

SUPPLEMENTARY MATERIAL: LEVERAGING FUTURE RELATIONSHIP REASONING FOR VEHICLE TRAJECTORY PREDICTION

1 ASSESSMENT OF ASSUMPTION VALIDITY

In our method, we assume that vehicles mainly follow lane centerlines. In fact, even if the influencer (giving interaction) does not follow lanes, our method is still valid at least if the reactor (receiving interaction) follow lanes. Nevertheless, we investigated the frequency of instances where this assumption is violated in the nuScenes trajectory dataset.

To determine if a vehicle is following a lane, we filtered lane candidates whose directions are within ± 30 degrees of the vehicle’s heading and selected the closest lane candidate to the vehicle’s position. We then examined all future time steps of the vehicle trajectory. If the distance between the vehicle’s position and the closest lane is smaller than half the width of the lane (generally 1.5m), we considered the vehicle to be following the designated lane.

Our analysis of the dataset showed that the average distance between the vehicle and the supposed following lane was 0.57m, and the vehicle followed the lane in 94% of the entire dataset. The assumption is violated only in a small fraction of cases, such as during U-turns or in noisy segments resulting from detection errors.

2 DETAILS OF ARCHITECTURE

In this section, we provide further details of the proposed architecture.

2.1 ENCODER

The encoder is composed of two components: Agent Encoder and Lane Encoder. The Agent Encoder encodes the agents’ past trajectories, while the Lane Encoder encodes sequences of lanes. Both encoders use a 1-layer GRU with a hidden dimension of 128, and we take the last values among output sequences. Followed by the GRU, the Agent Encoder has a 1-layer Multi-head attention (MHA) encoder layer. The MHA layer has 4 parallel heads and a feed-forward dimension of 256. The feed-forward and self-attention blocks have a 0.1 dropout layer, and the feed-forward layer has a ReLU activation. The Lane Encoder has a 2-layer Graph Attention Network (GAT). GAT follows a layer-wise propagation rule along predefined lane edges: successor, predecessor, left/right neighbor, and in the same intersection. Each GAT layer has a 0.5 dropout layer, and the output dimension is 128, same as the input.

2.2 WAYPOINT OCCUPANCY PREDICTION

Goal occupancy is predicted by a Multi-Layer Perceptron (MLP) from a concatenated feature of h_x of 128 dimensions and h_ℓ of 128 dimensions. The MLP has 2 layers with a hidden dimension of 128 and t_f output dimensions.

2.3 FUTURE RELATION MODULE

The Future Relation Module then utilizes a 2-layer Graph Convolutional Network (GCN) to smooth waypoint occupancy, which is predicted from past trajectory or obtained from the Ground Truth (GT) future trajectory. The GCN layer propagates 3-dimensional occupancy according to the adjacency matrix obtained from lane edges: successor, predecessor, left/right neighbor, and in the same

intersection. Therefore, the source waypoint occupancy is multiplied with a 5-dimensional trainable weight vector and then added to the destination occupancy feature. Lane-wise spatial proximity PR is then fed to a 1D convolutional layer with zero padding of 1, a kernel size of 2, and a stride of 2. For the posterior distribution, an MLP is applied to obtain a 64-dimensional feature, which is split into a 32-dimensional μ and σ . For the prior distribution, the output of the MLP is a $64 \times K$ -dimensional feature, which corresponds to K pairs of μ and σ . Here, we set K as 4. Another MLP is applied to get a 32-dimensional feature from h_x , and the obtained feature is Hadamard multiplied to the interaction feature h_R following Eq.(7) in the main paper.

2.4 DECODER

The decoder takes the feature of 288 dimensions, which is a concatenation of h_x of 128 dimensions, h_ℓ of 128 dimensions, and h_R of 32 dimensions. The feature is fed to a 2-layer MLP with 144 hidden dimensions and a 0.01 slope of the Leaky ReLU activation function. For the nuScenes dataset, the MLP outputs a 24-dimensional output, which is a 2-dimensional coordinate (x, y) of 12 time steps (6 seconds \times 2Hz). For the Argoverse dataset, the output decoder is a 30-dimensional feature, which corresponds to 3 seconds with 10 Hz. With a large number of predicted trajectories, we follow the clustering and scoring method to obtain 10 final trajectory outputs.

3 OTHER IMPLEMENTATION DETAILS

How to obtain lane connectivity

To obtain lane connectivity edges, we use predecessor and successor connectivity as defined in the nuScenes map API. We also define neighboring connectivity as lanes that are not predecessor or successor, but are within 4 meters of pairwise distance and have a yaw difference of less than 45 degrees. For *in same intersection* connectivity, lanes are not predecessor or successor, are not neighboring each other, and should be inside the same intersection polygon defined in the nuScenes API. During data augmentation, we randomly flip all coordinates of trajectories and lane information horizontally. When flipping lane information, the left and right neighbor connectivity is reversed.

How to obtain GT waypoint occupancy from GT future trajectory

To obtain the GT waypoint occupancy from the GT future trajectory, we use the location and heading of the vehicles at each future timestep. We choose a waypoint as the nearest lane with a direction within 45 degrees of the vehicle’s heading at each timestamp. If there are two lanes that are equidistant within a threshold of 0.1 meters, both lanes are chosen as waypoints.

4 METRICS DEFINITIONS

In this section, we provide definitions for all metrics used in this paper. The meaning of each symbol is consistent with the main paper.

minimum Average Displacement Error (mADE) The ADE measures the average L2 distances between the predicted trajectory $x_t^i = (x_t^i, y_t^i)$ and its corresponding ground truth \hat{x}_t^i for i -th agent and t -th time step. The mADE_k represents the minimum ADE over the k most likely predictions.

$$\text{ADE} = \frac{1}{N} \sum_{i=1}^N \frac{1}{t_f} \sum_{t=1}^{t_f} \|x_t^i - \hat{x}_t^i\|_2 \quad (1)$$

$$\text{mADE}_k = \min_k (\text{ADE}_{(1)}, \dots, \text{ADE}_{(k)}) \quad (2)$$

minimum Final Displacement Error (mFDE) The FDE measures the L2 distances between the final points of the prediction and ground truth. The mFDE_k represents the minimum FDE over the k most likely predictions.

$$\text{FDE} = \frac{1}{N} \sum_{i=1}^N \|x_{t_f}^i - \hat{x}_{t_f}^i\|_2 \quad (3)$$

$$\text{mFDE}_k = \min_k (\text{FDE}_{(1)}, \dots, \text{FDE}_{(k)}) \quad (4)$$

Table 1: Comparison with other methods on the Argoverse valid/test set in mADE₆

Category	Method	Val set	Test set	Decline ratio (from val to test)
Goal-based	TNT	0.73	0.94	-28.7%
	DenseTNT	0.73	0.88	-20.5%
	Ours-baseline	0.71	0.86	-21.2%
	Ours-full	0.68	0.82	-20.6%
	avg			-22.8%
Other	LaneRCNN	0.77	0.90	-14.4%
	TPCN	0.73	0.87	-19.2%
	Autobot	0.73	0.89	-21.9%
	mmTransformer	0.72	0.84	-16.6%
	SceneTransformer	-	0.80	-
	Multipath++	-	0.79	-
	HiVT	0.66	0.77	-16.6%
	avg			-17.7%

Miss Rate (MR) The MR is the proportion of missed predictions over all predictions. A prediction is considered a miss if its maximum pointwise L2 distance to the ground truth is greater than 2 meters, following the nuScenes benchmark. The MR_k takes the k most likely predictions and determine whether they are missed predictions or not. If there are m misses out of a total of n predictions, the MR would be $\frac{m}{n}$.

5 DERIVATION OF KULLBACK–LEIBLER (KL) TERM FOR OBJECTIVE

We assume a Gaussian mixture distribution as the prior distribution of interaction to allow multi-model interaction, given by:

$$p_{\theta}(\mathbf{z} \mid \mathbf{x}) = \sum_{i=1}^k \pi_i \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2) \quad (5)$$

where π_i is the mixture weight, $\boldsymbol{\mu}_i$ is the mean, and $\boldsymbol{\sigma}_i^2$ is the covariance matrix of the i -th Gaussian component.

On the other hand, we model the posterior distribution of interaction as a single Gaussian distribution, given by:

$$q_{\phi}(\mathbf{z} \mid \mathbf{x}, \mathbf{y}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2) \quad (6)$$

where $\boldsymbol{\mu}$ is the mean, and $\boldsymbol{\sigma}^2$ is the covariance matrix of the Gaussian distribution.

The Kullback–Leibler (KL) divergence is used to measure the difference between these two distributions, The Eqs.(7) express closed-form variational approximation of the KL-divergence.

$$D_{\text{KL}}(q \parallel p) \approx \sum_i \mathbf{w}_i \log \frac{\sum_{i'} \mathbf{w}_{i'} \exp(-D_{\text{KL}}(q_i \parallel q_{i'}))}{\sum_j \pi_j \exp(-D_{\text{KL}}(q_i \parallel p_j))} \quad (7)$$

Since we define the posterior distribution as a single Gaussian distribution, Eq.(7) can be simplified as:

$$D_{\text{KL}}(q \parallel p) \approx \log \frac{1}{\sum_j \pi_j \exp(-D_{\text{KL}}(q \parallel p_j))} = D_{\text{vKL}}(q \parallel p). \quad (8)$$

6 QUANTITATIVE RESULT ON ARGOVERSE

We report our results on the Argoverse test and validation set and compare them with those of other methods in terms of mADE₆. In addition to previously compared methods, we include Multipath++

(Varadarajan et al. (2022)) and SceneTransformer (Ngiam et al. (2022)) for comparison in Table 1 of the supplementary material. While our method is more effective for long-range prediction, it might be less effective in the Argoverse set, which focuses more on short-range prediction. However, our model (0.82) still performs comparably to SceneTransformer (0.80) and Multipath++ (0.79). HiVT shows the best performance (0.77), possibly due to its use of the surrounding agent’s trajectory for training, increasing the amount of training data.

The performance decline of our method from the validation set to the test set is not less than that of HiVT. Our analysis shows that goal-based models seem to have a larger performance drop in the test set, with an average mADE₆ drop of 22.8%, compared to 17.7% for other methods. This indicates that the distribution gap is larger in map data than in motion data in the Argoverse. Recent work Bahari et al. (2022) reports performance degradation due to map data distribution gap, but also proposes a learning method to mitigate this decline. We believe that our method could achieve better test results by adopting the learning method proposed in Bahari et al. (2022).

Despite the performance drop, our model still has strengths in terms of diversity and plausibility, thanks to the goal-based method. Please refer to the qualitative comparison with HiVT in Fig. 1 of the supplementary material for more details. These strengths are crucial in autonomous driving systems, where various risks must be taken into account.

7 QUALITATIVE RESULT

We present a qualitative comparison with HiVT in Fig. 1 on the Argoverse dataset. It is observed that goal-based methods exhibit a relatively larger performance drop from the validation set to the test set. However, in autonomous driving systems, AVs must consider various risks in the future, and the goal-based approach has an advantage in terms of diversity. Furthermore, as seen in the third result, the output from our method appears to be more plausible.

8 COMPUTATION

We compared the computational cost by examining the total FLOP count. For the Argoverse dataset, we estimated the total FLOP count by taking into account the computing capabilities of the GPUs used and the training time for both SceneTransformer and Autobot: SceneTransformer takes 108,000,000 TFLOPs (420 TFLOPS TPU-v3 X 73 hours), while Autobot takes 396,000 TFLOPs (11 TFLOPS Nvidia 1080ti X 10 hours). Our model, on the other hand, takes 6,739,200 TFLOPs (39 TFLOPS Nvidia A6000 X 48 hours). Ultimately, our method performs competitively with SceneTransformer but with much fewer FLOPs, while showing superior performance compared to Autobot, albeit with more FLOPs.

9 LIMITATION

We model the interaction in a lane-wise manner using prior knowledge of the vehicle. Assuming that an agent normally follows lanes is efficient for vehicle trajectory prediction, but it is not directly applicable for pedestrian trajectory prediction, which is also crucial for autonomous driving. Although we can predict waypoint occupancy for pedestrians, since connectivity between waypoints is not provided, it is challenging to consider a relationship that reflects traffic rules like the method using lanes. In addition, our proposed method is heavily influenced by the domain of map information. As shown in Tab.1 of the supplementary material, the goal-based approach is particularly affected by the domain gap between the validation and test sets of the map information. Because our method uses waypoints inferred from map information, it can have a negative impact on performance in cases where the distribution of map information differs significantly. This is considered a limitation of our method.

REFERENCES

Mohammadhossein Bahari, Saeed Saadatnejad, Ahmad Rahimi, Mohammad Shaverdikondori, Amir Hossein Shahidzadeh, Seyed-Mohsen Moosavi-Dezfooli, and Alexandre Alahi. Vehicle

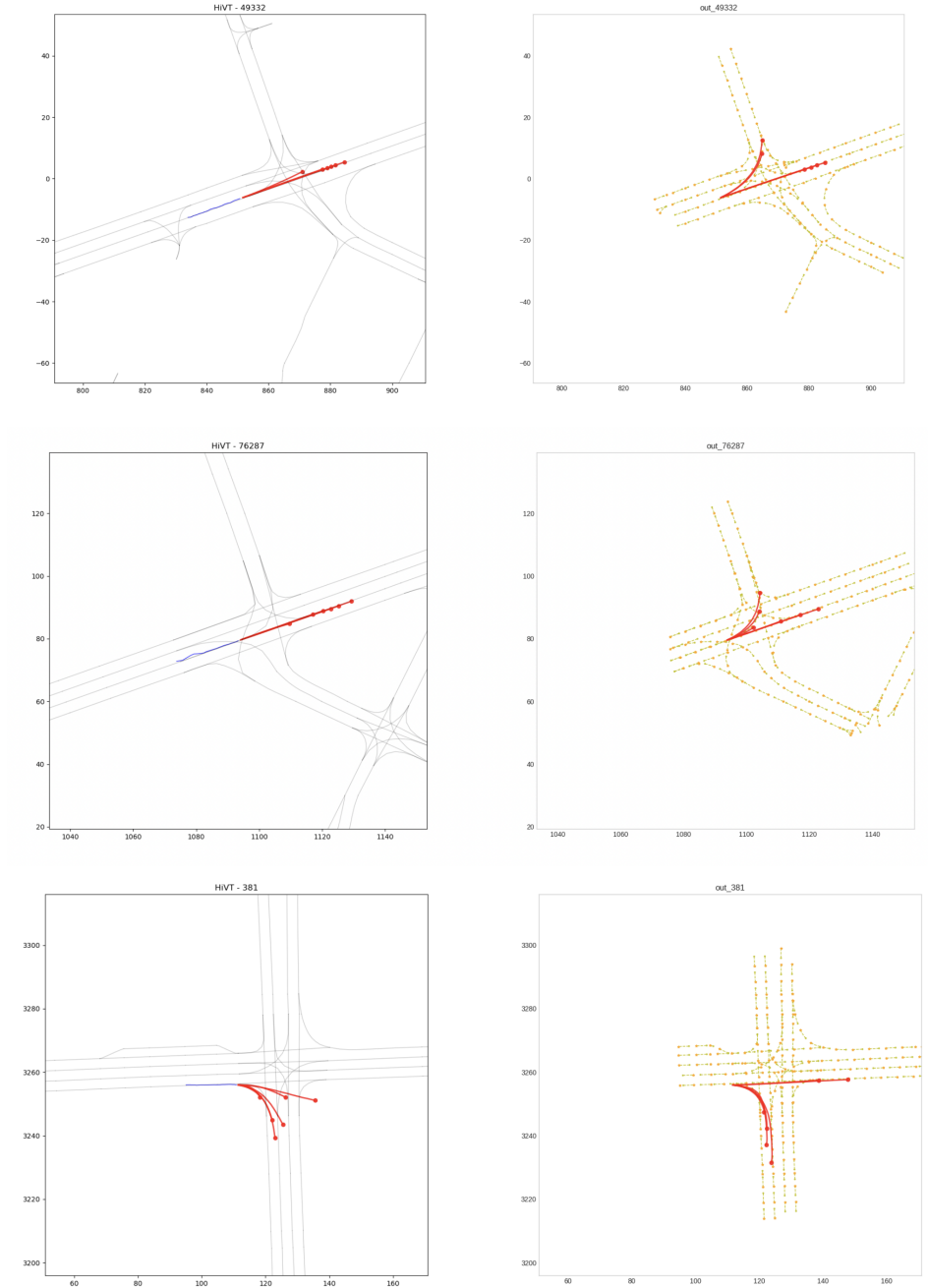


Figure 1: Qualitative comparison between HiVT (left) and our method (right) on the Argoverse dataset. While goal-based methods exhibit a relatively larger drop in performance from validation to test set, they have a clear advantage in terms of diversity, which is crucial in autonomous driving systems that must consider various future risks. Our method also produces more plausible trajectories, as seen in the third example.

trajectory prediction works, but not everywhere. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17123–17133, 2022.

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 *2022 International Conference on Learning Representations (ICLR)*, 2022. URL <https://openreview.net/forum?id=Wm3EA5O1HsG>.

Balakrishnan Varadarajan, Ahmed Hefny, Avikalp Srivastava, Khaled S Refaat, Nigamaa Nayakanti, Andre Cornman, Kan Chen, Bertrand Douillard, Chi Pang Lam, Dragomir Anguelov, et al. Multi-path++: Efficient information fusion and trajectory aggregation for behavior prediction. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 7814–7821. IEEE, 2022.