LAformer: Trajectory Prediction for Autonomous Driving with Lane-Aware Scene Constraints (Supplementary Material)

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

In this supplementary material, we provide more detailed information about the implementation of LAformer in Section A, the experimental results in Section B, and the discussion of limitations in Section C.

A. Implementation details

Format of vectorization. To obtain the sparse vector encoding of the input data–trajectories and lane segments–in a unified vector form, the input vector size is set to 32. Following the approach in [3, 4], for agent encodings, the first ten dimensions are dedicated to $v_t = [d_{t,s}, d_{t,e}, a]$, where $d_{t,s}$ denotes the start point (2-dim), $d_{t,e}$ denotes the end point (2-dim), a denotes an agent's trajectory attributes, *i.e.*, the timestamp (1-dim), object type (3-dim one-hot encoding for autonomous vehicles, target agent and others), vector length (1-dim), and relative time steps to the first observed time step (1-dim). The remaining dimensions are zero-padded. For lane segment encodings, the last eleven dimensions are dedicated to $v_n = [d_{n,s}, d_{n,e}, a, d_{n,\text{pre}}]$. Similarly, four dimensions are for the start and end points. a describes the lane segment attributes, i.e., its index among all the lane segments (1-dim), the semantic attributes of a lane segment such as if it is at an intersection (1-dim), if it belongs to a left-turn lane (1-dim) or a right-turn lane (1-dim), and if it has traffic control (1-dim). $d_{n,\text{pre}}$ (2-dim) indicates the predecessor of the start point. The remaining dimensions are zero-padded. It should be noted that the dimensions of the vector can be easily modified to include more or less attribute information.

Sliced lane segments. Following the approach in LTP [9] to model the intention of the target agent more precisely, we utilize sliced lane segments. Specifically, the length of each lane centerline segment is cut into $5\,\text{meters}$ or remains as the original length if the lane is shorter than $5\,\text{meters}$, with a $1\,\text{meter}$ resolution as the interval between two continuous centerline 2D positions. The ablation study on the length of sliced lane segments has been thoroughly conducted in [9]. Therefore, interested readers are referred to the paper for more detailed information.

Global Interaction Graph (GIG). Figure 1 denotes the operation of GIG. The unified vectorized trajectories and lane centerlines are encoded to learn agent-to-environment and agent-to-agent interactions via the attention mechanism [8]. First, two parallel embedding modules of identical structures extract spatial-temporal features from agent trajectory or lane segment vectors. Namely, a multi-layer perceptron (MLP) embeds each input vector into a fixed-length embedding independently. Subsequently, a gated recurrent unit (GRU) learns spatial-temporal features from the sequence of the trajectory or lane vectors and outputs their corresponding latent feature maps. The output features of agents' trajectories are h_i for $\forall i \in \{1, \ldots, N_{\text{traj}}\}$ and lane segments are c_j for $\forall j \in \{1, \ldots, N_{\text{lane}}\}$. Then, two three-layer cross-attention networks designed for modeling agent-to-environment interactions – *Agent2Lane* and *Lane2Agent* – have the agents attended to the lanes and vice versa. The outputs of *Agent2Lane* and *Lane2Agent* are used to update c_j and h_i, respectively. Afterwards, the GIG further explores the self-attention mechanism that also learns agent-to-agent interactions. Hence, the finally updated h_i and c_j after the GIG consider both agents' motion dynamics and scene contexts globally.

Ground truth label of the lane scores. The temporally dense lane scoring module employs a binary cross-entropy loss $\mathcal{L}_{\text{lane}}$ to optimize the motion and lane segment alignment probability estimation. To search for the ground truth lane segment at time step t, a distance function is used following the approach in [4]. Specifically, we use the shortest Euclidean distance between the set of all centerline 2D positions in the j-th lane segment $\mathbf{C}^j_{1:N}$ and the ground truth position Y_t as the distance metric, where N denotes the total vector length of this lane segment. The ground truth

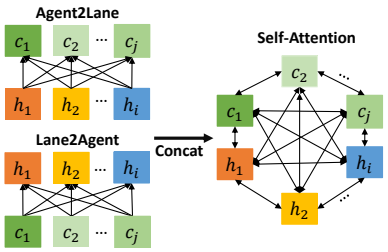

Figure 1. The agent-to-environment and agent-to-agent interaction module. First, the target's motion and the lane segments are encoded through Agent2Lane and Lane2Agent. Then, the encoded features are concatenated as the input for a global interaction graph with the self-attention mechanism.

label of the lane score $s_{j,t}$ equals one only if it has the minimum distance over all the other lane segments; otherwise, the ground truth label equals zero. It should be noted that this distance metric is applied to each time step for the motion and lane segment alignment.

Output of the final predicted trajectories. When utilizing the motion refinement module, the final predicted trajectories for each target agent are obtained by adding the predicted trajectories from the first stage (S1) and the predicted offset from the second stage (S2). It is defined as $\hat{Y} = \hat{Y}_{S1} + \Delta\hat{Y}_{S2}$.

Network setting. Only the lane segments that are within $50\,\mathrm{m}$ (Manhattan distance) of the target agent are sampled as the scene contexts. λ_1, λ_2, and λ_3 for the weights of the loss terms are set to 10, 5, and 2, respectively. The sensitivity analysis of these weights of the loss terms is presented in the main paper.

The hidden dimension of all the feature vectors in LAformer is set to 128. The activation functions used in the intermediate layers are Rectified Linear Units (ReLU). Similar to [11], we utilize the Softmax function to normalize the predicted $\hat{\pi}_m$ and ground truth probability π_m of each mode. For the Laplacian decoder, we use $\mathrm{ELU}(.)+1+\epsilon$, the Exponential Linear Unit (ELU) activation function with ϵ to ensure positive values for the probability estimation, where ϵ is set to $1e^{-3}$. In both Argoverse 1 and nuScenes, we set the learning rate to $1e^{-3}$.

We use a two-stage training scheme. In the first stage, all the modules except for the motion refinement module are trained using the Adam optimizer [5]. It should be noted that the temporally dense lane-aware module was trained jointly with the decoder. Only the refinement module was trained in the second stage because we first need to get decent predictions as anchors for the second stage. LAformer was trained for ten epochs for the first stage and nine epochs for the second stage on the Argoverse 1 dataset. It was trained for 50 epochs for both stages on the nuScenes dataset. The training time on 8xRTX3090 cards with each stage was about 8 hours. The source code of LAformer is available at https://github.com/mengmengliu1998/LAformer.

B. More details about the experimental results

The implementation of DenseTNT and HiVT. To ensure a fair comparison with DenseTNT and HiVT, we re-implemented them using their publicly available code. Since the trained model of DenseTNT is not provided, we trained the offline DenseTNT (w/ 100ms optimization) from scratch with the default settings [1]. The retrained DenseTNT achieves similar results to those reported in their paper. This retrained model is used to generate qualitative results for comparison. We use the publicly available trained model of HiVT [2] to generate qualitative results for comparison.

Figure 2 presents additional scenarios at different intersections in Argoverse 1. It can be observed that all the models can generate at least one mode of prediction that is close to the corresponding ground truth trajectories with respect to the moving directions. In general, LAformer achieves more accurate predictions with respect to the final steps. Compared to DenseTNT, LAformer generates more scene compliant predictions in scenarios ② and ⑤. It can also be observed in ② and ④ that the goal-based model DenseTNT may predict trajectories that are not smoothly connected to the selected goals by optimization. This is because the selected goals and the predicted complete trajectories are not generated sequentially simultaneously. Compared to HiVT, LAformer generates more diverse predictions in the lateral directions in ③ and ⑤, where the target vehicle has the potential to make a left or right turn while entering the respective intersections. However, HiVT only predicts driving straight forward. In ②, LAformer uses the temporally dense lane-aware module to select only the top aligned lane segments for scene-compliant trajectory prediction, while some of the modes predicted by HiVT are not feasible with respect to the lane boundaries.

C. Limitation

Failed cases. Figure 3 presents the failed scenarios in Argoverse 1. For example, in the leftmost scenario, LAformer predicts driving through or turning right, while the vehicle makes a left turn. A similar situation can be seen in the second leftmost scenario when the vehicle starts to make a slight left turn. In these turning scenarios, the observations do not provide sufficient cues (only with the observation of forward driving), and it is challenging to correctly predict the target vehicle's driving intention when it makes a

[1] https://github.com/Tsinghua-MARS-Lab/DenseTNT
[2] https://github.com/ZikangZhou/HiVT

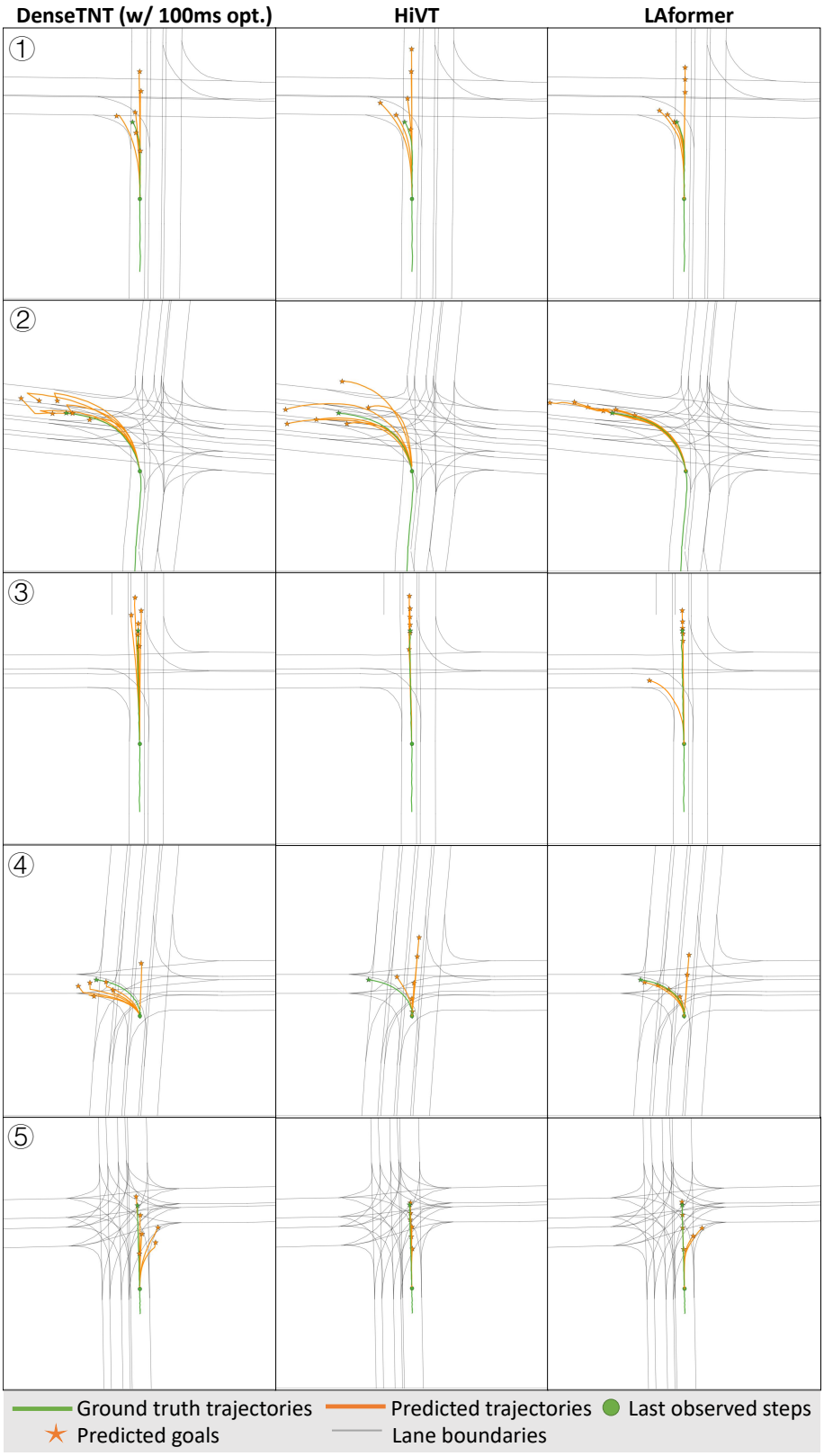

Figure 2. Qualitative comparison of different models in complex scenarios on the Argoverse 1 [2] validation set. Each row represents a unique scenario at an intersection and each column from left to right represents the results predicted by DenseTNT [4], HiVT [11], and LAformer, respectively. This figure is better visualized in zoomed-in and color mode.

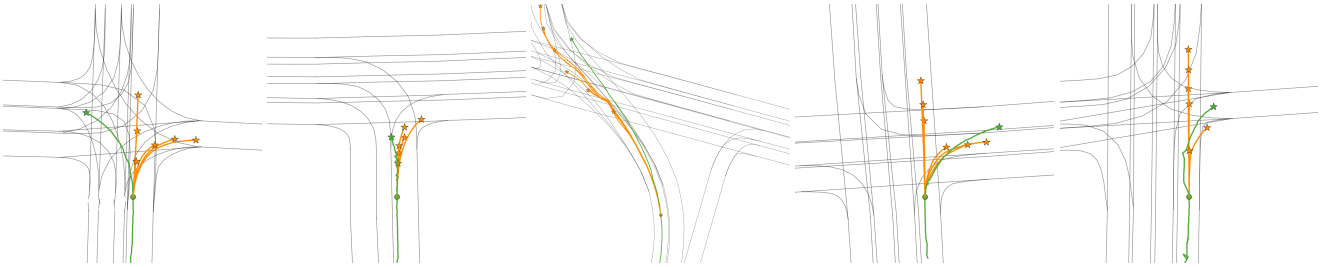

Figure 3. Failed prediction results on Argoverse 1 [2] validation set.

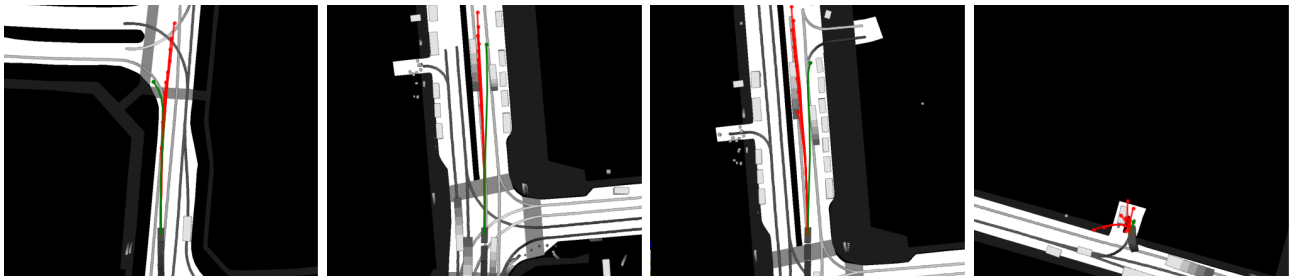

Figure 4. Failed prediction results on the nuScenes [1] validation set. Predicted trajectories are presented in red color and the corresponding ground truth trajectories are presented in green color.

turn at a later time point. In the middle scenario, although LAformer predicts the correct driving intention of the vehicle, its predictions do not well overlap with the vehicle's driving speed and heading. Predictions with large distance errors can be found in the two right scenarios as well. In these scenarios, the target vehicle makes a lane change at a later time point, leading to increased difficulties in capturing the vehicle's speed profile. In these challenging cases, additional information cues such as collaborative perception and vehicle-to-infrastructure communication may be necessary [10] to further facilitate trajectory prediction for autonomous driving.

Figure 4 presents the failed scenarios in nuScenes. Similar to the scenarios in Argverse 1, we can see that LAformer has difficulties predicting the target vehicle's driving intention when it makes a left turn in the rightmost scenario and makes a lane change in the two middle scenarios. This is because the vehicle was observed to be driving along on the road, and it made the change at a much later time point. LAformer misses the change cues and only predicts that the vehicle drives with the same moving pattern as it was observed. Interestingly, we also found that in the rightmost scenario, LAformer predicts some of the trajectories that are not compliant with the scene constraints. In this scenario, the vehicle is moving slowly in a dead-end. There is no further lane information available in the vehicle's driving direction. Hence, LAformer cannot make all scene-compliant predictions.

In addition, similar to [3, 4, 6, 7, 9], LAformer is an agent-centric approach that produces independent predictions for the target agent. Event though the inference speed of LAformer is decent, c.a. 10Hz for 12 agents, it is slower than the scene-centric approach HiVT (more information about the computational cost is given in the main paper). The lane selection module aims to let the model focus on the most potential lane segments to better align motion and lane information. To estimate the alignment, the module needs to compute the binary classification score at each time step for each potential lane, which is not optimal for saving memories and increasing speed.

References

[1] 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. 4

[2] 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. 3, 4

[3] 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, 4

[4] 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, 4

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 2

[6] 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. 4

[7] 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. 4

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 1

[9] 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, 4

[10] Runsheng Xu, Hao Xiang, Xu Han, Xin Xia, Zonglin Meng, Chia-Ju Chen, Camila Correa-Jullian, and Jiaqi Ma. The opencda open-source ecosystem for cooperative driving automation research. *IEEE Transactions on Intelligent Vehicles*, 2023. 4

[11] 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