Learning to Explain While Planning: Rule-Aligned Diffusion Planning for Autonomous Driving

Anonymous authors
Paper under double-blind review

Abstract

Diffusion planners exhibit strong capabilities in generating multimodal trajectories. However, existing methods primarily rely on expert demonstrations to fit trajectory distributions, learning statistical correlations among scenes, behaviors, and trajectories without explicitly modeling driving rules. In long-tail scenarios where expert data are scarce, the lack of behaviors to imitate may lead to trajectories that violate safety or compliance requirements. Moreover, their generation process lacks rule-level explanations, making it difficult to determine which rules drive trajectory adjustments, when they take effect, and how strongly they act, thereby limiting failure diagnosis, safety validation, and targeted improvement. To address these limitations, we propose the Rule-Aligned Diffusion Planner (RADP), which incorporates differentiable driving rules into the diffusion objective during training, turning rule knowledge into intrinsic behavioral principles beyond finite demonstrations. We further introduce Rule-Pressure Attribution (RPA), which constructs supervision signals from gradients of rule losses with respect to predicted trajectories and employs a lightweight attribution head to estimate the optimization pressure exerted by each rule online. To assess the closed-loop behavioral relevance of these attributions, we propose a temporal risk-alignment protocol that evaluates whether current rule pressures reflect corresponding risks during subsequent closed-loop execution. Experiments on nuPlan show that RADP improves closed-loop planning in challenging safety-critical scenarios, while RPA exhibits consistent temporal alignment with subsequent rule-specific risks, validating both intrinsic rule learning and rule-level interpretability.

1 Introduction

Diffusion models provide expressive representations of multimodal trajectory distributions and have achieved strong closed-loop planning performance. However, existing diffusion planners learn driving behavior mainly by imitating expert trajectories. Requirements such as collision avoidance, lane compliance, and speed-limit compliance are typically implicit in the data distribution rather than explicit training objectives. When expert demonstrations do not adequately cover complex interactions and long-tail states, a planner may learn superficial behavior patterns without acquiring transferable rule constraints. Although rewards or costs can guide diffusion sampling at inference time, such rules remain external optimization signals and do not directly shape the learned generative policy. Consequently, rule compliance remains decoupled from the model's learned generative capability, potentially leading to inconsistent behavior under distribution shifts and safety-critical situations insufficiently represented in the training data.

Diffusion planners also lack faithful rule-level explanations of their planning decisions. Existing interpretability techniques, including attention visualization and saliency mapping, are predominantly post-hoc: they analyze a decision after it has been produced and identify influential input features or internal activations. However, feature importance does not directly reveal the semantic objectives that drive trajectory optimization, nor does it quantify when and how strongly a particular driving rule acts on the generated trajectory. Such

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

post-hoc explanations are therefore weakly connected to the planner's actual optimization mechanism and cannot naturally provide rule-level explanations together with trajectory generation. In safety-critical autonomous driving, this limitation makes it difficult to determine why a behavior was selected, localize the causes of planning failures, validate safety properties, and perform targeted model improvement.

We therefore propose the Rule-Aligned Diffusion Planner (RADP), which jointly optimizes the diffusion objective and differentiable driving rules during training. The rules directly participate in learning the trajectory distribution, without requiring additional rule guidance or trajectory optimization at inference. Building on RADP, we introduce Rule-Pressure Attribution (RPA): the gradient magnitude of each rule cost with respect to the predicted trajectory represents its local optimization pressure and supervises a lightweight attribution head. The planner can thus produce a trajectory and its rule-level explanation within the same planning cycle.

To evaluate whether rule attributions reflect closed-loop driving conditions, we introduce a temporal risk-alignment protocol that associates each rule's current attribution with its corresponding risk over a future rollout window starting at the same planning frame. The protocol measures whether higher rule-specific risks are accompanied by higher attribution pressures. Experiments on nuPlan show that RADP maintains broadly comparable overall planning performance while improving collision and TTC metrics in challenging safety-critical scenarios. The attribution head approximates the gradient teacher, and its outputs exhibit stable temporal correspondence with subsequent rule-specific risks.

Our contributions are summarized as follows:

- We propose a rule-aligned diffusion planning framework that integrates differentiable driving rules into training, enabling rule-consistent planning without inference-time rule-based refinement.
- We introduce online rule-pressure attribution, which distills trajectory-gradient-based rule pressures into a lightweight head, providing efficient rule-aware explanations alongside trajectory generation.
- We develop a temporal risk-alignment protocol that relates rule pressures to corresponding future risks in closed-loop rollouts, providing empirical validation of the effectiveness and behavioral relevance of our attributions.

2 Related Work

Diffusion Planning. Diffusion models have become a strong framework for multimodal trajectory prediction and planning, extending denoising probabilistic modeling and Transformer denoisers (Ho et al., 2020; Peebles & Xie, 2023) to sequential decision-making and visuomotor policies (Ajay et al., 2023; Chi et al., 2023). MotionDiffuser supports controllable multi-agent sampling through differentiable costs (Jiang et al., 2023); Diffusion Planner uses a Transformer-based diffusion architecture for joint prediction and closed-loop planning (Zheng et al., 2025); and DiffusionDrive improves inference efficiency with multimodal anchors and a truncated diffusion schedule (Liao et al., 2025). Despite their planning performance, these methods learn behavior mainly from demonstrations and do not explicitly represent semantic rules for compliance reasoning or decision explanation.

Rule and Constraint Integration in Learned Planning. Beyond fitting expert trajectory distributions, prior work introduces additional objectives into generative planning to improve safety and controllability. Diffuser (Janner et al., 2022) guides sampling toward task-specific rewards; MotionDiffuser (Jiang et al., 2023) uses differentiable costs that encode physical priors and user preferences; Diffusion-ES (Yang et al., 2024) combines diffusion sampling with evolutionary search to support black-box and non-differentiable objectives; and guided conditional diffusion (Zhong et al., 2023) applies semantic constraints, including traffic rules, during denoising for controllable traffic simulation. SafeDiffuser embeds control-barrier conditions into denoising to enforce safety specifications (Ames et al., 2017; Xiao et al., 2025), while diffusion predictive control imposes explicit state and action constraints on sampled

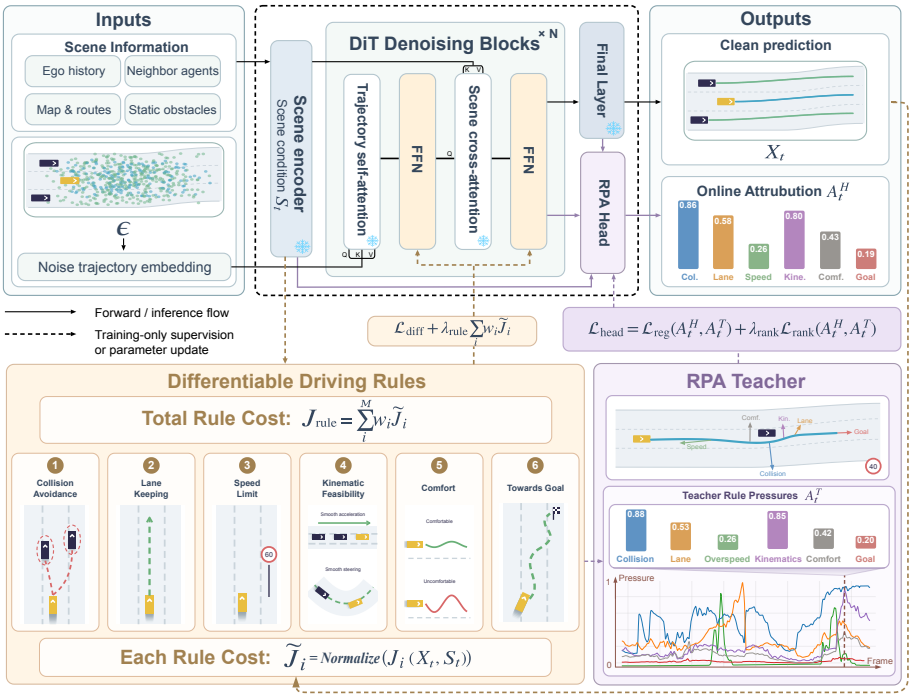


Figure 1: Overview of RADP.

controls (Römer et al., 2025); reward-based diffusion policy learning instead connects policy scores to action-value gradients (Psenka et al., 2024). Despite different formulations, these objectives primarily constrain or guide sampling after the diffusion model has been learned, rather than making semantic driving rules part of policy training.

Explainable Diffusion Planning. Existing explainability methods use gradients, saliency, and attention propagation to identify factors underlying neural-network outputs (Simonyan et al., 2013; Sundararajan et al., 2017) and related visual explanations (Selvaraju et al., 2017; Abnar & Zuidema, 2020; Arrieta et al., 2020); work on diffusion models further studies cross-attention, the denoising process, and training-data contribution (Tang et al., 2023; Park et al., 2024; Dai & Gifford, 2023). Driving-specific approaches expose interpretable cost volumes and intermediate predictions (Zeng et al., 2019), generate textual rationales grounded by visual attention (Kim et al., 2018), or rank nearby agents using interaction attention (Hazard et al., 2022). Interpretable trajectory representations (Ivanovic et al., 2021), action-oriented features (Xiao et al., 2021), and language aligned with intermediate driving outputs (Ding et al., 2025) provide complementary forms of transparency. Together with PlanT's object-level attention (Renz et al., 2023) and InterFuser's semantic intermediate features (Shao et al., 2023), these methods explain input objects, representations, or visual evidence. They do not directly quantify the corrective pressure exerted by individual driving rules. We instead use gradients of differentiable rules with respect to the trajectory as attribution supervision for online rule-pressure estimation.

3 Method

3.1 Problem Setup and Method Overview

Given the scene observation S_t at planning time t, RADP produces a future ego trajectory together with its online rule-level attribution within the same planning cycle, as illustrated in Figure 1. The observation contains the histories of the ego vehicle and other traffic participants, map elements, route information, and static obstacles; ϵ denotes the diffusion noise. The figure further summarizes how differentiable rules first adapt the planner and subsequently provide gradient-based supervision for the attribution head, with these supervision

paths used only during training. We write the overall input–output relation as

$$(X_t, A_t^H) = \mathcal{F}_{\theta,\phi}(S_t; \epsilon), \qquad X_t \in \mathbb{R}^{H\times d}, \quad A_t^H \in \mathbb{R}^M_{\geq 0}, \tag{1}$$

where $X_t = P_\theta(S_t; \epsilon)$ is the predicted ego trajectory over a horizon of H steps, and A_t^H is the non-negative rule-pressure vector produced by the online attribution head f_ϕ. We use $M = 6$ channels: physical collision avoidance, lane-boundary compliance, speed-limit compliance, kinematic feasibility, comfort, and goal progress.

The method is trained in two stages. Section 3.2 defines the differentiable rule costs

$$\mathcal{J} = \{J_i(X_t, S_t)\}_{i=1}^M, \tag{2}$$

and Section 3.3 incorporates them into the diffusion objective for lightweight planner adaptation. With the adapted planner frozen, Section 3.4 constructs gradient-based rule-pressure targets, and Section 3.5 distills them into an online attribution head. At inference, only the planner and attribution head are executed, without teacher gradients, rule guidance, or trajectory optimization.

3.2 Differentiable Driving Rules

All rules operate on trajectories in physical coordinates at 0.1 s resolution and are differentiable with respect to the predicted ego trajectory. For rule i, we first define a trajectory-level cost $J_i(X_t, S_t)$ that measures the overall degree to which trajectory X_t violates that rule in scene S_t:

$$J_i(X_t, S_t) = \mathrm{Agg}_{i,h,n}[m_{i,h,n}\, \varphi_{\sigma_i}(z_{i,h,n}(X_t, S_t))]\,. \tag{3}$$

Equation equation 3 compactly maps local violations to a trajectory-level cost. Reading it from the inside out, $z_{i,h,n}(X_t, S_t)$ is the local physical violation of rule i at future step h with respect to interaction entity n. The index n is used only for interactive rules such as collision avoidance and is omitted for rules that depend solely on the ego state. We use the convention $z_{i,h,n} > 0$ for a violation and $z_{i,h,n} \leq 0$ for rule satisfaction or a remaining safety margin. For example, collision uses the required safety distance minus the actual box separation, whereas speed-limit compliance uses the predicted speed minus the permitted speed.

To convert this signed, physically dimensioned quantity into a non-negative and differentiable local penalty, we use

$$\varphi_{\sigma_i}(z) = \left[\frac{1}{\beta}\,\mathrm{softplus}\left(\frac{\beta z}{\sigma_i}\right)\right]^2, \qquad \beta = 10. \tag{4}$$

Here σ_i is the physical scale of rule i. The function approximates $[\max(0, z/\sigma_i)]^2$: its value is close to zero when the rule is satisfied and increases smoothly with violation severity. In Eq. equation 3, $m_{i,h,n}$ is an applicability or risk gate that masks invalid or irrelevant local constraints, while Agg_i aggregates the valid time steps and interaction entities into the scalar J_i. We instantiate six semantic channels covering collision, lane, speed, kinematics, comfort, and goal progress, as summarized in Table 1. Their complete definitions are provided in Appendix B.

Because the resulting rule costs have different typical numerical ranges, we normalize them before combining them in the planner objective. For sample b,

$$\widetilde{J}_i^{(b)} = \frac{J_i^{(b)}}{s_i^{(r)}}, \tag{5}$$

Here $s_i^{(r)}$ is a stop-gradient EMA scale maintained separately for rule i during training (EMA decay 0.99, lower bounded by 10^{-3}). This normalization balances complete rule costs in joint training without changing their physical definitions. The attribution teacher uses the gradient of the unnormalized J_i.

As an example, collision avoidance uses oriented ego boxes constructed from the predicted rear-axle states and the nuPlan Pacifica geometry. Let $d_{h,n}^{\mathrm{box}}$ be the signed box separation

Table 1: Six rule channels for planning and attribution.

Rule channel	Violation signal	Purpose
Collision	Insufficient ego–agent clearance	Avoid imminent collisions
Lane	Route-corridor boundary violation	Maintain lane and route compliance
Speed	Speed above the legal limit	Enforce speed-limit compliance
Kinematics	Excess acceleration, braking, or steering	Ensure dynamic feasibility
Comfort	Excess jerk or control variation	Encourage smooth motion
Goal	Insufficient route progress	Preserve planning efficiency

computed by the separating-axis theorem. The collision violation is

$$z_{h,n}^{\mathrm{col}} = d_{h,n}^{\mathrm{safe}} - d_{h,n}^{\mathrm{box}}, \qquad d_{h,n}^{\mathrm{safe}} = d_0 + t_{\mathrm{head}} c_{h,n} + \frac{c_{h,n}^2}{2a_{\mathrm{brake}}}, \tag{6}$$

where $c_{h,n} \geq 0$ is the closing speed and $d_0 = 0.5\,\mathrm{m}$. Thus, $z_{h,n}^{\mathrm{col}} > 0$ indicates insufficient clearance. We use $\sigma_{\mathrm{col}} = 0.5\,\mathrm{m}$ and a detached gate based on distance, closing speed, TTC, and violation severity to suppress irrelevant interactions. The resulting penalties are aggregated into J_{col}, with other-agent trajectories treated as fixed context.

3.3 Rule-Aligned Planner Training

We build on Diffusion Planner (Zheng et al., 2025), which models the joint future trajectories of the ego vehicle and surrounding agents with an x_0-prediction objective. We retain its masked ego–neighbor reconstruction loss $\mathcal{L}_{\mathrm{diff}}$ and augment it with differentiable rule costs (full baseline formulation in Appendix A):

$$\mathcal{L}_{\mathrm{RADP}} = \mathcal{L}_{\mathrm{diff}} + \lambda_{\mathrm{rule}} \sum_{i=1}^{M} w_i \widetilde{J}_i, \tag{7}$$

where $M = 6$, $\widetilde{J}_i$ is the normalized cost of rule i, and w_i controls its contribution to planner training. We set $\lambda_{\mathrm{rule}} = 0.004$ and use $(w_{\mathrm{col}}, w_{\mathrm{lane}}, w_{\mathrm{speed}}, w_{\mathrm{kin}}, w_{\mathrm{comfort}}, w_{\mathrm{goal}}) = (2.5, 0.8, 0.8, 0.5, 0.8, 1.5)$. Starting from the baseline checkpoint, we fine-tune lightweight LoRA adapters in the decoder (Hu et al., 2022).

3.4 Gradient-Pressure Teacher

Rule-gradient teacher. For a generated trajectory X_t, the gradient of a differentiable rule cost indicates the first-order trajectory change required to reduce that cost. We therefore define the teacher pressure for rule i as the root-mean-square magnitude of its trajectory gradient:

$$g_i^T(t) = \left(\frac{1}{D} \left\| \nabla_{X_t} J_i(X_t, S_t) \right\|_2^2 \right)^{1/2}, \qquad D = Hd. \tag{8}$$

The resulting $g_i^T(t)$ is an independent, non-negative measure of the trajectory's sensitivity to rule i: a larger value means that this rule exerts stronger local pressure on the current plan.

Scale-calibrated attribution. Gradient magnitudes can differ systematically across rules. We estimate a positive calibration scale κ_i for each rule from the training split and define the teacher attribution as

$$A_i^T(t) = \log\left(1 + \frac{g_i^T(t)}{\kappa_i + \varepsilon}\right). \tag{9}$$

The logarithm attenuates rare extreme gradients while preserving their order. The resulting $A_i^T(t)$ is a calibrated, rule-specific measure of local trajectory sensitivity, which we use as supervision for the online attribution head.

3.5 Online Attribution Head

Computing Eq. equation 9 for all rules requires repeated backward passes. We amortize this computation with an attribution head f_ϕ. Its base input combines the ego token from the

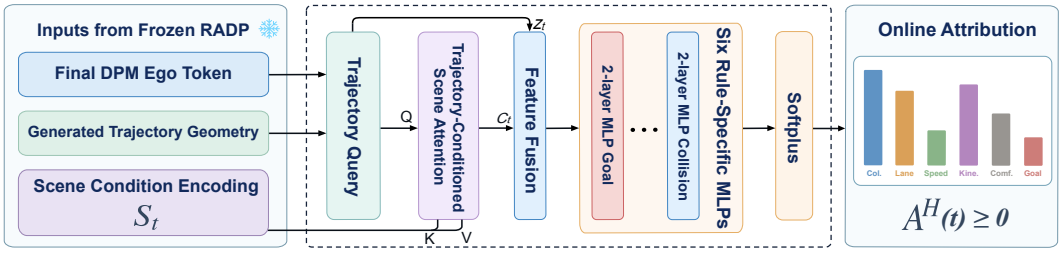


Figure 2: Online rule-pressure attribution head.

final DPM denoising call with geometric features of the generated trajectory:

$$z_t = [h_t^{\mathrm{DPM}}; q(X_t)]. \tag{10}$$

The head also receives the scene condition encoding S_t, which jointly contains surrounding-agent, map, route, and lane information. Using z_t as a query, it attends to the scene encoding and aggregates the context relevant to the current generated trajectory:

$$c_t = \mathrm{MHA}_{\mathrm{scene}}(z_t, S_t, S_t), \qquad A_i^H(t) = \mathrm{Softplus}(f_{\phi,i}([z_t; c_t])) \,. \tag{11}$$

Invalid tokens in the scene condition encoding are masked in attention. We instantiate one independent two-layer MLP for each rule. All six MLPs receive the shared representation $[z_t; c_t]$, but their output parameters are not shared, producing independent non-negative pressures. The scene encoding lets the head combine the current trajectory with agent interactions and road and route constraints, rather than relying only on a compressed ego feature. This computation is summarized in Figure 2.

The planner parameters θ are frozen, and only ϕ is trained on cached teacher targets. Our training objective combines weighted pressure regression with a ranking term:

$$\mathcal{L}_{\mathrm{head}} = \mathcal{L}_{\mathrm{reg}}(A^H, A^T) + \lambda_{\mathrm{rank}} \mathcal{L}_{\mathrm{rank}}. \tag{12}$$

We use a weighted Smooth-L_1 regression loss and a within-mini-batch pairwise ranking loss, with additional weight on safety-critical rules and high-pressure Teacher targets. The head is trained solely on cached gradient-teacher attributions. At inference, it reuses planner features in the same forward pass and does not alter the generated trajectory; closed-loop rollout risks are reserved exclusively for evaluating attribution quality.

4 Experiments

4.1 Experimental Setup

We evaluate RADP from three complementary perspectives. First, we assess closed-loop planning performance on the nuPlan val14, test14-random, and test14-hard reactive splits (Caesar et al., 2021; Karnchanachari et al., 2024). We compare RADP with Diffusion Planner across the overall score and all reported metric components, and further compare the overall score with representative learning-based planners. Non-reactive results are retained in Table 9 in the appendix. Second, on synchronized closed-loop planning frames, we evaluate how faithfully the online attribution head approximates the gradient-based Teacher. Third, we examine whether each rule-specific attribution is aligned with the corresponding driving condition observed during closed-loop execution. We evaluate six attribution channels against seven future-risk endpoints: each channel has one matched endpoint, while the collision channel is additionally evaluated against TTC risk as a complementary measure of imminent safety. Thus, TTC is an auxiliary evaluation endpoint rather than a seventh Head output. Detailed risk endpoints and statistical procedures are provided in Appendix B.

4.2 Planner Performance

Table 2: Closed-loop planning performance with reactive agents.

Split	Planner	Score	Collision	TTC	Drivable	Progress	Speed	Comfort
val14	Diffusion Planner	82.70	92.98	87.84	97.85	96.60	98.29	89.45
	RADP(ours)	81.37	94.01	89.53	98.03	94.90	98.87	89.27
test14-random	Diffusion Planner	82.82	94.06	89.66	98.08	97.70	97.07	85.44
	RADP(ours)	84.48	96.36	91.95	98.08	96.93	97.98	87.36
test14-hard	Diffusion Planner	68.94	86.95	79.78	94.85	92.65	97.39	85.29
	RADP(ours)	70.62	92.10	83.46	95.59	90.07	98.29	85.29

As reported in Table 2, RADP improves the reactive score and safety metrics on both test splits. On val14-r, safety also improves, but lower progress reduces the aggregate score, revealing a safety–efficiency trade-off. The broader comparison in Table 3 further shows that RADP achieves the highest reported scores on test14-random and test14-hard while remaining competitive on val14.

Table 3: Literature-reported reactive nuPlan scores.

Type	Planner	Val14-R	Hard-R	Random-R
Expert	Log-replay	80.32	68.80	75.86
Learned	PDM-Open	54.24	35.83	57.23
	UrbanDriver	64.11	49.95	67.15
	GameFormer	8.69	6.69	9.31
	PlanTF	76.95	61.61	79.58
	PLUTO	78.11	59.74	78.62
	Diffusion Planner	82.70	68.94	82.82
	RADP (ours)	81.37	70.62	84.48

4.3 Attribution Evaluation

We evaluate two complementary properties. On the same closed-loop planning frame, scene state, and generated trajectory, macro Spearman, MAE, and Top-1 agreement respectively measure whether the Head preserves the Teacher's pressure ordering, numerical values, and dominant-rule decision. Complete definitions are provided in Appendix B.5.

Attribution $A_i(t_k)$ describes the pressure exerted by rule i on the trajectory generated at planning time t_k. To determine whether this pressure corresponds to actual driving conditions, we construct future-risk targets from the same closed-loop rollout. Let $v_i^{\text{rollout}}(\tau)$ denote the instantaneous severity of rule-i risk observed at execution time τ. The future risk associated with the current planning frame is

$$R_i(t_k) = \text{Agg}_{\tau \in [t_k, t_k + H_i]} \, v_i^{\text{rollout}}(\tau), \tag{13}$$

where H_i is a rule-specific horizon and Agg reduces frame-wise severity within the window to a continuous risk value. Collision and TTC use a two-second window because they describe imminent hazards. Lane boundary, speed limit, kinematics, comfort, and goal progress use an eight-second window. Rule-specific severity functions and aggregation operators are fixed, together with all horizons and thresholds, before inspecting attribution results and are shared by Teacher and Head. Hence, $R_i(t_k)$ is an independent closed-loop evaluation target, not a training label.

We then test behavioral validity against $R_i(t_k)$. Scenario-wise temporal Spearman measures whether attribution and subsequent rule-matched risk rise and fall consistently; AUPRC evaluates retrieval of future high-risk frames across thresholds; and Lift@10% measures risk-event enrichment among the highest- attribution 10% of frames. Within-scenario temporal shuffling serves as a control. Complete thresholds, eligibility criteria, formulas, and bootstrap procedures are provided in Appendix B.6.

4.3.1 Closed-Loop Head–Teacher Fidelity

On synchronized reactive frames, Table 4 shows that the Head consistently tracks the Teacher in pressure ranking, absolute magnitude, and dominant-rule identification, while both remain positively aligned with subsequent rule-matched risks.

Table 4: Head–Teacher fidelity and future-risk alignment in reactive closed-loop simulation.

Reactive split	Head–Teacher fidelity			Future-risk alignment	
	Macro Spearman ↑	MAE ↓	Top-1 ↑	Teacher ρ_{macro} ↑	Head ρ_{macro} ↑
val14	0.536	0.216	70.6%	0.235	0.207
test14-random	0.570	0.200	71.7%	0.199	0.177
test14-hard	0.554	0.238	71.9%	0.221	0.192

4.3.2 Temporal Risk Alignment

Table 5 details the alignment of Teacher and Head pressures with matched driving conditions on val14-r, where collision pressure is evaluated against both physical contact and auxiliary TTC risk. Complete confidence intervals are provided in the appendix.

Table 5: Frame-level future-risk alignment on val14-r.

Channel	Risk endpoint	Pos. rate	AUPRC ↑		Lift@10% ↑		Temporal ρ ↑		Shuffled ρ	
			Teacher	Head	Teacher	Head	Teacher	Head	Teacher	Head
Collision	Physical collision	2.37%	0.540	0.514	1.823	1.598	0.234	0.160	0.001	−0.002
	TTC risk (aux.)	4.11%	0.555	0.489	1.926	1.610	0.237	0.158	−0.000	−0.000
Lane	Lane-boundary risk	49.31%	0.911	0.840	1.518	1.340	0.314	0.234	0.001	0.000
Speed	Overspeed risk	6.87%	0.779	0.761	1.890	1.895	0.303	0.297	−0.001	−0.001
Kinematics	Kinematic risk	0.013%	0.167	0.250	0.000	0.000	0.112	0.159	−0.000	0.000
Comfort	Comfort risk	0.102%	0.376	0.371	1.750	1.750	0.102	0.122	0.000	−0.000
Goal progress	Route-progress deficit	47.06%	0.850	0.798	1.804	1.626	0.343	0.316	0.005	0.027

For the five endpoints with sufficient events, Head AUPRC clearly exceeds the positive-rate baseline and Lift@10% indicates risk enrichment among high-pressure frames. Specifically, AUPRC should be compared with the positive event rate, Lift@10% with 1, and temporal correlation with 0. Across rule categories, the observed positive temporal correlations, together with near-zero correlations after temporal shuffling, further support a meaningful temporal association between predicted rule pressures and subsequent rule-specific risks.

4.4 Efficiency

As reported in Table 6, adding the attribution Head leaves end-to-end planning latency essentially unchanged: planner-only and planner-plus-Head inference require 251.05 and 245.50 ms per step respectively. Thus, online attribution introduces no measurable end-to-end overhead in this evaluation.

Table 6: Mean inference latency on an RTX 4080.

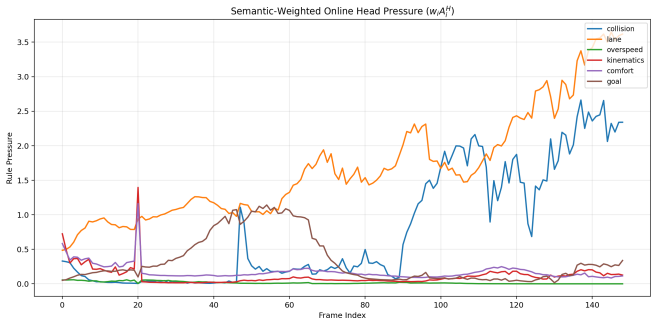
Method	Latency (ms)	Overhead
Planner only	251.05	—
Planner + Head	245.50	−1.64%
Planner + exact Teacher	465.29	+89.18%
Planner + gradient guidance	571.10	+130.76%
Exact Teacher on cached states	62.51	1.00×
Head on cached states	8.18	7.65× faster

In contrast, computing the exact Teacher online increases latency to 465.29 ms, and inference-time collision-gradient guidance requires 571.10 ms. The cached-state comparison excludes trajectory generation and measures only the additional attribution computation on the same stored planner states. Under this setting, the Head requires 8.18 ms compared with 62.51 ms for the exact Teacher, yielding a 7.65× attribution speedup.

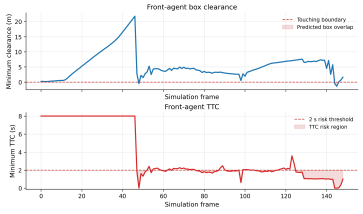
4.5 Qualitative Analysis

Left-turn scenario. The rollout contains two collision-critical interactions near frames 100 and 140, where front-agent clearance and TTC decrease (Figure 3(b)); after frame 80, the signed route-corridor clearance also falls and becomes negative (Figure 3(c)). Given these observed risks, the Head correctly raises collision pressure around both interactions and increases lane pressure as the boundary is approached, while keeping speed, kinematic, and comfort pressures comparatively low (Figure 3(a)). The matched snapshots further

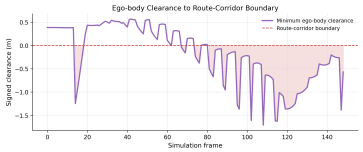
show that RADP maintains greater separation than the baseline (Figure 4), providing a qualitative counterpart to Table 4 and Table 5.



(a) online rule pressure A_i^H

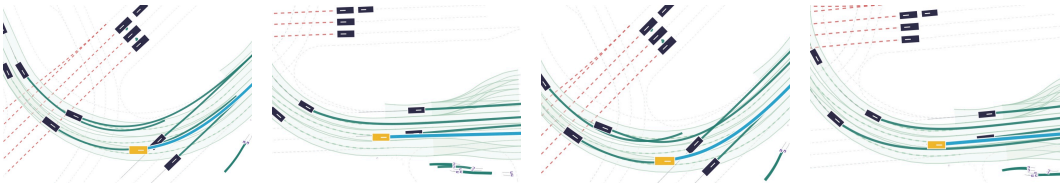


(b) Front-agent clearance and TTC



(c) Signed route-corridor clearance

Figure 3: A starting_left_turn case from test14-hard-r. (a) shows the Head's frame-wise rule pressures. (b) reports front-agent clearance and TTC. (c) shows signed ego-body clearance to the route-corridor boundary.



(a) Baseline, frame 100 (b) Baseline, frame 140 (c) RADP, frame 100 (d) RADP, frame 140

Figure 4: Matched closed-loop views. Baseline and RADP are shown near the two critical interactions; yellow and blue denote the ego vehicle and neighbors.

4.6 Ablation Study

Rule supervision improves planning and safety over the rule-free and unnormalized variants. Collision-only inference guidance performs substantially worse, supporting training-time rule integration (Table 7).

Table 7: Rule ablations on test14-hard-r (↑). Training variants use the same configuration; collision guidance is applied during DPM sampling.

Setting	Score	Collision	TTC	Drivable	Progress	Speed	Comfort
Collision-Guided Diffusion Planner	59.92	76.65	68.75	90.44	95.96	96.00	60.66
FFN-LoRA (no rules)	69.20	88.42	81.62	95.96	90.44	98.37	86.03
No normalization	67.78	85.11	77.57	93.75	93.38	95.93	84.56
RADP	70.62	92.10	83.46	95.59	90.07	98.29	85.29

5 Limitations and Conclusion

RADP internalizes differentiable driving rules during diffusion training and predicts rule-specific optimization pressures online. We use the trajectory-gradient magnitude $\|\nabla_X J_i\|$, rather than the rule cost J_i, as the attribution target: while J_i measures the current rule violation, its gradient more directly reflects how strongly the rule would locally modify the generated trajectory. Moreover, the current framework assumes a predefined rule set and fixed rule weights, and thus cannot adapt its rule priorities to changing scenarios or preferences. Comparisons with other attribution families are also non-trivial because their explanations are not directly aligned with our rule-specific risk endpoints. Future work will explore intervention-based validation, adaptive rule weighting, broader rule sets, and principled comparisons with other explanation methods.

AI Use Statement

We used generative AI tools to aid or polish the writing of the manuscript and to support retrieval and discovery, such as identifying potentially relevant related work. All AI-assisted text and retrieved references were reviewed and verified by the authors.

Ethics Statement

This work is computational and uses the publicly available nuPlan dataset and simulation framework. It does not involve human-subject experiments or private personal data. Because autonomous driving is safety-critical, the simulation results reported in this paper should not be interpreted as evidence of real-world deployment readiness. Practical deployment would require extensive real-world testing, safety validation, and appropriate human oversight.

Reproducibility Statement

We provide detailed descriptions of the planner architecture, differentiable driving rules, rule normalization, gradient-based Teacher, online attribution Head, and training procedure in the main paper and appendix. We also document the evaluation splits, closed-loop simulation settings, risk endpoints, attribution metrics, temporal controls, and statistical procedures. Code, configuration files, trained checkpoints, and evaluation scripts will be released to facilitate reproducibility.

References

Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 4190–4197. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.385. URL https://aclanthology.org/2020.acl-main.385/.

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? In International Conference on Learning Representations, 2023. URL https://openreview.net/forum?id=sP1fo2K9DFG.

Aaron D. Ames, Xiangru Xu, Jessy W. Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. IEEE Transactions on Automatic Control, 62(8):3861–3876, 2017. doi: 10.1109/TAC.2016.2638961.

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. Information Fusion, 58:82–115, 2020. doi: 10.1016/j.inffus.2019.12.012.

Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. In CVPR Workshop on Autonomous Driving, 2021. URL https://arxiv.org/abs/2106.11810.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In Robotics: Science and Systems, 2023. URL https://www.roboticsproceedings.org/rss19/p026.html.

Zheng Dai and David K. Gifford. Training data attribution for diffusion models. arXiv preprint arXiv:2306.02174, 2023. URL https://arxiv.org/abs/2306.02174.

Kairui Ding, Boyuan Chen, Yuchen Su, Huan-ang Gao, Bu Jin, Chonghao Sima, Xiaohui Li, Wuqiang Zhang, Paul Barsch, Hongyang Li, and Hao Zhao. Hint-ad: Holistically aligned interpretability in end-to-end autonomous driving. In Proceedings of The 8th Conference on Robot Learning, volume 270 of Proceedings of Machine Learning Research, pp. 3742–3765. PMLR, 2025. URL https://proceedings.mlr.press/v270/ding25a.html.

Christopher Hazard, Akshay Bhagat, Balarama Raju Buddharaju, Zhongtao Liu, Yunming Shao, Lu Lu, Sammy Omari, and Henggang Cui. Importance is in your attention: Agent importance prediction for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 2532–2535, 2022. URL https://openaccess.thecvf.com/content/CVPR2022W/Precognition/html/Hazard_Importance_Is_in_Your_Attention_Agent_Importance_Prediction_for_Autonomous_CVPRW_2022_paper.html.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Advances in Neural Information Processing Systems, volume 33, pp. 6840–6851, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In International Conference on Learning Representations, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Boris Ivanovic, Amine Elhafsi, Guy Rosman, Adrien Gaidon, and Marco Pavone. Mats: An interpretable trajectory forecasting representation for planning and control. In Proceedings of the Conference on Robot Learning, volume 155 of Proceedings of Machine Learning Research, pp. 2243–2256. PMLR, 2021. URL https://proceedings.mlr.press/v155/ivanovic21a.html.

Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In Proceedings of the 39th International Conference on Machine Learning, volume 162 of Proceedings of Machine Learning Research, pp. 9902–9915. PMLR, 2022. URL https://proceedings.mlr.press/v162/janner22a.html.

Chiyu Max Jiang, Andre Cornman, Cheolho Park, Ben Sapp, Yin Zhou, and Dragomir Anguelov. Motiondiffuser: Controllable multi-agent motion prediction using diffusion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9644–9653, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Jiang_MotionDiffuser_Controllable_Multi-Agent_Motion_Prediction_Using_Diffusion_CVPR_2023_paper.html.

Napat Karnchanachari, Dimitris Geromichalos, Kok Seang Tan, Nanxiang Li, Christopher Eriksen, Shakiba Yaghoubi, Noushin Mehdipour, Gianmarco Bernasconi, Whye Kit Fong, Yiluan Guo, and Holger Caesar. Towards learning-based planning: The nuplan benchmark for real-world autonomous driving. arXiv preprint arXiv:2403.04133, 2024. URL https://arxiv.org/abs/2403.04133.

Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. Textual explanations for self-driving vehicles. In Proceedings of the European Conference on Computer Vision, pp. 563–578, 2018. URL https://openaccess.thecvf.com/content_ECCV_2018/html/Jinkyu_Kim_Textual_Explanations_for_ECCV_2018_paper.html.

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, and Xinggang Wang. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12037–12047, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Liao_DiffusionDrive_Truncated_Diffusion_Model_for_End-to-End_Autonomous_Driving_CVPR_2025_paper.html.

Ji-Hoon Park, Yeong-Joon Ju, and Seong-Whan Lee. Explaining generative diffusion models via visual analysis for interpretable decision-making process. arXiv preprint arXiv:2402.10404, 2024. URL https://arxiv.org/abs/2402.10404.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4195–4205, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Peebles_Scalable_Diffusion_Models_with_Transformers_ICCV_2023_paper.html.

Michael Psenka, Alejandro Escontrela, Pieter Abbeel, and Yi Ma. Learning a diffusion model policy from rewards via q-score matching. In Proceedings of the 41st International Conference on Machine Learning, volume 235 of Proceedings of Machine Learning Research, pp. 41163–41182. PMLR, 2024. URL https://proceedings.mlr.press/v235/psenka24a.html.

Katrin Renz, Kashyap Chitta, Otniel-Bogdan Mercea, A. Sophia Koepke, Zeynep Akata, and Andreas Geiger. Plant: Explainable planning transformers via object-level representations. In Proceedings of The 6th Conference on Robot Learning, volume 205 of Proceedings of Machine Learning Research, pp. 459–470. PMLR, 2023. URL https://proceedings.mlr.press/v205/renz23a.html.

Ralf Römer, Alexander von Rohr, and Angela P. Schoellig. Diffusion predictive control with constraints. In Proceedings of the 7th Annual Learning for Dynamics and Control Conference, volume 283 of Proceedings of Machine Learning Research, pp. 791–803. PMLR, 2025. URL https://proceedings.mlr.press/v283/romer25a.html.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, pp. 618–626, 2017. URL https://openaccess.thecvf.com/content_iccv_2017/html/Selvaraju_Grad-CAM_Visual_Explanations_ICCV_2017_paper.html.

Hao Shao, Letian Wang, Ruobing Chen, Hongsheng Li, and Yu Liu. Safety-enhanced autonomous driving using interpretable sensor fusion transformer. In Proceedings of The 6th Conference on Robot Learning, volume 205 of Proceedings of Machine Learning Research, pp. 726–737. PMLR, 2023. URL https://proceedings.mlr.press/v205/shao23a.html.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034, 2013. URL https://arxiv.org/abs/1312.6034.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In Proceedings of the 34th International Conference on Machine Learning, volume 70 of Proceedings of Machine Learning Research, pp. 3319–3328. PMLR, 2017. URL https://proceedings.mlr.press/v70/sundararajan17a.html.

Raphael Tang, Linqing Liu, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Pontus Stenetorp, Jimmy Lin, and Ferhan Ture. What the daam: Interpreting stable diffusion using cross attention. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 5644–5659. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.310. URL https://aclanthology.org/2023.acl-long.310/.

Wei Xiao, Tsun-Hsuan Wang, Chuang Gan, Ramin Hasani, Mathias Lechner, and Daniela Rus. Safediffuser: Safe planning with diffusion probabilistic models. In International Conference on Learning Representations, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f95606d8e870020085990d9650b4f2a1-Abstract-Conference.html.

Yi Xiao, Felipe Codevilla, Christopher Pal, and Antonio M. López. Action-based representation learning for autonomous driving. In Proceedings of the Conference on Robot Learning, volume 155 of Proceedings of Machine Learning Research, pp. 232–246. PMLR, 2021. URL https://proceedings.mlr.press/v155/xiao21a.html.

Brian Yang, Huangyuan Su, Nikolaos Gkanatsios, Tsung-Wei Ke, Ayush Jain, Jeff Schneider, and Katerina Fragkiadaki. Diffusion-es: Gradient-free planning with diffusion for autonomous driving and zero-shot instruction following. arXiv preprint arXiv:2402.06559, 2024. URL https://arxiv.org/abs/2402.06559.

Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, and Raquel Urtasun. End-to-end interpretable neural motion planner. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8660–8669, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Zeng_End-To-End_Interpretable_Neural_Motion_Planner_CVPR_2019_paper.html.

Yinan Zheng, Ruiming Liang, Kexin Zheng, Jinliang Zheng, Liyuan Mao, Jianxiong Li, Weihao Gu, Rui Ai, Shengbo Eben Li, Xianyuan Zhan, and Jingjing Liu. Diffusion-based planning for autonomous driving with flexible guidance. In International Conference on Learning Representations, 2025. URL https://arxiv.org/abs/2501.15564.

Ziyuan Zhong, Davis Rempe, Danfei Xu, Yuxiao Chen, Sushant Veer, Tong Che, Baishakhi Ray, and Marco Pavone. Guided conditional diffusion for controllable traffic simulation. In 2023 IEEE International Conference on Robotics and Automation, pp. 3560–3566, 2023. doi: 10.1109/ICRA48891.2023.10161463. URL https://ieeexplore.ieee.org/document/10161463/.

A Baseline Diffusion Objective

Following Diffusion Planner (Zheng et al., 2025), the forward process and clean-trajectory prediction are

$$q(\mathbf{X}^{(k)} \mid \mathbf{X}^{(0)}) = \mathcal{N}\Big(\sqrt{\bar{\alpha}_k}\mathbf{X}^{(0)}, (1-\bar{\alpha}_k)\mathbf{I}\Big), \qquad \widehat{\mathbf{X}}^{(0)} = \mu_\theta(\mathbf{X}^{(k)}, k, S_t). \tag{14}$$

The masked reconstruction objective is

$$\mathcal{L}_{\text{diff}} = \mathcal{L}_{\text{nbr}} + 2\mathcal{L}_{\text{ego}}, \tag{15}$$

$$\mathcal{L}_{\text{ego}} = \mathbb{E}\left[\frac{1}{H}\sum_{h=1}^{H} \|\widehat{X}^{(0)}_{\text{ego},h} - X^{(0)}_{\text{ego},h}\|_2^2\right], \tag{16}$$

$$\mathcal{L}_{\text{nbr}} = \mathbb{E}\left[\frac{1}{|\mathcal{V}|}\sum_{(b,n,h)\in\mathcal{V}} \|\widehat{X}^{(0)}_{b,n,h} - X^{(0)}_{b,n,h}\|_2^2\right], \tag{17}$$

where $\mathcal{V}$ excludes padded agents and unavailable future states.

B Rule and Risk Definitions

Planner adaptation, gradient supervision, and Head outputs use the same six semantic channels listed in Table 1:

$$\mathcal{R}_{\text{train}} = \mathcal{R}_{\text{attr}} = \{\text{collision}, \text{lane}, \text{overspeed}, \text{kinematics}, \text{comfort}, \text{goal}\}. \tag{18}$$

Thus, the planner rule loss, Teacher targets, and Head outputs are fully aligned in semantics and dimensionality. Underspeed and traffic-light costs do not form independent training or attribution channels. TTC is likewise not an attribution channel: the TTC row uses collision pressure as its score and evaluates the same attribution against a distinct future-risk endpoint.

B.1 Trajectory Denormalization and Finite Differences

Rules operate on denormalized trajectories at 0.1 s resolution. Predicted states represent rear-axle position and heading; collision computation further uses the nuPlan Pacifica length, width, and rear-axle-to-center offset to construct oriented boxes. Speed, acceleration, jerk, curvature, and curvature rate are obtained by finite differences over the current ego state and future trajectory. Invalid map points, padded agents, and unavailable future states are excluded by masks.

B.2 Complete Instantiation of Rule Costs

Collision. In addition to Eq. equation 6, the implementation uses $t_{\text{head}} = 0.8\,\text{s}$ and $a_{\text{brake}} = 4.0\,\text{m/s}^2$. For each future time and obstacle,

$$\tau_{h,n} = \frac{[d^{\text{box}}_{h,n}]_+}{\max(c_{h,n}, 10^{-3})}. \tag{19}$$

A stop-gradient soft gate combines box separation, closing speed, a 4 s TTC threshold, and the safety-distance deficit, with a hard cutoff beyond 20 m. Active terms are combined by a softmax-weighted mean with temperature 8, so that the most critical interactions receive greater weight.

Lane. Each predicted point is matched to the nearest route-polyline point. Its tangent defines a normal direction and signed lateral error e_h. If the available left and right widths are w^L_h and w^R_h, respectively,

$$z^L_h = e_h + W_{\text{ego}}/2 - w^L_h, \tag{20}$$

$$z^R_h = -e_h + W_{\text{ego}}/2 - w^R_h, \tag{21}$$

$$J_{\text{lane}} = J(z^L; 0.5) + J(z^R; 0.5) + 0.05J(|e|; 2.0), \tag{22}$$

Table 8: Principal rule constants used in the implementation.

Rule	Threshold	Physical scale σ	Weight w_i	Other constants
Collision	$d_0 = 0.5$ m, $t_{\text{head}} = 0.8$ s, $a_{\text{brake}} = 4.0$ m/s^2	0.5 m	2.5	Active distance 20 m, risk TTC 4 s, softmax $\alpha = 8$
Lane	Ego-body boundary crossing	0.5 m	0.8	Centerline weight 0.05, center scale 2.0 m
Speed	$v > v^{\text{lim}}$	1.0 m/s	0.8	Time steps without a valid speed limit are masked out
Kinematics	$a_{\text{lon}} \le 6$, $-a_{\text{lon}} \le 8$, $\lvert a_{\text{lat}}\rvert \le 4.5$, $\lvert\kappa\rvert \le 0.35$	$1, 1, 1, 0.05$	0.5	—
Comfort	$\lvert j_{\text{lon}}\rvert, \lvert j_{\text{lat}}\rvert \le 8.37$, $\lvert\dot{\kappa}\rvert \le 0.30$	$1, 1, 0.1$	0.8	—
Goal progress	Reachable/expert progress deficit	5.0 m	1.5	Lateral weight 0.1, monotonicity weight 0.2, stopping buffer 2 m

where $J(z;\sigma)$ denotes Eq. equation 4 averaged over valid points. The final term is a weak centerline regularizer and does not replace the ego-body boundary constraints.

Speed. Let v_h^{lim} be the statutory speed limit of the nearest valid route lane. The current six-dimensional planner and attribution implementation uses only the atomic overspeed cost,

$$J_{\text{over}} = J(v - v^{\text{lim}}; 1.0). \tag{23}$$

Time steps without a valid route-lane speed limit are masked out. Road curvature, low-speed deviation, and free-road status do not enter this channel, preserving its semantics as pressure from exceeding the statutory speed limit.

Kinematics and comfort. The kinematic cost is

$$J_{\text{kin}} = J(a_{\text{lon}} - 6; 1) + J(-a_{\text{lon}} - 8; 1) \tag{24}$$

$$+ J(|a_{\text{lat}}| - 4.5; 1) + J(|\kappa| - 0.35; 0.05), \tag{25}$$

where acceleration is measured in m/s^2 and curvature in m^{-1}. The comfort cost is

$$J_{\text{comf}} = J(|j_{\text{lon}}| - 8.37; 1) + J(|j_{\text{lat}}| - 8.37; 1) + J(|\dot{\kappa}| - 0.30; 0.1). \tag{26}$$

Goal progress. Let s_h denote cumulative progress after projecting the trajectory onto the route. The target $s^\star$ is determined by the valid expert-future endpoint when available; otherwise, reachable progress is computed from the current speed and a nominal acceleration of 0.5 m/s^2 and clipped at the route endpoint. When a red light ahead is active, the target is capped at 2 m before the stop line. Traffic-light state therefore constrains a reasonable progress target but does not define a seventh attribution channel. The cost is

$$J_{\text{goal}} = J(s^\star - s_H; 5.0) + 0.1J(d_H^{\text{route}}; 0.5) + 0.2J(s_{h-1} - s_h; 5.0). \tag{27}$$

B.3 Rule Constants and Aggregation

All six differentiable rules use fixed physical thresholds and calibration constants, which are collected in Table 8.

Except for collision, costs are averaged over valid active elements; collision uses the softmax-weighted mean above. During planner training, a stop-gradient EMA scale $s_i^{(r)}$ is maintained for each complete rule cost with decay 0.99 and lower bound 10^{-3}, and $\widetilde{J}_i = J_i / \text{sg}(s_i^{(r)})$. Teacher targets are always computed from gradients of atomic costs J_i before semantic weighting or EMA normalization.

B.4 External Rollout-Risk Endpoints

The v_i^{rollout} in Eq. equation 13 is constructed from executed ego and agent states in the same closed-loop simulation. It is not the differentiable rule cost and is not used to train the head. Specifically: (i) physical collision is a positive-area overlap between ego and tracked-agent boxes; (ii) TTC risk uses the reciprocal of nuPlan's per-frame TTC; (iii) lane risk uses

the signed ego-body margin to the route/drivable boundary, with 0.30 m or less treated as an event; (iv) overspeed uses the positive part of the nuPlan speed-limit exceedance; (v) kinematics uses normalized violations of longitudinal/lateral acceleration and curvature; (vi) comfort uses the maximum ratio of longitudinal jerk, lateral jerk, and curvature rate to their thresholds; and (vii) goal risk uses incomplete reference-route progress, with an event threshold at an ego-to-expert progress ratio below 0.80.

For planning frame t_k, collision and TTC use a 2 s future window, while lane, speed, kinematics, comfort and goal use 8 s. Severity in each window is aggregated by a pre-specified rule-specific operator. Temporal Spearman is computed within each scenario having sufficient temporal variation and then averaged across scenarios. The shuffled control permutes attribution only within the same scenario, preserving its marginal distribution. Confidence intervals use 10,000 scenario-level bootstrap resamples.

B.5 Complete Head–Teacher Fidelity Metrics

Let $\mathcal{F}$ be the valid synchronized frames and M the number of rules. We compute per-rule correlation and its macro-average as

$$\rho_i^{H,T} = \text{Spearman}\left(\{A_i^H(t)\}_{t\in\mathcal{F}}, \{A_i^T(t)\}_{t\in\mathcal{F}}\right), \qquad \rho_{\text{macro}}^{H,T} = \frac{1}{M}\sum_{i=1}^{M}\rho_i^{H,T}. \tag{28}$$

Numerical agreement and dominant-rule agreement are

$$\text{MAE}^{H,T} = \frac{1}{M|\mathcal{F}|}\sum_{i=1}^{M}\sum_{t\in\mathcal{F}}|A_i^H(t) - A_i^T(t)|, \tag{29}$$

$$\text{Top-1}^{H,T} = \frac{1}{|\mathcal{F}|}\sum_{t\in\mathcal{F}}\mathbf{1}\Big[\arg\max_i A_i^H(t) = \arg\max_i A_i^T(t)\Big]. \tag{30}$$

B.6 Complete Temporal-Alignment Metrics

For rule i and scenario n, continuous temporal alignment is

$$\rho_{i,n}^{\text{temp}} = \text{Spearman}(\{A_{i,n}(t_k)\}_k, \{R_{i,n}(t_k)\}_k), \qquad \rho_i^{\text{temp}} = \frac{1}{|\mathcal{N}_i^\rho|}\sum_{n\in\mathcal{N}_i^\rho}\rho_{i,n}^{\text{temp}}, \tag{31}$$

where $\mathcal{N}_i^\rho$ contains scenarios with enough valid frames and non-constant attribution and risk sequences. For attribution source $Q \in \{T, H\}$, the reported macro score averages all seven evaluated endpoints,

$$\rho_{\text{macro}}^Q = \frac{1}{7}\sum_{e\in\mathcal{E}}\rho_e^Q, \quad \mathcal{E} = \{\text{collision, TTC, lane, speed, kinematics, comfort, goal}\}. \tag{32}$$

Physical collision and TTC are separate endpoints that share the collision attribution channel. With a pre-specified threshold, $y_{i,n}(t_k) = \mathbf{1}[R_{i,n}(t_k) > \delta_i]$. AUPRC uses attribution as the continuous event-retrieval score and is macro-averaged over eligible scenarios. If $\mathcal{T}_{i,n}^{10}$ is the top 10% of valid frames ranked by attribution, then

$$\text{Lift@10\%}_{i,n} = \frac{|\mathcal{T}_{i,n}^{10}|^{-1}\sum_{t_k\in\mathcal{T}_{i,n}^{10}} y_{i,n}(t_k)}{|\mathcal{T}_{i,n}|^{-1}\sum_{t_k\in\mathcal{T}_{i,n}} y_{i,n}(t_k)}, \qquad \text{Lift@10\%}_i = \frac{1}{|\mathcal{N}_i^L|}\sum_{n\in\mathcal{N}_i^L}\text{Lift@10\%}_{i,n}. \tag{33}$$

The random-ranking AUPRC baseline equals the positive-event rate. The temporal control permutes attribution within each scenario before recomputing Spearman. All 95% confidence intervals use 10,000 scenario-level bootstrap resamples, retaining all correlated frames of each sampled scenario.

C Additional Experimental Details

C.1 Non-Reactive Planning Results

For completeness, Table 9 reports the corresponding non-reactive aggregate scores. These results are supplementary to the reactive evaluation emphasized in the main text.

Table 9: Closed-loop planning scores with non-reactive agents (↑).

Split	Diffusion Planner	RADP
val14	89.65	88.72
test14-random	88.88	89.46
test14-hard	75.01	74.80

C.2 Planner Adaptation

We initialize from the released Diffusion Planner checkpoint. The DiT contains three blocks, each ordered as self-attention, mlp1, scene-conditioned cross-attention, and mlp2. All baseline parameters are frozen, and LoRA is inserted only into

$$\{\text{mlp1.fc1}, \text{mlp1.fc2}, \text{mlp2.fc1}, \text{mlp2.fc2}\} \tag{34}$$

of every block. For original linear layer W_0,

$$W_{\text{eff}}x = W_0x + \frac{\alpha}{r}BAx, \qquad r = 8, \quad \alpha = 16, \tag{35}$$

with LoRA dropout 0.05. Twelve linear modules are replaced, yielding 24 trainable A/B tensors; attention, AdaLN, the encoder, input projection, and final trajectory layer remain frozen. Adaptation uses batch size 128, learning rate 2×10^{-5}, and random seed 3407.

C.3 Teacher Calibration and Caching

To make gradients with different physical dimensions comparable, we collect RMS gradients for the six attribution channels over 100 calibration batches from the training split and use the 75th percentile of each channel's non-negative gradients as κ_i. Equation equation 9 then produces log-compressed absolute pressures without normalization across the rule dimension.

With the planner frozen, caching records the final-DPM denoising state used in closed-loop inference: final DiT agent tokens, ego-trajectory geometry, route polylines, neighbor masks, and teacher pressures. The training cache contains 100,000 samples from processed; the validation cache contains 20,000 non-overlapping samples from new_processed.

C.4 Trajectory-Conditioned Scene Attention Head Details

The attribution head operates on the final denoising representation produced by the frozen rule-aligned planner. The token dimension is 192. For each planning frame, we construct a trajectory-conditioned query z_t by combining the final-DPM ego token with geometric features extracted from the generated ego trajectory. The scene-condition representation consists of the valid context tokens provided by the planner encoder, covering surrounding agents, map elements, route information, and lane geometry. All planner-side representations are detached and remain frozen during attribution-head training.

To recover rule-relevant scene context, z_t attends to the scene-condition tokens through a six-head multi-head attention module,

$$c_t = \text{MHA}_{\text{scene}}(z_t, S_t, S_t),$$

where S_t denotes the valid scene-condition token set and the corresponding validity masks are applied during attention. This trajectory-conditioned attention allows the same generated trajectory to retrieve different contextual evidence depending on the surrounding traffic

and road geometry, rather than aggregating agent and route information through separate attention branches.

The query representation z_t and attended scene context c_t are concatenated and passed to six rule-specific prediction branches, corresponding to collision, lane, overspeed, kinematics, comfort, and goal progress. Each branch uses an independent two-layer MLP with hidden dimension 256 and an unshared scalar output layer. Softplus is applied to each scalar output to obtain a non-negative rule pressure. Thus, the six branches share the trajectory and scene representations but do not share their final prediction parameters.

The attribution head is trained from scratch for 50 epochs with batch size 512 using AdamW, a learning rate of 10^{-4}, and weight decay 10^{-4}. The Smooth-L1 regression loss uses transition parameter $\beta_{\text{Huber}} = 0.2$. To improve learning in the high-pressure tail, collision samples above the 75th percentile of positive collision pressure are sampled with a 3.0× multiplier and assigned a regression weight of 2.0. Lane samples above the 85th percentile are sampled with a 1.5× multiplier. For the remaining rule channels, samples above the corresponding 85th-percentile pressure threshold are sampled with a 1.5× multiplier and assigned a loss weight of 1.25. The pairwise-ranking loss is weighted by 0.2, uses temperature 0.2, and ignores teacher-pressure pairs whose absolute difference is below 0.05.

Throughout attribution-head training, all planner parameters, including the LoRA adapters, remain frozen. The head is supervised exclusively by the calibrated gradient-pressure Teacher targets cached from the frozen planner; future closed-loop rollout risks are never used as training supervision.

C.5 Evaluation Reproducibility

Teacher and head use the same planner checkpoint, scene state, and generated trajectory at each frame. Head–teacher fidelity is computed directly on these synchronized frames. For behavioral alignment, $A_i(t_k)$ is paired with future risk over $[t_k, t_k + H_i]$ in the same rollout. The random AUPRC baseline equals the positive-event rate, and Lift@10% uses the highest-attribution 10% of all valid frames. All thresholds, windows, and aggregation operators are fixed before inspecting head results and shared by teacher and head.

C.6 Latency Details

End-to-end latency is measured over five fixed test14-hard-r scenarios, while cached-state attribution is measured over 500 synchronized frames. The former includes trajectory generation, whereas the latter isolates attribution computation; all measurements use the same RTX 4080. The complete mean and standard-deviation results are reported in Table 10.

Table 10: Inference latency details on an RTX 4080. Values are mean ± standard deviation; lower is better.

Method	Latency (ms)	Overhead
Planner only	251.05 ± 39.04	—
Planner + Head	245.50 ± 30.43	$-1.64\% \pm 6.67\%$
Planner + exact Teacher	465.29 ± 66.46	$+89.18\% \pm 38.35\%$
Planner + gradient guidance	571.10 ± 52.64	$+130.76\% \pm 30.01\%$
Exact Teacher on cached states	62.51 ± 3.69	1.00×
Head on cached states	8.18 ± 1.20	7.65× faster

C.7 Complete Confidence Intervals for Main-Text Attribution Results

The scenario-bootstrap 95% confidence intervals omitted from the main text for readability are reported in Table 11 and Table 12. Resampling is performed at the scenario level so that correlated frames from the same scenario are not treated as independent observations.

Table 11: 95% confidence intervals for Head–Teacher fidelity and macro-averaged future-risk alignment on reactive splits.

Reactive split	Head–Teacher fidelity 95% CI			Future-risk alignment 95% CI	
	Macro Spearman	MAE	Top-1	Teacher ρ_{macro}	Head ρ_{macro}
val14	[0.525, 0.547]	[0.209, 0.222]	[69.4, 71.8]%	[0.213, 0.256]	[0.183, 0.229]
test14-random	[0.552, 0.588]	[0.188, 0.212]	[69.5, 73.9]%	[0.133, 0.258]	[0.119, 0.233]
test14-hard	[0.537, 0.571]	[0.224, 0.251]	[69.7, 74.0]%	[0.184, 0.255]	[0.151, 0.231]

Table 12: Scenario-bootstrap 95% confidence intervals for rule-wise future-risk alignment on val14-r.

Attribution channel	Risk endpoint	Teacher AUPRC	Head AUPRC	Teacher Lift@10%	Head Lift@10%	Teacher ρ	Head ρ
Collision	Physical collision	[0.481,0.604]	[0.452,0.578]	[1.388,2.257]	[1.184,2.019]	[0.158,0.302]	[0.077,0.252]
Collision	TTC risk	[0.516,0.598]	[0.450,0.527]	[1.640,2.212]	[1.334,1.896]	[0.196,0.274]	[0.115,0.198]
Lane	Lane-boundary risk	[0.894,0.928]	[0.815,0.864]	[1.422,1.617]	[1.260,1.428]	[0.283,0.345]	[0.206,0.265]
Speed	Overspeed risk	[0.725,0.829]	[0.701,0.816]	[1.553,2.254]	[1.556,2.285]	[0.198,0.402]	[0.200,0.388]
Kinematics	Kinematic risk	[0.167,0.167]	[0.250,0.250]	[0.000,0.000]	[0.000,0.000]	[0.079,0.143]	[0.123,0.194]
Comfort	Comfort risk	[0.162,0.679]	[0.156,0.598]	[0.000,4.375]	[0.000,3.500]	[0.072,0.133]	[0.089,0.155]
Goal progress	Route-progress deficit	[0.829,0.871]	[0.774,0.822]	[1.687,1.924]	[1.513,1.740]	[0.314,0.370]	[0.286,0.345]

D Additional Collision-Avoidance Cases

We further compare matched Baseline and RADP rollouts on four test14-hard-r scenarios for which the Baseline incurs an at-fault collision while RADP does not. For each scenario, the upper row shows the Baseline and the lower row shows RADP at exactly the same simulation indices. The four columns correspond to approximately three, two, and one second before the Baseline's first geometric contact, followed by the contact frame itself. Yellow boxes denote the ego vehicle, dark boxes denote other traffic participants, cyan curves show the trajectory planned at the displayed frame, and orange curves show the executed ego history. At the contact frame, the colliding participant and the Baseline ego outline are highlighted in red. All panels use the original nuPlan vector-map geometry and a fixed world-coordinate viewport within each scenario.

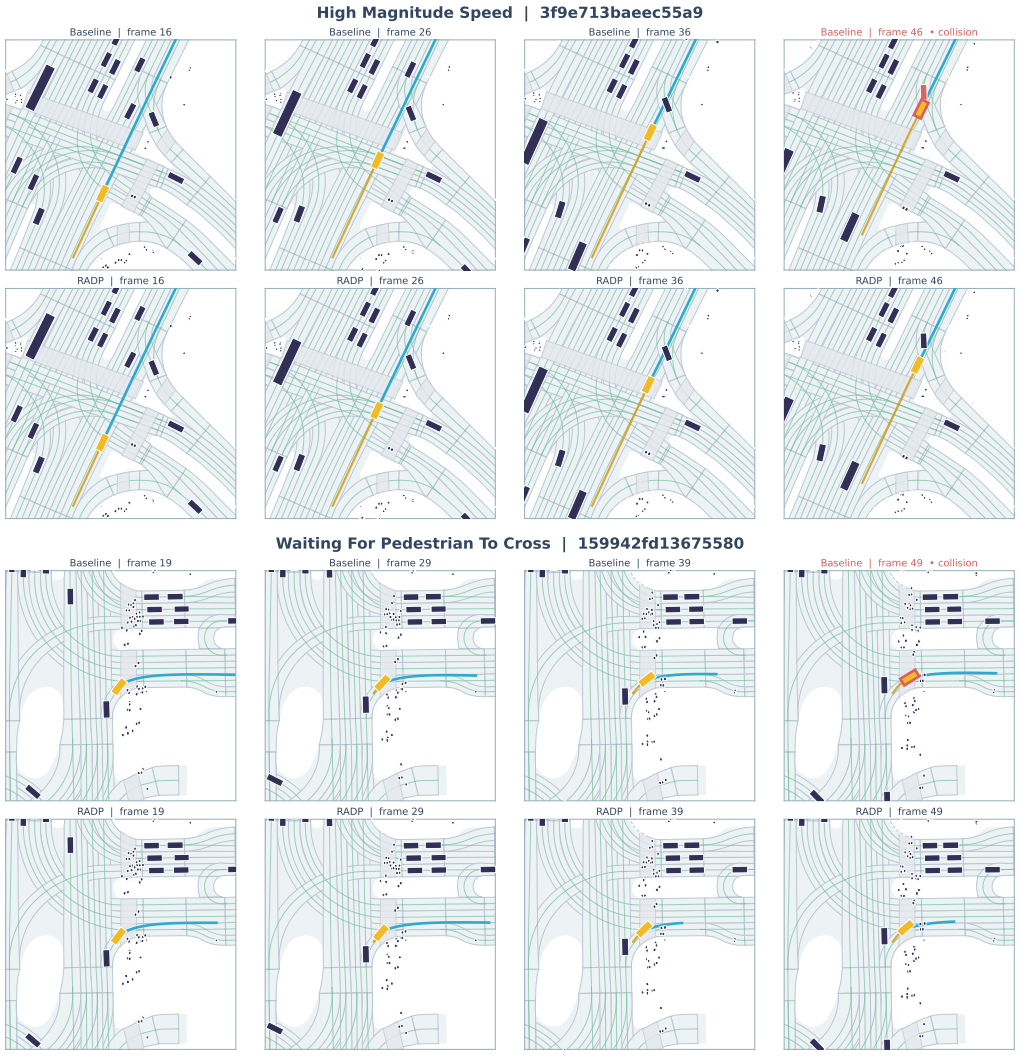


Figure 5: Matched collision-avoidance rollouts. Top: a high-speed vehicle interaction (token 3f9e713baeec55a9). Bottom: a pedestrian-crossing interaction (token 159942fd13675580). The Baseline makes contact in the final column, whereas RADP remains collision-free at the corresponding frame.

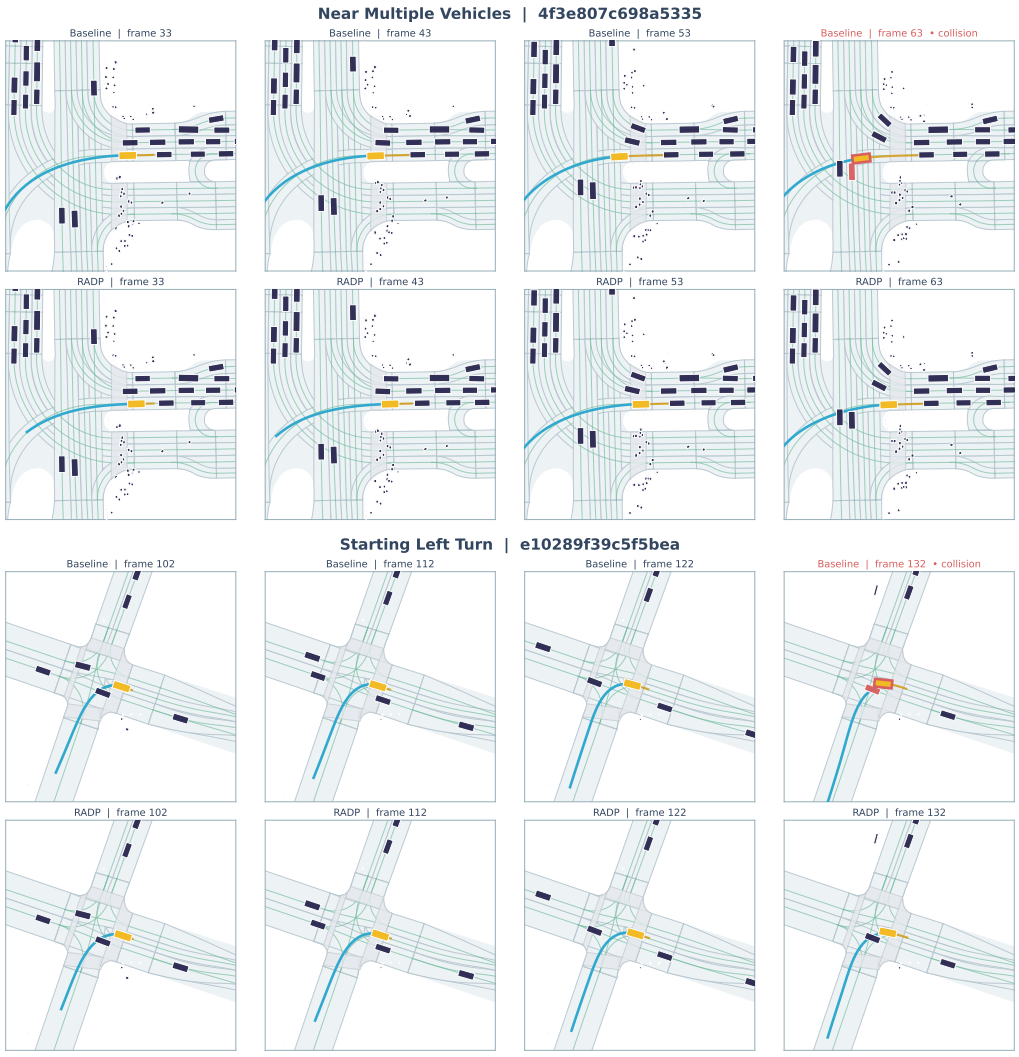


Figure 6: Additional matched collision-avoidance rollouts. Top: a dense multi-vehicle interaction (token 4f3e807c698a5335). Bottom: a left-turn vehicle conflict (token e10289f39c5f5bea). Identical frame indices are used for Baseline and RADP in every column.