CAR Planner: Constrained-Attention-Based Robust Imitation Learning for Autonomous Driving

Anonymous Author

***Abstract*—Imitation Learning (IL) for autonomous driving still suffers from shortcut learning, where policies rely on spurious correlations rather than causal driving behavior. In Transformer-based IL planners, such shortcut learning can appear as attention collapse, where attention weights become excessively concentrated on a small subset of input channels. This weakens robustness in challenging closed-loop scenarios. We propose CAR Planner, which mitigates shortcut learning by imposing a constrained optimization on ego-state cross-attention. The constraint penalizes only excessive concentration of attention weights, preserving task-driven channel-importance ordering while discouraging over-reliance on a few channels. On the nuPlan benchmark, CAR Planner shows substantially less degradation under ego-state channel reduction and achieves stronger closed-loop performance in challenging scenarios than strong baselines, with negligible training overhead and no inference-time cost. These results demonstrate that constrained attention is effective for robust imitation learning in autonomous driving.**

***Index Terms*—Motion and Path Planning, Imitation Learning, Planning under Uncertainty, Deep Learning Methods**

I. INTRODUCTION

ROBUST motion planning remains a central challenge in autonomous driving [1], [2]. Among data-driven approaches, imitation learning (IL) is attractive because it learns policies directly from expert demonstrations and has shown strong performance in modern driving benchmarks [3]–[5]. However, IL planners are vulnerable to shortcut learning, where the model relies on spurious correlations rather than causal driving cues [6]–[8]. This often creates a gap between nominal performance and robustness in challenging closed-loop scenarios.

In IL-based Transformer planners on nuPlan [9], shortcut learning is especially visible in ego-state aggregation. In this work, the current ego state is represented by six channels, (x, y, ψ, v, a, s), corresponding to position, heading, velocity, acceleration, and steering angle. The planner may overemphasize only a subset of these channels while underusing the others. Such attention collapse can still preserve strong open-loop accuracy, but often degrades performance in challenging splits and reactive closed-loop evaluation. Existing remedies, including heuristic penalties and the State Dropout Encoder (SDE) [10], address this issue only indirectly. Penalty terms can conflict with the task objective, and stochastic masking discards part of the input.

To address these limitation, we propose **CAR Planner**, a constrained-attention-based robust IL planner. **CAR Planner** constrains the deviation of ego-state attention weights from a uniform distribution and enforces this constraint with the Augmented Lagrangian Method (ALM) [11], [12]. The goal is not full uniformity but controlled dispersion. Task-driven channel-importance ordering is preserved, and only excessive concentration is penalized. This encourages balanced use of ego-state channels, preserves inference-time architecture, and yields stable runtime behavior in large-scale evaluation. We evaluate **CAR Planner** on the nuPlan benchmark using open-loop score (OLS), which measures prediction quality under logged trajectories, non-reactive closed-loop score (NR-CLS), which evaluates rollout performance without agent responses, and reactive closed-loop score (R-CLS), which further tests robustness under interactive closed-loop driving. Our evaluation therefore emphasizes robustness in increasingly challenging scenarios.

The main contributions of this work are as follows.

- **Constrained ego-state attention for shortcut mitigation.** We formulate ego-state attention dispersion as an inequality-constrained optimization problem and enforce it with ALM, preserving task-driven channel-importance ordering by penalizing only excessive concentration and preventing attention collapse.
- **Lightweight, architecture-preserving regularization.** The proposed constraint is applied only to the ego-state aggregation module and requires no auxiliary data, while preserving inference-time computation and stable runtime behavior.
- **Robustness gains on nuPlan across random and hard splits. CAR Planner** reduces sensitivity to ego-state ablations and improves closed-loop performance in challenging scenarios compared with strong baselines, including the base model and SDE-based regularization.

II. RELATED WORK

Autonomous driving planners are commonly grouped into rule-based, learning-based, and hybrid approaches. Rule-based methods such as IDM [13] and PDM-Closed [14] provide strong priors, but are often brittle in challenging driving situations. Learning-based planners learn policies directly from data. Representative models include RasterModel, UrbanDriver [15], GC-PGP [16], and PDM-Open [14]. Recent IL planners increasingly use Transformer architectures [17] to integrate ego-state, agent, and map features. While this improves interaction modeling, it can also cause ego-state attention to collapse onto a few channels and encourage shortcut learning through over-reliance on limited ego cues. PlanTF [10] is a representative shortcut-mitigation approach that applies State Dropout Encoder (SDE), but it addresses the problem only indirectly by masking input channels rather than directly regulating the attention distribution. Hybrid planners such as GameFormer [18], PDM-Hybrid [14], and Pluto [19]

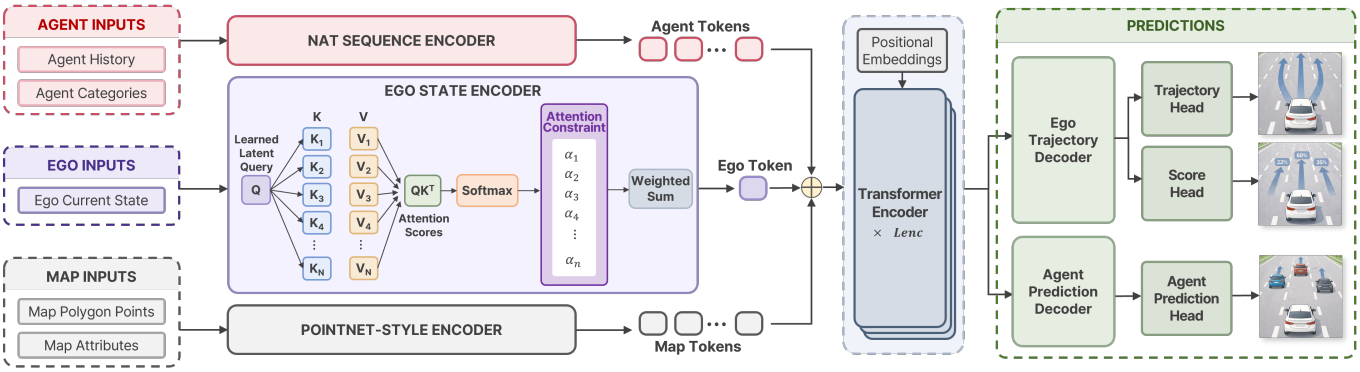

Fig. 1. Overview of the proposed **CAR Planner** architecture. Agent history, the current ego state, and local map polygons are encoded into agent, ego, and map tokens. The ego-state encoder uses a learned latent query with an attention constraint to regulate channel aggregation, and the resulting tokens are processed by a shared Transformer encoder for ego and non-ego trajectory prediction. Base and PlanTF share the same architecture shown here and differ only in ego-state regularization. Base uses no additional regularization, whereas PlanTF applies State Dropout Encoder regularization.

improve robustness with additional structure and objectives, often at higher inference cost. Other learning-based approaches using auxiliary objectives, reinforcement learning, adversarial training, or closed-loop training [8], [20]–[23] also modify the training or inference pipeline. By contrast, our method is a lightweight, architecture-preserving regularization scheme that directly constrains ego-state attention dispersion with ALM.

III. CAR PLANNER

A. Planner Architecture

For shortcut-learning analysis, we focus on three Transformer-based planner variants: **Base** (vanilla), **PlanTF** (with SDE), and **CAR Planner** (with ALM). These three variants share the same backbone. Figure 1 illustrates the **CAR Planner** instantiation. **Base** and **PlanTF** use the same architecture and differ only in ego-state regularization. The model takes three inputs: past states of surrounding agents, the current 6-dimensional ego state, and a local vectorized map. These are encoded into agent, ego, and map tokens, augmented with geometric embeddings, and processed by a shared Transformer encoder. The encoded ego token predicts multi-modal ego trajectories, while a lightweight head predicts future motions of non-ego agents. Table I includes additional baselines for overall benchmark comparison, but the shortcut-learning analysis focuses on the three IL-based Transformer variants, which enable a controlled comparison under the same backbone.

Common backbone. Agent-history tokens are produced from agent kinematic sequences, the ego token from the current 6-dimensional state (x, y, ψ, v, a, s), and the map from polygon features encoding lane geometry and semantics. For each token, a small MLP projects the pose descriptor $[x, y, \cos\theta, \sin\theta]$ into a geometric embedding. The resulting tokens are concatenated and processed by a shared Transformer encoder with key-padding masks for invalid inputs.

Prediction heads. From the encoded ego token, a trajectory head predicts M candidate future trajectories of length T, where each step is represented by $(x, y, \cos\psi, \sin\psi)$ with a mode score. At inference time, the most probable mode is selected and the heading is recovered by $\operatorname{atan2}(\sin\psi, \cos\psi)$. In parallel, a lightweight MLP predicts future positions for non-ego agents.

Difference between variants. The three planners differ only in ego-state regularization before the shared Transformer. **Base** uses no additional regularization, **PlanTF** applies State Dropout Encoder (SDE), and **CAR Planner** imposes a dispersion constraint on the same attention distribution. All other components are shared.

Default configuration. Unless otherwise specified, all experiments use d=128, L_e=4 Transformer layers, h=8 attention heads, M=6 trajectory modes, and T=80 future steps.

B. Objectives

The following objectives are shared by **Base**, **PlanTF (SDE)**, and **CAR Planner**. Let $\widehat{Y} \in \mathbb{R}^{M\times T\times 4}$ and $\pi \in \mathbb{R}^{M}$ denote the predicted ego trajectories and mode logits, respectively, and let $Y \in \mathbb{R}^{T\times 4}$ be the ground-truth ego trajectory. For non-ego agents, let $\widehat{P} \in \mathbb{R}^{(A-1)\times T\times 2}$ and $P \in \mathbb{R}^{(A-1)\times T\times 2}$ denote the predicted and ground-truth future positions, where A is the total number of agents including the ego agent.

Best-mode selection. During training, a single trajectory mode is selected by minimizing the displacement error in the position space:

$$m^* = \arg\min_m \sum_{t=1}^{T} \left\| \widehat{Y}_{m,t,1:2} - Y_{t,1:2} \right\|_2, \tag{1}$$

$$\widehat{\tau} = \widehat{Y}_{m^*}. \tag{2}$$

The selected mode index m^* is used both for ego-trajectory regression and for the classification target.

Ego imitation loss. The ego loss consists of a regression term and a classification term:

$$\mathscr{L}_{\text{reg}} = L1_{\text{smooth}}(\widehat{\tau}, Y), \tag{3}$$

$$\mathscr{L}_{\text{cls}} = \text{CE}(\pi, m^*). \tag{4}$$

Representing the heading as $(\cos\psi, \sin\psi)$ avoids angle wrap-around.

TABLE I
PERFORMANCE COMPARISON ON THE TEST14 BENCHMARK.

Planners		Test14-random			Test14-hard		
Type	**Method**	OLS	NR-CLS	R-CLS	OLS	NR-CLS	R-CLS
Expert	Log-replay	100.0	94.03	75.86	100.0	85.96	68.80
Rule-based	IDM	34.15	70.39	72.42	20.07	56.15	62.26
	PDM-Closed	46.33	90.05	91.64	26.43	65.08	75.19
Hybrid	PDM-Hybrid	81.26	90.10	91.29	74.08	65.99	76.07
Learning-based	RasterModel	62.93	68.71	67.52	52.38	50.65	51.41
	UrbanDriver	82.45	63.57	60.92	76.90	51.67	49.06
	GC-PGP	80.89	61.57	54.88	75.40	47.16	41.39
	PDM-Open	84.14	52.81	57.23	79.06	33.51	35.83
	Base	86.64	80.01	74.48	82.48	65.30	53.11
	PlanTF	86.27	85.23	79.36	83.34	70.03	59.83
	Pluto (w/o post.)	87.15	86.60	77.96	81.89	70.28	56.70
	CAR Planner (Ours)	87.67	85.91	78.31	86.31	70.81	66.12

Non-ego prediction loss. Non-ego future positions are supervised with smooth-L_1 loss over valid agents and time steps:

$$\mathscr{L}_{\mathrm{p}} = L1_{\mathrm{smooth}}\big(\widehat{P}, P\big). \tag{5}$$

Task objective. The final task objective is

$$\mathscr{L}_{\mathrm{task}} = \mathscr{L}_{\mathrm{reg}} + \mathscr{L}_{\mathrm{cls}} + \mathscr{L}_{\mathrm{p}}. \tag{6}$$

This task loss is used in all three planners. **CAR Planner** further imposes the constraint described in the following subsection and handles it using ALM.

C. Problem formulation as a constrained optimization

Let $a_\theta^{(n)} \in \Delta^{C-1}$ denote the ego-state attention weights for the n-th sample, where a single learned query attends to C=6 ego-state channels. To discourage excessive concentration on only a few channels, the deviation of the attention distribution from uniform is measured as

$$d\big(a_\theta^{(n)}\big) = \frac{1}{C}\sum_{i=1}^{C}\Big|a_{\theta,i}^{(n)} - \tfrac{1}{C}\Big|. \tag{7}$$

This quantity is then averaged over the batch:

$$D(\theta) = \frac{1}{N}\sum_{n=1}^{N} d\big(a_\theta^{(n)}\big). \tag{8}$$

The goal is not to force perfectly uniform attention, but to keep the deviation below a prescribed margin $\delta > 0$. With a positive margin, task-driven channel-importance ordering is preserved, and only excessive concentration is penalized. The constraint suppresses overly dominant channels and lifts severely underused ones, preventing attention collapse while preserving relative priority. This leads to the inequality constraint

$$g(\theta) = D(\theta) - \delta \le 0, \tag{9}$$

where δ controls the allowed level of concentration. In all experiments, δ=0.12 is used. A smaller value enforces more dispersed attention, while a larger value allows stronger selectivity.

D. ALM-based update rule

The resulting optimization problem is

$$\min_{\theta}\ \mathscr{L}_{\mathrm{task}}(\theta) \quad \text{subject to} \quad g(\theta) \le 0, \tag{10}$$

which encourages balanced ego-state attention while preserving the original planning objective.

This constrained problem is optimized with the Augmented Lagrangian Method (ALM). Using a nonnegative Lagrange multiplier λ and penalty coefficient $\rho > 0$, the augmented objective is

$$\mathscr{L}_{\mathrm{aug}}(\theta,\lambda) = \mathscr{L}_{\mathrm{task}}(\theta) + \lambda\,[g(\theta)]_+ + \frac{\rho}{2}\,[g(\theta)]_+^2. \tag{11}$$

Here, $[z]_+ := \max(0,z)$ ensures that only constraint violations are penalized. In all experiments, ρ=3 is used. At each iteration, the model parameters θ are updated by minimizing $\mathscr{L}_{\mathrm{aug}}$, and the multiplier is updated after the optimizer step according to

$$\lambda \ \leftarrow\ \max\big\{0,\ \lambda + \rho\,[g(\theta)]_+\big\}. \tag{12}$$

The constraint is applied only to the ego-state attention module, and the rest of the planner remains unchanged. Since the penalty is computed over only C=6 channels, the additional training cost is negligible. At inference time, neither the penalty term nor the multiplier update is used, so the runtime architecture is identical to the base planner.

IV. EXPERIMENTS

A. Experimental setup

The planners were trained for 20 epochs on NVIDIA RTX 4090 GPUs with a total batch size of 32, using Adam (learning rate 1e−3, weight decay 1e−4). During training, a state-perturbation augmentation [24] adds bounded noise to the current ego state, and all coordinates are re-normalized with respect to the perturbed ego frame.

Evaluation follows the nuPlan benchmark on two official test splits, `test14-random` and `test14-hard`, each with 280 scenarios across 14 scenario types.

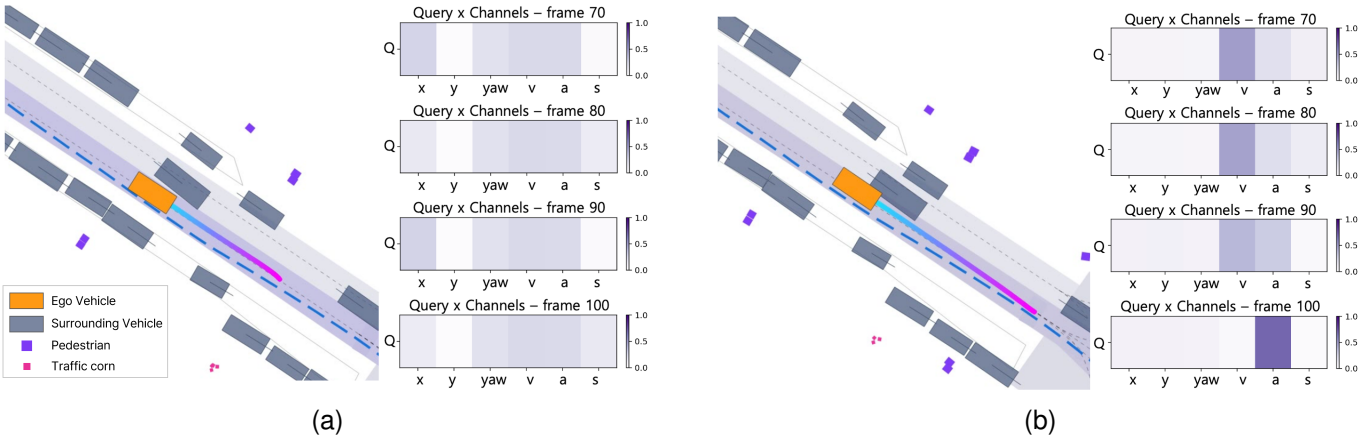

Fig. 2. **Ego-state attention comparison.** The scenario examples from nuPlan test14-hard split, R-CLS. (a) **CAR Planner**: attention over (x, y, ψ, v, a, s) stays dispersed with mild selectivity across frames, yielding a safe maneuver. (b) PlanTF: attention progressively collapses onto a few channels (notably a), reducing situational balance and culminating in a collision.

`test14-random` contains randomly sampled scenarios, whereas `test14-hard` is a curated set of challenging scenarios selected by PDM-Closed [14]. Reported metrics are OLS, NR-CLS, and R-CLS, where higher is better.

To analyze shortcut-learning mitigation, we focus on the following three model variants.

- **Base (Vanilla).** Uses the same single-query ego-state *state attention* encoder as the other variants, but without SDE or ALM. Agent/map encoders and the shared Transformer are identical to the other variants.
- **PlanTF (with SDE).** Uses the same backbone and single-query ego-state *state attention* encoder, with *stochastic state-channel dropout* (SDE) applied during training.
- **CAR Planner (with Constrained weights).** Uses the same backbone and single-query ego-state attention encoder, *regularized by a dispersion constraint* on the 6-way attention weights. Inference-time computation is unchanged.

In addition, Table I reports overall benchmark comparison against rule-based, hybrid, and learning-based baselines, while Table II provides a detailed reactive closed-loop sub-metric comparison among learning-based planners.

B. Experimental results

Table I summarizes overall OLS, NR-CLS, and R-CLS performance on both `test14-random` and `test14-hard`. **CAR Planner** remains competitive with state-of-the-art methods on `test14-random` and shows clearer gains on the more challenging `test14-hard` split, indicating stronger robustness in difficult conditions. This trend supports our interpretation that attention collapse drives shortcut learning, which becomes more harmful under closed-loop rollout and the resulting distribution shift. Table II further shows that **CAR Planner** achieves the best scores on key `test14-hard` safety-related sub-metrics, including Collisions and TTC, and remains close to the best on several others, indicating robust and stable behavior in difficult scenarios.

TABLE II
REACTIVE CLOSED-LOOP SUB-METRIC COMPARISON AMONG LEARNING-BASED PLANNERS ON THE TEST14-HARD BENCHMARK. HIGHER IS BETTER. BEST SCORES ARE SHOWN IN BOLD.

Planner	Collisions	TTC	Drivable	Comfort	Progress	Speed
RasterModel	80.67	74.50	78.48	78.88	86.05	97.10
UrbanDriver	69.85	63.97	82.72	**98.52**	**93.75**	86.13
GC-PGP	80.88	74.63	84.55	91.54	76.10	**98.54**
PDM-Open	77.20	69.11	84.19	**98.52**	73.89	97.56
Base	88.11	81.50	92.64	88.23	72.79	98.02
PlanTF	85.84	80.88	92.64	93.01	84.55	97.01
Pluto (w/o post.)	90.63	85.01	**97.00**	89.13	74.90	98.50
CAR Planner (Ours)	**91.83**	**86.49**	94.02	98.16	87.13	98.22

Figure 2 provides qualitative evidence from two nuPlan `test14-hard` scenarios. PlanTF shows progressive attention collapse to a few channels, indicating shortcut learning through over-reliance on limited ego-state cues. This weakens sensitivity to lateral cues (x, y, ψ) and can lead to unsafe maneuvers under closed-loop distribution shift. By contrast, **CAR Planner** maintains a dispersed yet selective pattern across frames, which suppresses shortcut learning and yields more balanced situational awareness and safer behavior.

Table III provides quantitative evidence through ego-state channel ablation. Rows `state5` and `state6` report scores with five and six channels, and the gap is defined as gap $=$ state6 $-$ state5, where smaller values indicate lower shortcut reliance. Figure 3 visualizes the same gap trend across metrics and splits. **CAR Planner** shows smaller or comparable gaps across both `test14-random` and `test14-hard`, indicating lower dependence on any single ego-state component and more stable performance under state reduction.

Figure 4 compares inference efficiency across planners. **CAR Planner** shows the most compact latency distribution and the lowest cumulative runtime, indicating efficient and stable inference over large-scale evaluation. Its runtime is comparable to, and slightly better than, PlanTF, consistent with our design in which the constraint is used only during training. By contrast, Pluto is a strong performance baseline but shows a

TABLE III
RESULTS ON SHORTCUT-LEARNING MITIGATION ACROSS THE NUPLAN TEST14-RANDOM AND TEST14-HARD SPLITS.

State	Method	Test14-random			Test14-hard		
		OLS	NR-CLS	R-CLS	OLS	NR-CLS	R-CLS
state5	Base	84.82	77.94	73.67	81.12	63.90	49.87
	PlanTF	87.81	83.96	72.72	84.14	69.27	56.91
	CAR Planner (Ours)	87.16	85.12	78.01	86.18	68.62	64.26
state6	Base	86.64	80.01	74.48	82.48	65.30	53.11
	PlanTF	86.27	85.23	79.36	83.34	70.03	59.83
	CAR Planner (Ours)	87.67	85.91	78.31	86.31	70.81	66.12

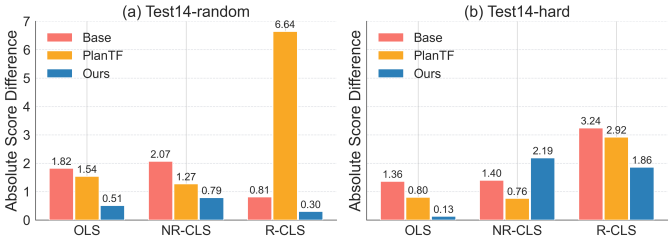

Fig. 3. Ego-state channel ablation gaps (state6 - state5) on nuPlan test14-random and test14-hard splits. Smaller gaps indicate reduced shortcut reliance.

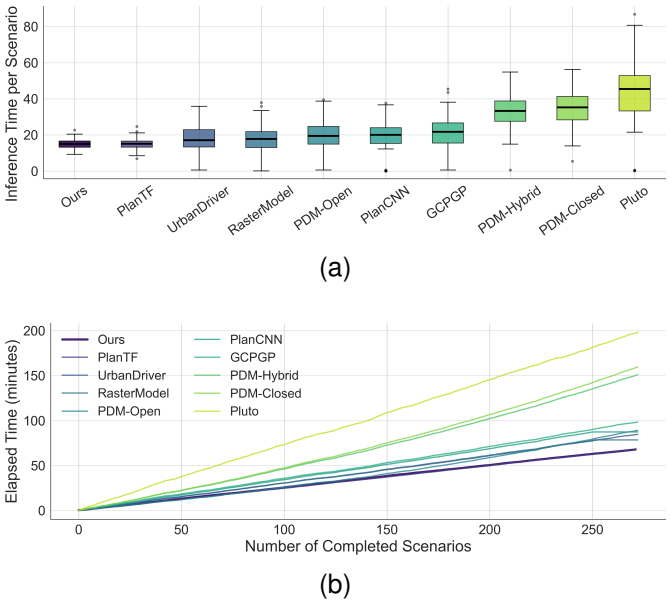

Fig. 4. Inference efficiency comparison on the evaluation scenarios. (a) Per-scenario inference latency distribution. (b) Cumulative elapsed runtime as scenarios are completed.

broader latency spread and substantially higher runtime. These results indicate that **CAR Planner** improves robustness while preserving efficient inference.

Figure 5 illustrates a typical failure-recovery contrast on the nuPlan test14-hard split. This case is consistent with the gains in R-CLS and comfort, highlighting improved closed-loop stability and safety in complex scenes.

V. CONCLUSION

This paper introduced **CAR Planner**, a constrained-attention-based framework for mitigating shortcut learning in imitation-learning planners. By formulating attention dispersion as a constrained optimization problem and enforcing it with ALM, the proposed approach suppresses attention collapse as a key mechanism of shortcut learning without discarding input information or altering inference-time computation.

Experiments on the nuPlan benchmark showed that **CAR Planner** consistently mitigates shortcut learning across both random and hard splits. It reduces degradation under reduced ego-state inputs, improves reactive closed-loop performance, and yields gains in safety- and comfort-related metrics. Runtime experiments further confirmed that these benefits are achieved with negligible inference overhead. Compared with strong baselines such as PlanTF, **CAR Planner** delivered more reliable performance in hard and reactive evaluations while preserving efficient and stable inference.

The current formulation is limited to ego-state cross-attention and has been validated only on nuPlan. Extending the constraint to interaction and map-level features, and validating on broader datasets or real-world testbeds, remains an important direction for future work.

REFERENCES

[1] A. Tampuu, T. Matiisen, M. Semikin, D. Fishman, and N. Muhammad, *A survey of end-to-end driving: Architectures and training methods*. IEEE, 2020, vol. 33, no. 4.

[2] K. Muhammad, A. Ullah, J. Lloret, J. Del Ser, and V. H. C. De Albuquerque, "Deep learning for safe autonomous driving: Current challenges and future directions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4316–4336, 2020.

[3] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," *Advances in neural information processing systems*, vol. 1, 1988.

[4] A. Amini, I. Gilitschenski, J. Phillips, J. Moseyko, R. Banerjee, S. Karaman, and D. Rus, "Learning robust control policies for end-to-end autonomous driving from data-driven simulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1143–1150, 2020.

[5] M. Zare, P. M. Kebria, A. Khosravi, and S. Nahavandi, "A survey of imitation learning: Algorithms, recent developments, and challenges," *IEEE Transactions on Cybernetics*, 2024.

[6] 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.

[7] C.-C. Chuang, D. Yang, C. Wen, and Y. Gao, "Resolving copycat problems in visual imitation learning via residual action prediction," in *European Conference on Computer Vision*. Springer, 2022, pp. 392–409.

[8] 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.

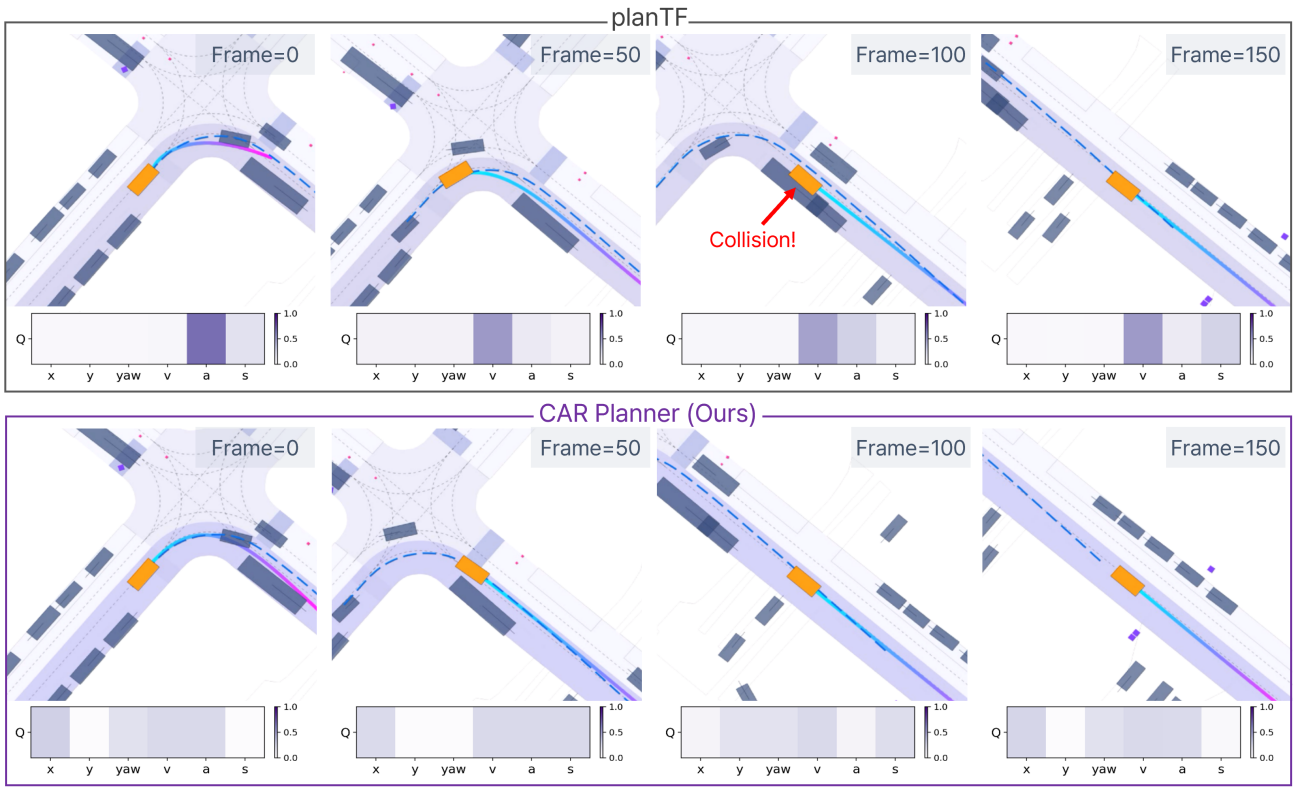

Fig. 5. A scenario example from the nuPlan test14-hard split. PlanTF clips a parked car during a right turn, while our method negotiates the corner without incident.

[9] 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," *arXiv preprint arXiv:2106.11810*, 2021.

[10] J. Cheng, Y. Chen, X. Mei, B. Yang, B. Li, and M. Liu, "Rethinking imitation-based planners for autonomous driving," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 14 123–14 130.

[11] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 2006.

[12] D. P. Bertsekas, "Nonlinear programming," *Journal of the Operational Research Society*, vol. 48, no. 3, pp. 334–334, 1997.

[13] 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.

[14] D. Dauner, M. Hallgarten, A. Geiger, and K. Chitta, "Parting with misconceptions about learning-based vehicle motion planning," in *Conference on Robot Learning*. PMLR, 2023, pp. 1268–1281.

[15] 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.

[16] M. Hallgarten, M. Stoll, and A. Zell, "From prediction to planning with goal conditioned lane graph traversals," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2023, pp. 951–958.

[17] 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.

[18] Z. Huang, H. Liu, and C. Lv, "Gameformer: Game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3903–3913.

[19] J. Cheng, Y. Chen, and Q. Chen, "Pluto: Pushing the limit of imitation learning-based planning for autonomous driving," *arXiv preprint arXiv:2404.14327*, 2024.

[20] 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.

[21] Y. Lu, J. Fu, G. Tucker, X. Pan, E. Bronstein, R. Roelofs, B. Sapp, B. White, A. Faust, S. Whiteson *et al.*, "Imitation is not enough: Robustifying imitation with reinforcement learning for challenging driving scenarios," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 7553–7560.

[22] Z. Zhang, R. Yang, K. Wu, Z. Xu, J. Liu, L. Mu, Z. Gan, and W. Ding, "Drive in corridors: Enhancing the safety of end-to-end autonomous driving via corridor learning and planning," *IEEE Robotics and Automation Letters*, 2025.

[23] 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.

[24] 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.