LAformer: Trajectory Prediction for Autonomous Driving with Lane-Aware Scene Constraints

Mengmeng Liu[1,*], Hao Cheng[1,*], Lin Chen[2], Hellward Broszio[2],
Jiangtao Li[3], Runjiang Zhao[3], Monika Sester[4], Michael Ying Yang[5]
[1]Uni. of Twente, [2]VISCODA GmbH, [3]PhiGent Robotics, [4]Leibniz Uni. Hannover, [5]Uni. of Bath
[*]Equal contribution, h.cheng-2@utwente.nl

Abstract

Existing trajectory prediction methods for autonomous driving typically rely on one-stage trajectory prediction models, which condition future trajectories on observed trajectories combined with fused scene information. However, they often struggle with complex scene constraints, such as those encountered at intersections. To this end, we present a novel method, called LAformer. It uses an attention-based temporally dense lane-aware estimation module to continuously estimate the likelihood of the alignment between motion dynamics and scene information extracted from an HD map. Additionally, unlike one-stage prediction models, LAformer utilizes predictions from the first stage as anchor trajectories. It leverages a second-stage motion refinement module to further explore temporal consistency across the complete time horizon. Extensive experiments on nuScenes and Argoverse 1 demonstrate that LAformer achieves excellent generalized performance for multimodal trajectory prediction. The source code of LAformer is available at https://github.com/mengmengliu1998/LAformer.

1. Introduction

Accurate trajectory prediction is paramount for enabling autonomous driving in diverse traffic scenarios involving interactions with various road agents. Due to the stochastic behaviors of agents and their mutual influences, and the varying environmental scene contexts, trajectory prediction remains a challenging task. Therefore, this task necessitates effective learning of an agent's motion dynamics and interactions with other agents, as well as careful consideration of scene constraints.

Numerous data-driven approaches have been developed to tackle trajectory prediction by extracting motion dynamics from sequential trajectories and scene contexts from rasterized map data, and then fusing them in a latent space as the input to a multimodal decoder, as demonstrated by, *e.g.* Trajectron++ [41], CoverNet [36], and AgentFormer [51]. However, these approaches fail to utilize spatial and temporal information at an early stage for the subsequent decoding module. Also, a rasterized map requires large receptive filters and computational cost to perceive the scene context, which may not provide accurate road structure features at complex intersections, especially for vehicle trajectory prediction. Consequently, the decoder may generate trajectory predictions that are non-compliant with the scene. To mitigate this problem, VectorNet [12] proposes to unify trajectory and high-definition (HD) map data into a consistent vectorized form. This vectorization enables trajectories and lane segments based on HD maps to be easily processed and fused using the same encoder.

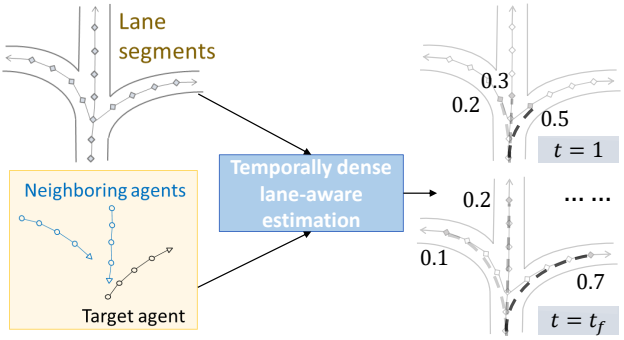

Figure 1. A temporally dense lane-aware estimation module estimates the step-wise likelihood of lane segments aligning with motion dynamics, *e.g.*, selecting the most scene-compliant lane segments at each future time step to facilitate the decoding process.

There are already quite a lot of attempts to explore lane segments, including deep feature fusion, *e.g.* [22, 52] and heuristic searching, *e.g.* [11]. We further categorize the mainstream methods into *spatially* and *temporally* dense methods. Most of the current methods belong to the former one, which estimates dense probabilistic goal candidates [17, 54], segment proposals for endpoints [13, 14, 45]

or the whole sequence encoding [11] projected on the given scene. We argue that these methods are suboptimal because compound prediction errors could occur if the prediction is inaccurate in the initial steps. In contrast, the temporally dense methods seek to estimate the likelihood of motion states aligning with lane positions at each time step. Hence, the decoder has a better chance to adjust its predictions if the motion states and lane segments divert from each other as time unfolds. However, this is not trivial because the estimation module needs to account for the variability of lane segments and uncertainty of motion states. Also, the alignment simply based on distance metrics [14] is insufficient when the ego vehicle is at an intersection with multiple parallel lanes, or when the ego vehicle makes a lane change or a turn. Nevertheless, not much research has been done in exploring the temporally dense methods.

To this end, we propose a temporally dense method, called LAformer. The essence of LAformer is illustrated in Fig. 1. It utilizes a lane-aware estimation module to select only the top-k highly potential lane segments at each time step, which effectively and continuously aligns motion dynamics with scene information. As shown in Fig. 1, LAformer utilizes a lane-aware estimation module to estimate the likelihood of each potential lane segment at each time step. In the decoding process, only the top-k highly potential lane segments are fed to the decoder. Specifically, we employ a novel attention-based encoder, termed Global Interaction Graph (GIG), to extract spatial-temporal features from the unified vectorized trajectories and HD map. Different from the spatially dense methods such as [11, 13, 14, 45], we train a binary classifier using the lane information extracted from the GIG module and the target agent's motion including speed and orientation information for step-wise lane selection throughout the prediction time horizon. Then, we introduce a Laplacian Mixture Density Network (MDN) to generate scene-compliant multimodal trajectory predictions aligned with only the lane segments with the highest likelihoods.

Additionally, to exploit the temporal consistency over the complete time horizon, we introduce a motion refinement module. LAformer utilizes predictions from the first stage as anchor trajectories, which distinguishes itself from anchor-points-based trajectory prediction methods using predefined anchor points [3, 43]. Compared to the first stage, the second-stage motion refinement module takes as input both the observed and predicted trajectories to further reduce prediction offsets.

Our **key contributions** are threefold:

- We propose a novel temporally dense lane-aware selection method to identify the top-k highly potential lane segments at each predicted time step, which is different from previous spatially dense approaches. This selection method improves the accuracy of the lane-conditioned decoder for trajectory prediction.
- We leverage the predicted trajectories from the first stage as anchor trajectories and introduce a second-stage motion refinement module that considers both observed and predicted trajectories. This refinement module further explores the temporal consistency across the past and future time horizons to further reduce prediction errors.
- We demonstrate the effectiveness of LAformer on two benchmark datasets, *i.e*. Argoverse 1 [4] and nuScenes [2]. It achieves good performances on both benchmarks and shows superior generalized performance for the multimodal motion prediction task.

2. Related Work

Interaction modeling. Agents are interconnected for social connections and collision avoidance [19, 35]. Most deep learning models, *e.g*. [1, 18, 25, 40, 41, 51], use agents' hidden states to aggregate the interaction information. Common aggregation strategies include pooling [1, 9, 12], message passing [50, 52, 53] using graph convolutional networks (GCNs) [47], and attention mechanisms [6, 28, 51]. To differentiate the impacts of surrounding agents based on their relative positions and attributes, we propose the use of attention mechanisms for interaction modeling in this work.

Predicting multimodal trajectories. In the context of trajectory prediction for autonomous driving, predicting diverse multimodal trajectories is more favorable than single-modal trajectories to cope with agents' uncertain behaviors and scene constraints. Generative models, *e.g*., Generative Adversarial Nets (GANs) [16], Variational Auto-Encoder (VAE) [23] and conditional-VAE [24], and Flows [37], use sampling-based approaches to generate multiple predictions [7, 18, 25, 38]. However, they do not provide a straightforward estimation of the likelihood of each mode. Although Gaussian Mixture Density Networks (MDNs) can provide a probability density function to learn the mode distribution, similar to the generative models, they often suffer from the so-called mode-collapse problem [39] when only a single ground truth trajectory is used for supervised learning. To mitigate the mode-collapse problem, this paper explores the use of a Laplacian MDN with a winner-takes-all strategy [5, 11, 30, 46, 56, 57]. Additionally, in order to increase modality diversity, some approaches generate a plethora of predictions and employ ensembling techniques such as clustering or Non-Maximum Suppression to reduce the predictions into a limited number of modalities [43, 45]. Nevertheless, this ensembling process is time-consuming and impractical for real-time autonomous vehicles [56]. Therefore, this paper refrains from adopting this technique.

Scene-aware modeling. To predict scene-compliant trajectories, scene contexts must be considered. Convolutional neural networks (CNNs) are commonly used to ex-

tract scene contexts from bird's-eye view images, such as RGB images with general contexts [25, 40] and semantic maps with different scene categories [36, 41, 51]. However, CNNs struggle to capture fine-grained scene information like lane geometry and traffic regulations. Furthermore, the sparse information on rasterized data leads to less computational efficiency, requiring a powerful fusion module to align heterogeneous motion and scene information for the prediction module. To address these challenges, a unified vectorization scheme [12] can be used to align trajectories and lanes from an HD map. Both trajectories and scene contexts, denoted by points, polylines, and polygons, are coded in a unified vector with coordinate information and various agent or lane attributes [6, 11, 17, 26, 33, 45, 56]. We adopt this representation in our scene-aware trajectory prediction approach.

Furthermore, lane-based scene information is proposed to guide the prediction process. Proposal-based models [9, 42] classify an agent's maneuvers and then predict subsequent trajectories accordingly. Goal-based models predict feasible goals [8, 13, 14, 17, 38, 54] that lie in plausible lanes, and then generate complete trajectories. Other methods use a fixed set of anchors corresponding to trajectory distribution modes to regress predicted multimodal trajectories [3, 43]. Alternatively, [48, 55] propose a method that treats the collection of historical trajectories or predecessors' trajectories [27] at an agent's current location as prior information to narrow down the search space for potential future trajectories. We categorize these methods as spatially dense lane-based methods, as they focus on generating a probabilistic distribution of candidate goals or full trajectories over the space. However, these methods do not fully explore temporal information to account for motion uncertainty and scene variability as time progresses. Additionally, the prediction module must implicitly filter out irrelevant scene information, which can be challenging in complex scene constraints, such as those present at intersections.

3. Method

3.1. Problem formulation

Following the mainstream works, *e.g.* [11, 12, 17, 45, 56], we assume that detecting and tracking road agents, as well as perceiving the environment, provides high-quality trajectory and HD map data in a 2D coordinate system. Namely, for agent i, the x- and y-positions within a given time horizon $\{-t_h+1, \cdots, 0, 1, \cdots, t_f\}$ are obtained, along with the HD map $\mathbf{C}$ of the scene. The downstream task is to predict the subsequent trajectories $\mathbf{Y}^i_{1:t_f}$ by leveraging the HD map and observed trajectories of all agents in the given scenario, including the target agent's trajectory $\mathbf{X}^i_{-t_h+1:0}$.

Both agents' past trajectories and lane centerlines are represented as vectors. To be more specific, for agent i, its history trajectory X_i is represented as an ordered sequence of sparse trajectory vectors $\mathbf{A}^i_{-t_h+1:0} = \{v^i_{-t_h+2}, v^i_{-t_h+3}, ..., v^i_0\}$ over the past t_h time steps. Each trajectory vector v^i_t is defined as $v^i_t = [d^i_{t,s}, d^i_{t,e}, a^i]$, where $d^i_{t,s}$ and $d^i_{t,e}$ denote the start and end points, respectively, and a_i corresponds to agent i's attribute features, such as timestamp and object type (*i.e.*, autonomous vehicles, target agent, and others). In addition, lane centerlines are further sliced into predefined segments to capture fine-grained lane information in order to model an agent's intention precisely. Similar to the trajectory vector, a lane centerline segment is represented as $\mathbf{C}^j_{1:N} = \{v^j_1, v^j_2, ..., v^j_N\}$, where N denotes the total vector length. Each lane vector $v^j_n = [d^j_{n,s}, d^j_{n,e}, a_j, d^j_{n,\text{pre}}]$ adds $d^j_{n,\text{pre}}$ to indicate the predecessor of the start point. The lane vectors are connected end-to-end to obtain the HD map's structural features. To ensure input feature invariance concerning an agent's location, the coordinates are normalized to be centered around the target agent's last observed position.

Fig. 2 presents the framework of LAformer. It takes vectorized trajectories and an HD map as input and outputs multimodal trajectories for the target agent. Each module is explained in detail below.

3.2. Agent motion and scene encoding

We design a novel attention-based Global Interaction Graph (GIG) to encode agent motion and scene information. Concretely, we process trajectory vectors $\mathbf{A}^i$ and lane vectors $\mathbf{C}^j_{1:N}$ using a Multi-Layer Perceptron (MLP) and a Gated Recurrent Unit (GRU) layer sequentially. The output encodings of these layers are represented as h_i for $\forall i \in \{1, \ldots, N_{\text{traj}}\}$ and c_j for $\forall j \in \{1, \ldots, N_{\text{lane}}\}$. N_{traj} and N_{lane} denote the number of trajectories and lane segments in a given scenario, respectively. To fuse these encodings, we design a symmetric cross attention mechanism that operates on h_i and c_j as follows: $h_i = h_i + \text{CrossAtt}\{h_i, c_j\}$ for $j \in \{1, \ldots, N_{\text{lane}}\}$ and $c_j = c_j + \text{CrossAtt}\{c_j, h_i\}$ for $i \in \{1, \ldots, N_{\text{traj}}\}$. Afterward, the GIG further explores self-attention and skip-connection to learn the interactions among agents. Namely, $h_i = \text{ConCat}[h_i, c_j]$ for $j \in \{1, \ldots, N_{\text{lane}}\}$ and is updated as $h_i = h_i + \text{SelfAtt}\{h_i\}$ for $i \in \{1, \ldots, N_{\text{traj}}\}$. More details about the GIG are provided in the supplementary material.

3.3. Temporally dense lane-aware estimation

The temporally dense lane-aware probability estimation module uses attention to guide a target agent towards the most influential lane segments for its future trajectories at each future time step $t \in \{1, \ldots, t_f\}$. We predict lane probabilities using a lane-scoring head and an attention mechanism. The key (K) and value (V) vectors are linear projections of the agent motion encoding h_i, while the

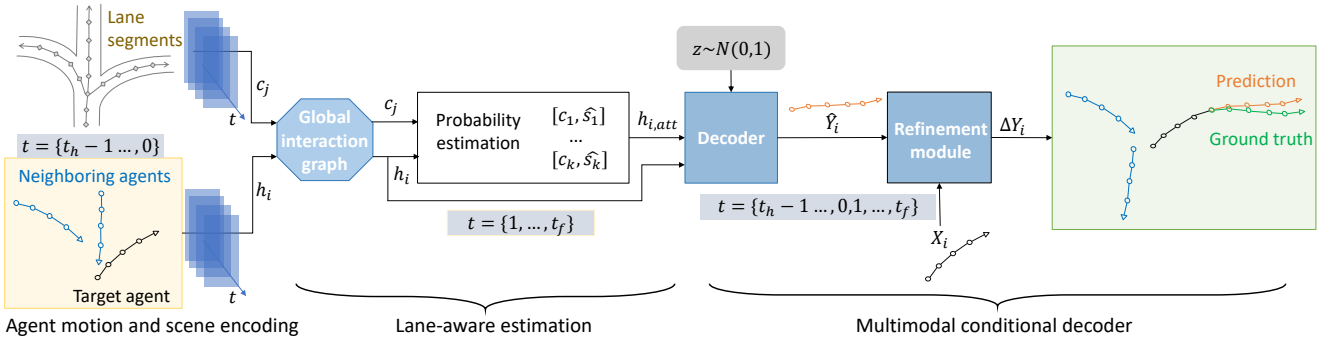

Figure 2. LAformer takes vectorized trajectories and HD map lane segments as input. The connected dots indicate the trajectory or map lane positions and the arrows indicate the heading directions of the agents. The agent motion and scene encodings are represented as h_i and c_j, respectively, and are later fused by the attention-based global interaction graph. At each time step t, the decoder then takes the target agent's past trajectory h_i, the updated motion information aligned with the candidate lane information $h_{i,\text{att}}$ from the lane-aware estimation module, and a random latent variable z as inputs, and predicts the future trajectory $\hat{Y}_i$. The refinement module further reduces the predicted offset ΔY to improve prediction accuracy.

query (Q) vector is a linear projection of the lane encoding c_j. They are then fed into a scaled dot-product attention block $A_{i,j} = \text{softmax}(QK^T/\sqrt{d_k})V$, resulting in the predicted score of the j-th lane segment at t given by $\hat{s}_{j,t} = \frac{\exp(\phi\{h_i, c_j, A_{i,j}\})}{\sum_{n=1}^{N_{\text{lane}}} \exp(\phi\{h_i, c_n, A_{i,n}\})}$, where ϕ denotes a two-layer MLP.

To balance the variability of lane segments and uncertainty of motion dynamics, we select the top-k lane segments $\{c_1, \ldots, c_k\}$ with the k highest scores $\{\hat{s}_1, \ldots, \hat{s}_k\}$ as the candidate lane segments. We then concatenate the candidate lane segments and associated scores over the future time steps to obtain $C = \text{ConCat}\{c_{1:k}, \hat{s}_{1:k}\}_{t=1}^{t_f}$. Next, we perform cross attention to project the target agent's past trajectory encoding h_i as the query vector, and the candidate lane encodings C as the key and value vectors. The output $h_{i,\text{att}}$ is the updated motion information aligned with the lane information. This cross-attention further explores scene information in spatial and temporal dimensions.

The lane scoring module uses a binary cross-entropy loss $\mathcal{L}_{lane} = \sum_{t=1}^{t_f} \mathcal{L}_{\text{CE}}(s_t, \hat{s}_t)$ to optimize the probability estimation. The ground truth value s_t is set to 1 for the lane segment that is closest to the trajectory's truth position, and 0 for all other lanes. It is worth mentioning that no additional labeling is needed; s_t can be identified easily using a distance metric, such as the Euclidean distance.

3.4. Multimodal conditional decoder

This section introduces a Laplacian mixture density network (MDN) decoder that is conditioned on the encodings of the target agent's past trajectory h_i and the updated motion information aligned with the candidate lane information $h_{i,\text{att}}$. To further preserve the diversity of multimodalities, we sample a latent vector z from a multivariate normal distribution, which serves as an additional condition added to the encodings for the predictions. The decoder predicts a set of trajectories $\{(\pi; \text{Laplace}(\mu, b))\}_{m=1}^{M}$, where $\hat{\pi}_m$ denotes the probability of each mode indexed by m among the M predicted modes and $\sum_{m=1}^{M} \hat{\pi}_m = 1$. $(\mu, b) \in \mathbb{R}^{2 \times t_f}$, representing the location and scale parameters of each Laplace component. We use an MLP to predict $\hat{\pi}_m$, a GRU to recover the time dimension t_f of the predictions, and two side-by-side MLPs to predict μ and b.

We train the decoder by minimizing a *regression loss* $\mathcal{L}_{\text{reg}}$ and a *classification loss* $\mathcal{L}_{\text{cls}}$. $\mathcal{L}_{\text{reg}}$ is computed using the Winner-Takes-All strategy [11, 30, 56] $\mathcal{L}_{\text{reg}} = \frac{1}{t_f} \sum_{t=1}^{t_f} -\log P(Y_t | \mu_t^{m^*}, \mathbf{b}_t^{m^*})$, where Y represents the ground truth position and m^* represents the mode with the minimum L_2 error among the M predictions. The cross-entropy loss is used to optimize the mode classification and is defined as $\mathcal{L}_{\text{cls}} = \sum_{m=1}^{M} -\pi_m \log(\hat{\pi}_m)$. We adopt the soft displacement error, following [56], as our target probability π_m. The total loss for the motion prediction in the first stage is given by $\mathcal{L}_{\text{S1}} = \lambda_1 \mathcal{L}_{\text{lane}} + \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{cls}}$, where λ_1 controls the relative importance of $\mathcal{L}_{\text{lane}}$.

3.5. Motion refinement

A second-stage motion refinement is introduced to further explore the temporal consistency for predicting more accurate future trajectories. The goal is to reduce the offset between the ground truth trajectory $Y_{1:t_f}$ and the predicted trajectory $\hat{Y}_{1:t_f}$. In this stage, we leverage the complete trajectory $\{\{X\}_{-t_h+1}^{0}, \{\hat{Y}\}_{1}^{t_f}\}$ as the input to extract the motion encoding $\dot{h}_i$ using a similar temporal encoder as in the first stage. Then, a regression head constructed by a two-layer MLP takes as input all the motion encodings $[h_i, h_{i,\text{att}}, \dot{h}_i]$ in both stages and predicts the offset $\Delta Y = Y - \hat{Y}_m$ between the ground truth and predicted trajectories. We use L_2 loss to optimize the offset $\mathcal{L}_{\text{off}} = \frac{1}{t_f} \sum_{t=1}^{t_f} ||\Delta \hat{Y}_t - \Delta Y_t||_2$. Furthermore, we use a

cosine function, $\mathcal{L}_{\text{angle}} = \frac{1}{t_f}\sum_{t=1}^{t_f} -cos(\hat{\theta}_t - \theta_t)$ to explicitly aid the model in learning the turning angle from the last observed position. It measures the difference between the ground truth angle $\theta_t = \text{arctan2}(Y_t - X_0)$ and the predicted angle $\hat{\theta}_t = \text{arctan2}(\hat{Y}_t - X_0)$. Here, we employ a Winner-Takes-All strategy to optimize the offset and angle losses, similar to the first stage. The total loss in the second is $\mathcal{L}_{\text{S2}} = \mathcal{L}_{\text{S1}} + \lambda_2\mathcal{L}_{\text{off}} + \lambda_3\mathcal{L}_{\text{angle}}$, where λ_2 and λ_3 control the relative importance of the corresponding loss terms.

4. Experiments

4.1. Datasets and evaluation metrics

Datasets. The proposed approach is developed and evaluated on two widely used benchmarks for autonomous driving: *nuScenes* [2] and *Argoverse 1* [4]. These benchmarks provide trajectories of various types of road agents with an HD map of the given scene. To ensure a fair comparison, the official data partitioning and online test server of both benchmarks are used for the training and test setting, respectively.

Evaluation metrics: We adopt the standard evaluation metrics to measure prediction performance, including FDE_{K} and ADE_{K} for measuring L_2 errors at the final step and averaged at each step, respectively, for predicting K modes. Here, the minimum error of the K modes is reported. Both ADE and FDE are measured in meters. The miss rate MR_{K} measures the percentage of scenarios for which the final-step error is larger than $2.0\,\text{m}$. K is set to 5 and 10 in nuScenes and 6 in Argoverse 1 for the multimodal trajectory prediction. For all the evaluation metrics, the lower the better.

4.2. Quantitative results

Tables 1 and 2 present the results obtained on the *Argoverse 1* validation and online test sets, and the *nuScenes* online test set, respectively. The leaderboard results are updated up to 2024-03-09.

LAformer yields the best results on the Argoverse 1 validation set, surpassing previous models, such as DenseTNT [17], LaneGCN [26], FRM [34], and HiVT [56] with a clear margin in ADE and FDE. It also achieves good results on the test set, on par with HiVT and ProphNet [46] in ADE, whereas it falls behind the most recent query-based model QCNet [57].

In the nuScenes benchmark, LAformer achieves competitive performance, only slightly inferior to FRM [34] and falls behind the latest goal-based model Goal-LBP [48]. FRM introduces relationship reasoning to help understand future interactions between the target and other agents, while LAformer relies on attention mechanisms to learn interactions between agents and focuses more on scene constraints. Goal-LBP [48] further accumulates the observed trajectories in the neighborhood of the target agent and uses this information to estimate the potential goal positions of the target agent. This difference in approach may contribute to the performance difference. However, LAformer outperforms other lane-based models, *e.g.*, LaneGCN [26, 55] and PGP [11], with a clear margin, indicating that our lane-aware estimation is more effective than the other distance-based or heuristic lane searching.

Paper	Val set ADE	Val set FDE	Test set ADE	Test set FDE
THOMAS [13]‡	-	-	0.94	1.44
TNT [54]	0.73	1.29	0.91	1.45
LaneRCNN [52]	0.77	1.19	0.90	1.45
GOHOME [14]‡	-	-	0.89	1.29
DenseTNT [17]	0.73	1.05	0.88	1.28
LaneGCN [26]	0.71	1.08	0.87	1.36
LaneGCN [26, 55]♢‡	-	-	0.84	1.30
mmTrans [28]	-	-	0.84	1.34
MultiModalTrans [20]			0.84	1.29
LTP [45]	-	-	0.83	1.29
Goal-LBP [48]‡	-	-	0.83	1.33
TPCN [49]	0.73	1.15	0.82	1.24
FRM [34]‡	0.68	0.99	0.82	1.27
SceneTrans [33]	-	-	0.80	1.23
Multipath++ [43]	-	-	0.79	1.21
HiVT [56]	0.66	0.96	0.77	1.17
ProphNet [46]	-	-	0.77	1.14
QCNet [57]	-	-	**0.73**	**1.07**
LAformer (Ours)‡	**0.64**	**0.92**	0.77	1.16

Table 1. The results *Argoverse 1* [4] with $\mathsf{K} = 6$. The best/second-best values are highlighted in boldface/underlined.

It is worth noting that when compared to the models (marked by ‡) tested on both benchmarks, namely THOMAS [13], GOHOME [14], LaneGCN ♢ enhanced by local behavior data LBA [55] as well as Goal-LBP [48], and FRM, LAformer shows a more generalized performance across the benchmarks. This suggests that the proposed temporally dense lane-aware estimation module effectively aligns scene constraints with motion dynamics, even though the trajectories provided in Argoverse 1 and nuScenes include locations in different cities and driving directions. Further evidence supporting the efficacy of this module can be found in the following ablation study presented in Table 3.

4.3. Qualitative results

Figure 3 presents qualitative results of LAFormer compared to DenseTNT and HiVT on the *Argoverse 1* validation set[1].

[1]More qualitative results are presented in the supplementary material.

Paper	K = 5		K = 10	
	ADE	MR	ADE	MR
Multipath [3]	2.32	-	1.96	-
CoverNet [36]	1.96	0.67	1.48	-
Trajectron++ [41]	1.88	0.70	1.51	0.57
AgentFormer [51]	1.86	-	1.45	-
ALAN [32]	1.87	0.60	1.22	0.49
SG-Net [44]	1.86	0.67	1.40	0.52
WIMP [21]	1.84	0.55	1.11	0.43
MHA-JAM [31]	1.81	0.59	1.24	0.46
CXX [29]	1.63	0.69	1.29	0.60
LaPred [22]	1.53	-	1.12	-
P2T [10]	1.45	0.64	1.16	0.46
LaneGCN [26, 55]♢‡	-	0.49	0.95	0.36
GOHOME [14]‡	1.42	0.57	1.15	0.47
Autobot [15]	1.37	0.62	1.03	0.44
THOMAS [13]‡	1.33	0.55	1.04	-
PGP [11]	1.30	0.61	1.00	0.37
Q-EANet [5]	1.18	0.48	1.02	0.44
FRM [34]‡	1.18	0.48	**0.88**	0.30
Goal-LBP [48]‡	**1.02**	**0.32**	0.93	**0.27**
LAformer (Ours)‡	1.19	0.48	0.93	0.33

Table 2. The results on the *nuScenes* [2] benchmark online test set. The best/second best values are highlighted in boldface/underlined.

They produce multimodal predictions for the target agent in various traffic scenarios at intersections, such as turning right ①and left ②, ④, or driving straight with acceleration ③. But LAformer generates more accurate predictions in the right-turn scenario ① and the acceleration scenario ③, while other models tend to predict decelerating or turning modes.

Furthermore, when the temporally dense lane-aware module is deactivated (i.e., models w/o D vs. w/o S2 without the second refinement model), LAformer generates less diverse predictions in the lateral directions. This reflects our concern that when the decoder initially deviates from the lane segments in its predictions, it becomes less likely to consider alternative lanes as time progresses. Therefore, the temporally dense lane-aware module operates more effectively compared to the spatially dense method. Moreover, the complete model with the second-stage refinement module S2 shown in the rightmost column maintains good prediction diversity and accuracy. It is interesting to observe that with the addition of the second refinement module further reducing the offsets from the ground truth, the diversity of predictions has decreased (w/o S2 vs. LAformer). The remaining predictions are now better aligned with the ground truth.

Figure 4 presents additional scenarios at various intersections in nuScenes. It can be clearly seen that LAformer predicts accurate and diverse trajectories for the target vehicle, including driving at a roundabout, curving along a road, traversing at intersections with multiple arms, and turning right or left.

4.4. Ablation study

Considering the data scale and availability of ground truth, we carry out the ablation study on the Argoverse 1 validation set with 39,472 sequences. The Baseline model predicts future trajectories only conditioned on the observed trajectories of the target and its neighboring agents, with the second refinement module (S2) and lane-aware estimation module removed. LAformer (Spa.) only estimates the likelihood of goal position aligning with the lane information, similar to the spatially dense models. In contrast, LAformer (Tem.) estimates the likelihood of the position at each time step aligning with the temporally dense lane information. LAformer (Full) is the complete proposed model.

From Table 3, the performance of the Baseline is much inferior to the other models. The comparison of Baseline vs. Baseline+S2 and LAformer (Tem.) vs. LAformer (Full) demonstrates the performance gain of S2, *e.g.*, ca. 3% in FDE. The comparison of LAformer (Spa.) vs. LAformer (Tem.) shows that our temporally dense method is more effective than the spatially dense method, reducing ADE by about 4% and FDE by about 8%.

Table 4 shows the analysis of the effectiveness $\mathcal{L}_{\text{angle}}$ and $\mathcal{L}_{\text{off}}$ losses in the second stage. Including both angle and offset losses helps to improve the prediction accuracy by about 2% in ADE/FDE. In summary, the enhanced performances validate the effectiveness of the temporally dense lane selection module and the second refinement module.

Name	S2	Goal	Dense lane	ADE_6	FDE_6
Baseline	-	-	-	0.72	1.12
Baseline+S2	√	-	-	0.71	1.08
LAformer (Spa.)	-	√	-	0.69	1.03
LAformer (Tem.)	-	-	√	0.66	0.95
LAformer (Full)	√	-	√	0.64	0.92

Table 3. Ablation study on the lane estimation and refinement modules.

$\mathcal{L}_{\text{angle}}$	$\mathcal{L}_{\text{off}}$	ADE_6	FDE_6
√	-	0.65	0.94
-	√	0.65	0.92
√	√	0.64	0.92

Table 4. Ablation study on the loss functions for angle $\mathcal{L}_{\text{angle}}$ and offset $\mathcal{L}_{\text{off}}$ errors in the second stage.

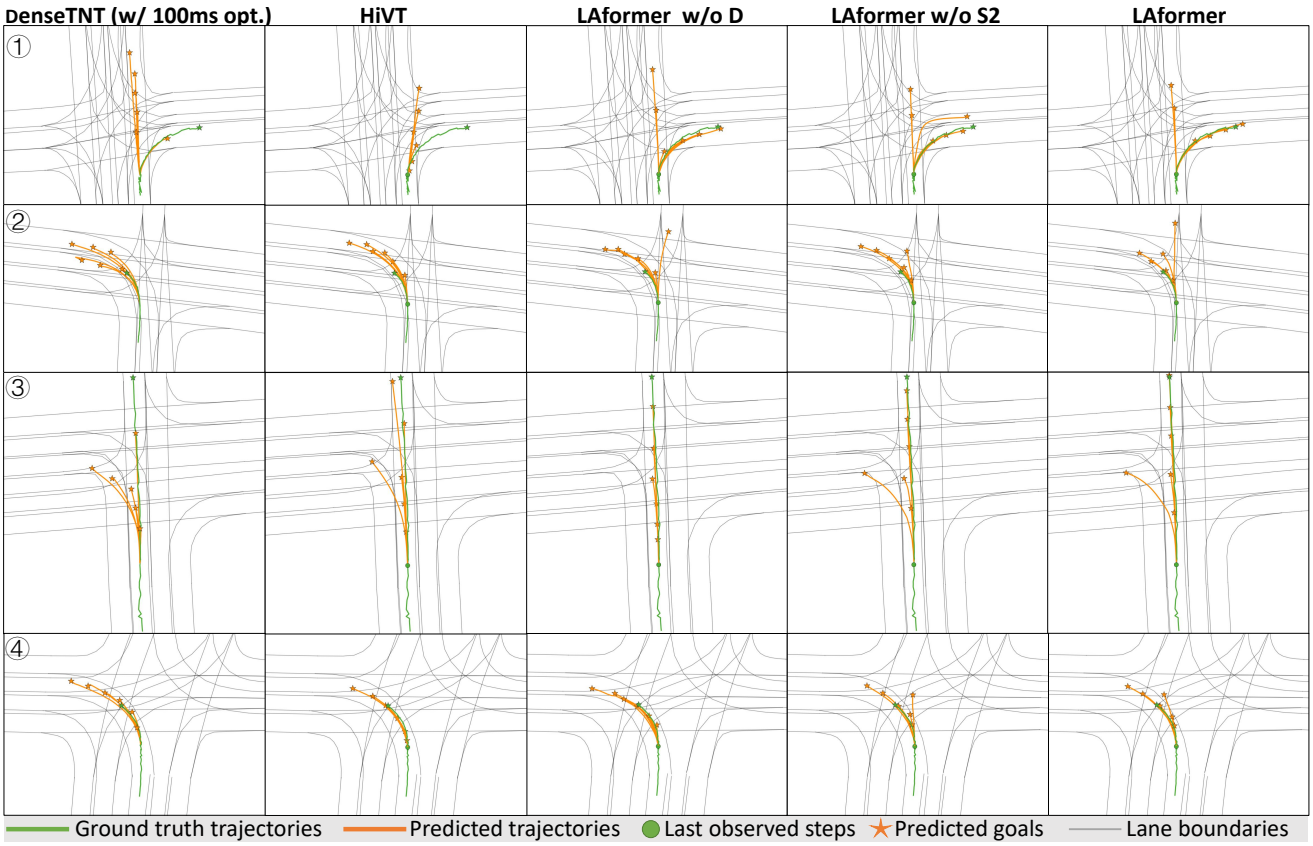

Figure 3. Qualitative comparison of models in complex scenarios, with each row representing a unique intersection scenario and each column representing results predicted by the same model.

We also ablate the latent variable z added to the input of the multimodal conditional decoder. It is sampled from a multivariate normal distribution with its dimension setting to 2. However, as shown in Table 5, we find that inserting z only leads to a marginal performance improvement (less than $0.5\,\mathrm{cm}$). Our conjecture is that we have used the Laplace MDN decoder to explicitly learn agents' multimodal behaviors. The addition of the latent variable has less influence on generating more diverse predictions.

z	ADE_6	FDE_6
-	0.692	1.035
$\surd$	0.690	1.032

Table 5. Ablation study on the latent variable z.

The number of top-k lane segments and the weights of losses are crucial hyper-parameters for LAformer. To examine their impact, we conduct an empirical study by varying their values around the experimental settings indicated by an underline in Tables 6 and 7.

As shown on the left side of Table 6, increasing the number of lane segments from 1 to 4 initially results in a performance gain up to $k = 3$, but after that, it starts to decline. In the second stage, $k = 2$ provides better results than 3. Using a larger k increases the chances of including irrelevant lane segments, while a relatively small k enables the decoder to focus on the most relevant lane segments.

k	ADE_6	FDE_6
1	0.68	1.00
<u>2</u>	0.66	0.95
3	0.65	0.95
4	0.66	0.96

k	ADE_6	FDE_6
<u>2</u>	0.64	0.92
3	0.64	0.93

Table 6. The number of top lane segments that impact the prediction performance. Left: first stage, Right: second stage.

We also vary the loss weights λ_1 in Eq. (9)) and λ_2, λ_3 in Eq. (12)). As shown in Table 7, we only observe marginal performance differences, *e.g.*, ADE_6 fluctuates within $1\,\mathrm{cm}$.

4.5. Computational performance

As reported in Table 8, LAformer has 2,645K parameters, similar to HiVT-128 but larger than LTP and DenseTNT. Its

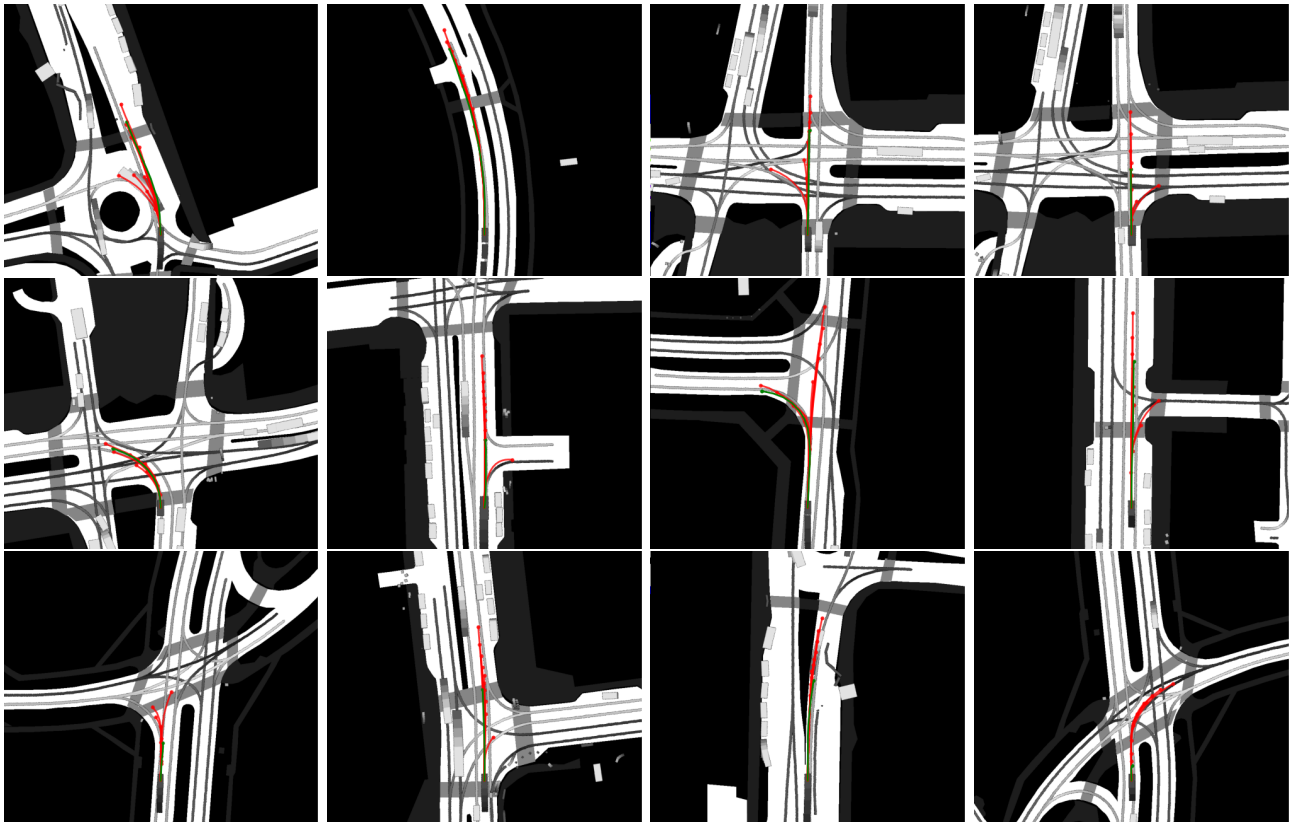

Figure 4. The qualitative comparison of the prediction results on the nuScenes [2] validation set. Predicted trajectories are presented in red color and the corresponding ground truth trajectories are presented in green color.

λ_1	ADE_6	FDE_6
8	0.70	1.05
9	0.70	1.02
<u>10</u>	0.69	1.01
11	0.70	1.05
12	0.70	1.05

λ_2	ADE_6	FDE_6
1	0.68	1.01
<u>5</u>	0.68	1.00
10	0.69	1.02
λ_3	ADE_6	FDE_6
1	0.68	1.00
<u>2</u>	0.68	1.00
10	0.68	1.01

Table 7. The loss weights that impact the prediction performance. Left: first stage, Right: second stage.

inference time for a scenario with an average of 12 agents is around 115 ms, which is not as good as HiVT. But this inference speed is comparable to LTP and faster than DenseTNT and PGP, making LAformer close to real-time use cases at 10 Hz.

In the supplementary material, we provide more detailed information about the implementation of the proposed model LAformer as well as extra qualitative results. There, we also report the failed cases LAformer encounters and discuss its limitations.

Model	#Params	Inference speed	
		Batch size	Time (ms)
LTP [45]	1,100k	8	92
DenseTNT [17]	1,103K	32	531
HiVT-128 [56]	2,529K	32	38
PGP [11]	-	12	215
LAformer (Ours)	2,654K	12	115

Table 8. Computational performance.

5. Conclusion

The paper presents LAformer, an end-to-end attention-based trajectory prediction model that takes observed trajectories and an HD map as input and outputs a set of multi-modal predicted trajectories. A Transformer-based temporally dense lane-aware module and a second-stage motion refinement module are used to improve prediction accuracy. LAformer demonstrates a superior generalized performance on Argoverse 1 and nuScenes. Moreover, extensive ablation and sensitivity studies verify the efficacy of the lane-aware and motion refinement modules.

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 961–971, 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, 5, 6, 8

[3] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *Proceedings of the Conference on Robot Learning*, pages 86–99. PMLR, 2020. 2, 3, 6

[4] Ming-Fang Chang, John W Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 2, 5

[5] Jiuyu Chen, Zhongli Wang, Jian Wang, and Baigen Cai. Q-eanet: Implicit social modeling for trajectory prediction via experience-anchored queries. *IET Intelligent Transport Systems*, 2023. 2, 6

[6] Yuxiao Chen, Boris Ivanovic, and Marco Pavone. Scept: Scene-consistent, policy-based trajectory predictions for planning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17103–17112, 2022. 2, 3

[7] Dooseop Choi and KyoungWook Min. Hierarchical latent structure for multi-modal vehicle trajectory forecasting. In *European Conference on Computer Vision*, pages 129–145. Springer, 2022. 2

[8] Marcos V Conde, Rafael Barea, Luis M Bergasa, and Carlos Gómez-Huélamo. Improving multi-agent motion prediction with heuristic goals and motion refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5322–5331, 2023. 3

[9] 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 Workshops*, pages 1468–1476, 2018. 2, 3

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

[11] Nachiket Deo, Eric Wolff, and Oscar Beijbom. Multimodal trajectory prediction conditioned on lane-graph traversals. In *Conference on Robot Learning*, pages 203–212. PMLR, 2022. 1, 2, 3, 4, 5, 6, 8

[12] 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. 1, 2, 3

[13] Thomas Gilles, Stefano Sabatini, Dzmitry Tsishkou, Bogdan Stanciulescu, and Fabien Moutarde. Thomas: Trajectory heatmap output with learned multi-agent sampling. In *International Conference on Learning Representations*, 2021. 1, 2, 3, 5, 6

[14] Thomas Gilles, Stefano Sabatini, Dzmitry Tsishkou, Bogdan Stanciulescu, and Fabien Moutarde. Gohome: Graph-oriented heatmap output for future motion estimation. In *2022 International Conference on Robotics and Automation*, pages 9107–9114. IEEE, 2022. 1, 2, 3, 5, 6

[15] Roger Girgis, Florian Golemo, Felipe Codevilla, Martin Weiss, Jim Aldon D'Souza, Samira Ebrahimi Kahou, Felix Heide, and Christopher Pal. Latent variable sequential set transformers for joint multi-agent motion prediction. In *International Conference on Learning Representations*, 2021. 6

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2

[17] Junru Gu, Chen Sun, and Hang Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15303–15312, 2021. 1, 3, 5, 8

[18] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2255–2264, 2018. 2

[19] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995. 2

[20] Zhiyu Huang, Xiaoyu Mo, and Chen Lv. Multi-modal motion prediction with transformer-based neural network for autonomous driving. In *2022 International Conference on Robotics and Automation*, pages 2605–2611. IEEE, 2022. 5

[21] Siddhesh Khandelwal, William Qi, Jagjeet Singh, Andrew Hartnett, and Deva Ramanan. What-if motion prediction for autonomous driving. *arXiv preprint arXiv:2008.10587*, 2020. 6

[22] ByeoungDo Kim, Seong Hyeon Park, Seokhwan Lee, Elbek Khoshimjonov, Dongsuk Kum, Junsoo Kim, Jeong Soo Kim, and Jun Won Choi. Lapred: Lane-aware prediction of multi-modal future trajectories of dynamic agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14636–14645, 2021. 1, 6

[23] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Inernational Conference on Learning Representations*, 2014. 2

[24] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. *Advances in Neural Information Processing Systems*, 27, 2014. 2

[25] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. Desire:

Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 336–345, 2017. 2, 3

[26] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *European Conference on Computer Vision*, pages 541–556. Springer, 2020. 3, 5, 6

[27] Mengmeng Liu, Hao Cheng, and Michael Ying Yang. Tracing the influence of predecessors on trajectory prediction. In *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 3245–3255. IEEE, 2023. 3

[28] Yicheng Liu, Jinghuai Zhang, Liangji Fang, Qinhong Jiang, and Bolei Zhou. Multimodal motion prediction with stacked transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7577–7586, 2021. 2, 5

[29] 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. 6

[30] Osama Makansi, Eddy Ilg, Ozgun Cicek, and Thomas Brox. Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7144–7153, 2019. 2, 4

[31] Kaouther Messaoud, Nachiket Deo, Mohan M Trivedi, and Fawzi Nashashibi. Multi-head attention with joint agent-map representation for trajectory prediction in autonomous driving. *arXiv preprint arXiv:2005.02545*, 2020. 6

[32] Sriram Narayanan, Ramin Moslemi, Francesco Pittaluga, Buyu Liu, and Manmohan Chandraker. Divide-and-conquer for lane-aware diverse trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15799–15808, 2021. 6

[33] Jiquan Ngiam, Vijay Vasudevan, Benjamin Caine, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, David J Weiss, Ben Sapp, Zhifeng Chen, and Jonathon Shlens. Scene transformer: A unified architecture for predicting future trajectories of multiple agents. In *International Conference on Learning Representations*, 2022. 3, 5

[34] Daehee Park, Hobin Ryu, Yunseo Yang, Jegyeong Cho, Jiwon Kim, and Kuk-Jin Yoon. Leveraging future relationship reasoning for vehicle trajectory prediction. In *International Conference on Learning Representations*, 2023. 5, 6

[35] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision*, pages 261–268. IEEE, 2009. 2

[36] 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. 1, 3, 6

[37] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538. PMLR, 2015. 2

[38] Nicholas Rhinehart, Rowan McAllister, Kris Kitani, and Sergey Levine. Precog: Prediction conditioned on goals in visual multi-agent settings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2821–2830, 2019. 2, 3

[39] Eitan Richardson and Yair Weiss. On gans and gmms. In *Advances in Neural Information Processing Systems*, 2018. 2

[40] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 1349–1358, 2019. 2, 3

[41] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *European Conference on Computer Vision*, pages 683–700. Springer, 2020. 1, 2, 3, 6

[42] Haoran Song, Wenchao Ding, Yuxuan Chen, Shaojie Shen, Michael Yu Wang, and Qifeng Chen. Pip: Planning-informed trajectory prediction for autonomous driving. In *European Conference on Computer Vision*, pages 598–614. Springer, 2020. 3

[43] Balakrishnan Varadarajan, Ahmed Hefny, Avikalp Srivastava, Khaled S Refaat, Nigamaa Nayakanti, Andre Cornman, Kan Chen, Bertrand Douillard, Chi Pang Lam, Dragomir Anguelov, et al. Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction. In *2022 International Conference on Robotics and Automation*, pages 7814–7821. IEEE, 2022. 2, 3, 5

[44] Chuhua Wang, Yuchen Wang, Mingze Xu, and David J Crandall. Stepwise goal-driven networks for trajectory prediction. *IEEE Robotics and Automation Letters*, 7(2):2716–2723, 2022. 6

[45] Jingke Wang, Tengju Ye, Ziqing Gu, and Junbo Chen. Ltp: Lane-based trajectory prediction for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17134–17142, 2022. 1, 2, 3, 5, 8

[46] Xishun Wang, Tong Su, Fang Da, and Xiaodong Yang. Prophnet: Efficient agent-centric motion forecasting with anchor-informed proposals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21995–22003, 2023. 2, 5

[47] Max Welling and Thomas N Kipf. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. 2

[48] Zhen Yao, Xin Li, Bo Lang, and Mooi Choo Chuah. Goal-lbp: Goal-based local behavior guided trajectory prediction for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 2023. 3, 5, 6

[49] Maosheng Ye, Tongyi Cao, and Qifeng Chen. Tpcn: Temporal point cloud networks for motion forecasting. In *Proceed-*

ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11318–11327, 2021. 5

[50] Cunjun Yu, Xiao Ma, Jiawei Ren, Haiyu Zhao, and Shuai Yi. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *European Conference on Computer Vision*, pages 507–523. Springer, 2020. 2

[51] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9813–9823, 2021. 1, 2, 3, 6

[52] Wenyuan Zeng, Ming Liang, Renjie Liao, and Raquel Urtasun. Lanercnn: Distributed representations for graph-centric motion forecasting. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 532–539. IEEE, 2021. 1, 2, 5

[53] Pu Zhang, Wanli Ouyang, Pengfei Zhang, Jianru Xue, and Nanning Zheng. Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12085–12094, 2019. 2

[54] Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Ben Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, et al. Tnt: Target-driven trajectory prediction. In *Conference on Robot Learning*, pages 895–904. PMLR, 2021. 1, 3, 5

[55] Yiqi Zhong, Zhenyang Ni, Siheng Chen, and Ulrich Neumann. Aware of the history: Trajectory forecasting with the local behavior data. In *European Conference on Computer Vision*, pages 393–409. Springer, 2022. 3, 5, 6

[56] Zikang Zhou, Luyao Ye, Jianping Wang, Kui Wu, and Kejie Lu. Hivt: Hierarchical vector transformer for multi-agent motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8823–8833, 2022. 2, 3, 4, 5, 8

[57] Zikang Zhou, Jianping Wang, Yung-Hui Li, and Yu-Kai Huang. Query-centric trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17863–17873, 2023. 2, 5