

PBP: Path-based Trajectory Prediction for Autonomous Driving

Sepideh Afshar*, Nachiket Deo*, Akshay Bhagat, Titas Chakraborty, Yunming Shao,
Balarama Raju Buddharaju, Adwait Deshpande, Henggang Cui
Motional

{sepideh.afshar, nachiket.deo, henggang.cui}@motional.com

Abstract—Trajectory prediction plays a crucial role in the autonomous driving stack by enabling autonomous vehicles to anticipate the motion of surrounding agents. Goal-based prediction models have gained traction in recent years for addressing the multimodal nature of future trajectories. Goal-based prediction models simplify multimodal prediction by first predicting 2D goal locations of agents and then predicting trajectories conditioned on each goal. However, a single 2D goal location serves as a weak inductive bias for predicting the whole trajectory, often leading to poor *map compliance*, i.e., part of the trajectory going off-road or breaking traffic rules. In this paper, we improve upon goal-based prediction by proposing the *Path-based prediction (PBP)* approach. PBP predicts a discrete probability distribution over reference paths in the HD map using the path features and predicts trajectories in the path-relative Frenet frame. We applied the PBP trajectory decoder on top of the HiVT scene encoder and report results on the Argoverse dataset. Our experiments show that PBP achieves competitive performance on the standard trajectory prediction metrics, while significantly outperforming state-of-the-art baselines in terms of map compliance.

I. INTRODUCTION

To safely navigate through traffic while offering passengers a smooth ride, autonomous vehicles need the ability to predict the trajectories of surrounding agents. There is inherent uncertainty in predicting the future, making this a challenging task. Agent trajectories tend to be highly non-linear over long prediction horizons. Additionally, the distribution of future trajectories is multimodal; in a given scene an agent could have multiple plausible goals and could take various paths to each goal.

In spite of these challenges, agent motion is not completely unconstrained. Vehicles tend to follow the direction of motion ascribed to their lanes, make legal turns and lane changes, and stop at stop signs and crosswalks. Bicyclists tend to use the bike lane, and pedestrians tend to walk along sidewalks and crosswalks. High-definition (HD) maps of traffic scenes efficiently represent such constraints on agent motion and have thus been a critical component of autonomous driving datasets [1], [2], [3], [4], [5]. In fact, it has been shown in many prior works [6], [7], [8], [9], [10], [11], [12] that a key requirement of the trajectory prediction task for a real-world autonomous driving system is to predict *map-compliant* trajectories – trajectories that don't go off-road or violate traffic rules over long prediction horizons. For example, incorrectly predicting a non-map-compliant trajectory that encroaches into the oncoming traffic

lane could cause the ego vehicle to brake hard or even make dangerous maneuvers on the road. As a result, prediction map compliance w.r.t. the provided HD map is central to our proposed approach and experimental evaluation.

Prior works have leveraged HD maps for trajectory prediction in two distinct ways. First, the HD map is often used as an input to the model. Early works [13], [14], [15] use rasterized HD maps and CNN encoders. More recent works directly encode vectorized HD maps using PointNet encoders [16], [17], graph neural networks [18] or transformer layers [19], [20], [21], [22]. The map encoding is then used by a multimodal prediction header to output K trajectories and their probabilities. A drawback of multimodal prediction headers is that they need to learn a complex one-to-many mapping from the entire scene context to multiple future trajectories, often leading to non-map-compliant predictions.

To address this shortcoming, a few recent works additionally use the HD map for *goal-based prediction* [23], [24], [25], [26], [27]. Goal-based prediction models associate each mode of the trajectory distribution to a 2D goal location sampled from the HD map. They predict a discrete distribution over the sampled goals, and then predict trajectories conditioned on each goal. This simplifies the mapping learned by the prediction header, and also makes each mode of the trajectory distribution more interpretable. However, 2D goal locations serve as a weak inductive bias to condition predictions, and may lead to imprecise trajectories for each goal.

In this work, we seek to improve upon goal-based trajectory prediction. We argue that *reference paths* rather than 2D goals are the appropriate HD map element to condition predicted trajectories. We define reference paths as segments of lane centerlines close to the agent of interest that the agent may follow over the prediction horizon. We propose a novel path classifier that predicts a discrete probability distribution over the candidate reference paths and a trajectory completion module that predicts trajectories conditioned on each path in the Frenet frame. Figure 1 shows an overview of our approach. In particular, our approach has two key advantages over goal-based prediction:

- 1) **Path features instead of goal features:** We predict trajectories conditioned on feature descriptors of the entire reference path instead of just 2D goal locations. This is a more informative feature descriptor and leads to more map-compliant trajectories over longer prediction horizons compared to goal-based prediction.

*Authors contributed equally.

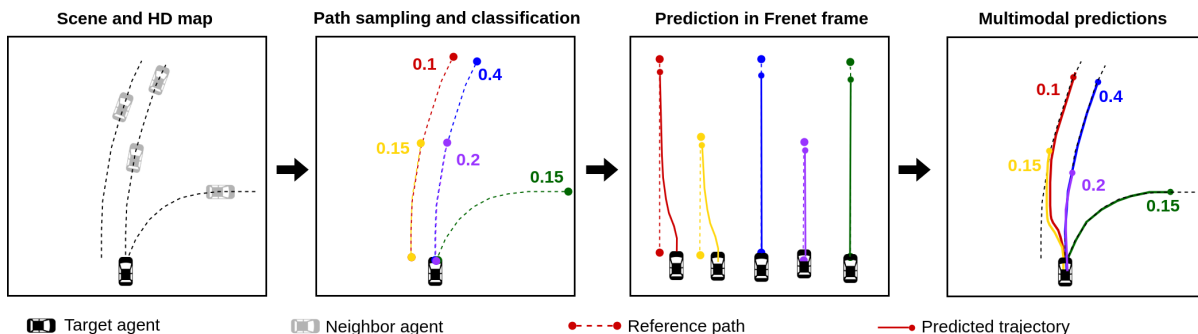


Fig. 1: **Overview of path-based prediction.** Path-based prediction predicts trajectories conditioned on *reference paths* rather than 2D goals. We sample reference paths using the lane network from HD maps, predict a discrete distribution over the sampled paths, and predict future trajectories in the Frenet frame relative to the paths. Finally, we transform the trajectories back to the Cartesian frame relative to the target agent to obtain multimodal predictions.

- 2) **Prediction in the Frenet frame:** The reference paths allow us to predict trajectories in the Frenet frame relative to the path. Compared to the Cartesian frame with varying lane locations and curvatures, predictions in the Frenet frame have much lower variance, which leads to more map-compliant trajectories that better generalize to novel scene layouts.

Our path-based trajectory decoder is modular by design and could be used with any existing scene encoder such as VectorNet [17], LaneGCN [18], Scene Transformer [19], Wayformer [21], etc. Here, we build our decoder on top of the recently proposed HiVT encoder [22] that achieved competitive results on the Argoverse dataset [1] and has a publicly available code base. Our results on the Argoverse dataset show that our path-based decoder achieves competitive performance in terms of the standard minADE, minFDE, and miss rate metrics, while significantly outperforming the HiVT baseline and goal-based prediction in terms of map compliance metrics.

Our contributions can be summarized as follows:

- We propose a novel path-based trajectory prediction (PBP) approach that improves upon traditional goal-based prediction.
- We applied our PBP trajectory decoder on top of the HiVT [22] scene encoder. The resulting model achieves the best map compliance metric on the Argoverse leaderboard while being competitive in terms of prediction error metrics.
- We present extensive ablation studies comparing different trajectory decoder approaches on the Argoverse validation set.

II. RELATED WORK

Map-compliant trajectory prediction: Leveraging the HD-map and predicting map-compliant trajectories has been the focus of a large number of works on trajectory prediction. Several works have proposed novel HD map encoders [28], [17], [18], [16], [19], [22], trajectory decoders conditioned on HD maps [23], [24], [26], [27], [29], [30], [31], and

even novel metrics and auxiliary loss functions for map-compliance [6], [7], [8], [9], [10], [11], [12]. In this work, we propose a path-based prediction approach that significantly improves prediction map compliance.

Goal-free multimodal prediction: The distribution of future trajectories is multimodal due to unknown intents of agents. Machine learning models for trajectory prediction thus need to learn a one-to-many mapping from the HD map and past states of agents, to multiple future trajectories. Prior work has addressed this using two approaches. The first approach is to implicitly learn the trajectory distribution using latent variable models such as GANs [32], [33], [34], CVAEs [35], [36], and normalizing flows [6], [37], where samples from the model represent plausible future trajectories. The other common approach is to use a multimodal regression header that outputs a fixed number of trajectories along with their probabilities [13], [18], [22], [19]. Such models are trained using the winner takes all/variety loss [32]. Some recent works [21], [38], [39], use DETR-like learned tokens [40] to output K distinct trajectories.

Goal-based prediction: Goal-based prediction models [23], [25], [26], [27], [24] partly address the above limitations by associating each mode of the trajectory distribution to a 2D goal in the HD map. TNT [23] samples a sparse set of goals along lane centerlines. LaneRCNN [25] uses nodes in a lane graph to predict goal locations. HOME [26] and GoHOME [27] predict goal heatmaps along a grid and graph representation of the HD map, and sample goal locations to optimize for the minFDE or miss rate metrics. Finally, DenseTNT [24] first predicts a dense goal heatmap along lanes, before using a second learned model to sample goals from the heatmap. We improve upon goal-based prediction models by conditioning our predictions on reference paths in the HD map rather than goals. Reference paths provide our trajectory decoder with more informative feature descriptors than 2D goal coordinates, and additionally allow us to predict in the path-relative Frenet frame.

Frenet frame trajectory decoding: There are some existing models that predict trajectories in path-relative Frenet frame, such as GoalNet [31], DAC [41], and WIMP [42]. PBP

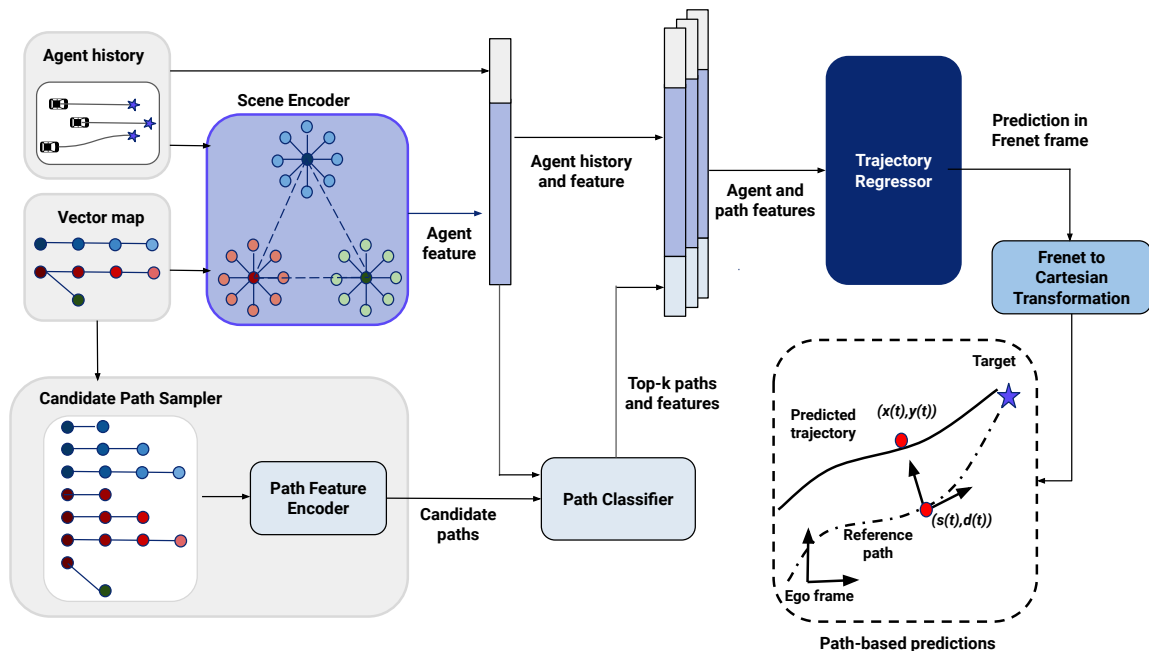


Fig. 2: **Model architecture:** Our model consists of four key modules. The *scene encoder* encodes the agent history and HD map information (Section III-C). The *candidate path sampler* samples candidate paths for each agent from the lane graph (Section III-D). The *path classifier* predicts a discrete distribution over the reference paths (Section III-E). Finally, the *trajectory regressor* decodes trajectory predictions in the path-relative Frenet frame conditioned on the paths (Section III-F).

has two key differences from those works. First, PBP has a different definition of its reference paths from those works. The reference paths in GoalNet, DAC, and WIMP are fixed-lengthed paths in the lane level. To generate the reference paths, GoalNet and DAC start from the agent’s current position and search along the lane graph for a fixed distance. Such reference paths only capture the agent’s high-level intention (e.g., go straight or turn right) but do not capture other uncertainties such as change of speed profiles. As a result, GoalNet, DAC, and WIMP all predict M trajectory modes within each reference path to achieve multimodal prediction. On the other hand, PBP’s reference paths are sequences of lane segments with variable lengths, and PBP relies entirely on its path classification to achieve multimodal prediction since a reference path can uniquely define a predictive mode. To highlight the difference, PBP considers around 600 candidate reference paths per agent, while GoalNet and DAC only consider less than three reference paths per agent. Second, DAC [41] and WIMP [42] do not have a learned path classification module to predict path probabilities or a path classification loss as a training objective. DAC uses a heuristic algorithm to rank paths based on the distance-along-lane score and centerline-yaw score, and WIMP finds only one single closest reference path for each agent using a heuristic algorithm. On the other hand, PBP has a path classification module that predicts the probability distribution over all candidate paths.

PRIME [43] also predicts trajectories in the Frenet frame, but it uses a model-based trajectory generator (a quartic poly-

nomial) to sample trajectories. In contrast, PBP’s trajectory generator is entirely learned, allowing it to generate a variety of motion profiles in the Frenet frame.

III. PBP: PATH-BASED PREDICTION

A. Problem statement

The objective of a trajectory prediction model is to forecast the future trajectories of a set of agents in the scene, given their past history positions and map context. We denote the past history positions of an agent a by $\{\mathbf{P}^a\}_{Past} = \{\mathbf{P}_{-T'+1}^a, \mathbf{P}_{-T'+2}^a, \dots, \mathbf{P}_0^a\}$ where $\mathbf{P}_t^a = (x_t^a, y_t^a)$ is a 2-D coordinate position, and $T' > 0$ is the past history length. The map context \mathcal{M} is represented as a set of discretized lane segments $\{l_j\}_{j=1}^L$ and their connections. The prediction model is required to forecast the future state of each agent $\{\mathbf{P}^a\}_{Future} = \{\mathbf{P}_1^a, \mathbf{P}_2^a, \dots, \mathbf{P}_T^a\}$ over the time horizon $T > 0$. In order to capture the uncertainties of the agents’ future behaviors, the model will output K trajectory predictions and their probabilities $\{p_k\}_{k=1}^K$ for each agent.

B. Overall architecture

The overall architecture of our PBP model is illustrated in Figure 2, which consists of four main components. The scene encoder generates agent and map embeddings from agent-map and agent-agent interactions (Section III-C). The candidate path sampler samples the candidate paths from the map for each agent (Section III-D). The path classifier predicts the probability of each sampled path (Section III-E). Finally, the trajectory regressor decodes trajectories conditioned on the selected paths (Section III-F).

C. Scene encoding

The scene encoder module creates agent feature vectors from the scene for each agent. In this work, we borrowed the scene encoder module from the HiVT model [22], a recently proposed trajectory prediction model that achieves state-of-the-art performance on Argoverse. The HiVT scene encoder represents each scene as a set of vectorized entities. It uses this representation to encode the scene by hierarchical aggregation of the spatial-temporal information. First, rotational invariant local feature vectors are encoded for each agent with a transformer module to aggregate neighboring agents' information as well as local map structure. Next, global interactions between agents are aggregated into each agent's feature vector to capture the scene-level context. The outputs of the encoder are the feature vectors for each agent denoted by \mathbf{F}_a .

D. Candidate sampling

The objective of the candidate sampling module is to create a set of candidate reference paths for each agent by traversing the lane graph. A reference path is defined as a sequence of connected lane segments $r_i = \{l_{i,1}, l_{i,2}, \dots, l_{i,R_i}\}$. The starting point of the reference path for an agent a is supposed to be in the vicinity of the agent's current location \mathbf{P}_0^a , and the endpoint is supposed to be in the vicinity of the agent's future trajectory endpoint \mathbf{P}_T^a , as is illustrated in Figure 1.

To select the candidate reference path for an agent a , we first select a set of *seed lane segments* that will be considered as the path starting points. We used a simple heuristic to select the seed lane segments by picking the lane segments that are within a distance range of the agent's current location and have their lane directions within a range of the agent's current heading. By picking the seed lanes this way, we will have candidate paths starting from not only the agent's current lane but also the neighbor lanes, which allows the model to predict lane-changing trajectories.

From the seed lane segments, we run a breadth-first search to find the candidate paths. The output of the candidate sampling module is a set of candidate reference paths for each agent, denoted as $\mathcal{R}^a = \{r_i^a\}$.

E. Path classification

Given the set of candidate reference paths, the path classification module predicts the probability distribution over them using the agent and path features.

To encode the features $\mathbf{F}_{p,i}$ of a path $r_i = \{l_{i,1}, l_{i,2}, \dots, l_{i,R_i}\}$, we pick the start segment $l_{i,1}$, the middle segment $l_{i,R_i}/2$, and the end segment l_{i,R_i} of the path, and use their coordinates and direction vectors as the raw feature. We encode those raw features with an MLP to a feature vector \mathbf{F}_p .

In addition to the agent and path features, we also create an agent-path pair feature that captures the interactions between the agent and the path. We use the distance vectors and angle deltas from the agent's current location to the start, middle, and end segments of the path as the raw features. We then

use another MLP network to encode them to an agent-path pair feature vector $\mathbf{F}_{a,(p,i)}$

We concatenate the agent feature \mathbf{F}_a , path feature \mathbf{F}_p , and agent-path pair feature $\mathbf{F}_{a,(p,i)}$ together and run them through another MLP network to predict the probability distribution over all candidate paths of the agent, trained with the cross-entropy loss as \mathcal{L}_{cls} . We decide the ground-truth reference path r_{GT}^a of the agent a based on its ground-truth future trajectory $\{\mathbf{P}^a\}_{Future}$, similar to the ground-truth goal selection in goal-based prediction. At inference time, we use non-maximum suppression (NMS) to sample a set of K diverse paths to decode the trajectory predictions.

F. Frenet frame trajectory decoding

The trajectory regressor module decodes trajectories conditioned on the reference paths. One key difference between our trajectory regressor and the one used in traditional goal-based prediction [23], [25], [26], [27], [24] is that it has the information of the whole reference path instead of just the final goal endpoint. To leverage this path information, we designed our trajectory regressor to decode trajectories in the path-relative Frenet frame.

For each selected reference path r_i^a , the trajectory regressor predicts a trajectory in path-relative Frenet frame, with longitudinal component $\{\hat{s}_t^a\}_{t=1\dots T}$ and lateral component $\{\hat{d}_t^a\}_{t=1\dots T}$, whose inputs include agent features \mathbf{F}_a , path features $\mathbf{F}_{p,i}$, and agent history in Frenet frame $\mathbf{P}_{Past,r_i^a}^a$.

During training, we use a teacher-forcing technique and train the trajectory regressor using the ground-truth reference path r_{GT}^a . We transform the ground-truth trajectory \mathbf{P}_{Future}^a to the Frenet frame w.r.t. r_{GT}^a , with longitudinal component $\{s_t^a\}_{t=1\dots T}$ and lateral component $\{d_t^a\}_{t=1\dots T}$.

The loss function is defined as smooth L1 losses of the longitudinal and lateral components in the Frenet frame:

$$\mathcal{L}_{reg}^a = \sum_{t=1}^T \mathcal{L}_{L1}(s_t^a, \hat{s}_t^a) + \lambda_{lateral} \mathcal{L}_{L1}(d_t^a, \hat{d}_t^a) \quad (1)$$

The total loss is a weighted sum of the path classification loss and the trajectory regression loss over all agents:

$$\mathcal{L}_{pbp} = \sum_{a \in \text{Agents}} \lambda_{cls} \mathcal{L}_{cls}^a + \mathcal{L}_{reg}^a \quad (2)$$

After predicting the trajectories in the Frenet frame, we transform them back to the Cartesian frame using the corresponding reference path, using the formulas in [44].

G. Path-free prediction for non-map-compliant agents

In order to robustly handle non-map-compliant agents (i.e., agents whose behaviors are not compliant with the annotated map), we additionally train a path-free trajectory decoder with the same architecture as the original HiVT decoder [22]. We also train a binary classifier to select the predictions between the two decoders for each agent. The path-free decoder and its classifier share the same scene encoder as the PBP decoder and use the agent feature vector \mathbf{F}_a as the input. During training, we label an agent as a path-free agent if its ground-truth trajectory is more than 5 meters away from any candidate reference path.

TABLE I: Decoder ablations on Argoverse validation set.

Decoder	minFDE ₁	MR ₁	minFDE ₆	MR ₆	Offroad rate	Lane dev.
Multimodal regression	2.93	0.481	0.996	0.101	0.069	0.510
Anchor-based	2.93	0.491	1.019	0.096	0.068	0.503
Goal-based	2.82	0.488	1.095	0.107	0.008	0.386
PBP in Cartesian frame	2.84	0.479	1.048	0.099	0.005	0.389
PBP (Ours)	2.82	0.473	1.008	0.095	0.004	0.386

TABLE II: Comparison to the state-of-the-art models on the Argoverse leaderboard

Model	minADE ₁	minFDE ₁	MR ₁	minADE ₆	minFDE ₆	MR ₆	DAC
TNT [23]	2.174	4.959	0.710	0.910	1.446	0.166	0.9889
DenseTNT [24]	1.679	3.632	0.584	0.882	1.282	0.126	0.9875
GoHOME [27]	1.689	3.647	0.572	0.943	1.450	0.105	0.9811
PRIME [43]	1.911	3.822	0.587	1.219	1.558	0.115	0.9898
HiVT-128 [22]	1.598	3.532	0.547	0.773	1.169	0.127	0.9888
MultiPath++ [38]	1.623	3.614	0.564	0.790	1.214	0.132	0.9876
DCMS [45]	1.477	3.251	0.532	0.766	1.135	0.109	0.9902
Wayformer [21]	1.636	3.656	0.572	0.767	1.162	0.119	0.9893
QCNet [46]	1.523	3.342	0.526	0.734	1.067	0.106	0.9887
PBP (Ours)	1.626	3.562	0.535	0.855	1.325	0.145	0.9930

IV. EXPERIMENTS

A. Dataset

We evaluate our model using the public Argoverse dataset [1]. Argoverse includes track histories of agents published at 10 Hz and vectorized HD maps. The task involves predicting the future trajectory of a focal agent in each scenario over a prediction horizon of 3 seconds, conditioned on 2 seconds of track histories and the HD map of the scene.

B. Implementation details

We implemented our path-based prediction decoder on top of the open-source HiVT-64 scene encoder [22]. We followed a similar training scheme as the original HiVT model for PBP and its variants. We used 8 AWS T4 GPUs for model training and evaluation. We trained each model for 64 epochs with a batch size of 4 and the Adam optimizer with a learning rate of 0.0005 and a decay weight of 0.0001.

C. Metrics

Best-of-K metrics: We report results using the standard metrics used for multimodal trajectory prediction: minADE_K, minFDE_K and miss rate (MR_K). The standard metrics compute prediction errors using the best of K predicted trajectories, in order to not penalize diverse but plausible modes predicted by the model. The minADE_K metric averages the L2 norms of displacement errors between the ground truth and the best mode over the prediction horizon. The minFDE_K metric computes the L2 norm of the displacement error between the final predicted waypoint of the best mode and the final waypoint in the ground truth. Finally, miss rate computes the fraction of all predictions where none of the K predicted trajectories are within 2 meters of the ground truth. We report results for $K=1$ and $K=6$, following the convention used in Argoverse.

Map compliance metrics: A key limitation of the standard best-of-k metrics is that they fail to penalize implausible predictions, even if they veer off-road or violate lane directions. Ideally, we want all K predictions to be plausible and map-compliant. Thus, we additionally report two map-compliance metrics. *Offroad rate* measures the fraction of the predicted waypoints at a given horizon falling outside the drivable area. This is closely related to Argoverse’s drivable area compliance (DAC) metric, but our offroad rate metric measures each individual waypoint and can report map compliance as a function of the prediction horizon as in Figure 3. *Lane deviation* measures the L2 distance between a predicted waypoint and the nearest lane centerline. It captures map compliance signals even when the waypoint is inside the drivable area. We report the two map-compliance metrics averaged over all waypoints along the whole prediction horizon and all $K = 6$ trajectories.

D. Decoder ablation study

We first perform a set of controlled experiments comparing our PBP model with path classification and Frenet frame trajectory decoder against the following alternative prediction decoders.

- *Multimodal regression:* This is the original HiVT-64 model [22]. It directly regresses multimodal predictions with the winner-takes-all loss.
- *Anchor-based:* This decoder is used in MultiPath [14]. It predicts offsets with respect to fixed anchor trajectories. We obtain the anchors using K-means clustering on the train set.
- *Goal-based:* The goal-based prediction decoder [23], [25], [24] uses only the goal endpoint features (no path features) in its goal classification module and decodes trajectories conditioned on goal endpoints (no Frenet frame).

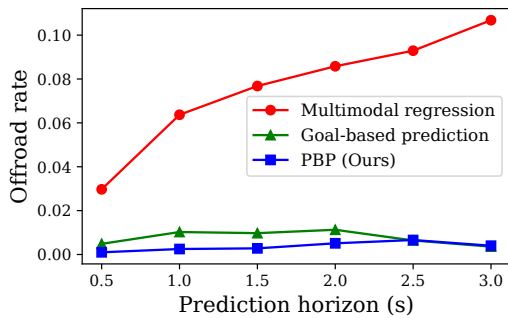


Fig. 3: Offroad rate.

- *PBP in Cartesian frame*: This decoder performs path classification as in PBP but decodes trajectories in the Cartesian frame instead of the Frenet frame.

For fair comparisons, we implemented all decoders using the same HiVT-64 encoder as PBP. The results are shown in Table I, and we observe the following.

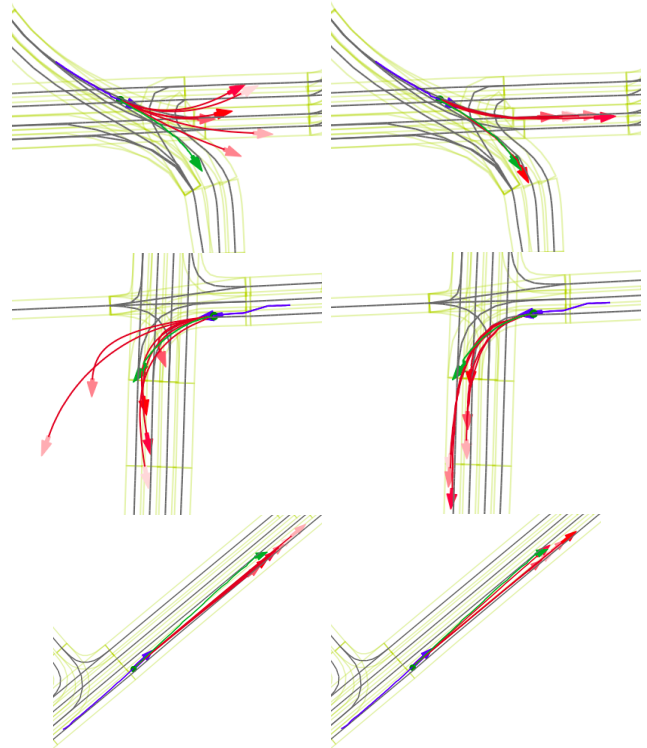
Significantly better map compliance. PBP and goal-based prediction achieve significantly lower offroad rates and lane deviation errors than multimodal regression and anchor-based decoders. This effect is even more pronounced over longer prediction horizons, as shown in Figure 3.

Advantage over goal-based prediction. Compared to goal-based prediction, PBP achieves overall lower prediction errors in terms of minFDE and MR and better map compliance metrics, because of the usage of richer path features. From Figure 3, goal-based prediction has strong map compliance at the final waypoint (i.e., goal endpoint), but it has higher offroad rates at the intermediate waypoints than PBP because of the missing path information.

Slightly worse mode diversity than goal-free decoders. PBP’s minFDE₆ metric is slightly worse than the multimodal regression baseline by 1%. This lower diversity is because PBP’s predictions are constrained to lanes (as is shown in Figure 4). We argue that it is a fair trade-off to have more map-compliant predictions for real-world autonomous driving applications.

E. Comparison against the state-of-the-art

We submitted our PBP model to the Argoverse leaderboard. Table II reports our results along with the top entries on the leaderboard. Our model achieves the highest drivable area compliance (DAC) on the leaderboard, outperforming state-of-the-art in terms of map compliance, while being competitive in terms of minADE₁, minFDE₁, and MR₁. Those results are consistent with our ablation study results on the validation set. PBP’s top-6 metrics are slightly worse than the top leaderboard submissions, but note that most of them used extensive model ensembling (e.g., [21], [38], [46], [47], [45]), while our submission used only one single pair of encoder and decoder. Our inference latency is 72.7 ms on an AWS T4 GPU, with 43.0 ms on the scene encoder and 29.7 ms on the trajectory decoder.



(a) HiVT-64

(b) PBP

Fig. 4: **Qualitative comparison between original HiVT-64 and PBP.** The first column shows the predictions from HiVT-64, and the second column shows the predictions from PBP. The blue, green, and red lines represent past history, ground-truth, and top-6 prediction trajectories, respectively.

F. Qualitative examples

Figure 4 shows a few qualitative comparisons between the HiVT-64 baseline (using multimodal regression) and PBP. The results show PBP predicts map-compliant trajectories from all modes, while HiVT-64 has many offroad predictions. The example on the last row shows that PBP is able to correctly predict lane-changing trajectories because the path candidates also contain paths on the neighbor lanes.

V. CONCLUSION

In this paper, we propose PBP, a novel path-based prediction approach. In contrast to the traditional goal-based prediction approaches, PBP performs classification on the whole reference path instead of just the goal endpoint. The additional reference path information improves the path classification accuracy and allows PBP to decode trajectories in the path-relative Frenet frame. Evaluation results show that the path-based prediction approach makes the trajectory predictions significantly more map-compliant compared to the traditional multimodal regression and goal-based prediction approaches, while maintaining competitive or better prediction accuracy.

REFERENCES

- [1] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, *et al.*, “Argoverse: 3d tracking and forecasting with rich maps,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8748–8757.
- [2] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, “nusenes: A multimodal dataset for autonomous driving,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.
- [3] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou, *et al.*, “Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9710–9719.
- [4] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, *et al.*, “Argoverse 2: Next generation datasets for self-driving perception and forecasting,” in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.
- [5] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari, “nuPlan: A closed-loop ml-based planning benchmark for autonomous vehicles,” *arXiv preprint arXiv:2106.11810*, 2021.
- [6] N. Rhinehart, K. M. Kitani, and P. Vernaza, “R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 772–788.
- [7] M. Niedoba, H. Cui, K. Luo, D. Hegde, F.-C. Chou, and N. Djuric, “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*, 2019.
- [8] F. A. Boulton, E. C. Grigore, and E. M. Wolff, “Motion prediction using trajectory sets and self-driving domain knowledge,” *arXiv preprint arXiv:2006.04767*, 2020.
- [9] H. Cui, H. Shajari, S. Yalamanchi, and N. Djuric, “Ellipse loss for scene-compliant motion prediction,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8558–8564.
- [10] D. Ridel, N. Deo, D. Wolf, and M. Trivedi, “Scene compliant trajectory forecast with agent-centric spatio-temporal grids,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2816–2823, 2020.
- [11] 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.
- [12] D. Zhu, M. Zahran, L. E. Li, and M. Elhoseiny, “Motion forecasting with unlikelihood training in continuous space,” in *Conference on Robot Learning*. PMLR, 2022, pp. 1003–1012.
- [13] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric, “Multimodal trajectory predictions for autonomous driving using deep convolutional networks,” in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2090–2096.
- [14] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, “Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction,” in *Conference on Robot Learning (CoRL)*. PMLR, 2020, pp. 86–99.
- [15] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff, “CovertNet: Multimodal behavior prediction using trajectory sets,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 14 074–14 083.
- [16] M. Ye, T. Cao, and Q. Chen, “Tpcn: Temporal point cloud networks for motion forecasting,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 318–11 327.
- [17] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, “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.
- [18] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun, “Learning lane graph representations for motion forecasting,” in *European Conference on Computer Vision*. Springer, 2020, pp. 541–556.
- [19] J. Ngiam, B. Caine, V. Vasudevan, Z. Zhang, H.-T. L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal, *et al.*, “Scene transformer: A unified multi-task model for behavior prediction and planning,” *arXiv e-prints*, pp. arXiv–2106, 2021.
- [20] Y. Liu, J. Zhang, L. Fang, Q. Jiang, and B. Zhou, “Multimodal motion prediction with stacked transformers,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7577–7586.
- [21] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp, “Wayformer: Motion forecasting via simple & efficient attention networks,” *arXiv preprint arXiv:2207.05844*, 2022.
- [22] Z. Zhou, L. Ye, J. Wang, K. Wu, and K. Lu, “Hivt: Hierarchical vector transformer for multi-agent motion prediction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8823–8833.
- [23] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid, *et al.*, “Tnt: Target-driven trajectory prediction,” in *Conference on Robot Learning*. PMLR, 2021, pp. 895–904.
- [24] J. Gu, C. Sun, and H. Zhao, “Densentn: End-to-end trajectory prediction from dense goal sets,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 303–15 312.
- [25] W. Zeng, M. Liang, R. Liao, and R. Urtasun, “Lanercnn: Distributed representations for graph-centric motion forecasting,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 532–539.
- [26] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, “Home: Heatmap output for future motion estimation,” in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 500–507.
- [27] —, “Gohome: Graph-oriented heatmap output for future motion estimation,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 9107–9114.
- [28] N. Djuric, V. Radosavljevic, H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, N. Singh, and J. Schneider, “Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 2095–2104.
- [29] J. Wang, T. Ye, Z. Gu, and J. Chen, “Ltp: Lane-based trajectory prediction for autonomous driving,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 134–17 142.
- [30] N. Deo, E. Wolff, and O. Beijbom, “Multimodal trajectory prediction conditioned on lane-graph traversals,” in *Conference on Robot Learning*. PMLR, 2022, pp. 203–212.
- [31] L. Zhang, P.-H. Su, J. Hoang, G. C. Haynes, and M. Marchetti-Bowick, “Map-adaptive goal-based trajectory prediction,” in *Conference on Robot Learning*. PMLR, 2021, pp. 1371–1383.
- [32] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, “Social gan: Socially acceptable trajectories with generative adversarial networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2255–2264.
- [33] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezaatfighi, and S. 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*, 2019, pp. 1349–1358.
- [34] T. Zhao, Y. Xu, M. Monfort, W. Choi, C. Baker, Y. Zhao, Y. Wang, and Y. N. Wu, “Multi-agent tensor fusion for contextual trajectory prediction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 126–12 134.
- [35] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, “Desire: Distant future prediction in dynamic scenes with interacting agents,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 336–345.
- [36] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, “Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data,” in *European Conference on Computer Vision*. Springer, 2020, pp. 683–700.
- [37] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine, “Precog: Prediction conditioned on goals in visual multi-agent settings,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2821–2830.

- [38] B. Varadarajan, A. Hefny, A. Srivastava, K. S. Refaat, N. Nayakanti, A. Cornman, K. Chen, B. Douillard, C. P. Lam, D. Anguelov, *et al.*, “Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7814–7821.
- [39] X. Wang, T. Su, F. Da, and X. Yang, “Prophnet: Efficient agent-centric motion forecasting with anchor-informed proposals,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 995–22 003.
- [40] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *European conference on computer vision*. Springer, 2020, pp. 213–229.
- [41] S. Narayanan, R. Moslemi, F. Pittaluga, B. Liu, and M. Chandraker, “Divide-and-conquer for lane-aware diverse trajectory prediction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 799–15 808.
- [42] S. Khandelwal, W. Qi, J. Singh, A. Hartnett, and D. Ramanan, “What-if motion prediction for autonomous driving,” *arXiv preprint arXiv:2008.10587*, 2020.
- [43] H. Song, D. Luan, W. Ding, M. Y. Wang, and Q. Chen, “Learning to predict vehicle trajectories with model-based planning,” in *Conference on Robot Learning*. PMLR, 2022, pp. 1035–1045.
- [44] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, “Optimal trajectory generation for dynamic street scenarios in a frenet frame,” in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 987–993.
- [45] M. Ye, J. Xu, X. Xu, T. Cao, and Q. Chen, “Dcms: Motion forecasting with dual consistency and multi-pseudo-target supervision,” *arXiv preprint arXiv:2204.05859*, 2022.
- [46] Z. Zhou, J. Wang, Y.-H. Li, and Y.-K. Huang, “Query-centric trajectory prediction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 863–17 873.
- [47] Y. Wang, H. Zhou, Z. Zhang, C. Feng, H. Lin, C. Gao, Y. Tang, Z. Zhao, S. Zhang, J. Guo, *et al.*, “Tenet: Transformer encoding network for effective temporal flow on motion prediction,” *arXiv preprint arXiv:2207.00170*, 2022.