbf{g}'$. To obtain realizations of the distribution $P(\mathbf{x}_{t+1:T}^n | \mathbf{g}', l', \mathbf{x}_{1:t}^{1:N})$, we first deploy equidistant anchors along l' up to the location of $\mathbf{g}'$. We use as many anchors as time steps will be predicted. We then find the coordinates of the predicted trajectory by predicting the offset from each anchor, i.e. $\hat{\mathbf{x}}_t^n = (\Delta\hat{x}_t, \Delta\hat{y}_t) + (a_t^x, a_t^y)$.

The architecture chosen for the offset prediction task is a 2-layer MLP that takes as inputs the last observed location of the target, the locations of all the anchors, the lane contextual features, and the target agent features obtained from the lane encoder.

The loss for the predicted trajectory is then:

$$\mathcal{L}_{\text{traj}} = \sum_{k=t+1}^{T} \mathcal{L}_{\text{reg}}(\hat{\mathbf{x}}_k^n, \mathbf{x}_k^n) \tag{6}$$

where $\mathcal{L}_{\text{reg}}$ is the smooth L1 Loss.

G. Learning

The loss function for the complete architecture is:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{lane}} + \lambda_2 \mathcal{L}_{\text{EP}} + \lambda_3 \mathcal{L}_{\text{traj}} \tag{7}$$

We use the values $\lambda_1 = 5$, $\lambda_2 = 1$, $\lambda_3 = 5$. During training, we always sample a single ground truth lane. Otherwise, it would not be possible to train the network's second stage. For the regression task, we apply teacher forcing to the endpoint. Each of the discussed layers is followed by layer normalization and a ReLU non-linearity, except for the output layers. The complete architecture is trained end-to-end for 36 epochs with an Adam optimizer. The learning rate is set to 0.001 and the batch size to 32.

IV. EXPERIMENTAL EVALUATION

In this section, we evaluate the performance of the proposed forecasting architecture. In the first place, we are interested in measuring the ability of the model to predict the lane intentions of the target. Ideally, the model should unequivocally identify a given candidate lane when the dynamics of the target agree only with that lane, and spread the probability across multiple lanes when all of them are suitable options.

Secondly, we evaluate the effectiveness of MultiLane to forecast *sensible* trajectories that follow the corresponding lane and react to the presence of other road users.

A. Experimental setup

a) Dataset: We evaluate our work using the Argoverse motion forecasting dataset [1]. This dataset provides a graph of lane centerlines (as visualized in Fig. 3). The dataset contains 205942 training samples, 39472 validation samples and 78143 test samples. For each sample, a 5s 2D trajectory of each obstacle is provided, where the class of the obstacle (vehicle, pedestrian, etc.) is not given. The observed trajectory corresponds to the first 2s and the goal is to predict the following 3s. Only a single agent of interest is considered for the evaluation of the forecast.

b) Model variations: In Tab. I we perform an ablation study with different variations of the model presented in section III. The variations affect only the regression network. We consider the case where the regression network forecasts the trajectory using only the target vehicle features and the location of the endpoint (first row). This variation aims to

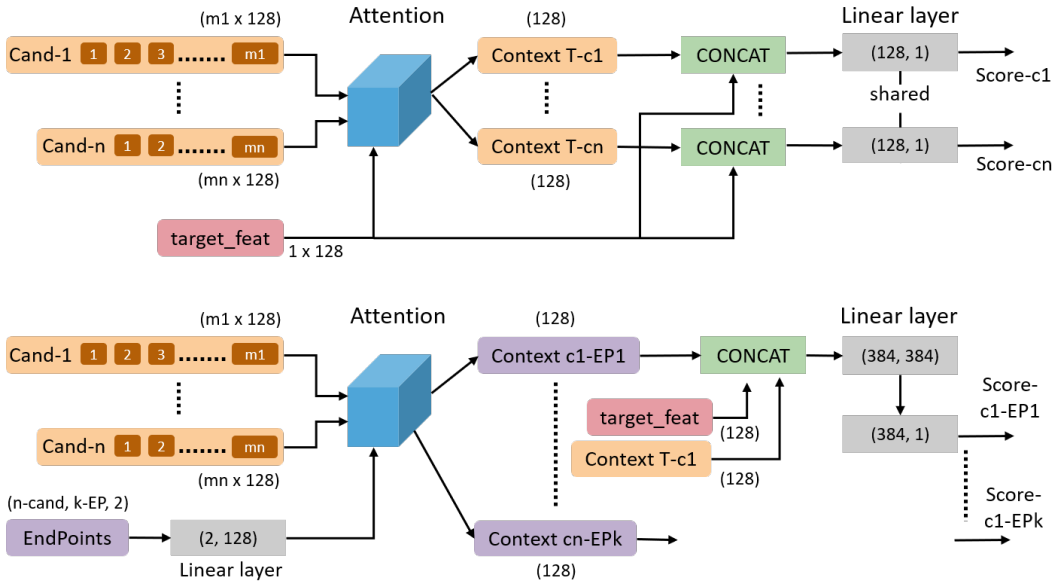

Fig. 4: (*top*) Architecture for the lane intention prediction stage. (*bottom*) Architecture for the trajectory endpoint prediction stage.

mimic endpoint-oriented approaches. The second variation (second row) additionally provides the regression network with the contextual features of the lane, so as to evaluate if these features help in producing better trajectories. Finally, we consider the model from section III, where anchors are deployed along the sampled centerline and the regression network predicts the offset to the real trajectory. For this last case, we show the results with an endpoint grid of 1m resolution and widths 2 and 6 meters.

c) Metrics: We report all widely adopted motion forecasting metrics. The Final Deviation Error (FDE) corresponds to the L2 error at the last time step. The Average Deviation Error (ADE) is the average L2 error across all predicted time steps. These two metrics are calculated for the single-prediction case (K = 1) and the multi-modal prediction case (K = 6). For the multi-modal case, only the minimum across all predictions is considered, yielding the minFDE and minADE metrics. We also report the Miss Rate (MR) metric, that corresponds to the ratio of samples for which the FDE of the prediction was larger than 2m. Finally, the Drivable Area Compliance (DAC), represents the fraction of the forecast trajectories that do not exit the drivable area. A raster of the drivable area with a resolution of 1m is used to calculate this metric.

For our approach, we also report the accuracy in lane and endpoint prediction. We define the lane accuracy as the fraction of samples in which the lane with the highest probability was labeled as a ground truth lane. The definition is similar for endpoints, noting that all endpoints within 1m of the last position of the target are labeled as ground truth endpoints. Finally, we also report the average distance of the endpoint with the highest probability to the ground truth endpoint.

B. *Qualitative analysis*

Figure 5 presents two prediction examples of MultiLane on the validation set. The notation is the same for Figs. 1, 5 and 6. Triangles represent the last observation, i.e. the location of all agents at t=2s. Circles represent the location of all agents at t=5s. Magenta represents the prediction and red represents the ground truth for the agent of interest. In black, we represent the rest of agents (for which the forecasts are not evaluated); for these agents, we only show the trajectory after t=2s to minimize the clutter in the scene. Crosses represent the nodes that compose each centerline; the probability predicted for each lane is displayed in the same color of its corresponding nodes. Finally, in some figures we display the predicted endpoint probability distribution as a heatmap for lanes with a high predicted probability.

The example on the left in Fig. 5 shows only one prediction. The anchors deployed along the blue centerline to regress the prediction are shown as green crosses. As we can see, the model learned to smoothly interpolate the trajectory between the last observation and the sampled endpoint. Trajectories that cut corners are rather common in the dataset.

The example on the right in Fig. 5 shows another left-turn example. Given the dynamics of the target, the right-lane is predicted to be more likely (0.63) than the left lane (0.16). Looking at the endpoint distribution for the right-lane, we see that a certain lane-keeping behavior is predicted, where distance is maintained with the traffic ahead. In contrast, for the left-lane a more aggressive acceleration profile is predicted, consistent with an overtake behavior.

Fig. 6 shows additional illustrative examples for samples in the validation split. In this figure, all trajectories are compared to the ones predicted by LaneGCN [4]. The top-left example highlights how the lane intent prediction spreads the probability across multiple feasible modalities, leading

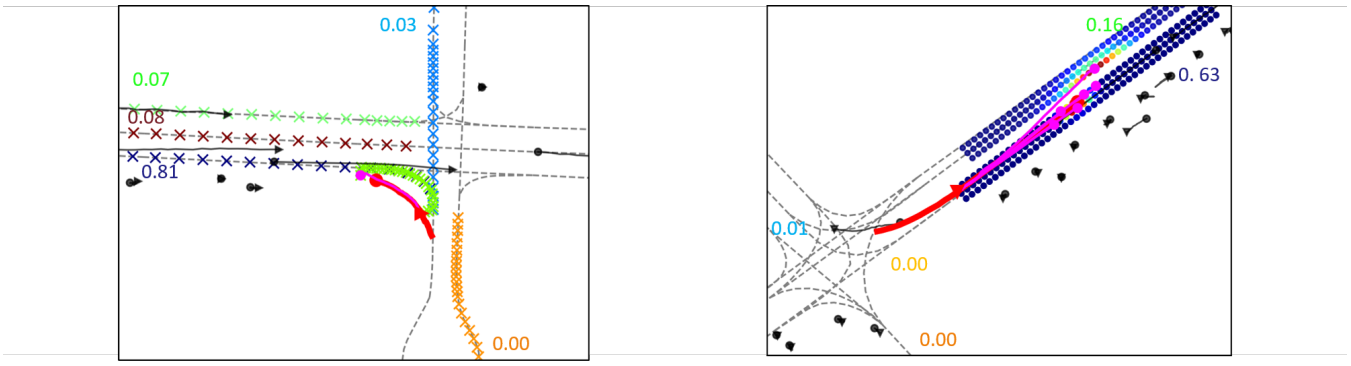

Fig. 5: Two qualitative trajectory forecasting examples. *(left)* A single trajectory forecast is shown, along with the anchors used for regressing it. Additionally, the crosses represent the nodes of the candidate lane centerlines, with their predicted probabilities. *(right)* This example highlights the multi-modal capabilities of the proposed approach. In this left-turn, a high probability is predicted for the right-most lane candidate. The endpoint distribution is consistent with a distance-keeping behavior given the traffic ahead. A lane change is also assigned a non-negligible amount of probability. In this case, the endpoint distribution displays a more aggressive acceleration profile, consistent with an overtake maneuver.

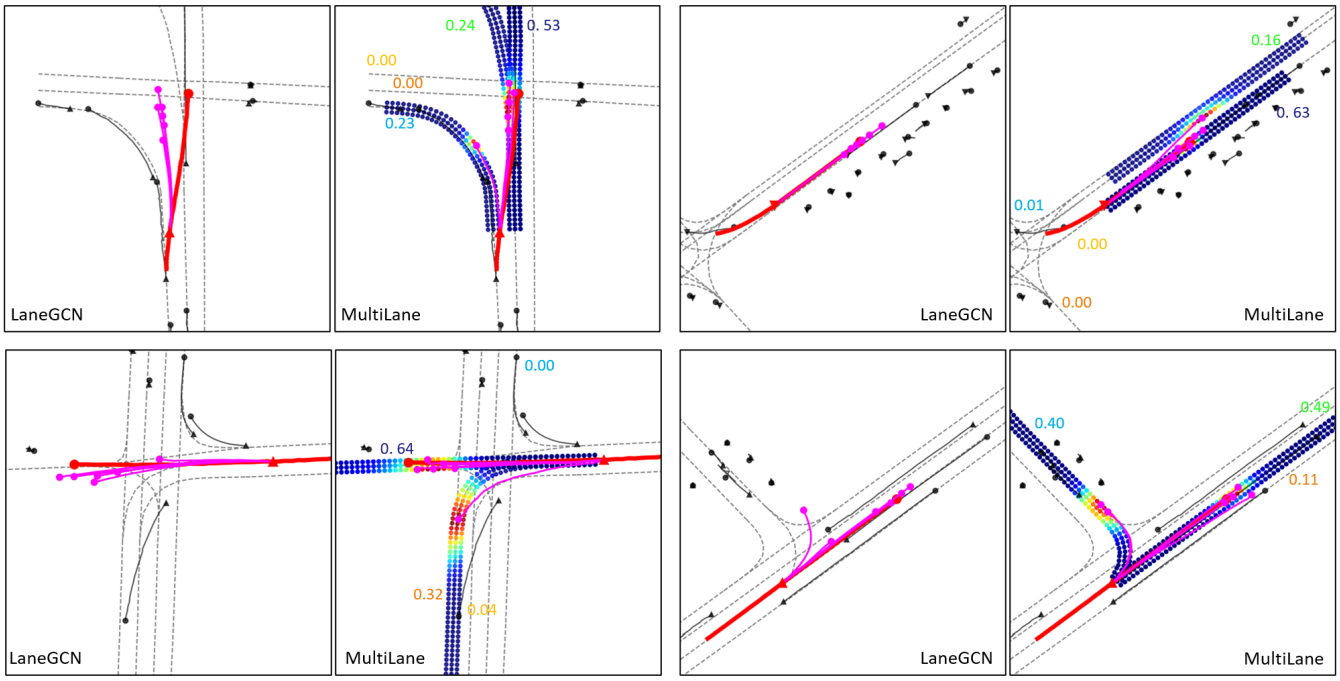

Fig. 6: Comparisons between trajectory forecasts of the proposed model and forecasts obtained using LaneGCN [4]. While LaneGCN achieves better metric scores, it does so by implicitly spreading its trajectory forecasts along different acceleration or turn profiles, yielding sometimes unreasonable trajectories. In contrast, MultiLane covers all likely lane modalities and generates sensible, lane-oriented and interaction-aware trajectories.

to sensible multi-modal trajectory forecasts. In contrast, LaneGCN fails to properly consider the lane structure in this sample. The top-right example in Fig. 6 corresponds to the one discussed above for Fig. 5-right. We see here how LaneGCN does not output an overtake modality. The bottom-left and bottom-right examples in Fig. 6 further illustrate the ability of the intent prediction model to identify high-probability lanes, and how this leads to sensible motion forecasts. In contrast, LaneGCN spreads its forecasts, leading sometimes to unlikely trajectories that go out of the road. Additional examples can be seen at `https://youtu.be/XuzCCbGYQ5A`.

C. Quantitative analysis

We present the results obtained in the validation split in Tab. I and the results obtained in the test split in Tab. II.

We start by analyzing the ablation study shown at the top in Tab. I. As it can be seen, including the lane contextual features as an input in the regression task leads to an improvement across all metrics (reduced errors and miss-rate). Moreover, posing the regression as an offset prediction task from anchors deployed along the centerline further improves the results. This is a promising result, as anchors can also

	Model			K = 1			K = 6			Acc. Lane	Acc. EP	Dist. EP
	Lane Feat.	EP	Anchors	ADE	FDE	MR	minADE	minFDE	MR			
MultiLane (W=2)		✓		1.62	3.45	0.57	0.86	1.52	0.16	0.82	0.21	3.04
	✓	✓		1.53	3.29	0.55	**0.82**	1.47	0.15	0.83	0.22	2.93
	✓	✓	✓	1.48	3.18	0.54	0.87	1.44	0.13	0.83	0.22	2.91
MultiLane (W=6)	✓	✓	✓	**1.42**	**3.08**	**0.51**	0.84	**1.42**	**0.11**	**0.83**	**0.25**	**2.82**
Lane-oriented	Luo et al. [21]			1.46	3.27	-	1.05	2.06	-			
	WIMP [22]			**1.40**	**3.01**	-	0.75	**1.14**	0.12			
	LaPred [23]			1.48	3.29	-	**0.71**	1.44	-			
EP-oriented	TNT (S2) [18]			-	-	-	0.87	1.63	0.21			
	TNT (S3) [18]			-	-	-	**0.72**	1.29	0.09			
	DenseTNT [19]			-	-	-	0.73	**1.05**	0.09			
	HOME [20]			-	3.02	0.50	-	1.28	**0.06**			
Unconstrained	LaneGCN [4]			**1.36**	**3.02**	0.50	0.71	1.08	0.10			
	VectorNet [3]			1.66	3.67	-	-	-	-			

TABLE I: Trajectory forecasting results on the validation split of the Argoverse dataset. Bold figures denote the best result within each category.

	Model	K = 1				K = 6			
		ADE	FDE	MR	DAC	minADE	minFDE	MR	DAC
	MultiLane (W=2)	1.83	3.99	0.62	0.991	1.09	1.92	0.21	**0.989**
	MultiLane (W=6)	1.83	4.00	0.61	0.989	1.04	1.79	0.17	0.981
Lane Oriented	Luo et al. [21]	1.91	4.31	0.66	-	0.99	1.71	0.19	-
	WIMP [22]	1.82	4.03	0.62	-	0.9	1.42	0.17	0.981
EP Oriented	TNT [18]	2.17	4.95	0.70	-	0.94	1.54	0.13	**0.989**
	DenseTNT [19]	1.67	**3.63**	**0.58**	-	0.88	**1.28**	0.12	0.988
	HOME [20]	1.73	3.73	**0.58**	-	0.94	1.45	**0.10**	0.983
Unconstrained	LaneGCN [4]	**1.72**	3.82	0.59	0.985	**0.87**	1.37	0.16	0.981

TABLE II: Trajectory forecasting results on the test split of the Argoverse dataset.

be used to perform auxiliary tasks that could potentially improve the results even further [28]. Finally, we see that using a wider grid of endpoints, leads to reduced errors and an increased endpoint prediction accuracy. We found the grid of width 6m and resolution 1m to have the best performance across all the settings we tried.

Next, we look at the metrics for the lane and endpoint prediction tasks. Our best model achieves a lane intent prediction accuracy of 83% and endpoint accuracy of 25%. The endpoint accuracy appears to be low; however, it should be noted that this metric involves finding the ground truth endpoint across typically hundreds of endpoints per candidate. The average endpoint distance is 2.82m and represents the value that would be obtained for the single-prediction FDE metric if the lane prediction accuracy were always 1.

Comparing to alternative lane-oriented approaches, we see that the proposed approach performs on-par or better than Luo et al [21] and LaPRed [23] for the single-prediction and multi-modal cases. Comparing to WIMP [22], the validation metrics are on-par for the single-prediction task, but lag significantly behind in the multi-modal case. We hypothesize that this is due to the endpoint sampling algorithm, where we iteratively select the endpoint that has the highest probability, and then zero all endpoints within a 2m bubble. This technique has been shown to improve the MR metric, but harm the FDE [20]. In contrast, WIMP optimizes the 6 forecasts jointly during training, so the model automatically learns how to optimally spread the trajectories.

The multi-modality gap in performance applies also to the comparison with endpoint-oriented and unconstrained approaches. However, it should be noted that our approach performs better than TNT before their endpoint regression and trajectory scoring layers (S3) [18]. This further confirms that our bottleneck is in the endpoint sampling procedure. Note also that Dense TNT and HOME perform sophisticated endpoint annotation and post-processing techniques, which we have not yet integrated into our architecture [19], [20].

LaneGCN continues to be one of the top-performing approaches, despite the qualitative results shown in subsection IV-B. This is due to the fact that LaneGCN, just like WIMP, optimizes the K-predicted modalities simultaneously, implicitly making the model learn how to spread the trajectory forecasts. However, as we have seen, this does not necessarily lead to a forecasting model that would be suitable for deployment, as the number of false-positive alarms of dangerous situations would be unacceptable.

The same trends that we have discussed extend also to the results obtained on the test set, shown in Tab. II. In this case, we also show the DAC metric. For the single-prediction modality, MultiLane shows a clear improvement in DAC over LaneGCN, with the error metrics trailing not far behind. For the multi-modal case, MultiLane matches the DAC of LaneGCN with the wider grid of endpoints, and surpasses it with the narrower grid. We believe that this phenomenon is caused by the model learning to cut corners during turns, which is rather common in the dataset (see Fig. 5), and it gets exacerbated as the endpoints are located further away from the centerline. Without access to semantic information, the model has no way to determine if cutting a corner is reasonable in a given scene.

V. CONCLUSION

In this work, we have shown how to predict the lane intentions of a target vehicle based on features learned from

a graph encoder. The proposed model achieves an accuracy of 83% on the Argoverse validation dataset, and is capable of realistically identifying multiple alternative futures given the past trajectory of the target. This is a key requirement for multi-modal and long-term motion forecasting.

In addition to the lane intention prediction model, we have proposed an approach to forecast complete trajectories conditioned on the predicted lanes. Performance-wise, our trajectories score on-par or better than previous lane-oriented forecasting approaches for the single trajectory task. When predicting multiple modalities, our model lags behind alternative approaches. Our hypothesis is that this is caused by the lack of sophisticated post-process trajectory endpoint optimization [20], [19]. Additionally, we have seen that by producing lane- and goal-oriented trajectories, the drivable area compliance improves over that of our base graph-encoder; this is an important requirement if the model is to be deployed. Furthermore, we believe additional improvement in this regard could be achieved by leveraging the semantics of the environment and we plan to investigate this line of research in future work.

VI. ACKNOWLEDGEMENTS

We would like to thank Gabriel Othmezouri from TME for the helpful discussions during the preparation of this work. The experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see `https://www.grid5000.fr`).

REFERENCES

[1] M. Chang, J. Lambert, P. Sangkloy, *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019, pp. 8748–8757.

[2] H. Caesar, V. Bankiti, A. H. Lang, *et al.*, "nuscenes: A multimodal dataset for autonomous driving," in *CVPR*, 2020.

[3] J. Gao, C. Sun, H. Zhao, *et al.*, "Vectornet: Encoding hd maps and agent dynamics from vectorized representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 525–11 533.

[4] M. Liang, B. Yang, R. Hu, *et al.*, "Learning lane graph representations for motion forecasting," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part II*, ser. Lecture Notes in Computer Science, A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, Eds., vol. 12347. Springer, 2020, pp. 541–556.

[5] D. Sierra González, J. S. Dibangoye, and C. Laugier, "High-Speed Highway Scene Prediction Based on Driver Models Learned From Demonstrations," in *Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC 2016)*, Rio de Janeiro, Brazil, Nov. 2016.

[6] A. Lawitzky, D. Althoff, C. F. Passenberg, *et al.*, "Interactive scene prediction for automotive applications," in *2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast City, Australia, June 23-26, 2013*, 2013, pp. 1028–1033.

[7] W. Schwarting and P. Pascheka, "Recursive conflict resolution for cooperative motion planning in dynamic highway traffic," in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, Oct 2014, pp. 1039–1044.

[8] N. Deo, A. Rangesh, and M. M. Trivedi, "How would surround vehicles move? A unified framework for maneuver classification and motion prediction," *IEEE Trans. Intell. Veh.*, vol. 3, no. 2, pp. 129–140, 2018.

[9] A. Alahi, K. Goel, V. Ramanathan, *et al.*, "Social lstm: Human trajectory prediction in crowded spaces," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 961–971.

[10] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," *CoRR*, vol. abs/1805.06771, 2018.

[11] K. Messaoud, I. Yahiaoui, A. Verroust-Blondet, and F. Nashashibi, "Non-local Social Pooling for Vehicle Trajectory Prediction," in *Intelligent Vehicles Symposium (IV)*, Paris, France, 2019. [Online]. Available: https://hal.inria.fr/hal-02160409

[12] J. Mercat, T. Gilles, N. E. Zoghby, *et al.*, "Multi-Head Attention for Joint Multi-Modal Vehicle Motion Forecasting," in *IEEE International Conference on Robotics and Automation*, Paris, France, May 2020, virtual conference. [Online]. Available: https://hal-centralesupelec.archives-ouvertes.fr/hal-02860895

[13] J. Hong, B. Sapp, and J. Philbin, "Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019, pp. 8454–8462.

[14] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," in *CoRL*, 2019.

[15] R. Greer, N. Deo, and M. Trivedi, "Trajectory prediction in autonomous driving with a lane heading auxiliary loss," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4907–4914, 2021.

[16] M. Niedoba, H. Cui, K. Luo, *et al.*, "Improving movement prediction of traffic actors using off-road loss and bias mitigation," in *Workshop on Machine Learning for Autonomous Driving at Conference on Neural Information Processing Systems (ML4AD)*, 2019.

[17] S. H. Park, G. Lee, J. Seo, *et al.*, "Diverse and admissible trajectory forecasting through multimodal context understanding," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., 2020.

[18] H. Zhao, J. Gao, T. Lan, *et al.*, "TNT: target-driven trajectory prediction," in *4th Conference on Robot Learning, CoRL 2020, 16-18 November 2020, Virtual Event / Cambridge, MA, USA*, ser. Proceedings of Machine Learning Research, J. Kober, F. Ramos, and C. J. Tomlin, Eds., vol. 155. PMLR, 2020, pp. 895–904.

[19] J. Gu, C. Sun, and H. Zhao, "Densetnt: End-to-end trajectory prediction from dense goal sets," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15 283–15 292, 2021.

[20] T. Gilles, S. Sabatini, D. Tsishkou, *et al.*, "HOME: Heatmap Output for future Motion Estimation," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC'2021)*, Indianapolis, United States, Sept. 2021.

[21] C. Luo, L. Sun, D. Dabiri, and A. Yuille, "Probabilistic multi-modal trajectory prediction with lane attention for autonomous vehicles," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2370–2376.

[22] S. Khandelwal, W. Qi, J. Singh, *et al.*, "What-if motion prediction for autonomous driving," *arXiv preprint arXiv:2008.10587*, 2020.

[23] B. Kim, S. Park, S. Lee, *et al.*, "Lapred: Lane-aware prediction of multi-modal future trajectories of dynamic agents," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 2021, pp. 14 636–14 645.

[24] L. Zhang, P. Su, J. Hoang, *et al.*, "Map-adaptive goal-based trajectory prediction," in *4th Conference on Robot Learning, CoRL 2020, 16-18 November 2020, Virtual Event / Cambridge, MA, USA*, ser. Proceedings of Machine Learning Research, J. Kober, F. Ramos, and C. J. Tomlin, Eds., vol. 155. PMLR, 2020, pp. 1371–1383.

[25] N. Deo, E. Wolff, and O. Beijbom, "Multimodal trajectory prediction conditioned on lane-graph traversals," in *Proceedings of the 5th Conference on Robot Learning*, 2022.

[26] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[27] A. Cui, A. Sadat, S. Casas, *et al.*, "Lookout: Diverse multi-future prediction and planning for self-driving," *International Conference on Computer Vision (ICCV)*, 2021.

[28] K. Chitta, A. Prakash, and A. Geiger, "Neat: Neural attention fields for end-to-end autonomous driving," in *International Conference on Computer Vision (ICCV)*, 2021.