Is Ego Status All You Need for Open-Loop End-to-End Autonomous Driving?

Supplementary Material

A. Implementation Details

ST-P3 uses partially incorrect training and evaluation data. For the common practice, the future GT planning trajectory is generated from the ego locations of the samples in the subsequent 3 seconds. However, since one nuScenes clip is usually a 20s video, which means that the samples at the tail of the video (within 17s-20s) cannot produce a complete future trajectory, normal methods [3, 4] will perform special processing on these special samples by using masks, but ST-P3 [2] did not do this. ST-P3 mistakenly used samples from other scenes while generating GT of these tail samples, so errors occurred during training and testing. Related issue: https://github.com/OpenDriveLab/ST-P3/issues/24.

Ego Status Usage Details For UniAD (ID-1) and VAD-Base (ID-4), in order to exclude ego status from the Bird's Eye View (BEV) generation phase, we set the use_can_bus flag to False. Conversely, for UniAD (ID-3), to incorporate ego status into its planner, we adhered to the methodology used in VAD, which involves concatenating the ego status vector with the query features.

B. Metrics Details.

Collision Rate. While current methods tend to evaluate the collision rates of planned trajectories [1–5, 7], there are issues in both the definition and implementation of this metric in existing approaches. First of all, in open-loop end-to-end autonomous driving, other agents do not provoke a response from the ego car. Instead, they strictly adhere to their predetermined trajectories. Consequently, this leads to a bias in the calculation of collision rates. The second issue arises from the fact that the planning predictions generated by current methods consist solely of a series of trajectory points. As a consequence, in the final collision calculation, the yaw angle of the ego car is not taken into account. Instead, it is assumed to remain unchanged. This assumption leads to erroneous results, particularly in turning scenarios, as shown in Fig. 1.

There are also problems in the current implementation. The current definition of the collision rate of each single sample is:

$$CR(t) = \frac{\sum_{i=0}^{N} \mathbb{I}_i}{N}, N = t/0.5, \tag{1}$$

N represents the number of steps at intervals of t seconds, and $\mathbb{I}_i$ denotes whether the ego car at step i will intersect

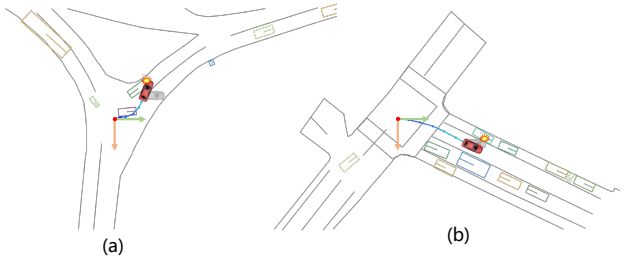

Figure 1. Current methods [2–4] neglect to consider yaw angle variations of the ego vehicle, consistently preserving a 0 yaw angle (depicted by the gray vehicle), thereby resulting in an increased incidence of false negatives (a) and false positives in (b) collision detection. In this paper, we improve collision detection accuracy by estimating the vehicle's yaw angle from variations in its trajectory (depicted by the red vehicle).

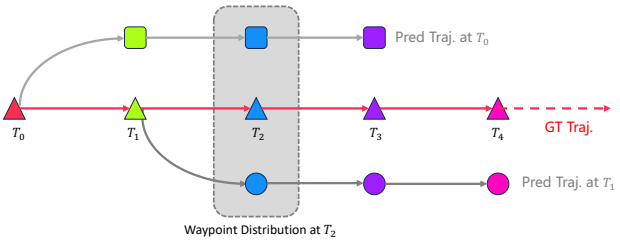

Figure 2. In open-loop autonomous driving approaches, the future trajectory is forecasted from the starting location of the ego vehicle. Within the imitation learning paradigm, the predicted trajectory ideally should closely align with the actual ground truth trajectory. Furthermore, trajectories forecasted at successive time steps should maintain consistency, thereby guaranteeing the continuity and smoothness of the driving strategy. Consequently, the predicted trajectories depicted in red boxes of **??** not only deviate from the ground truth trajectory but also demonstrate significant divergence at various timestamps.

with other agents. In this paper, we modify the definition of collision to

$$CR(t) = (\sum_{i=0}^{N} \mathbb{I}_i) > 0, N = t/0.5. \tag{2}$$

For previous implementation, they assumed that collisions at each moment were mutually independent, which does not align with real-world scenarios. Our modified version yields values that more precisely indicate the collision rate occurring along the predicted trajectory.

Trajectory Smoothness We also assessed the stability of the model's predicted trajectories. Given that the model predicts the trajectory for the next three seconds at each moment, it means that for every absolute moment in time t,

Method	L2 (m) ↓				σ_{wd}↓			
	1s	2s	3s	Avg.	1s	2s	3s	Avg.
Baseline	0.30	0.52	0.85	0.56	0.03	0.19	0.70	0.31

Table 1. The smoothness σ_{wd} of predicted trajectories.

the model predicted multiple waypoints at time t from various preceding times. We see these different waypoints as a distribution. In non-extreme conditions, this distribution should be as concentrated as possible to ensure smoothness in the driving process, as shown in Fig. 2. To quantitatively analyze this distribution, we calculated the squared deviation distance of these distribution points, as shown in Tab. 1. We found that this smoothness metric does not convey more information than the L2 metric, and we believe that this requires more exploration to verify the rationality of an metric.

Valid Samples We discussed above that for the tail samples without complete GT trajectories. The normal method will use the mask for special identification. During evaluation, the previous methods have different processing methods. One is that if a sample does not have a complete GT future trajectory, it will not be considered during evaluation. The second strategy only considers the valid part of the GT future trajectory if the length of the GT trajectory is less than 3s. In this paper, we follow the first strategy. For 6019 samples of nuScenes val split, the number of final valid samples is 5119 (85% of all samples). The reason why we didn't reproduce the correct version of ST-P3 is that the definition of valid samples of ST-P3 is different from others. Valid samples of ST-P3 must use sufficient historical data, so ST-P3 does not predict trajectories for the first few samples of each clip. Even if we reproduce the correct ST-P3, we cannot compare it with other methods for a fair comparison.

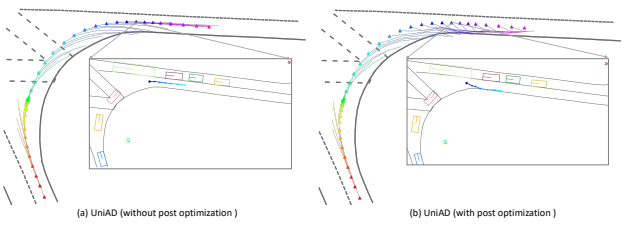

Figure 3. In order to avoid collisions as much as possible, post-optimization is introduced in UniAD [3] to keep the predicted trajectory away from other vehicles. However, during actual traffic driving, other factors need to be considered, such as road conditions. As shown in figure (b), UniAD rushed to the road boundary in order to avoid the possible danger caused by the opposite lane and actually caused another accident.

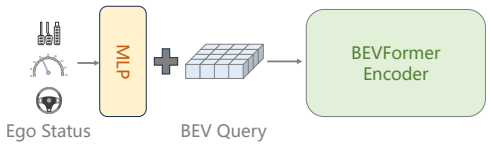

Figure 4. BEVFormer incorporates ego status information during the initialization of BEV queries, a nuance not addressed by current end-to-end autonomous driving approaches [3, 4, 7].

C. Neglected Ego Status in Perception Stage.

In fact, a crucial question is whether the method really completely eliminates the influence of ego status. The pipeline of the existing open-loop end-to-end autonomous driving methods [2–4, 7] basically follows the **??** (b). Given that ego status exerts a substantial influence on the planning results, these methods actually have clear explanations on whether to introduce ego status in the planner. However, methods [3, 4] ignored the impact of introducing ego status in the early perception stage on the planning results. In detail, both UniAD [3] and VAD [4] utilize BEVFormer [6] as their BEV generation module. For BEVFormer, it involves projecting the ego status onto the hidden features and incorporating it into the BEV query, as shown in Fig. 4. This trick exerts a marginal effect on perception performance, as shown in Tab. 2. However, when BEVFormer is integrated into an end-to-end pipeline, the introduction of ego status at this initial stage can wield a substantial influence on the ultimate planning performance. As shown in **??**, upon the removal of the ego status input during the BEV stage, the planning performance of both VAD and UniAD exhibits a marked decline. It is important to clarify that our position is not opposed to the use of ego status; rather, we argue that within the context of current datasets and evaluation metrics, the integration of ego status can significantly impact, and even determine, the planning results. Unfortunately, the incorporation of ego status within the perception module is often overlooked in the existing end-to-end autonomous driving methods. Therefore, it is essential in comparative analyses of different methodologies to carefully examine the role and impact of ego status to ensure fairness and consistency in the evaluations.

Methods	Ego Status	mAP↑	NDS↑
BEVFormer	✓	41.6	51.7
BEVFormer	✗	41.3	51.5

Table 2. The integration of ego status within BEVFormer exerts only a marginal effect on the perception performance.

D. Post Optimization of UniAD.

As demonstrated in Fig. 3, we observe that while UniAD utilizes collision optimization, the resulting optimized tra-

jectory tends to intersect with the road boundary at a higher rate. This occurs because the collision optimizer overlooks map priors. In its effort to avoid collisions, the optimizer disregards other factors that could pose safety risks. However, if the optimizer were to consider all relevant factors, it would more closely resemble traditional Planning and Navigation Control (PNC) systems, contradicting the fundamental motivation of end-to-end autonomous driving.

E. Dropping Cameras

Referencing Table 2 in our main paper, it is observed that when VAD incorporates ego status as an input, the removal of camera input does not markedly impair its performance. A parallel experiment was conducted with VAD [4] devoid of ego status. We also provide visualization results in Fig. 5. As delineated in Tab. 3, excluding camera inputs in VAD without ego status leads to a significant decline in performance, particularly regarding L2 distance and collision rate metrics. Intriguingly, this decrease was not mirrored in the Intersection rate with road boundary metric.

In fact, when the model operates without using ego status and with the camera input removed, it relies solely on driving commands to guide its future direction. In this scenario, the fact that the intersection rate with road boundaries does not increase is counter-intuitive. This counter-intuitive phenomenon has driven us to delve deeper into the evaluation process. As shown in Tab. 4, we compiled statistics on the intersection metric under different driving commands. The Intersection-LR metrics show that the model, when operating without camera input, significantly increases the probability of interacting with boundaries in turning scenarios. This is also consistent with our observations from visualization. The real reason lies in the fact that in straight-driving scenarios (87% of all evaluation samples), removing the camera input leads the model to adopt a relatively conservative straight-driving strategy, making it less likely to intersect with road boundaries (indicated by Intersection-ST). Since straight-driving scenarios constitute a large proportion of the *val* split, this results in the model achieving better overall average results when operating without camera input.

Failure Cases Although the majority of scenarios in the nuScenes dataset are relatively straightforward, it does include certain challenging scenes, notably those involving continuous cornering. As shown in Fig. 6, we can observe that methods with various settings all yielded suboptimal predicted trajectories when navigating high-curvature bends. For challenging scenarios like cornering, where the system must continuously make evolving decisions, evaluating open-loop autonomous driving systems poses a significant challenge. One limitation of open-loop methods is that they do not suffer from cumulative errors. In detail, in the case of an extremely erroneous trajectory predicted at a given timestep, the trajectory starting point for the next timestep is still based on the GT trajectory. The metric we utilize, CCR, is adept at identifying low-quality trajectories. However, an appropriate metric that can effectively highlight high-quality trajectories remains an intriguing direction for further exploration.

Method	Img Corruption	Ego Status	L2 (m) ↓ 1s	2s	3s	Avg.	Collision (%) ↓ 1s	2s	3s	Avg.	Intersetion (%) ↓ 1s	2s	3s	Avg.	Det. (NDS)	Map (mAP)
VAD-Base	-	✓	**0.17**	**0.34**	**0.60**	**0.37**	0.04	**0.27**	**0.67**	**0.33**	**0.21**	**2.13**	**5.06**	**2.47**	**45.5**	47.0
VAD-Base	Blank	✓	0.19	0.41	0.77	0.46	**0.00**	0.40	1.21	0.54	0.35	3.05	7.73	3.71	0.0	0.0
VAD-Base	-	✗	0.69	1.22	1.83	0.06	0.68	2.52	0.84	0.37	1.02	3.44	7.00	3.82	45.1	**53.7**
VAD-Base	Blank	✗	2.59	4.32	6.09	4.33	2.29	7.89	12.7	7.63	1.07	3.73	6.64	3.81	0.0	0.0

Table 3. Omitting camera inputs in the VAD model, when it does not utilize ego status, results in a marked reduction in performance, as evidenced by the metrics for L2 distance and collision rate.

Method	Img Corruption	Ego Status	CCR ↓ 1s	2s	3s	Avg.	CCR-ST (%) ↓ 1s	2s	3s	Avg.	CCR-LR (%) ↓ 1s	2s	3s	Avg.	Det. (NDS)	Map (mAP)
VAD-Base	-	✗	1.02	3.44	7.00	3.82	2.49	8.50	16.4	9.13	0.95	2.70	5.50	3.05	45.1	53.7
VAD-Base	Blank	✗	1.07	3.73	6.64	3.81	2.63	18.2	32.1	17.6	0.83	1.51	2.72	1.69	0.0	0.0

Table 4. CCR-ST is the CCR rate with going straight driving commands. CCR-LR is the CCR rate with turning left/right commands.

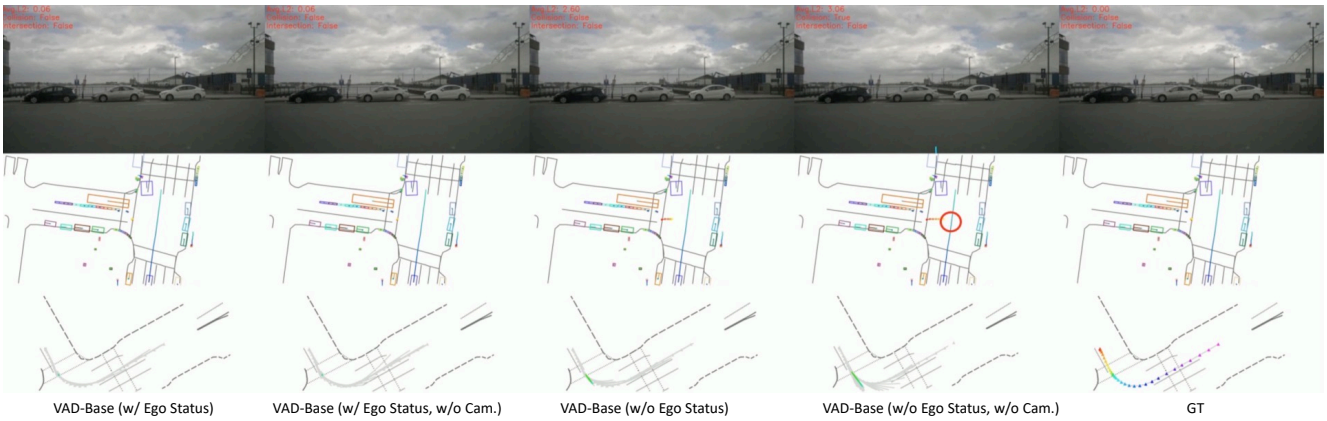

Figure 5. When the model uses ego status as an input, removing the camera does not significantly impact its performance. However, without ego status, omitting camera inputs makes the model more prone to erroneous planning. Red circles indicate potential collisions.

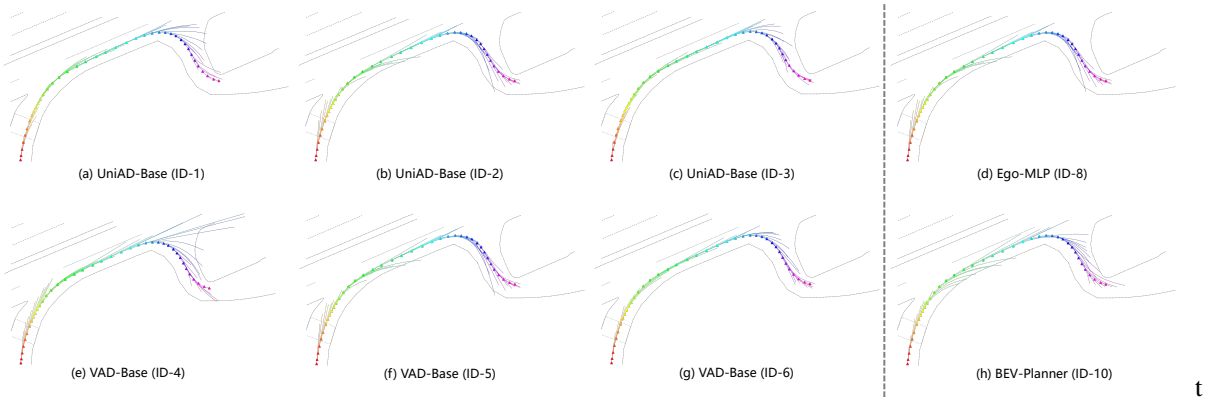

Figure 6. In scenarios that necessitate continuous turning, all methods predict suboptimal trajectories.

References

[1] Peiyun Hu, Aaron Huang, John Dolan, David Held, and Deva Ramanan. Safe local motion planning with self-supervised freespace forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12732–12741, 2021. 1

[2] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *European Conference on Computer Vision*, pages 533–549. Springer, 2022. 1, 2

[3] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17853–17862, 2023. 1, 2

[4] 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. *arXiv preprint arXiv:2303.12077*, 2023. 1, 2, 3

[5] Tarasha Khurana, Peiyun Hu, Achal Dave, Jason Ziglar, David Held, and Deva Ramanan. Differentiable raycasting for self-supervised occupancy forecasting. In *European Conference on Computer Vision*, pages 353–369. Springer, 2022. 1

[6] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. BEVFormer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *arXiv preprint arXiv:2203.17270*, 2022. 2

[7] Tengju Ye, Wei Jing, Chunyong Hu, Shikun Huang, Lingping Gao, Fangzhen Li, Jingke Wang, Ke Guo, Wencong Xiao, Weibo Mao, et al. Fusionad: Multi-modality fusion for prediction and planning tasks of autonomous driving. *arXiv preprint arXiv:2308.01006*, 2023. 1, 2