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

Jiyun Kim[1] and Kyunghwan Choi[2*]

Abstract— **Imitation Learning (IL) has been widely applied to autonomous driving, but still suffers from shortcut learning, where policies rely on spurious correlations rather than causal driving behavior. This limits generalization and reduces robustness, defined here as the ability to sustain safe and reliable performance under distribution shifts and out-of-distribution (OOD) scenarios. We propose CAR Planner, which mitigates shortcut learning by enhancing the attention mechanism, widely adopted in recent planners, with a constrained optimization formulation. Specifically, we impose an inequality constraint on the mean deviation of ego-state cross-attention weights from a uniform distribution, and solve it using the Augmented Lagrangian Method (ALM). This regularizer discourages over-reliance on a few channels and promotes balanced state representations, with negligible training overhead and no inference-time cost. On the nuPlan benchmark, CAR Planner exhibits substantially less degradation when ego-state channels are reduced, providing strong evidence of robustness against shortcut reliance. Furthermore, it consistently outperforms both the baseline (without constraints) and state-of-the-art dropout-based methods in challenging scenarios, while also producing smoother and more stable driving. These results highlight the effectiveness of CAR Planner in enabling robust imitation learning for autonomous driving.**

I. INTRODUCTION

Achieving human-like motion planning that remains robust across diverse and complex real-world scenarios is a central challenge in autonomous driving [1], [2]. Among data-driven approaches, Imitation Learning (IL) has emerged as a popular paradigm for training driving policies by mimicking expert demonstrations [3], [4], [5] . Despite its simplicity and effectiveness, IL faces two well-known challenges: compounding error and shortcut learning. Compounding error occurs when small deviations from expert behavior accumulate over time, causing cascading failures [6]. This problem is typically alleviated by data augmentation [7], [8], for example through noise injection or domain randomization. More critically, shortcut learning arises when the policy exploits spurious correlations in the training data rather than acquiring causal driving behavior [9], [10], [11]. Shortcut learning substantially limits robustness, here defined as the planner's ability to sustain safe and reliable performance under distribution shifts and out-of-distribution (OOD) scenarios.

In the nuPlan benchmark [12], shortcut learning is particularly evident in ego-state representations. A planner may overfit to a subset of ego features, such as velocity or steering angle, while neglecting other relevant states. This behavior can yield deceptively high open-loop scores but leads to severe degradation in closed-loop evaluations, especially in challenging test splits (e.g., test14-hard) and reactive closed-loop simulations (R-CLS). A common manifestation of this problem is attention collapse, where cross-attention weights concentrate excessively on a few ego-state channels. While such collapse can appear effective in nominal cases, it undermines robustness under distribution shifts and out-of-distribution scenarios. Prior attempts to address this issue include heuristic penalty-based regularization or the State Dropout Encoder (SDE) [13], which stochastically masks ego-state channels. However, penalty terms often interfere with the task loss and require careful tuning, while SDE discards part of the input and therefore cannot guarantee consistent performance in challenging conditions.

To address these limitations, we propose CAR Planner, a constrained-attention-based robust imitation learning framework. CAR Planner formulates shortcut mitigation as a constrained optimization problem: an inequality constraint is imposed on the mean deviation of ego-state cross-attention weights from a uniform distribution, ensuring balanced feature utilization. The constraint is enforced during training using the Augmented Lagrangian Method (ALM) [14], [15], a classical technique for constrained optimization. This design prevents attention collapse without sacrificing input information, incurs negligible training overhead, and preserves inference-time efficiency. The effectiveness of CAR Planner is demonstrated on the nuPlan benchmark [12], which provides large-scale real-world driving scenarios across multiple cities and supports both open-loop and closed-loop evaluations, including random and hard splits that probe distribution shift. Following standard protocol, we report performance on the open-loop score (OLS), the non-reactive closed-loop score (NR-CLS), and the reactive closed-loop score (R-CLS).

The main contributions of this work are as follows:

- **ALM-constrained ego-state attention:** To mitigate shortcut learning, we prevent attention collapse by formulating dispersion as a constrained optimization problem. Unlike auxiliary-loss-based regularization, which interferes with the task loss and requires careful tuning, our approach introduces a direct constraint that avoids these issues. Furthermore, unlike the SDE [13], which discards part of the input through stochastic masking, our method leverages all input information, leading to

*This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2025-00554087).

[1] Jiyun Kim is with the Artificial Intelligence Graduate School, Gwangju Institute of Science and Technology, Gwangju, Republic of Korea jiyun6606@gm.gist.ac.kr

[2] Kyunghwan Choi is with the Cho Chun Shik Graduate School of Mobility, Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea kh.choi@kaist.ac.kr

more consistent robustness under distribution shifts.

- **Lightweight and architecture-preserving design:** The constraint is applied only to the ego-state aggregation module, leaving the rest of the planner unchanged. No auxiliary losses or additional data are required, and inference cost remains identical to the base model.
- **Principled and tunable mechanism:** The formulation provides two interpretable parameters—the dispersion margin m and penalty coefficient ρ—which set the target spread of attention weights, while the Lagrange multiplier λ reflects constraint activity. This enables principled analysis as well as practical tuning.
- **Shortcut mitigation and robustness:** Experiments on the nuPlan benchmark demonstrate that the proposed method exhibits substantially less degradation under reduced ego-state inputs, evidencing robustness against shortcut reliance. Furthermore, it outperforms both the base model and SDE in challenging scenarios, while producing smoother and more stable driving.

II. RELATED WORK

Early autonomous driving systems were dominated by rule-based planners, which follow deterministic pipelines and handcrafted heuristics. Classical examples include IDM [16], a car-following model with longitudinal rules, and PDM-Closed [17], which combines curated logic with safety checks. These methods are interpretable and safe under nominal flow, but they are brittle under distribution shift and lack adaptability to novel scenarios.

In contrast, learning-based planners learn policies directly from data, either expert demonstrations or simulations. UrbanDriver [18] encodes surrounding agents and map context to generate multi-step ego trajectories, achieving strong closed-loop performance. GC-PGP [19] leverages goal-conditioned probabilistic generation for multi-modal trajectory prediction, while PDM-Open [17], incorporates learned scoring in an open-loop framework. Despite their success, learning-based planners are still prone to shortcut learning, where the model overfits to spurious correlations and shows degraded robustness under distribution shift.

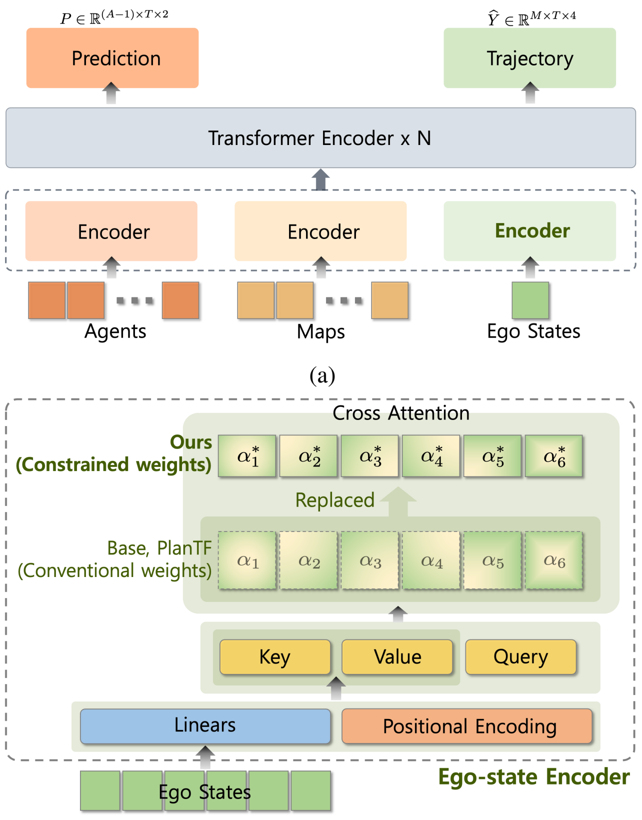
Recent work has turned to Transformer-based IL planners, which employ attention mechanisms to integrate ego-state, agent, and map features. Several approaches attempt to regularize attention to avoid over-reliance, such as heuristic penalty terms or auxiliary loss functions. planTF [13] represents a state-of-the-art baseline in this direction, introducing the SDE to mitigate shortcut learning by stochastically masking ego-state channels. This method has shown effectiveness in reducing over-reliance while preserving a pure IL formulation. However, because part of the input is intentionally discarded, its performance may be limited under challenging distribution shifts such as nuPlan's test14-hard split or other OOD scenarios. In contrast, our method retains all ego-state information and regulates attention dispersion through a constrained optimization framework, aiming to mitigate shortcut reliance without sacrificing input coverage.

Beyond pure IL, hybrid planners combine rule-based modules with learning-based policies. GameFormer [20] augments learning with game-theoretic reasoning to better capture multi-agent interactions, while PDM-Hybrid [17] integrates deterministic logic with learned scoring components. These approaches improve robustness but increase system complexity. Our work, by contrast, remains within the pure IL paradigm, showing that robustness can be enhanced by principled attention regularization without altering the planner architecture.

Finally, other approaches go beyond imitation learning to overcome IL's inherent limitations [6], [11], [21]. These include incorporating environment-aware auxiliary losses [4], [5], reinforcement learning [22], [23], and adversarial or closed-loop training strategies [24]. While such methods broaden the design space, our work focuses on pure IL-based planners, demonstrating that carefully designed attention regularization can substantially improve robustness.

III. CAR PLANNER

A. Planners Architecture



(a)

(b)

Fig. 1: Planner architecture of **Base** (vanilla), **PlanTF** (with SDE), **Ours: Car Planner** (with constrained attention weights): (a) Architecture overview and (b) ego state encoder.

All planners—**Base** (vanilla), **PlanTF** (SDE), and **Ours: CAR Planner** (ALM)— use an identical Transformer-based encoder-decoder architecture. The inputs are: (i) past states of all agents, (ii) the current 6-dimensional ego state, and (iii) a local map represented as lane polygons. Agent histories and map polygons are encoded by dedicated encoders into d-dimensional tokens. For each token, the model also adds a positional MLP that projects $[x, y, \cos\theta, \sin\theta]$ and is additively fused with the token features. The ego, agent, and map tokens are concatenated and processed by a shared Transformer encoder (4 layers, 8 heads) with key-padding masks to ignore invalid tokens. From the encoded ego token, a trajectory decoder outputs M candidate futures of length T with $(x, y, \cos\psi, \sin\psi)$ and their logits. At test time the model selects the most probable mode and recovers headings via $\operatorname{atan2}(\sin\psi, \cos\psi)$. In parallel, a shallow MLP predicts $T \times 2$ future positions for non-ego agents. All variants use the same state perturbation [7], [13] for data augmentation during training, and also identical objective functions.

Tokenization. (i) *Agent-history tokens* are built from agent kinematic sequences and embedded by its own encoder into d-dimensional vectors. (ii) *Ego-state token* summarizes the *current* ego state composed of K=6 scalar channels using a single learnable query that attends over the K channel embeddings. (iii) *Map-polygon tokens* encode local lane geometry and attributes with a dedicated polygon encoder, incorporating relevant spatial and semantic features.

Positional encoding. To preserve global geometry, for each token the model takes its last pose and for each map polygon its center pose, encodes them as $[x, y, \cos\theta, \sin\theta]$, and projects them through a small MLP. The resulting embeddings are additively fused with the corresponding token features.

Joint encoding. The concatenated sequence (ego, agents, map) is processed by a shared Transformer encoder with L_e=4 layers and h=8 heads per layer (pre-norm), using key-padding masks to ignore invalid tokens. This stage models agent–map interactions via quadratic self-attention over the active tokens.

Decoding. From the encoded ego token, a trajectory decoder predicts M candidate future trajectories of length T, each with L=4 channels per step $(x, y, \cos\psi, \sin\psi)$, along with mode logits. At test time, the most probable mode is selected and headings are recovered via $\operatorname{atan2}(\sin\psi, \cos\psi)$. In parallel, a shallow MLP predicts $T \times 2$ future positions for non-ego agents.

Training-time augmentation. All variants use the same state perturbation [7], [13] augmentation during training: we add small, bounded noise to the current ego state (and aligned history channels when applicable) without changing the inference graph.

Default configuration. The default configuration for all experiments use d=128, L=4, h=8, M=6, and T=80.

B. Objectives

The following objectives are identical for **Base**, **PlanTF (SDE)**, and **Ours: CAR Planner**.

Let $\widehat{Y} \in \mathbb{R}^{M \times T \times 4}$ denote ego futures with channels $(x, y, \cos\psi, \sin\psi)$ and $\pi \in \mathbb{R}^{M}$ the corresponding mode logits (pre-softmax). The ground-truth ego trajectory is $Y \in \mathbb{R}^{T \times 4}$, where the heading is represented by $(\cos\psi^{\text{gt}}, \sin\psi^{\text{gt}})$ for continuity. For non-ego agents, $P \in \mathbb{R}^{(A-1) \times T \times 2}$ and P^{gt} denote the predicted and ground-truth future positions, respectively. For training, a single supervised ego trajectory hypothesis $\widehat{\tau} \in \mathbb{R}^{T \times 4}$ is selected from $\widehat{Y}$ (selection rule described elsewhere).

Imitation loss. The imitation objective consists of two parts: (i) a regression term that matches the supervised ego hypothesis $\widehat{\tau}$ to the ground-truth trajectory Y in $(x, y, \cos\psi, \sin\psi)$ space, and (ii) a classification term that allocates probability mass to the label associated with this hypothesis. The regression uses $L1_{\text{smooth}}$ (Huber, δ=1) computed element-wise and averaged over valid steps/channels; representing the heading by $(\cos\psi, \sin\psi)$ avoids angle wrap-around. The classification applies cross-entropy to the mode logits $\pi \in \mathbb{R}^{M}$. Formally,

$$\mathscr{L}_{\text{reg}} = L1_{\text{smooth}}\big(\widehat{\tau}, Y\big), \tag{1}$$

$$\mathscr{L}_{\text{cls}} = \text{CE}\big(\pi, y^{\text{mode}}\big), \tag{2}$$

$$\mathscr{L}_{\text{i}} = \mathscr{L}_{\text{reg}} + \mathscr{L}_{\text{cls}}, \tag{3}$$

where $y^{\text{mode}} \in \{1, \ldots, M\}$ is the training label aligned with $\widehat{\tau}$ (its construction is described elsewhere).

Non-ego prediction loss. Encoded representations of non-ego agents are passed through a shallow two-layer MLP to produce single-modal future positions $P \in \mathbb{R}^{(A-1) \times T \times 2}$. This auxiliary objective supplies dense supervision across agents and time, encouraging scene-consistent plans and supporting downstream feasibility checks. The loss compares predictions to ground truth with smooth-L_1, averaged over valid agents and time steps (validity masks are applied via the index set $\mathscr{M}$ defined in $L1_{\text{smooth}}$):

$$\mathscr{L}_{\text{p}} = L1_{\text{smooth}}\big(P, P^{\text{gt}}\big). \tag{4}$$

Task objective. The training objective consists of two parts: the ego imitation loss, which fits the supervised ego trajectory and its mode score, and the non-ego prediction loss, which provides dense scene supervision and encourages the shared encoder to learn interaction-aware features. Reductions are averaged over valid time steps, and for the non-ego prediction loss, over valid agents as well. Unless otherwise specified, both components are given equal weight. The term $\mathscr{L}_{\text{task}}$ is used as the task objective in the augmented-Lagrangian formulation; the ALM constraint penalty and dual update are applied separately (see Section *ALM-based update rule*) and do not affect inference-time computation.

$$\mathscr{L}_{\text{task}} \;=\; \mathscr{L}_{\text{i}} \;+\; \mathscr{L}_{\text{p}}. \tag{5}$$

Definition of $L1_{\text{smooth}}$. The smooth-L_1 loss (Huber loss with threshold δ=1) is computed element-wise over the set of valid indices $\mathscr{M}$, which may include selected time steps, channels, or agents. The loss is normalized by $|\mathscr{M}|$ to ensure scale invariance with respect to prediction horizon and the number of valid elements. The quadratic region near zero provides stable gradients, while the linear region improves robustness to outliers:

$$L1_{\text{smooth}}(A,B) = \frac{1}{|\mathscr{M}|} \sum_{i \in \mathscr{M}} \begin{cases} \frac{1}{2}(A_i - B_i)^2, & \text{if } |A_i - B_i| < \delta, \\ |A_i - B_i| - \frac{\delta}{2}, & \text{otherwise.} \end{cases} \tag{6}$$

Notes. The threshold δ is adjustable, with a default value of δ=1. Representing the heading as $(\cos\psi, \sin\psi)$ helps avoid discontinuities and keeps residuals bounded, which improves numerical stability. If needed, channel-wise weights can be used to balance position and heading errors, but by default, all channels are equally weighted.

In conclusion, the base training objective consists of three parts: ego trajectory regression for the selected mode, mode classification, and non-ego agent regression. The loss is

$$\mathscr{L}_{\text{task}}(\theta) = L1_{\text{smooth}}\big(\widehat{Y}_{m^*}, Y\big) + \text{CE}\big(\pi, m^*\big) + L1_{\text{smooth}}\big(P, P^{\text{gt}}\big), \tag{7}$$

where $Y \in \mathbb{R}^{T\times 4}$ uses $(\cos\psi^{\text{gt}}, \sin\psi^{\text{gt}})$ for the heading, $\widehat{Y} \in \mathbb{R}^{M\times T\times 4}$ represents ego futures, and $\pi \in \mathbb{R}^{M}$ are the mode logits. For non-ego agents, $P, P^{\text{gt}} \in \mathbb{R}^{(A-1)\times T\times 2}$ denote the predicted and ground-truth positions.

The mode selection process identifies the best trajectory hypothesis by minimizing the average displacement error (ADE) over valid time steps:

$$m^* = \arg\min_{m} \ \text{ADE}_m, \tag{8}$$

$$\text{ADE}_m = \frac{1}{\sum_{t=1}^{T} V_t} \sum_{t=1}^{T} V_t \left\| \widehat{Y}^{(m)}_{t,1:2} - Y_{t,1:2} \right\|_2, \tag{9}$$

where $V_t \in \{0,1\}$ is a validity mask, and the selected trajectory is $\widehat{\tau} = \widehat{Y}_{m^*}$.

C. Problem formulation as a constrained optimization

Let $a_\theta^{(n)} \in \Delta^{C-1}$ denote the ego-state cross-attention over C=6 channels for the n-th sample (single learned query attending six channel embeddings). To suppress excessive peaking while allowing modest concentration, we measure the per-sample deviation from uniform as

$$d\big(a_\theta^{(n)}\big) \ = \ \frac{1}{C} \sum_{i=1}^{C} \left| a_{\theta,i}^{(n)} - \tfrac{1}{C} \right|. \tag{10}$$

The batch-averaged deviation is

$$D(\theta) \ = \ \frac{1}{N} \sum_{n=1}^{N} d\big(a_\theta^{(n)}\big), \tag{11}$$

and the inequality constraint is defined by

$$g(\theta) \ = \ D(\theta) - m \ \le \ 0, \qquad [z]_+ := \max(0,z), \tag{12}$$

where m>0 is a fixed margin (default m=0.12). Equivalently, in compact form,

$$g(\theta) \ = \ \frac{1}{NC} \sum_{n=1}^{N} \sum_{i=1}^{C} \left| a_{\theta,i}^{(n)} - \tfrac{1}{C} \right| \ - \ m \ \le \ 0, \tag{13}$$

which is algebraically identical to the definitions above. Note that for C=6 a one-hot attention yields $\frac{1}{C}\sum_{i=1}^{C} \left| a_i - \frac{1}{C} \right| = \frac{10}{36} \approx 0.278$, whereas the uniform distribution yields 0.

D. ALM-based update rule

The constrained optimization problem for Car Planner is solved using ALM with a single nonnegative lagrangian multiplier λ and penalty $\rho > 0$ (default ρ=3):

$$\mathscr{L}_{\text{aug}}(\theta, \lambda) = \mathscr{L}_{\text{task}}(\theta) + \lambda\, [g(\theta)]_+ + \frac{\rho}{2}\, [g(\theta)]_+^2. \tag{14}$$

At each iteration, parameters θ are updated by minimizing $\mathscr{L}_{\text{aug}}$, and the lagrangian multiplier is updated after the optimizer step using the violation:

$$\lambda \ \leftarrow \ \max\big\{0, \ \lambda + \rho\, [g(\theta)]_+\big\}. \tag{15}$$

The constraint is applied exclusively to the ego-state cross-attention weights. The shared Transformer encoder remains standard self-attention. Computation is negligible ($O(C)$ per batch for C=6), and inference is unchanged since neither the penalty nor the lagrangian multiplier update is used at test time.

Finally, the parameters θ are learned under an attention-dispersion constraint:

$$\min_{\theta} \ \mathscr{L}_{\text{task}}(\theta) \quad \text{subject to} \quad g(\theta) \le 0, \tag{16}$$

where $g(\theta)$ encodes the dispersion constraint and is enforced using the hinge-augmented ALM

IV. EXPERIMENTS

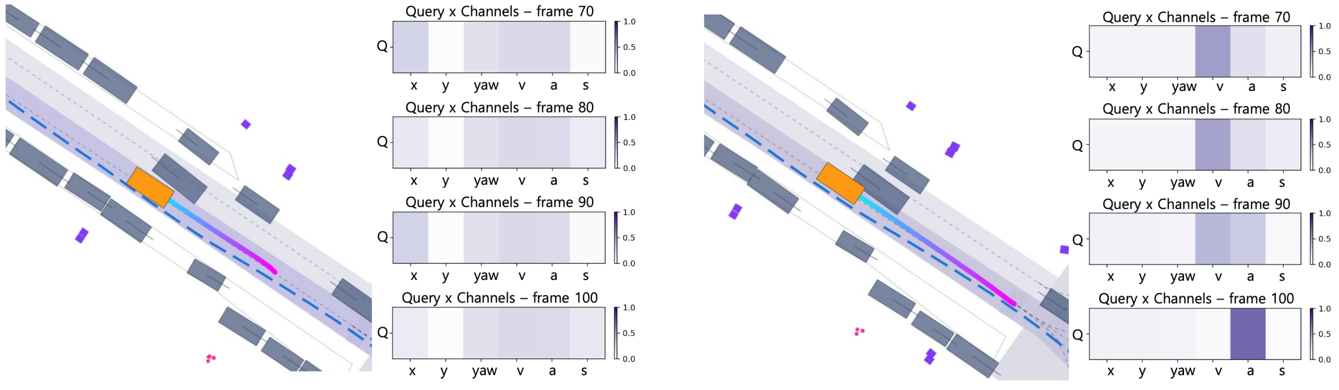
A. Experimental setup

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 is applied which adds bounded noise to the current ego state and and all coordinates are re-normalized with respect to the perturbed ego frame.

Evaluation follows the nuPlan benchmark. Results are reported on two official test splits, `test14-random` and `test14-hard`, each comprising 280 scenarios across 14 scenario types. `test14-random` consists of randomly sampled scenarios (fixed after selection), whereas `test14-hard` is a curated set of challenging scenarios selected by PDM-Closed [17]. Reported metrics are OLS (open-loop score against logged trajectories), NR-CLS (closed loop with nonreactive log-replay agents), and R-CLS (closed loop with reactive agents); higher is better for all metrics.

Here are the model variants used in the experiments:

- **Base (Vanilla).** Ego 6-D state is summarized by a small MLP (no attention), and there is no SDE or ALM. Agent/map encoders and the shared Transformer are identical to the other variants.
- **PlanTF (with SDE).** Same backbone, but the ego state uses a single-query *state attention* encoder with *stochastic state-channel dropout* (SDE) during training.
- **Ours — CAR Planner (with Constrained weights).** Same backbone with single-query ego-state attention, *regularized by an augmented-Lagrangian (ALM) dispersion constraint* on the 6-way attention weights. Inference-time computation is unchanged.



(a)

(b)

Fig. 2: **Ego-state attention comparison.** The scenario examples from nuPlan test14-hard split, R-CLS. (a) Car Planner: attention over (x,y,yaw,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.

B. *Experimental results*

Figure 2 shows two typical scenarios from the nuPlan test14-hard split, illustrating the difference in ego-state attention behavior between **PlanTF** (with SDE) and **CAR Planner** (with ALM). In both cases the model architecture is identical and only the ego-state attention module differs. PlanTF employs a single-query state attention trained with stochastic channel dropout but without dispersion control, and its attention progressively collapses onto a few channels (notably acceleration), diminishing sensitivity to lateral cues (x, y, yaw) and leading to unsafe maneuvers under dense traffic and occlusions. By contrast, CAR Planner applies an augmented-Lagrangian inequality to the 6-way attention, which maintains a dispersed yet selective pattern across frames, yielding balanced situational awareness and safe driving.

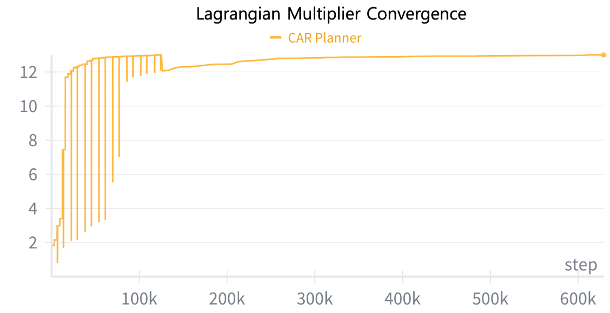
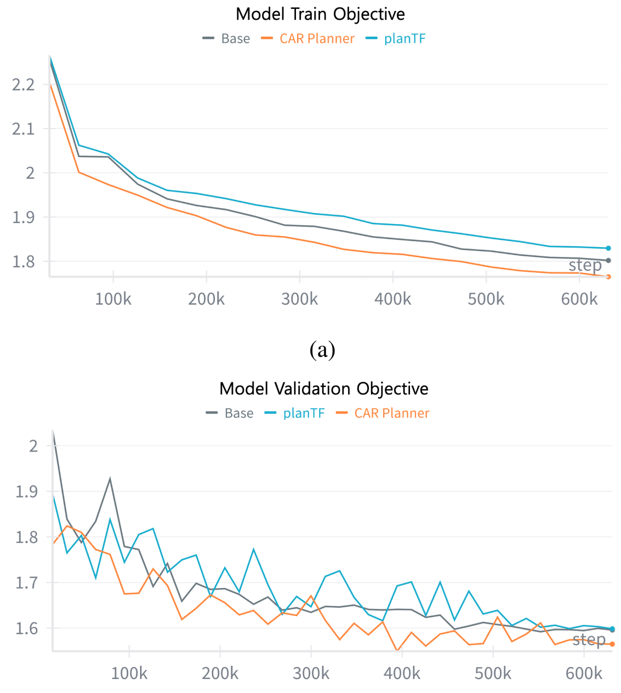


Fig. 3: Convergence of the Lagrange multiplier λ during training, showing stable behavior and confirming that the ALM mechanism enforces the constraint as intended.

The Figure 3 shows the stable convergence of the Lagrange multiplier λ during training across multiple runs, confirming that the ALM mechanism enforces the constraint as intended.

Figure 4 compares the objectives of the three planners during training. For Base and PlanTF the objective is $\mathscr{L}_{\text{task}}$, while for CAR Planner it is $\mathscr{L}_{\text{aug}} = \mathscr{L}_{\text{task}} + \lambda[g]_+ + \frac{\rho}{2}[g]_+^2$. CAR Planner stays lowest throughout and at convergence (train ≈ 1.78, val ≈ 1.57). Whereas PlanTF remains higher and shows larger fluctuations on validation, consistent with stochastic channel dropout. The consistent ordering between (a) and (b) indicates effective optimization and regularization by the ego-state attention constraint.



(a)

(b)

Fig. 4: Training (a) and validation (b) objectives versus training steps for Base, PlanTF, and CAR Planner. CAR Planner consistently achieves a lower objective.

Table I examines shortcut reliance by ablating ego-state channel. Rows `state5` and `state6` report scores with

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	84.91	78.31	86.31	69.48	64.64

TABLE I: Experimental results for evaluate solving shortcut learning on nuPlan test14-random and test14-hard splits.

five and six channels, respectively, and the gap is defined as gap $=$ state6 $-$ state5 (smaller is better). As shown in Figure 5, across both splits, **CAR Planner (ALM)** exhibits the smallest gaps on all metrics. On `test14-random`, the gaps are (+0.51, −0.21, +0.30) for (OLS, NR-CLS, R-CLS), and on `test14-hard` (+0.13, +0.86, +0.34). **PlanTF (SDE)** shows sensitivity on R-CLS (+6.64 random, +2.92 hard) and even a negative OLS gap on random (−1.54), indicating instability to the channel set. **Base** is moderately sensitive (e.g., NR-CLS +2.07 and R-CLS +0.81 on random; R-CLS +3.24 on hard). The consistently small gaps of CAR Planner indicate reduced dependence on any single state component and improved robustness to missing or perturbed inputs, aligning with the goal of mitigating shortcut learning.

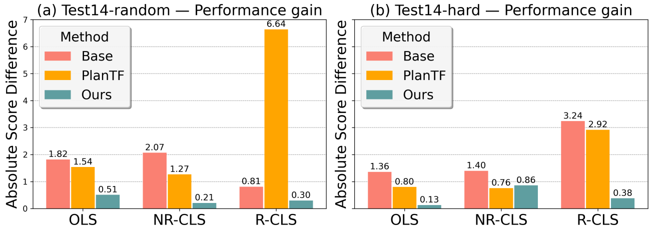


Fig. 5: Ego-state channel ablation gaps (state6 - state5) on nuPlan test14-random and test14-hard splits. Smaller gaps indicate reduced shortcut reliance.

Table II summarizes planner performance. CAR Planner is comparable to PlanTF on `test14-random`, and shows robust performance on `test14-hard`, most notably on the reactive closed-loop metric (R-CLS: 64.64 vs. 59.83, +4.81 pts) and on OLS (+2.97 pts). NR-CLS on `test14-hard` is similar (69.48 vs. 70.03). These results indicate stronger behavior in challenging scenarios while remaining competitive on nominal routes.

Tables III and IV detail closed-loop and open-loop submetrics on test14-hard. On R-CLS (Table III), CAR Planner ranks first on five of six metrics: Collisions (90.63), TTC (85.49), Drivable (94.02), Comfort (98.16), and Speed (98.22). PlanTF achieves the highest Progress (84.55), with CAR Planner a close second (84.28). This result shows improved safety and comfort without loss of efficiency. Open-loop results (Table IV) show CAR Planner leading all five metrics: ADE, FDE, AHE, FHE, and MR. The largest gains are on ADE and FDE compared to both baselines, reflecting more accurate trajectory prediction consistent with the superior closed-loop behavior.

Method	Test14-random			Test14-hard		
	OLS	NR-CLS	R-CLS	OLS	NR-CLS	R-CLS
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	84.91	78.31	86.31	69.48	64.64

TABLE II: Comparison of planner performance on nuPlan test14-random and test14-hard splits.

Method	Collisions	TTC	Drivable	Comfort	Progress	Speed
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
CAR Planner	**90.63**	**85.49**	**94.02**	**98.16**	84.28	**98.22**

TABLE III: Reactive Closed-loop sub-metric comparison on nuPlan test14-hard split. Higher is better.

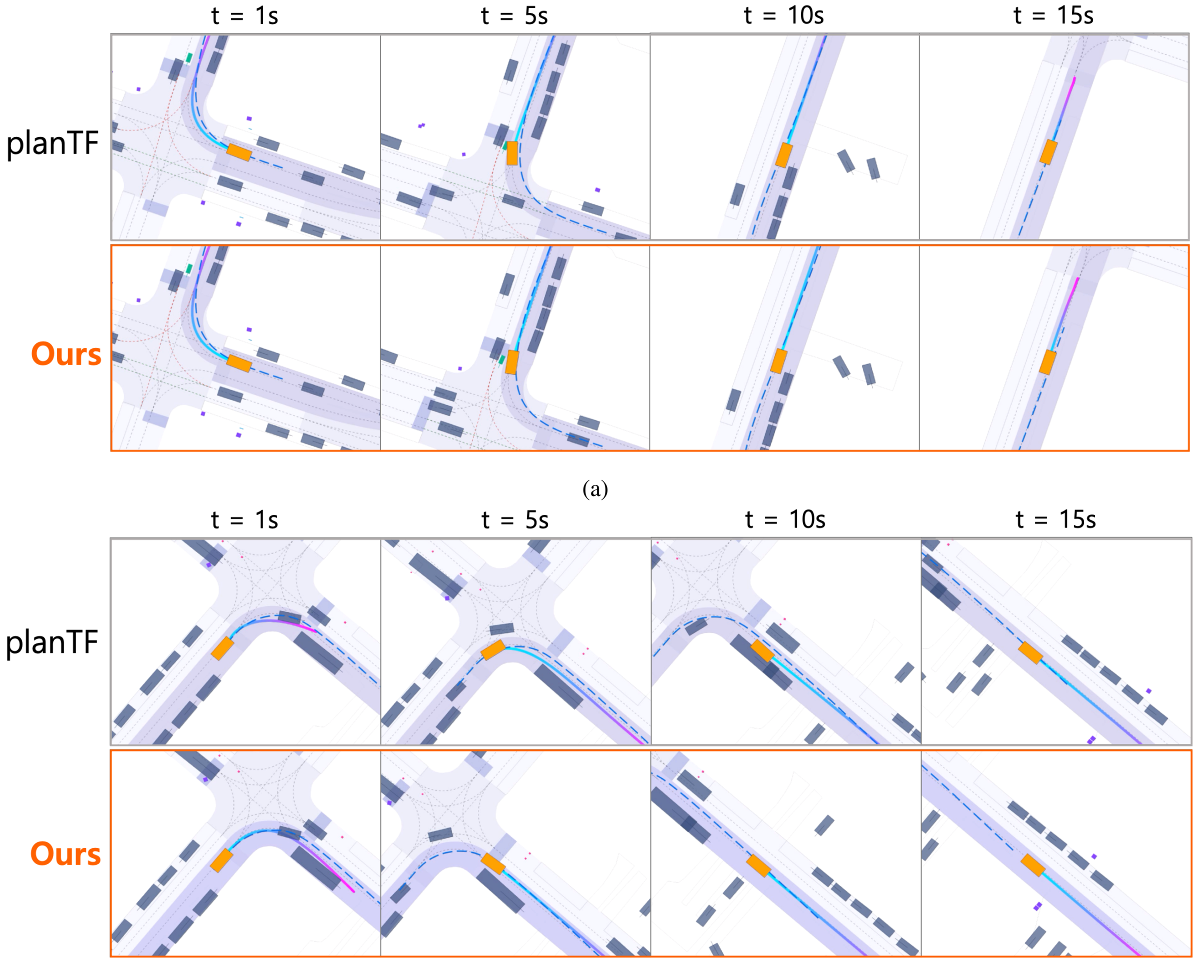
Method	ADE	FDE	AHE	FHE	MR
Base	79.65	61.34	93.91	91.05	95.95
planTF	81.93	64.31	93.43	90.64	96.32
CAR Planner	**84.99**	**67.81**	**94.11**	**91.55**	**97.06**

TABLE IV: Open-loop sub-metric comparison on nuPlan test14-hard split. Higher is better.

Figure 6 illustrates typical failure-recovery contrasts. In (a), PlanTF turns right and collides with a motorcycle waiting at the signal, whereas our policy safely avoids the obstacle. In (b), PlanTF clips a parked car during a right turn. Our method negotiates the corner without incident. These cases mirror the quantitative gains in R-CLS and comfort, emphasizing improved stability and safety in complex scenes.

V. CONCLUSION

This paper introduced CAR Planner, a constrained-attention-based framework to mitigate shortcut learning in



(a)

(b)

Fig. 6: The scenario examples from nuPlan test14-hard split. (a) The planTF collides with a motorcycle waiting at the signal while our policy safely avoids the obstacle. (b) The planTF clips a parked car during a right turn while our method negotiates the corner without incident.

imitation-learning planners. By formulating attention dispersion as a constrained optimization problem and enforcing it with an Augmented Lagrangian Method (ALM), the proposed approach prevents attention collapse without discarding input information or interfering with the task loss.

Experiments on the nuPlan benchmark demonstrated that CAR Planner achieves competitive or superior results across both random and hard splits. In particular, it substantially reduces degradation under reduced ego-state inputs and improves robustness in reactive closed-loop (R-CLS) evaluations, where shortcut reliance typically leads to failures. Submetric analysis further confirmed smoother and more stable driving, with gains in comfort and safety-related metrics. Compared against state-of-the-art baselines such as PlanTF with SDE, CAR Planner showed stronger generalization under distribution shifts and out-of-distribution (OOD) scenarios.

While effective, the current formulation is limited to ego-state cross-attention and has been validated only on the nuPlan benchmark. Extending the constraint to interaction and map-level features, and validating on broader datasets or real-world testbeds, therefore represents a natural next step to further strengthen robustness and scalability of imitation-learning planners.

Appendix

The results of **planTF** and **CAR Planner** are compared with the other models whose results are reported in [13].

Planners		**Test14-hard**		
Type	**Method**	OLS	NR-CLS	R-CLS
Learning-based	Urbandriver	76.90	51.54	49.07
	GC-PGP	73.78	43.22	39.63
	PDM-Open	79.06	33.51	35.83
	planTF	83.34	70.03	59.83
	CAR Planner	86.31	69.48	64.64

TABLE V: Comparison of planner performance on the nuPlan test14-hard split.

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] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[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] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

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

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.