Bridging Past and Future: End-to-End Autonomous Driving with Historical Prediction and Planning

Supplementary Material

1. Methodology

1.1. Model details

The perception component of our model follows a sparse paradigm [16, 17, 21]. For detection, after obtaining the initialized object queries Q_{obj} and multi-view visual features $\mathcal{F}$, several decoder layers are applied. These layers include attention mechanisms across object queries, deformable aggregation with visual features, and a feedforward network [16]. The Historical Mot2Det Fusion Module, designed by us, follows the above modules to refine the object queries and detection outputs using historical prediction. For online mapping, the structure is similar to that used in detection [21]. For multi-head attention, Flash Attention [3] is adopted to save GPU memory.

1.2. Loss function

As stated in the End-to-End Learning section of the main paper, the loss function for each task is divided into regression and classification components. The losses are defined as follows:

$$\begin{aligned}
\mathcal{L}_{det} &= \lambda_{det_reg}\mathcal{L}_{det_reg} + \lambda_{det_cls}\mathcal{L}_{det_cls}, \\
\mathcal{L}_{map} &= \lambda_{map_reg}\mathcal{L}_{map_reg} + \lambda_{map_cls}\mathcal{L}_{map_cls}, \\
\mathcal{L}_{mot} &= \lambda_{mot_reg}\mathcal{L}_{mot_reg} + \lambda_{mot_cls}\mathcal{L}_{mot_cls}, \\
\mathcal{L}_{plan} &= \lambda_{plan_reg}\mathcal{L}_{plan_reg} + \lambda_{plan_cls}\mathcal{L}_{plan_cls}, \\
\mathcal{L}_{total} &= \mathcal{L}_{det} + \mathcal{L}_{map} + \mathcal{L}_{mot} + \mathcal{L}_{plan}.
\end{aligned} \tag{1}$$

The loss weights are set as follows: $\lambda_{det\_reg} = 0.25$, $\lambda_{det\_cls} = 2.0$, $\lambda_{map\_reg} = 10.0$, $\lambda_{map\_cls} = 1.0$, $\lambda_{mot\_reg} = 0.05$, $\lambda_{mot\_cls} = 0.1$, $\lambda_{plan\_reg} = 1.0$, $\lambda_{plan\_cls} = 0.5$.

1.3. Notations

As shown in Table 1, we provide a lookup table for notations used in the paper.

2. Experiments

2.1. Evaluation metrics

Open-loop evaluation. We provide evaluation metrics for perception, prediction, and planning tasks. The detection and tracking evaluation adheres to standard protocols [2]. For detection, we use mean Average Precision (**mAP**) and nuScenes Detection Score (**NDS**). For tracking, Average Multi-object Tracking Accuracy (**AMOTA**), Average Multi-object Tracking Precision (**AMOTP**), and Identity Switches (**IDS**). The online mapping [21] and motion

Notation	Description
N_{a}	the number of surrounding agents
M_{mot}	the number of prediction modes
C	the feature channels
T_{mot}	the number of future time steps for prediction
M_{plan}	the number of planning modes
T_{plan}	the number of future time steps for planning
K	the number of historical motion planning frames stored in the memory queue
N_{img}	the number of camera views
$\mathcal{F}$	multi-view visual features
Q_{obj}	object queries
B_{obj}	object anchor boxes
Q_{mot}	motion queries
Q_{plan}	planning queries
Q_{m2d}	historical motion queries used in the Historical Mot2Det Fusion Module
T_{m2m}	the number of time steps that interact with historical motion queries
Q_{m2m}	historical motion queries used in the History-Enhanced Motion Prediction Module
T_{p2p}	the number of time steps that interact with historical planning queries
Q_{p2p}	historical planning queries used in the History-Enhanced Planning Module
Q^{*}_{mot}	selected motion queries used in the Step-Level Mot2Plan Interaction Module

Table 1. Notations used in the paper.

prediction [7, 21] evaluations are consistent with previous works. For online mapping, we use the Average Precision (**AP**) for three map classes: lane divider, pedestrian crossing, and road boundary. The mean Average Precision (**mAP**) is then calculated by averaging the AP across all classes. For motion prediction, we use the minimum Average Displacement Error (**ADE**), minimum Final Displacement Error (**FDE**), Miss Rate (**MR**), and End-to-End Prediction Accuracy (**EPA**) as proposed in ViP3D [5]. For planning, we use the L2 Displacement Error metric, as used in VAD [12], and the Collision Rate, as defined in [14, 21]. The Collision Rate addresses two issues in the previous benchmark [7, 12]: false collisions in certain cases and the exclusion of the ego vehicle's heading.

Closed-loop evaluation on NeuroNCAP. Following the official definition [19], a NeuroNCAP score is computed for each scenario. A full score is awarded only if a collision is completely avoided, while partial scores are granted for successfully reducing impact velocity. Inspired by the 5-star Euro NCAP rating system [4], the NeuroNCAP score is calculated as:

$$\text{NNS} = \begin{cases} 5.0 & \text{if no collision,} \\ 4.0 \cdot \max(0, 1 - v_i/v_r) & \text{otherwise.} \end{cases} \tag{2}$$

where v_i is the impact speed as the magnitude of relative velocity between ego-vehicle and colliding actor, and v_r is the reference impact speed that would occur if no action is performed. In other words, the score corresponds to a 5-star rating if collision is entirely avoided, and otherwise the rating is linearly decreased from four to zero stars at (or exceeding) the reference impact speed.

2.2. Implementation details

As stated in the Implementation Details section of the main paper, training is conducted in two stages. The first stage focuses on the perception task with a batch size of 8 for 100 epochs, while the second stage focuses on end-to-end training with a batch size of 4 for 15 epochs. The total training time is approximately 1.5 days. For the model settings, the number of object queries and map queries is set to 900 and 100, respectively. The feature dimension C is 256. The backbone, ResNet101, uses pre-trained weights from the nuImage dataset.

2.3. Online mapping results

The online mapping results on the nuScenes [2] validation dataset, compared to other methods, are shown in Table 2.

2.4. Comparison with other baselines

We compare our model with two other common methods, and the results are shown in Table 3.

2.5. Analysis for moving agents

Following the reviewer's suggestion, we evaluate our model using a more suitable metric proposed by [25], which better reflects the end-to-end nature of the task (see Table 4). As

Method	AP_{ped} ↑	$AP_{divider}$ ↑	$AP_{boundary}$ ↑	mAP ↑
HDMapNet [13]	14.4	21.7	33.0	23.0
VectorMapNet [18]	36.1	47.3	39.3	40.9
MapTR [15]	56.2	59.8	60.1	58.7
VAD† [12]	40.6	51.5	50.6	47.6
SparseDrive [21]	49.9	57.0	58.4	55.1
BridgeAD-S (Ours)	51.8	56.4	57.5	55.2
BridgeAD-B (Ours)	52.0	57.1	57.9	55.7

Table 2. Comparison of online mapping results for state-of-the-art online mapping and end-to-end methods. † indicates evaluation with the official checkpoint.

Method	L2 (m) ↓				Col. Rate (%) ↓			
	$1s$	$2s$	$3s$	Avg.	$1s$	$2s$	$3s$	Avg.
BEVPlanner [14]	0.27	0.54	0.90	0.57	0.04	0.35	1.80	0.73
BEVPlanner* [14]	0.28	0.42	0.68	0.46	0.04	0.37	1.07	0.49
PARA-Drive* [23]	0.25	0.46	0.74	0.48	0.14	0.23	0.39	0.25
BridgeAD	0.29	0.57	0.92	0.59	0.01	0.05	0.22	0.09

Table 3. Comparison with other baselines. "*" denotes use ego status as input.

shown, our model outperforms UniAD and ViP3D on these metrics, which specifically focus on moving agents.

Method	mAP_f ↑	minADE ↓	minFDE ↓	MR ↓
ViP3D [5]	0.034	3.540	5.943	0.432
UniAD [7]	0.117	1.842	3.258	0.228
BridgeAD	0.139	1.733	3.098	0.210

Table 4. Motion forecasting results with more adapted metrics. We use 6 modes by default.

2.6. Safety assessments

Following the reviewer's suggestion, we conduct a safety assessment of our method, including an analysis of its robustness to images, as shown in Table 5. Additionally, we provide an analysis of failure cases and limitations in the supplementary material.

Image Corruption	L2 (m) ↓ Avg.	Col. Rate (%) ↓ Avg.
Only front view	0.68	0.22
Blank	2.76	1.83
Default	0.59	0.09

Table 5. Our model's robustness to images on nuScenes.

2.7. Experiments on the Bench2Drive dataset

We conduct experiments on CARLA v2 simulator using the Bench2Drive benchmark [11], as shown in Table 6. Our method outperforms UniAD and VAD in both open-loop and closed-loop evaluations, showcasing the model's generalization ability.

Method	Open-loop	Closed-loop	
	Avg. L2 ↓	DS ↑	SR (%) ↑
AD-MLP [26]	3.64	18.05	0.00
UniAD [7]	0.73	45.81	16.36
VAD [12]	0.91	42.35	15.00
BridgeAD	0.71	50.06	22.73
TCP* [24]	1.70	40.70	15.00
TCP-ctrl* [24]	-	30.47	7.27
TCP-traj* [24]	1.70	59.90	30.00
ThinkTwice* [10]	0.95	62.44	31.23
DriveAdapter* [9]	1.01	64.22	33.08

Table 6. Experiment on CARLA v2 using the Bench2Drive benchmark. "DS" indicates Driving Score, "SR" indicates Success Rate. "*" denotes expert feature distillation.

2.8. Ablation study

Effects of self-attention in motion prediction. We conduct an ablation study to evaluate the effects of step-level and mode-level self-attention in the motion prediction module, as shown in Table 7, similar to Table 7 in the main paper. Both types of self-attention propagate historical information across prediction steps and modes, enhancing the accuracy of motion prediction.

SLA	MLA	ADE (m) ↓ Car / Ped	FDE (m) ↓ Car / Ped	EPA ↑ Car / Ped
✓		0.65 / 0.71	1.02 / 1.00	0.49 / 0.42
	✓	0.64 / 0.71	1.00 / 1.01	0.48 / 0.42
✓	✓	0.62 / 0.70	0.98 / 0.99	0.50 / 0.44

Table 7. Ablation study on step-level self-attention (SLA) and mode-level self-attention (MLA).

Effects of the number of historical frames. We conduct an ablation study on the number of historical frames K, as shown in Table 8. The results show that $K = 3$ achieves the best balance between efficiency and performance.

2.9. Qualitative results

We present additional qualitative results from both the open-loop and closed-loop evaluations on the nuScenes [2]

HisFrame	Avg. L2 (m) ↓	Avg. Col. Rate (%) ↓
2	0.64	0.13
3	0.59	0.09
4	0.62	0.10

Table 8. Ablation study on the number of historical frames.

dataset. The open-loop evaluation results are shown in Figure 2. The closed-loop evaluation results, obtained using the NeuroNCAP [19] simulator, are shown in Figures 3, 4, and 5. Notably, the red line in the closed-loop evaluation represents the reference trajectory under normal driving conditions, where no safety risk is present.

2.10. Failure cases

We present the failure cases observed in both open-loop and closed-loop evaluations.

The failure cases from the open-loop evaluation are shown in Figure 6. In both the first and second cases, the planned trajectories veer off the road at the curbs (road boundaries). Adding constraints or post-processing techniques to keep the planned trajectories on the road could prevent these failures.

The failure case from the closed-loop evaluation is shown in Figure 7. The planned trajectories steer to avoid the front truck, but insufficient steering and a lack of deceleration still result in a crash. Providing more training data focused on deceleration or applying post-processing techniques to enforce slowing down could prevent this failure.

3. Limitations and future work

The results of closed-loop testing indicate that our model still struggles to handle safety-critical scenarios and relies heavily on complex post-processing. This limitation is a common issue among existing end-to-end methods. Our approach mitigates safety-critical scenarios to some extent by aggregating historical planning information to produce coherent driving actions that avoid collisions. However, this remains insufficient. Exploring effective and efficient solutions, such as training with more data in these situations or integrating the end-to-end pipeline with reinforcement learning or rule-based planning, is a promising direction for future research.

4. Discussion

4.1. Further explanation about our BridgeAD

To better illustrate our method, we provide a further explanation of our key idea. As shown in Figure 1 (a), unlike previous methods [7, 12, 21], we represent motion and planning queries as multi-step queries. In contrast to previous

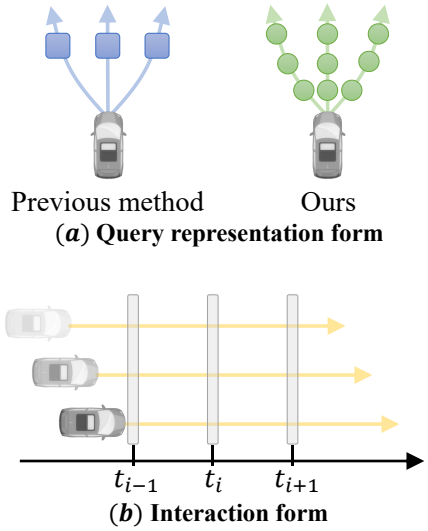

Figure 1. Further explanation about our BridgeAD.

approaches that use a single query to represent an entire trajectory instance, our method utilizes multiple queries for a single trajectory. For example, in the planning task on the nuScenes dataset, where a 3-second future trajectory is planned at 2 Hz, six queries are used to represent one trajectory instance.

Regarding the interaction mechanism in our method, as shown in Figure 1 (b), queries are grouped based on time steps, and those corresponding to the same time step interact through our designed modules. This approach is applied to both motion queries for surrounding agents and planning queries for the ego agent.

4.2. Discussion about belief states

Belief states represent an agent's probabilistic estimation of the true state of the environment, given past observations and actions. They are commonly used in decision-making under uncertainty, where the full state is not directly observable. By maintaining and updating a belief state, an agent can make more informed and robust decisions in dynamic or partially observable environments. Some methods [1, 6, 8] explore its potential for planning and decision-making in autonomous driving. Huang *et al.* [8] proposes a neural memory-based belief update model for online behavior prediction and a macro-action-based MCTS planner guided by deep Q-learning. By leveraging long-term multi-modal trajectory predictions and optimizing decision-making under uncertainty, the approach enhances both efficiency and performance in autonomous driving scenarios.

Our BridgeAD can essentially be seen as encoding belief states. By leveraging historical prediction and planning, it incorporates belief states into perception, prediction, and planning, enhancing end-to-end autonomous driving perfor-

mance.

4.3. Discussion about historical predictions

In the motion prediction task, recent works have explored leveraging historical predictions to improve performance. HPNet [22] utilizes historical predictions to achieve more stable and accurate motion forecasts, while RealMotion [20] operates in a streaming fashion to enhance motion prediction. In contrast, our BridgeAD incorporates both historical prediction and planning to optimize the entire pipeline of end-to-end autonomous driving.

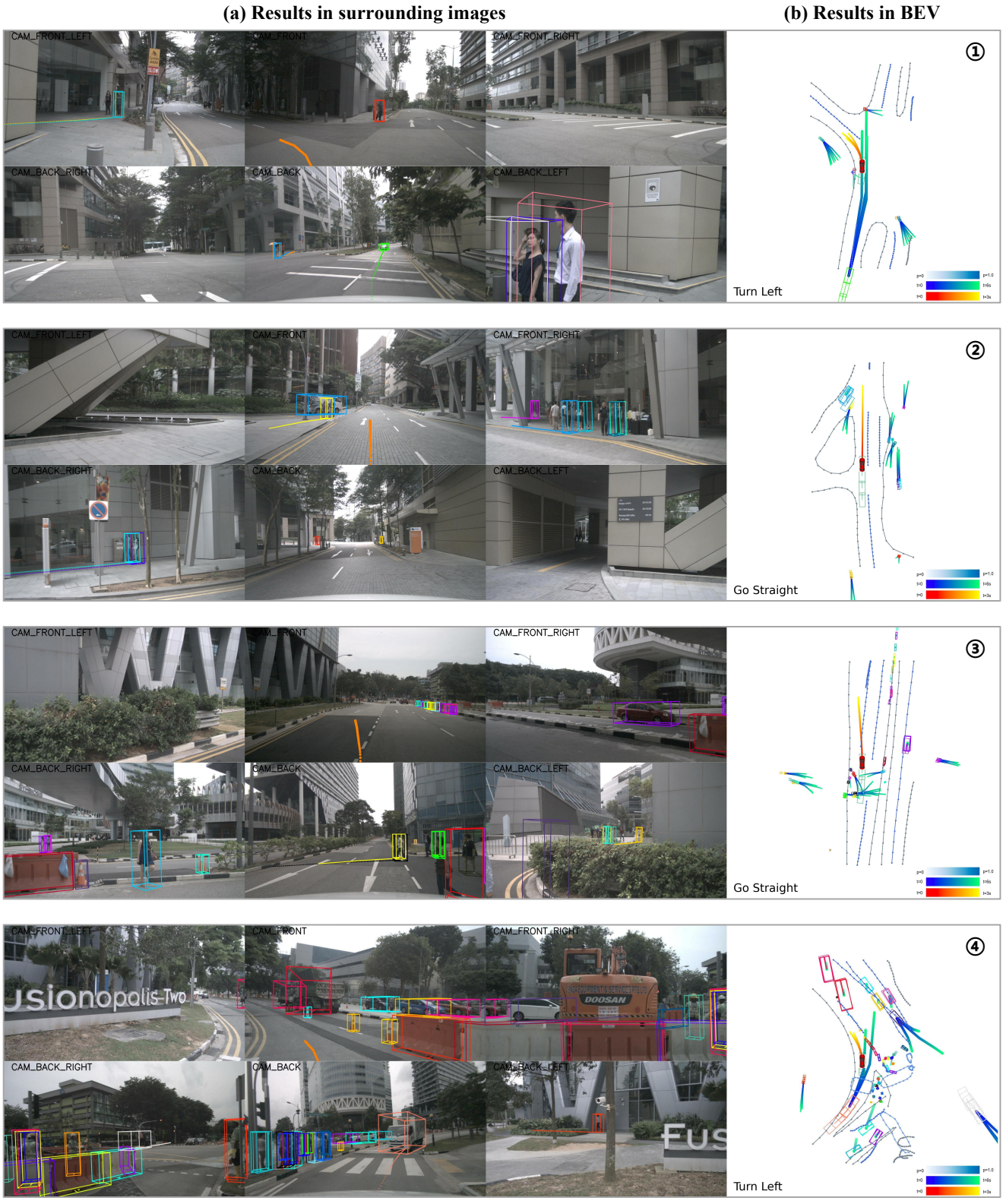

Figure 2. Qualitative results in the **open-loop** evaluation.

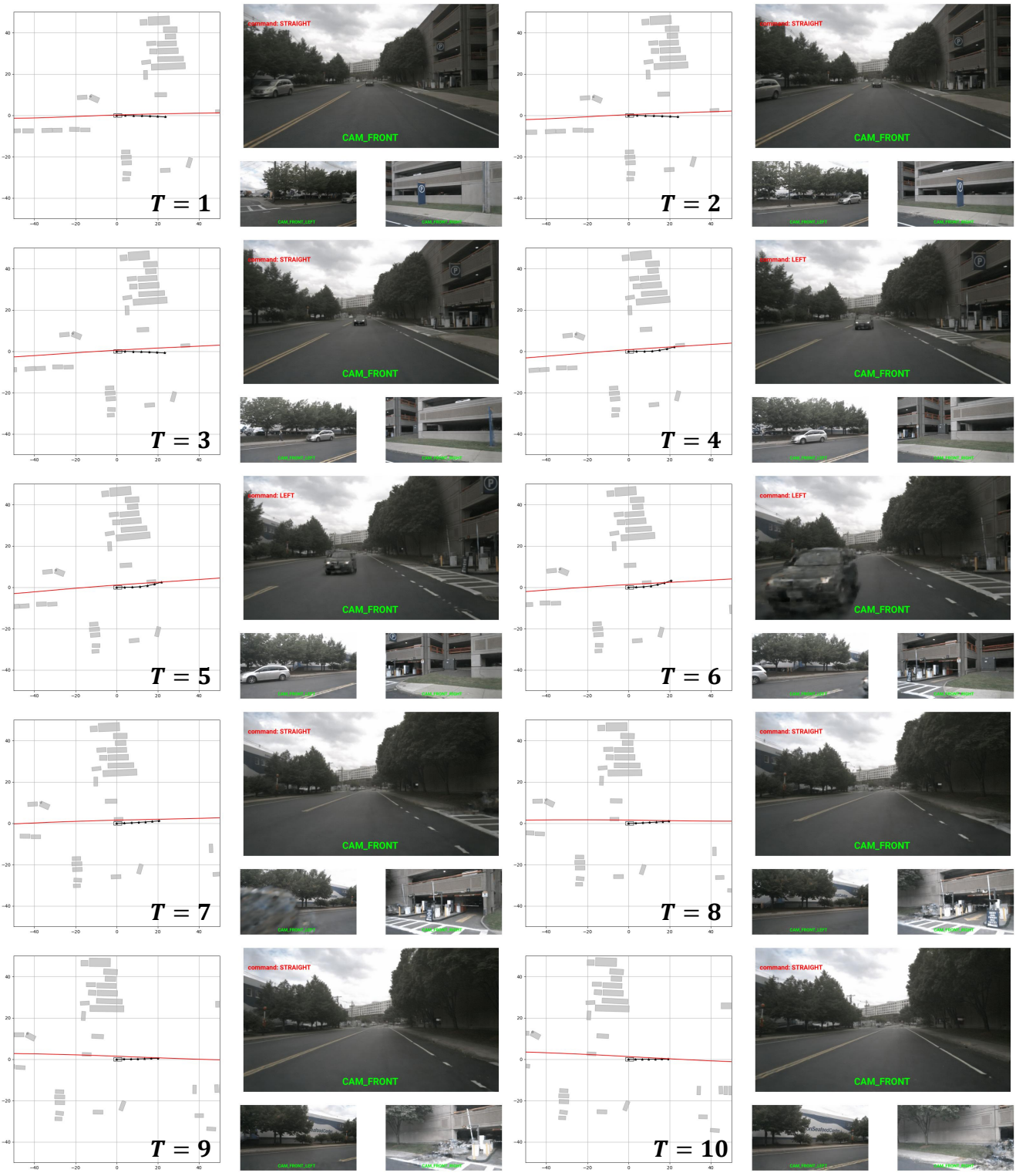

Figure 3. Qualitative result 1 in the **closed-loop** evaluation.

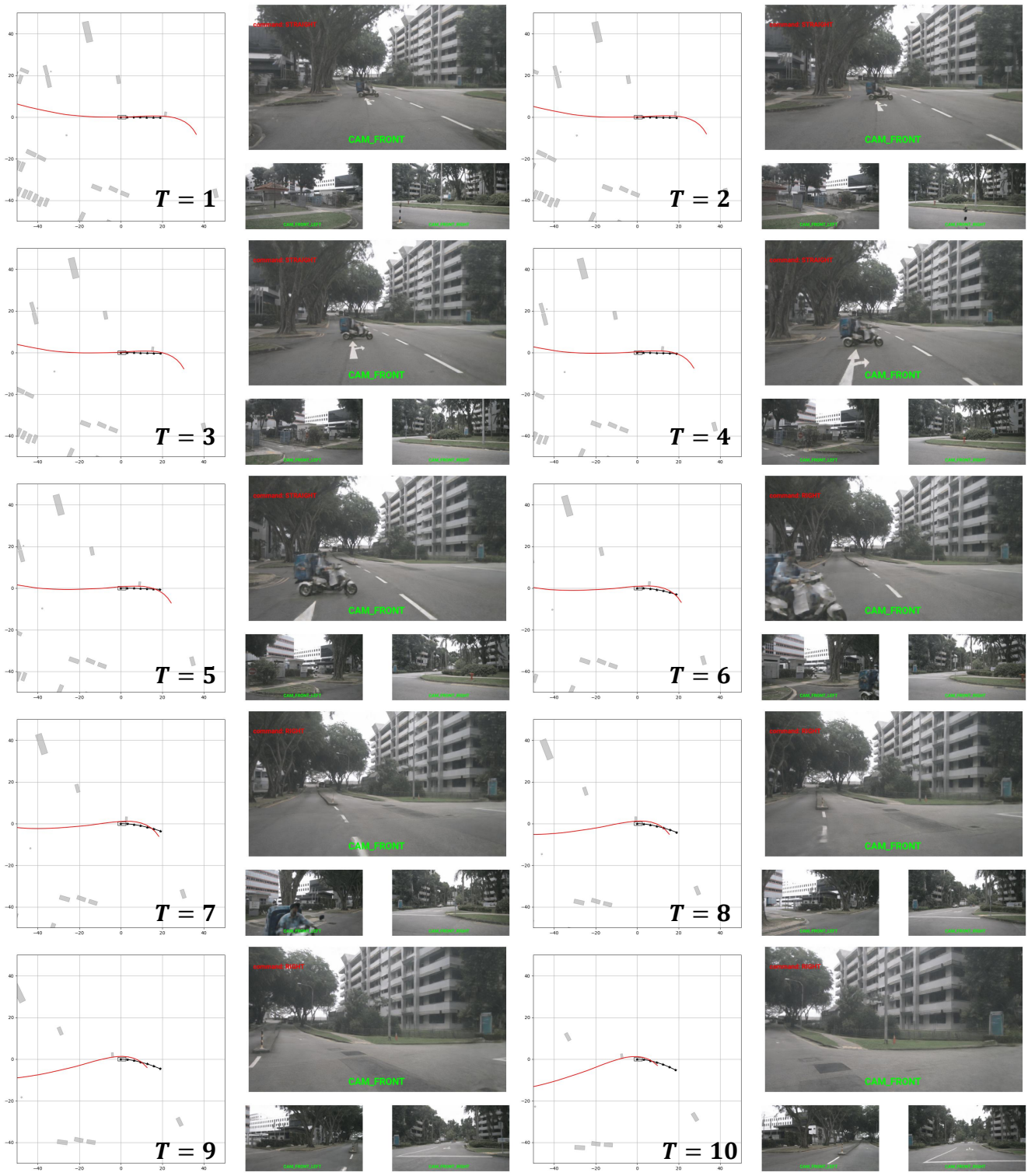

Figure 4. Qualitative result 2 in the **closed-loop** evaluation.

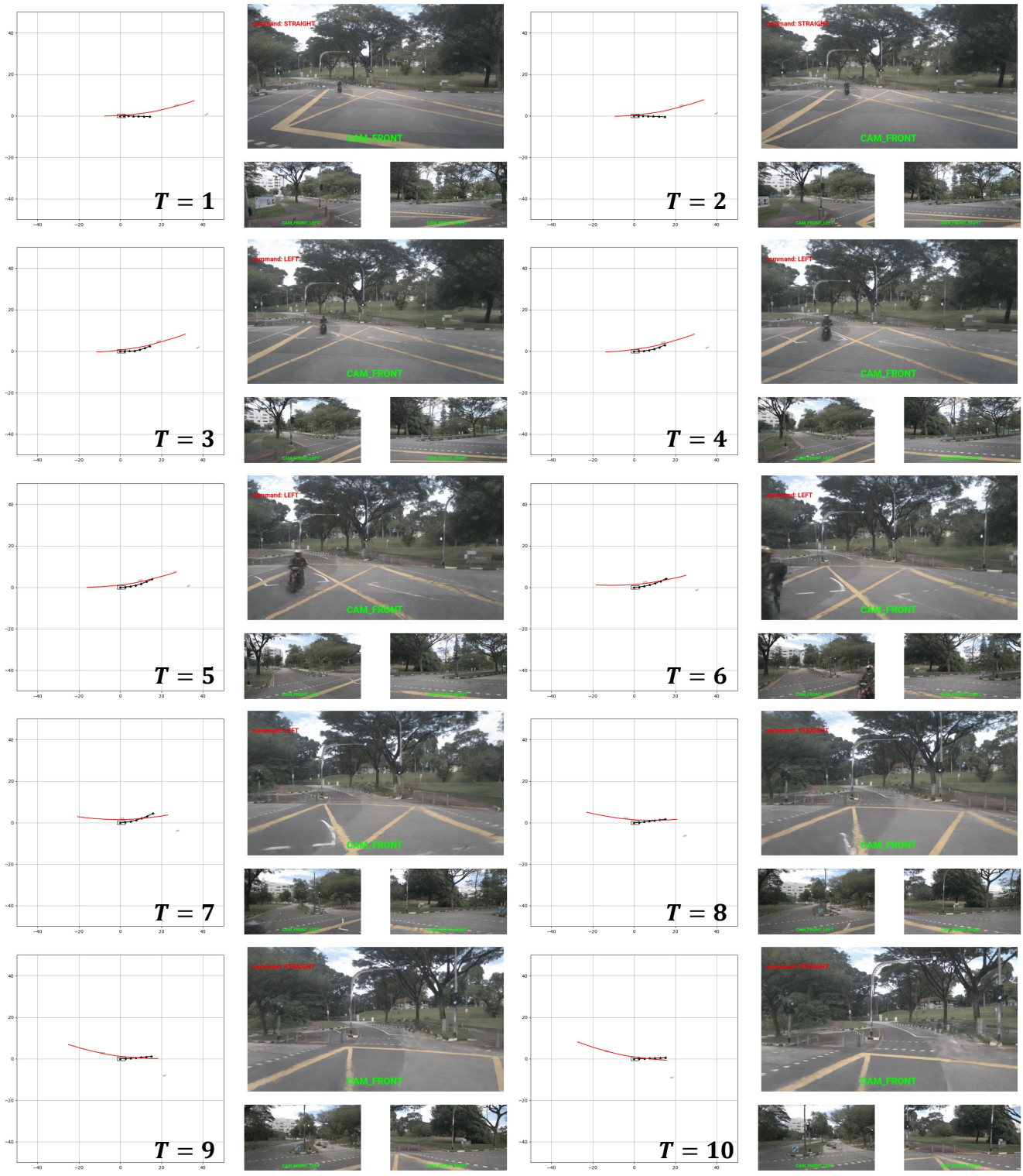

Figure 5. Qualitative result 3 in the **closed-loop** evaluation.

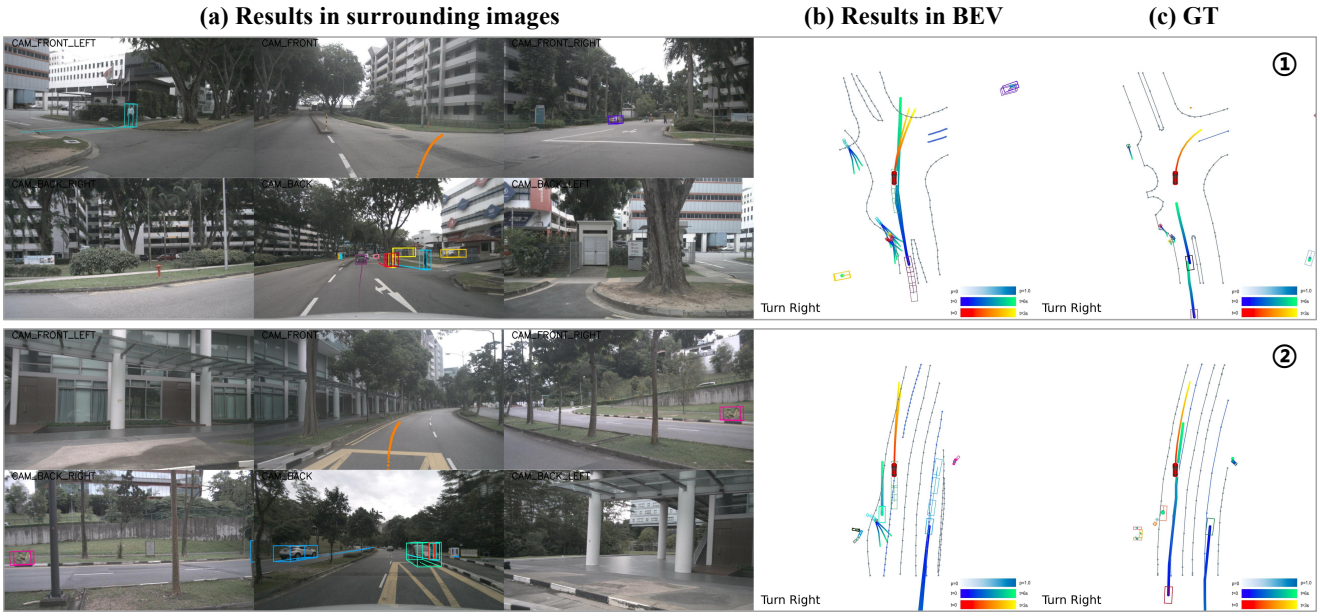

Figure 6. Failure cases in the **open-loop** evaluation.

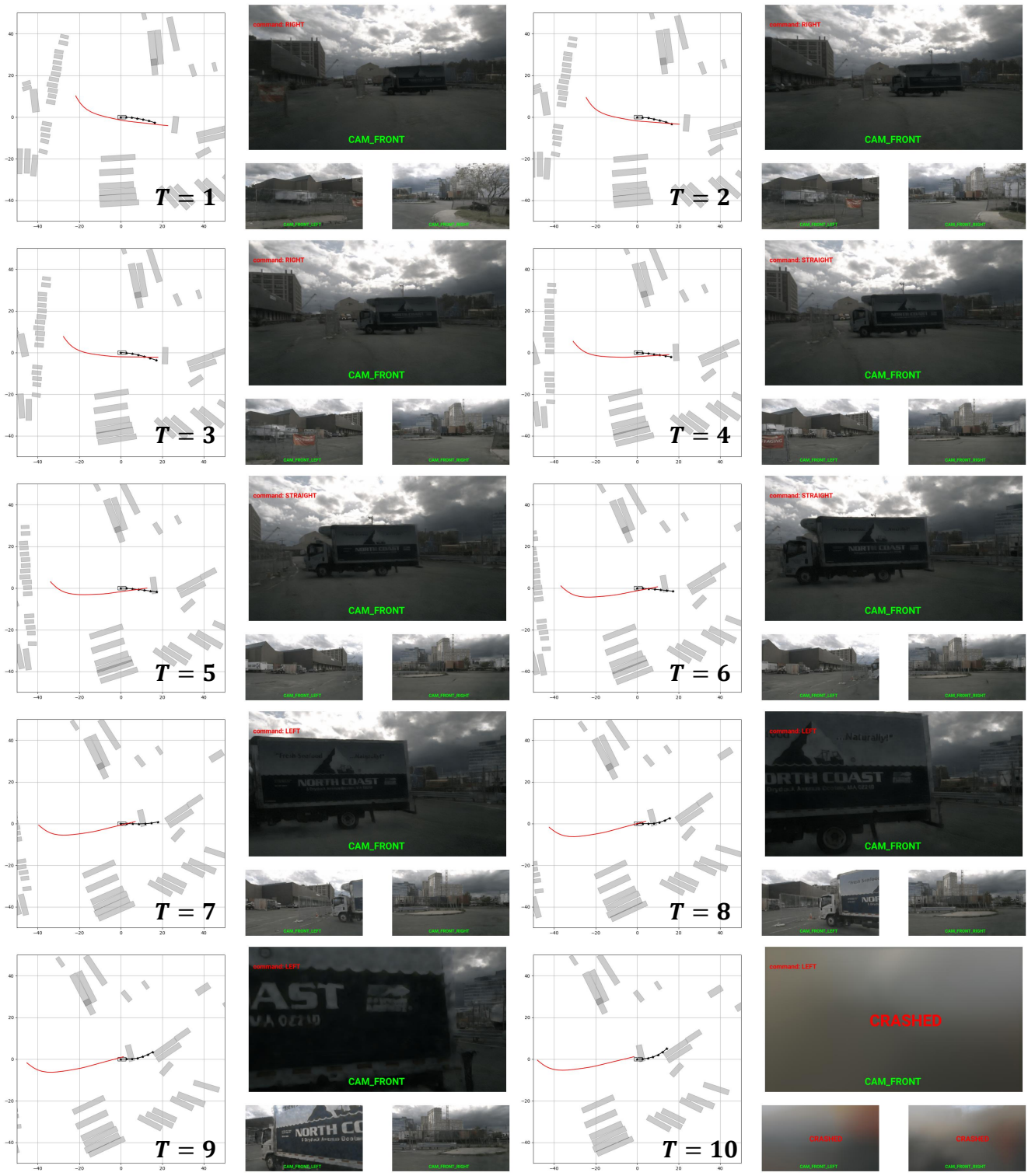

Figure 7. Failure case in the **closed-loop** evaluation.

References

[1] Maxime Bouton, Akansel Cosgun, and Mykel J Kochenderfer. Belief state planning for autonomously navigating urban intersections. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017. 4

[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 *CVPR*, 2020. 1, 2, 3

[3] Tri Dao, Daniel Y Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *NeurIPS*, 2022. 1

[4] EuroNCAP. Assessment protocol – safety assist - collision avoidance, 2023. https://www.euroncap.com/media/79866/euro-ncap-assessment-protocol-sa-collision-avoidance-v104.pdf. 2

[5] Junru Gu, Chenxu Hu, Tianyuan Zhang, Xuanyao Chen, Yilun Wang, Yue Wang, and Hang Zhao. Vip3d: End-to-end visual trajectory prediction via 3d agent queries. In *CVPR*, 2023. 2, 3

[6] Ziqing Gu, Yujie Yang, Jingliang Duan, Shengbo Eben Li, Jianyu Chen, Wenhan Cao, and Sifa Zheng. Belief state separated reinforcement learning for autonomous vehicle decision making under uncertainty. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021. 4

[7] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *CVPR*, 2023. 2, 3, 4

[8] Zhiyu Huang, Chen Tang, Chen Lv, Masayoshi Tomizuka, and Wei Zhan. Learning online belief prediction for efficient pomdp planning in autonomous driving. *IEEE RA-L*, 2024. 4

[9] Xiaosong Jia, Yulu Gao, Li Chen, Junchi Yan, Patrick Langechuan Liu, and Hongyang Li. Driveadapter: Breaking the coupling barrier of perception and planning in end-to-end autonomous driving. In *ICCV*, 2023. 3

[10] Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In *CVPR*, 2023. 3

[11] Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. In *NeurIPS*, 2024. 3

[12] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *ICCV*, 2023. 2, 3, 4

[13] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *ICRA*, 2022. 3

[14] Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahan Li, Jan Kautz, Tong Lu, and Jose M Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? In *CVPR*, 2024. 2, 3

[15] Bencheng Liao, Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Chang Huang. Maptr: Structured modeling and learning for online vectorized hd map construction. In *ICLR*, 2023. 3

[16] Xuewu Lin, Tianwei Lin, Zixiang Pei, Lichao Huang, and Zhizhong Su. Sparse4d v2: Recurrent temporal fusion with sparse model. *arXiv preprint*, 2023. 1

[17] Xuewu Lin, Zixiang Pei, Tianwei Lin, Lichao Huang, and Zhizhong Su. Sparse4d v3: Advancing end-to-end 3d detection and tracking. *arXiv preprint*, 2023. 1

[18] Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. Vectormapnet: End-to-end vectorized hd map learning. In *ICML*, 2023. 3

[19] William Ljungbergh, Adam Tonderski, Joakim Johnander, Holger Caesar, Kalle Åström, Michael Felsberg, and Christoffer Petersson. Neuroncap: Photorealistic closed-loop safety testing for autonomous driving. In *ECCV*, 2024. 2, 4

[20] Nan Song, Bozhou Zhang, Xiatian Zhu, and Li Zhang. Motion forecasting in continuous driving. In *NeurIPS*, 2024. 5

[21] Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. *arXiv preprint*, 2024. 1, 2, 3, 4

[22] Xiaolong Tang, Meina Kan, Shiguang Shan, Zhilong Ji, Jinfeng Bai, and Xilin Chen. Hpnet: Dynamic trajectory forecasting with historical prediction attention. In *CVPR*, 2024. 5

[23] Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. In *CVPR*, 2024. 3

[24] Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. In *NeurIPS*, 2022. 3

[25] Yihong Xu, Loïck Chambon, Éloi Zablocki, Mickaël Chen, Alexandre Alahi, Matthieu Cord, and Patrick Pérez. Towards motion forecasting with real-world perception inputs: Are end-to-end approaches competitive? In *ICRA*, 2024. 2

[26] Jiang-Tian Zhai, Ze Feng, Jinhao Du, Yongqiang Mao, Jiang-Jiang Liu, Zichang Tan, Yifu Zhang, Xiaoqing Ye, and Jingdong Wang. Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes. *arXiv preprint*, 2023. 3