Rethinking Imitation-based Planner for Autonomous Driving

Jie Cheng[1], Yingbing Chen[1], Xiaodong Mei[1], Bowen Yang[1], Bo Li[2] and Ming Liu[1,3]

imitation gap

Abstract—In recent years, imitation-based driving planners have reported considerable success. However, due to the absence of a standardized benchmark, the effectiveness of various designs remains unclear. The newly released nuPlan addresses this issue by offering a large-scale real-world dataset and a standardized closed-loop benchmark for equitable comparisons. Utilizing this platform, we conduct a comprehensive study on two fundamental yet underexplored aspects of imitation-based planners: the essential features for ego planning and the effective data augmentation techniques to reduce compounding errors. Furthermore, we highlight an imitation gap that has been overlooked by current learning systems. Finally, integrating our findings, we propose a strong baseline model—PlanTF. Our results demonstrate that a well-designed, purely imitation-based planner can achieve highly competitive performance compared to state-of-the-art methods involving hand-crafted rules and exhibit superior generalization capabilities in long-tail cases. Our models and benchmarks are publicly available. Project website **https://jchengai.github.io/planTF**.

I. INTRODUCTION

Learning-based planners are considered a potentially scalable solution for autonomous driving, supplanting traditional rule-based planners [1]–[3]. This has sparked significant research interest in recent years. In particular, imitation-based planners [4]–[12] are reported to achieve notable success in simulations and real-world scenarios. Nevertheless, these planners are predominantly trained and evaluated in diverse custom conditions (*e.g.* varying datasets, metrics, and simulation setups) owing to the absence of a standardized benchmark. Consequently, it becomes challenging to compare and summarize effective design choices for constructing practical learning-based systems.

Recently, the release of the large-scale nuPlan [13] dataset, alongside a standardized simulation benchmark, has provided a new opportunity for advancing learned motion planners. Enabled by this fresh benchmark, we conduct in-depth investigations on several common and critical yet not fully studied design choices of the learning-based planner, aiming to provide constructive suggestions for future research. This paper concentrates on two overarching and fundamental facets of the imitation-based planner: the requisite ego features for planning and the efficacious techniques of data augmentation.

The majority of imitation-based planning models [5]–[11] follow the success of prediction models and inherently incorporate the past trajectory of the autonomous vehicle

(AV) as an input feature, though imitation learning (IL) has frequently been noted for its tendency to acquire shortcuts from historical observations [4], [14]–[16]. Our research reaffirms that the past motion of the AV leads to significant closed-loop performance degradation. The planner achieves enhanced performance by solely utilizing the AV's present state. Surprisingly, it attains better closed-loop performance purely using the AV's current pose (position and heading). This implies that additional kinematic attributes typically deemed crucial for planning, such as velocity, acceleration, and steering, lead to a performance decline. To gain deeper insights into this phenomenon, we perform a sensitivity analysis to assess the impact of the AV's states on the resulting trajectory. Our experiments reveal that the planner can learn to exploit shortcuts from its kinematic states, even when past motion data is absent. To mitigate this challenge, we introduced a straightforward yet highly effective attention-based state dropout encoder, enabling the planner that utilize kinematic states to achieve optimal overall performance.

Imitation learning is also known to have compounding errors [17]. Perturbation-based augmentations [4]–[6] are a commonly employed strategy to instruct the planner on recovering from deviations. We conduct comprehensive experiments exploring various augmentation techniques, including history perturbation, state perturbation, and future correction. Additionally, we demonstrate the indispensability of proper normalization for the effectiveness of augmentation. Furthermore, we identify an ignored imitation gap within current learning frameworks and illustrate its potential impact.

Finally, by combining our findings, we provide a pure learning-based baseline model that demonstrates strong performance against state-of-the-art competitors on our standardized nuPlan benchmark. Our contributions are summarized as follows:

1) We perform an in-depth investigation on necessary features for ego planning, yielding counter-intuitive results contrary to mainstream practices. Furthermore, we introduced an effective attention-based state dropout encoder that attains the highest overall performance.
2) We conducted a comprehensive array of experiments involving various augmentation techniques, thereby elucidating an effective strategy to mitigate compounding errors. Additionally, we identified an overlooked imitation gap in current learning frameworks.
3) By combining our findings, we provide an open baseline model with strong performance. All our code, benchmarks, and models will be publicly released, as a reference for future research.

[1]Jie Cheng, Yingbing Chen, Xiaodong Mei and Bowen Yang are with the Hong Kong University of Science and Technology, Hong Kong SAR, China.{jchengai,ychengz,xmeiab,byangar}@connect.ust.hk
[2]Bo Li is with Lotus Technology Ltd. libo@lotuscar.com.cn
[3]Ming Liu is also with The Hong Kong University of Science and Technology (Guangzhou), Nansha, Guangzhou, China. eelium@ust.hk

augmentation techniques

"   "

II. RELATED WORK

Imitation-based planners are highly favored among learning-based planners due to their ease of convergence and typical scalability with data. They can be categorized into two distinct groups based on their input types:

1) *End-to-end*. End-to-end (E2E) methods [18]–[29] directly produce future trajectories using raw sensor inputs. Leveraging the closed-loop CARLA benchmark [30] and the collaborative efforts of the open-source community, E2E methods have achieved remarkable advancements within a short span of time: evolving from initial basic CNN-based approaches (LBC [23], CILRS [22]) to encompass multi-modal fusion (Transfuser [26], NEAT [25], MMFN [27], Interfuser [26], ThinkTwice [29]), as well as incorporating integrated perception and planning strategies (LAV [24], ST-P3 [19], VAD [21]). However, due to limitations posed by the simulated environment, these methods typically function at low vehicle speeds, and the behavior of the simulated traffic agents lacks realism and diversity. Emerging and intriguing research, such as data-driven traffic simulation [31], [32] and realistic sensor emulation [33], [34], holds the potential to mitigate these issues.

2) *Mid-to-mid*. These approaches [4], [6]–[12], [35], [36] utilize post-perception outcomes as input and can directly learn from recorded real-world data. Chauffernet [4] introduces the synthesis of perturbed trajectories to mitigate co-variate shift, a practice that becomes common in subsequent studies. [5] further augment the training data with on-policy rollouts. Several works have demonstrated the capability to operate real vehicles (SafetyNet [6], UrbanDriver [7], SafetyPathNet [9]). Many include a post-optimizer (DIPP [11], GameFormer [10], hotplan [35], pegasus [37]) to enhance the planner's robustness. All the abovementioned methods except hotplan use AV's history motion. Our study focuses on this category and provides an in-depth investigation of several critical design choices based on standardized data and benchmarks.

Beyond imitation. Another line of research aims to overcome the inherent limitations [14]–[16] of pure imitation learning (IL), such as utilizing environmental losses [4], [5], integrating IL with reinforcement learning [38]–[40], and incorporating adversarial training, also known as closed-loop training [41]–[43]. Our work shows that the pure IL-based planner has not reached its limit and can be significantly improved with appropriate design.

III. RETHINK IMITAION-BASED PLANNER

We consider the task of urban navigation employing a learned planner, trained by imitating the expert trajectory from the dataset. At each planning iteration, the planner receives various inputs, such as tracking data of surrounding objects up to a 2-second historical window, the current and past kinematic states of the ego vehicle, information about traffic lights, high-definition (HD) maps, speed limits, and the designated route. The planner is tasked with generating a trajectory for the subsequent 8 seconds. It is essential to
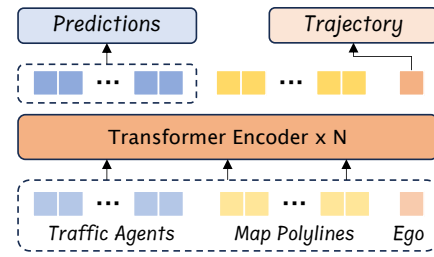
ILPlanner   2s

8s

Fig. 1. A brief overview of our baseline model. Agents, map, and ego features are separately encoded and then concatenated, which are subsequently processed by a stack of transformer encoder layers. The baseline model jointly predicts traffic agents and plans for ego vehicle at the scene level.

note that, unless otherwise stated, we employ the unaltered trajectory output from the planner in this paper. We intentionally avoid incorporating performance-enhancing techniques, such as rule-based emergency stops or post-optimization, to assess the planner's inherent performance.

nuPlan [13] is a large-scale closed-loop ML-based planning benchmark for autonomous vehicles. The dataset encompasses 1300 hours of recorded driving data collected from four urban centers, segmented into 75 scenario types using automated labeling tools.

Simulation. We use nuPlan's closed-loop simulator as our simulation environment. Each simulation entails a 15-second rollout at a rate of 10 Hz. It employs an LQR controller for trajectory tracking, while the control commands are utilized to update the state of the autonomous vehicle through an internal kinematic bicycle model. The behavior of background traffic varies based on the simulation mode, which can be non-reactive (log-replay) or reactive.

Metrics. We employ the official evaluation metrics provided by nuPlan, which include the open-loop score (OLS), non-reactive closed-loop score (NR-CLS), and reactive closed-loop score (R-CLS). R-CLS and NR-CLS share identical calculation methodologies, differing only in that R-CLS incorporates background traffic control via an Intelligent Driver Model (IDM) [44] during simulations. The closed-loop score is a comprehensive composite score, achieved through a weighted combination of factors such as traffic rule adherence, human driving resemblance, vehicle dynamics, goal attainment, and other metrics specific to the scenario. The score scales from 0 to 100. For a detailed description and calculation of the metrics, please refer to [45].

Baseline. As a baseline, we have adapted the motion-forecasting backbone model from our prior work [46] to address the planning task. Figure 1 provides a concise overview of the baseline model. Despite its simplicity, the architecture primarily comprises multiple Transformer encoders [47], demonstrating significant modeling capacity. We direct interested readers to the code base for details.

Benchmark. For all experiments, we standardize the data split for training and evaluation. For the training phase, we utilize all 75 scenario types in the nuPlan training set, limiting the total number of scenarios to 1M frames. For the

Models		Test14-random			Test14-hard		
Input feature	Variants	OLS	NR-CLS	R-CLS	OLS	NR-CLS	R-CLS
w/ history	shared encoder	90.20	56.50	56.28	**88.25**	48.60	51.32
	seperate encoder	**90.28**	61.02	59.85	86.77	51.98	49.34
w/o history	state3 (x, y, yaw)	81.13	**85.99**	**79.38**	71.43	68.44	**63.14**
	state4 (+vel.)	86.42	81.32	75.75	82.30	68.15	62.51
	staet5 (+acc.)	87.71	81.76	74.51	84.54	**68.67**	54.91
	state6 (+steer)	88.45	83.32	77.52	85.93	65.15	55.99

TABLE I. Results of different input features. For models with history input, "shared" and "separate" encoders indicate whether both the agent and ego vehicles utilize a shared history encoder or distinct ones. For models without history input, + refers to the inclusion of an additional state variable compared to the preceding model in the table. Higher values indicate better performance for all metrics, with the best metric highlighted in **bold**.

evaluation phase, we employ 14 scenario types specified by the nuPlan Planning Challenge, each comprising 20 scenarios. We examine two scenario selection schemes: (1) **Test14-random**: scenarios are randomly sampled from each type and fixed after selection, and (2) **Test14-hard**: in order to investigate the planner's performance on long-tail scenarios, we execute 100 scenarios of each type using a state-of-the-art rule-based planner (PDM-Closed [48]), subsequently selecting the 20 least-performing scenarios of each type. Example scenarios can be found on the project page. As the online leaderboard submission is closed, all evaluations are conducted on the nuPlan public test set.

A. Input feature makes a difference

This section aims to address the following questions: (1) *Is historical motion data essential for planning?* (2) *If not, do all current states of autonomous vehicles contribute to improving the planner's performance?* To address these inquiries, we conducted an investigation involving two sets of variants derived from our baseline model. The results on Test14-random and Test14-hard benchmarks are presented in Table I. Among the two historical variants, one shares its history encoder with other traffic agents, while the other employs a distinct history encoder for the ego vehicle's past motion. In the case of state-only models, we scrutinized several pivotal state variables essential for conventional planners, encompassing vehicle pose, velocity, acceleration, and steering angle. Based on the experimental results, we have the following findings:

History is not necessary. While models incorporating historical motion data exhibit superior off-policy evaluation performance (OLS), they manifest significantly poorer performance in closed-loop metrics compared to state-based models. This phenomenon may attributed to the well-established "copycat" problem [16] or learning shortcuts [49], wherein the planner relies on extrapolation from historical data without a comprehensive grasp of the underlying causal factors. Furthermore, the advantage in open-loop performance of historical models diminishes rapidly as the number of states increases in state-only models. Therefore, we conclude that history motions are not necessary for planning models.
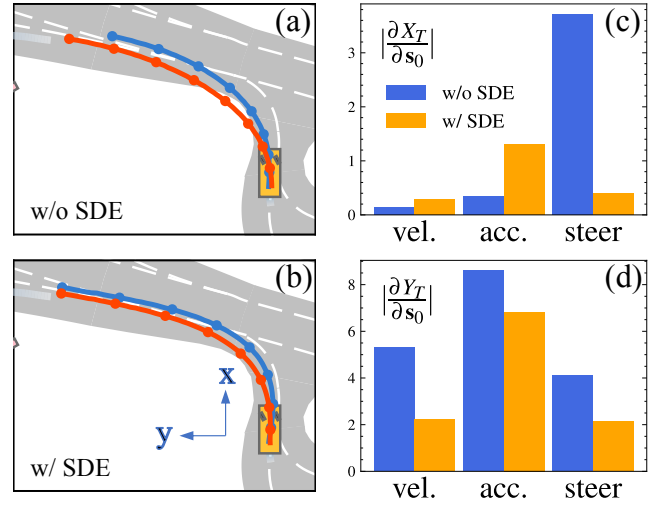

Fig. 2. The left side shows the planning trajectory of the *state6* model by adjusting AV's steering angle from 0.15 to 0.5 rad. The right side illustrates the magnitude of the gradient concerning the trajectory endpoint's position in relation to the AV's kinematic states.
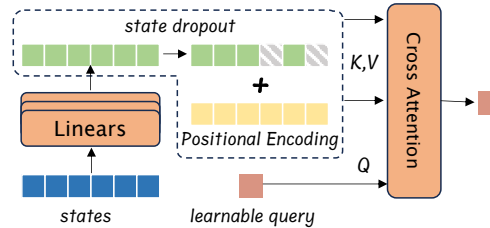

Fig. 3. Illustration of the attention-based state dropout encoder.

Shortcut learning in kinematic states. Kinematic states, such as velocity and acceleration, serve as vital initial boundary conditions for ensuring safety and comfort in trajectory planning. Nevertheless, we are surprised to find that the *state3* model, which exclusively relies on the autonomous vehicle's (AV) pose (comprising position and heading), significantly outperforms other models incorporating kinematic states in terms of CLS metrics. To gain deeper insights into this phenomenon, we study a left-turn case of the *state6* model. As depicted in Figure 2a, the model generates an undesired off-road trajectory when changing the steering

	SDE	Test14-random			Test14-hard		
		OLS	NR-CLS	R-CLS	OLS	NR-CLS	R-CLS
staet3	-	81.13	85.99	79.38	71.43	68.44	63.14
state5	✗	87.71	81.76	74.51	84.54	68.67	54.91
	✓	88.80 (+1.09)	86.73 (+4.97)	75.75 (+1.24)	84.29 (-0.25)	71.28 (+2.61)	61.88 (+6.97)
state6	✗	88.55	83.19	74.79	85.89	67.57	58.99
	✓	87.07 (-1.48)	86.48 (+3.29)	80.59 (+5.80)	83.32 (-2.57)	72.68 (+5.11)	61.70 (+2.71)

TABLE II. Experimental results of the state dropout encoder (SDE) on Test14-random and Test14-hard benchmark. Models with SDE gain significant improvements on CLS while maintaining high performance on OLS.

angle from 0.15 (blue) to 0.5 (red) rad. We hypothesize the model still learns false correlation from the kinematics even without the present of the past observation.

State dropout encoder. To confront our assumption, we propose an attention-based state dropout encoder (SDE), as shown in Figure 3. Each state variable undergoes individual embedding through a linear layer before being combined with positional encoding. A learnable query aggregates state embeddings through a cross-attention module. During training, each embedded state (except position and heading) token will be dropped with a certain probability. The encoder compels the model to unveil the root causes of behaviors by imposing partial constraints on its access to auxiliary information. Meanwhile, the model can enhance its planning capabilities when kinematic attributes are accessible. We implement the state dropout encoder in the *state5* and *state6* models, and the results are depicted in Table II. The results indicate that the utilization of SDE significantly enhances the closed-loop performance of the models. Importantly, when compared to *state3*, *state5* and *state6* models augmented with SDE exhibit not only improved closed-loop score but also substantially higher open-loop score, providing compelling evidence for the efficacy of SDE. We point out that *state3* model is fundamentally ambiguous as it loses all kinematic information (supported by its poor OLS performance). Figure 2(a)(b) displays the comparative planning results of *state6* model, while Figure 2(c)(d) presents comparative results for the magnitude of the gradient of the endpoint's position (X_T, Y_T) w.r.t. the initial kinematic states s_0. The results demonstrate that the model employing SDE is less sensitive to variations in kinematic states, resulting in more resilient planning outcomes.

B. Data augmentation and normalization

Data augmentation is a common practice for IL-based models to learn how to recover from deviations. In this section, we conduct comprehensive experiments on different data augmentation techniques, aiming to explore effective strategies to mitigate compounding errors. Different augmentation strategies are displayed in Figure 4. In Figure 4(a), an example driving scenario is depicted, with all coordinates normalized relative to the autonomous vehicle's center. In Figure 4(b), randomly sampled noise (perturbation)
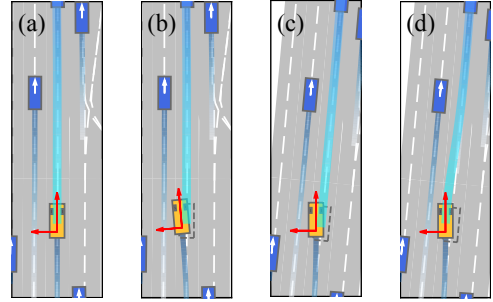

Fig. 4. (a) The original scenario. (b) Random noise is added to the AV's current state and history motion is smoothed. (c) The coordinates of the scenario are re-normalized based on the perturbed position of the AV. (d) A corrected future trajectory is generated using constrained nonlinear optimization.

	P	RN	FC	OLS	NR-CLS	R-CLS
history	✗	✗	✗	88.99	65.84	65.58
	✓	✓	✓	89.94	65.14	66.03
state3	✗	✗	✗	78.92	71.86	70.87
	✓	✗	✗	80.85	74.28	71.69
	✓	✓	✗	81.13	85.99	79.38
	✓	✓	✓	79.28	81.35	76.60
state5	✗	✗	✗	88.44	80.67	74.50
	✓	✗	✗	89.20	79.85	72.43
	✓	✓	✗	87.71	81.76	74.51
	✓	✓	✓	86.43	82.10	74.71
state6 +SDE	✗	✗	✗	88.33	77.28	74.10
	✓	✗	✗	87.71	77.70	75.18
	✓	✓	✗	87.07	86.48	80.59
	✓	✓	✓	85.50	82.95	76.09

TABLE III. Results of different augmentation and normalization combinations on **Test14-random** benchmark. P: **P**erturbation; RN: **Re**-Normalization; FC: **F**uture **C**orrection.

is added to the AV's current state, and its history states are smoothed accordingly. In Figure 4(c), it is demonstrated that the scenario's coordinates are re-normalized with respect to the autonomous vehicle's center after perturbation. Figure 4(d) showcases the generation of a rectified future trajectory through nonlinear optimization. Essentially, both strategies depicted in Figure 4(b) and 4(d) serve the common objective
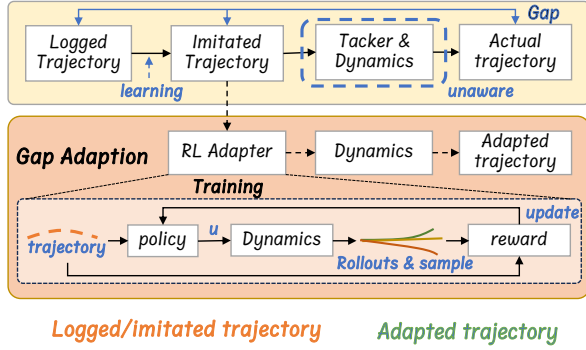
Fig. 5. Illustration of the imitation gap and the proposed RL adapter.

	Test14-random		Test14-hard	
Log-replay +	NR-CLS	R-CLS	NR-CLS	R-CLS
Perfect tracking	96.63	77.38	91.61	71.34
LQR	94.03	75.86	85.96	68.80
RL Adapter	96.3	77.13	91.65	71.62

TABLE IV. Exprimental results of the log-replay planner (perfect imitation) with different trackers on Test14-random and Test14-hard benchmarks. LQR is the default tracker used by the nuPlan benchmark and the RL adapter is our proposed method to address the imitation gap.

of guiding the vehicle back to the expert trajectory.

Table III displays the outcomes of experiments conducted with various augmentation strategies on four model variants. Based on these results, the following findings emerge: (1) In the case of the *history* and *state5* models, none of the data augmentations exhibit substantial enhancements. We postulate that the primary challenge faced by these two models is the issue of causal confusion, *i.e.* extrapolation from either historical or kinematic states. (2) For *state3* and *state6+SDE* models, perturbation is of great importance, but only works with proper normalization. For example, *state3* model's NR-CLS score boosts from 71.86 to 85.99 with perturbation and re-normalization, which is much higher than solely using perturbation (74.28). This implies it is important to keep the data distribution close between training and testing. (3) Providing a corrected guiding future trajectory does not serve a positive effect. One possible reason is that the manually generated trajectory does not align with the expert's trajectory distribution. Directly using the expert trajectory as supervision is a more effective choice as it keeps the original distribution, and small deviations can be easily fixed by the tracker.

C. The hidden imitation gap

The imitation gap. Within the most popular imitation learning frameworks, models imitate the logged expert's footprints from the dataset. We argue that this learning framework gives

Reward Term	Expression	Weight
Position Tracking	$e^{-15||\mathbf{P}_{xy} - \mathbf{P}_{xy}^*||}$	1.0
Action	$||\mathbf{u}||^2$	-0.01
Action Rate	$||\dot{\mathbf{u}}||^2$	-0.1
Lon. Acc. limit	$\mathbb{1}(\dot{v} > 2.4)$	-1
Jerk limit	$\mathbb{1}(||\ddot{v}|| > 4.0)$	-1
Yaw rate	$\mathbb{1}(||\dot{\theta}|| > 0.95)$	-0.5

TABLE V. The reward terms and expression of the RL adapter. Action $\mathbf{u}$ contains acceleration and steering rate. v and θ refers to the longitudinal and heading angle of the AV.

rise to a concealed gap in imitation, potentially leading to notable performance degradation. As illustrated in Figure 5, the recorded trajectory, commonly known as the expert trajectory, serves as the ground truth during the training of the imitation-based planner. The generated imitated trajectory is subsequently processed by the downstream tracker and the underlying system dynamics, yielding the final trajectory of the AV. Nevertheless, owing to the lack of knowledge about the tracker and dynamics during training, the actual trajectory may substantially deviate from the recorded trajectory, even when imitation is flawless. This assertion finds support in the experimental findings presented in Table IV. Notably, the NR-CLS of the *Log-replay + LQR* method on Test14-hard exhibits a significant decrease of 5.65 in comparison to perfect tracking.

RL Adapter. One possible solution is to directly imitate the control command rather than the trajectory points. Nevertheless, this approach is heavily reliant on the specific vehicle model, making it less generalizable and interpretable than the trajectory-based method. An alternative approach involves incorporating a differentiable kinematic model into the trajectory decoder [5], [51]. However, the kinematic model is often oversimplified to ensure differentiability. To tackle this challenge, we introduce a reinforcement learning-based trajectory adapter (RL Adapter) designed to bridge this gap. The RL Adapter transforms the imitated trajectory into the relevant control commands while accounting for the underlying dynamics. The benefits are two-folds. First, it can adapt to various vehicle models without retraining the planner. Second, it imposes no constraints on the vehicle model and remains compatible with non-differentiable vehicle models (*e.g.* high-fidelity real vehicle dynamics model). The training process of the adapter is displayed in Figure 5 and the rewards are shown in Table V. We use PPO [52] for policy optimization and the training finishes in 80K steps with a learning rate of 1e-3. As depicted in Table IV, the RL Adapter performs similarly to perfect tracking, highlighting its capacity to bridge the imitation gap. We notice that it can be integrated into the training process of the planner and leave this as future work.

IV. COMPARISON TO STATE OF THE ART

Implementation details. Integrating our findings, we propose a fully learning-based baseline planning model called

Planners		Test14-random			Test14-hard			
Type	Method	OLS	NR-CLS	R-CLS	OLS	NR-CLS	R-CLS	Time(ms)
Expert	Log-replay	100.0	94.03	75.86	100.0	85.96	68.80	-
Rule-based	IDM [44]	34.15	70.39	72.42	20.07	56.16	62.26	32
	PDM-Closed [48]	46.32	90.05	**91.64**	26.43	65.07	75.18	140
Hybrid†	GameFormer [10]	79.35	80.80	79.31	75.27	66.59	68.83	443
	PDM-Hybrid [48]	82.21	**90.20**	91.56	73.81	65.95	**75.79**	152
Learning-based	RasterModel [13]	62.93	69.66	67.54	52.4	49.47	52.16	82
	UrbanDriver [7]	82.44	63.27	61.02	76.9	51.54	49.07	124
	GC-PGP [50]	77.33	55.99	51.39	73.78	43.22	39.63	160
	PDM-Open [48]	84.14	52.80	57.23	79.06	33.51	35.83	101
	PlanTF (Ours)	**87.07**	86.48	80.59	**83.32**	**72.68**	61.7	155

TABLE VI. Comparison with state-of-the-arts. The runtime includes feature extraction and model inference based on Python code.

† *indicates these methods' final output trajectory relies on rule-based strategies or post-optimization.*

Planning Transformer (**PlanTF**). Specifically, we employ the *state6* model, incorporating a state attention dropout encoder with a dropout rate of 0.75. During training, we apply state perturbation with a probability of 0.5. The model is trained using a batch size of 128 and a weight decay of 1e-4 for 25 epochs. The initial learning rate is set to 1e-3, decaying to zero in a cosine manner.

Methods. We compare PlanTF's performance with several state-of-the-art planners. (1) **RasterModel** is a CNN-based planner provided in [13]. (2) **UrbanDriver** [7] is a vectorized planner based on PointNet-based polyline encoders and Transformer. Here we use its open-loop re-implementation and history perturbation is employed during training. (3) **GameFormer** [10] is a DETR-like interactive prediction and planning framework based on the level-k game, which incorporates a post-optimizer to generate the final trajectory. (4) **PDM*** [48] is the winning solution of the 2023 nuPlan Planning Challenge. **PDM-Closed** is a purely rule-based approach that ensembles the IDM [44] with different hyperparameters. **PDM-Hybrid** is a variant of PDM-closed that adds an offset predictor to improve its open-loop prediction performance. **PDM-Open** is the pure learning component without the IDM-based planner. Results are reproduced using their publicly available code and trained on our standard 1M data split.

Results. Table VI presents comparative results for the Test14-random and Test14-hard benchmarks. First, the proposed PlanTF significantly outperforms all other pure imitation-based methods across all metrics, particularly in terms of closed-loop performance. It is also the only learning-based method that surpasses the widely recognized IDM, highlighting the importance of proper design in IL. Second, when compared to rule-based and hybrid methods, PlanTF delivers outstanding OLS while maintaining highly competitive CLS, without the need for any tricky handcrafted rules or strategies. Notably, our approach achieves the highest NR-CLS on the Test14-hard benchmark, indicating that although rule-based methods perform well in ordinary scenarios (Test14-random), they struggle to generalize in long-tail situations (Test14-hard). In contrast, PlanTF demonstrates stronger generalization capabilities.

V. CONCLUSION

In this study, we systematically examine several crucial design aspects of imitation-based planners by utilizing the standardized nuPlan benchmark. Our findings reveal that catastrophic shortcut learning generally occurs for input features, such as historical motions and single-frame kinematic states. This leads to the unexpected outcome that planning solely based on the AV's current position results in superior closed-loop performance. To mitigate this issue, we introduce a straightforward attention-based state dropout encoder (SDE) that effectively addresses the shortcut learning problem. With the implementation of SDE, the *state6* model achieves the best overall performance. Data augmentation is another significant factor in imitation-based planners. Our results demonstrate that perturbation is vital for reducing compounding errors, but only effective with appropriate feature normalization. Furthermore, we observe that the original expert trajectory remains a reliable training ground truth, even when subjected to perturbation. In addition to these findings, we identify a neglected imitation gap caused by the model's lack of awareness of the underlying system dynamics, which considerably impacts the planner's performance. To rectify this issue, we propose a reinforcement learning-based adapter. By incorporating our findings, the proposed purely learning-based baseline model, PlanTF, demonstrates impressive performance compared to state-of-the-art approaches and is on par with methods that employ intricate rule-based strategies or post-optimization. This highlights the importance of proper design choices for imitation learning-based planners.

Limitation and future work. Despite pushing the boundaries of pure imitation-based planners, our method is constrained by the fundamental mismatch between *open-loop*

training and *closed-loop testing*. Incorporating closed-loop information and system dynamics into the training process constitutes our future research direction.

APPENDIX

Additional results on Val14 benchmark. We present the comparative results (Table. VII) on the **Val14** [48] benchmark. **Val14** contains 1180 scenarios from 14 scenario types.

Method	OLS	NR-CLS	R-CLS
Log-replay	100	94	80
IDM [44]	38	77	76
GC-PGP [50]	82	57	54
PlanCNN [13]	64	73	72
PDM-Hybrid [48]	84	93	92
PlanTF (Ours)	89.18	84.83	76.78

TABLE VII. Comparision to SOTAs on the Val14 benchmark. The results of other methods are taken from [48].

Ablation on the state dropout rate. Table VIII shows the ablation study on different dropout rate the of *state6+SDE* model.

Model	dropout	OLS	NR-CLS	R-CLS
state6	-	88.33	77.28	74.10
	0.25	89.11	81.70	78.44
	0.50	89.12	83.71	77.52
	0.75	87.07	86.48	80.59

TABLE VIII. Ablation study on the state dropout rate of the SDE.

REFERENCES

[1] L. Chen, P. Wu, K. Chitta, B. Jaeger, A. Geiger, and H. Li, "End-to-end autonomous driving: Challenges and frontiers," *arXiv preprint arXiv:2306.16927*, 2023.

[2] S. Hagedorn, M. Hallgarten, M. Stoll, and A. Condurache, "Rethinking integration of prediction and planning in deep learning-based automated driving systems: A review," *arXiv preprint arXiv:2308.05731*, 2023.

[3] J. Cheng, Y. Chen, Q. Zhang, L. Gan, C. Liu, and M. Liu, "Real-time trajectory planning for autonomous driving with gaussian process and incremental refinement," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8999–9005.

[4] M. Bansal, A. Krizhevsky, and A. Ogale, "Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst," *arXiv preprint arXiv:1812.03079*, 2018.

[5] J. Zhou, R. Wang, X. Liu, Y. Jiang, S. Jiang, J. Tao, J. Miao, and S. Song, "Exploring imitation learning for autonomous driving with feedback synthesizer and differentiable rasterization," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 1450–1457.

[6] M. Vitelli, Y. Chang, Y. Ye, A. Ferreira, M. Wołczyk, B. Osiński, M. Niendorf, H. Grimmett, Q. Huang, A. Jain, *et al.*, "Safetynet: Safe planning for real-world self-driving vehicles using machine-learned policies," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 897–904.

[7] O. Scheel, L. Bergamini, M. Wolczyk, B. Osiński, and P. Ondruska, "Urban driver: Learning to drive from real-world demonstrations using policy gradients," in *Conference on Robot Learning*. PMLR, 2022, pp. 718–728.

[8] J. Cheng, R. Xin, S. Wang, and M. Liu, "Mpnp: Multi-policy neural planner for urban driving," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 10 549–10 554.

[9] S. Pini, C. S. Perone, A. Ahuja, A. S. R. Ferreira, M. Niendorf, and S. Zagoruyko, "Safe real-world autonomous driving by learning to predict and plan with a mixture of experts," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 10 069–10 075.

[10] Z. Huang, H. Liu, and C. Lv, "Gameformer: Game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving," *arXiv preprint arXiv:2303.05760*, 2023.

[11] Z. Huang, H. Liu, J. Wu, and C. Lv, "Differentiable integrated motion prediction and planning with learnable cost function for autonomous driving," *IEEE transactions on neural networks and learning systems*, 2023.

[12] K. Guo, W. Jing, J. Chen, and J. Pan, "Ccil: Context-conditioned imitation learning for urban driving," *arXiv preprint arXiv:2305.02649*, 2023.

[13] 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," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2021.

[14] U. Muller, J. Ben, E. Cosatto, B. Flepp, and Y. Cun, "Off-road obstacle avoidance through end-to-end learning," *Advances in neural information processing systems*, vol. 18, 2005.

[15] D. Wang, C. Devin, Q.-Z. Cai, P. Krähenbühl, and T. Darrell, "Monocular plan view networks for autonomous driving," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 2876–2883.

[16] C. Wen, J. Lin, T. Darrell, D. Jayaraman, and Y. Gao, "Fighting copycat agents in behavioral cloning from observation histories," *Advances in Neural Information Processing Systems*, vol. 33, pp. 2564–2575, 2020.

[17] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[18] W. Zeng, W. Luo, S. Suo, A. Sadat, B. Yang, S. Casas, and R. Urtasun, "End-to-end interpretable neural motion planner," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8660–8669.

[19] S. Hu, L. Chen, P. Wu, H. Li, J. Yan, and D. Tao, "St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning," in *European Conference on Computer Vision*. Springer, 2022, pp. 533–549.

[20] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, *et al.*, "Planning-oriented autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 853–17 862.

[21] B. Jiang, S. Chen, Q. Xu, B. Liao, J. Chen, H. Zhou, Q. Zhang, W. Liu, C. Huang, and X. Wang, "Vad: Vectorized scene representation for efficient autonomous driving," *arXiv preprint arXiv:2303.12077*, 2023.

[22] F. Codevilla, E. Santana, A. M. López, and A. Gaidon, "Exploring the limitations of behavior cloning for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9329–9338.

[23] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl, "Learning by cheating," in *Conference on Robot Learning*. PMLR, 2020, pp. 66–75.

[24] D. Chen and P. Krähenbühl, "Learning from all vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 222–17 231.

[25] K. Chitta, A. Prakash, and A. Geiger, "Neat: Neural attention fields for end-to-end autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 793–15 803.

[26] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger, "Transfuser: Imitation with transformer-based sensor fusion for au-

tonomous driving," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[27] Q. Zhang, M. Tang, R. Geng, F. Chen, R. Xin, and L. Wang, "Mmfn: Multi-modal-fusion-net for end-to-end driving," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 8638–8643.

[28] H. Shao, L. Wang, R. Chen, H. Li, and Y. Liu, "Safety-enhanced autonomous driving using interpretable sensor fusion transformer," in *Conference on Robot Learning*. PMLR, 2023, pp. 726–737.

[29] X. Jia, P. Wu, L. Chen, J. Xie, C. He, J. Yan, and H. Li, "Think twice before driving: Towards scalable decoders for end-to-end autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 983–21 994.

[30] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*. PMLR, 2017, pp. 1–16.

[31] M. Igl, D. Kim, A. Kuefler, P. Mougin, P. Shah, K. Shiarlis, D. Anguelov, M. Palatucci, B. White, and S. Whiteson, "Symphony: Learning realistic and diverse agents for autonomous driving simulation," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2445–2451.

[32] Q. Li, Z. Peng, L. Feng, C. Duan, W. Mo, B. Zhou, *et al.*, "Scenarionet: Open-source platform for large-scale traffic scenario simulation and modeling," *arXiv preprint arXiv:2306.12241*, 2023.

[33] A. Amini, T.-H. Wang, I. Gilitschenski, W. Schwarting, Z. Liu, S. Han, S. Karaman, and D. Rus, "Vista 2.0: An open, data-driven simulator for multimodal sensing and policy learning for autonomous vehicles," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022.

[34] Z. Wu, T. Liu, L. Luo, Z. Zhong, J. Chen, H. Xiao, C. Hou, H. Lou, Y. Chen, R. Yang, Y. Huang, X. Ye, Z. Yan, Y. Shi, Y. Liao, and H. Zhao, "Mars: An instance-aware, modular and realistic simulator for autonomous driving," *CICAI*, 2023.

[35] Y. Hu, K. Li, P. Liang, J. Qian, Z. Yang, H. Zhang, W. Shao, Z. Ding, W. Xu, and Q. Liu, "Imitation with spatial-temporal heatmap: 2nd place solution for nuplan challenge," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2023.

[36] K. Renz, K. Chitta, O.-B. Mercea, A. S. Koepke, Z. Akata, and A. Geiger, "Plant: Explainable planning transformers via object-level representations," in *Conference on Robot Learning*. PMLR, 2023, pp. 459–470.

[37] W. Xi, L. Shi, and G. Cao, "An imitation learning method with data augmentation and post processing for planning in autonomous driving," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2023.

[38] Y. Lu, J. Fu, G. Tucker, X. Pan, E. Bronstein, B. Roelofs, B. Sapp, B. White, A. Faust, S. Whiteson, *et al.*, "Imitation is not enough: Robustifying imitation with reinforcement learning for challenging driving scenarios," *NeurIPS 2022 Workshop on Machine Learning for Autonomous Driving*, 2022.

[39] H. Liu, Z. Huang, J. Wu, and C. Lv, "Improved deep reinforcement learning with expert demonstrations for urban autonomous driving," in *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2022, pp. 921–928.

[40] Z. Huang, J. Wu, and C. Lv, "Efficient deep reinforcement learning with imitative expert priors for autonomous driving," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[41] N. Baram, O. Anschel, I. Caspi, and S. Mannor, "End-to-end differentiable adversarial imitation learning," in *International Conference on Machine Learning*. PMLR, 2017, pp. 390–399.

[42] E. Bronstein, M. Palatucci, D. Notz, B. White, A. Kuefler, Y. Lu, S. Paul, P. Nikdel, P. Mougin, H. Chen, *et al.*, "Hierarchical model-based imitation learning for planning in autonomous driving," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 8652–8659.

[43] G. C. K. Couto and E. A. Antonelo, "Hierarchical generative adversarial imitation learning with mid-level input generation for autonomous driving on urban environments," *arXiv preprint arXiv:2302.04823*, 2023.

[44] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.

[45] Motional. nuplan metrics. [Online]. Available: https://nuplan-devkit. readthedocs.io/en/latest/metrics_description.html

[46] J. Cheng, X. Mei, and M. Liu, "Forecast-MAE: Self-supervised pretraining for motion forecasting with masked autoencoders," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[47] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[48] D. Dauner, M. Hallgarten, A. Geiger, and K. Chitta, "Parting with misconceptions about learning-based vehicle motion planning," in *Conference on Robot Learning (CoRL)*, 2023.

[49] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, no. 11, pp. 665–673, 2020.

[50] M. Hallgarten, M. Stoll, and A. Zell, "From prediction to planning with goal conditioned lane graph traversals," *arXiv preprint arXiv:2302.07753*, 2023.

[51] H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, J. Schneider, D. Bradley, and N. Djuric, "Deep kinematic models for kinematically feasible vehicle trajectory predictions," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 10 563–10 569.

[52] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.