WHEN CAPABILITIES BECOME ACTIONABLE: CAPABILITY GROUNDING IN DRIVING FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Driving foundation models increasingly augment planning with richer representations, auxiliary supervision, and predictive objectives. Yet a model can benefit from such information without necessarily using it in the intended way when generating a trajectory. To address this problem, we formulate *capability grounding* as the question of whether capability-relevant information actually shapes the planned action in a way that is appropriate for the driving situation. We introduce **Capability Grounding Attribution (CGA)**, a controlled evaluation framework for testing this relationship. CGA constructs controlled comparisons that change capability-relevant evidence or supervision while holding alternative factors fixed, and examines how the planned action changes. Using CGA on four representative driving policies, we find that evidence of sensitivity, learning, or planning benefit can diverge from evidence that the intended information actually guides action. We further compare CGA diagnostics with closed-loop outcomes and find that the two capture complementary aspects of driving behavior: CGA identifies how information shapes planning, whereas closed-loop evaluation measures what happens when those plans are executed.

1 INTRODUCTION

Driving foundation models increasingly add information beyond task-specific perception and planning. Vision-language-action (VLA) policies use pretrained world knowledge and language reasoning to generate trajectories (Zhou et al., 2025; Li et al., 2026b). Other systems add spatial and geometric representations, latent reasoning about space and time, or structured world information (Li et al., 2026c; Yu et al., 2026; Luo et al., 2026; Huang et al., 2026a). World-model approaches train on future prediction to improve scene representations and planning (Li et al., 2025a; 2026a). Although their architectures differ, these methods share an expectation: richer information should become useful for driving decisions.

The expectation itself is reasonable; the difficulty is verifying that the added information is actually used by the policy. A trajectory may change when a relevant input changes, additional supervision may update states shared with the planner, or an auxiliary training objective, such as future prediction, may improve planning performance. These observations show that the model is sensitive to the input, that the supervision can affect its representation, or that the training objective is useful. But none of them, by itself, tells us whether the policy is relying on the intended information when choosing its action. Information recoverable from an internal representation or demonstrated through question answering has the same limitation: it shows that the information is available, not that it controls behavior.

Prior studies show why this distinction matters. Driving VLMs can answer from language priors when visual evidence is degraded or absent (Xie et al., 2025); reasoning-oriented planners can exploit history shortcuts (Li et al., 2026d); and agreement between a generated reasoning trace and an action can differ from the trace's causal influence (Nguyen et al., 2026). Visual perturbation studies further show that policies can depend very differently on different parts of their input (He et al., 2026). These works expose specific shortcuts or sources of behavioral dependence, but they

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.

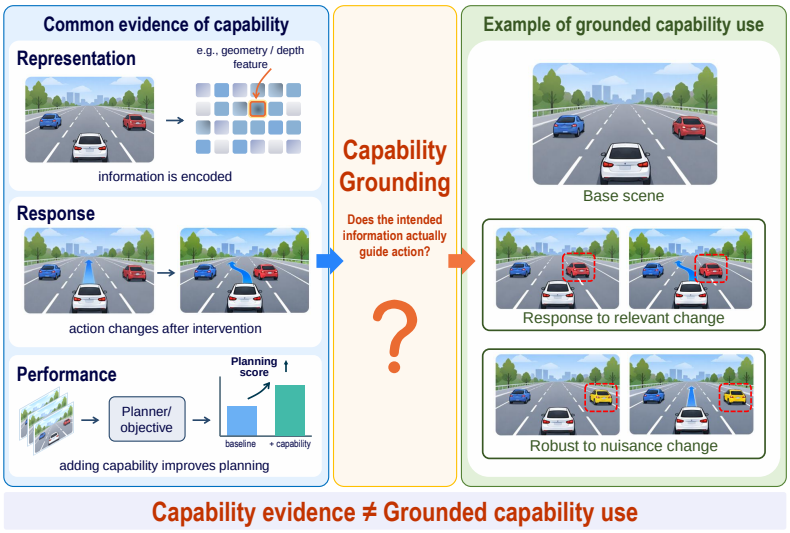

Figure 1: Motivation for capability grounding. Evidence that a policy encodes information, changes its trajectory after an intervention, or benefits from added supervision does not by itself show that it uses the claimed capability to choose actions. The conceptual example on the right illustrates the additional question: does the policy respond appropriately when a driving-relevant spatial relation changes while remaining stable when only vehicle appearance changes?

do not provide a common criterion for deciding when a claimed semantic, geometric, or predictive capability is actually used for action.

Consider a planner whose trajectory changes substantially when another vehicle moves into the ego lane. At first glance, this seems to show that the planner is using spatial information about the increased risk. But the trajectory change alone is not enough: the planner might simply be highly sensitive to visual changes in general. To distinguish the two, we need to ask whether it responds appropriately to changes that matter for driving, while remaining relatively stable to changes that should not matter, such as the vehicle's appearance. This is the question we call *capability grounding* (Figure 1): is the policy's action actually guided by the information that the claimed capability is supposed to provide, and does it use that information in the way the driving task requires?

To test this link, we introduce **Capability Grounding Attribution (CGA)**. We compare planned actions after changing task-relevant scene evidence and, separately, irrelevant appearance; after adding capability-specific supervision in matched training runs; and after changing the presence or content of predictive targets (Figure 2). Each comparison tests whether an observed response or planning gain depends on the information the claimed capability is meant to provide. Across representative driving policies (Zou et al., 2025; Zhou et al., 2025; Li et al., 2026b;a), these comparisons show that strong trajectory responses or planning gains do not by themselves establish that the intended information guides action.

Finally, we test whether the action-level patterns exposed by CGA are reflected in closed-loop behavior. The experiments show that their relationship with driving outcomes varies across scenarios. This indicates that closed-loop performance alone does not reveal how capability-relevant information shapes planning, motivating capability grounding to be evaluated alongside driving performance rather than inferred from it.

Our contributions are threefold:

- We formulate capability grounding as an explicit evaluation question: whether a policy uses capability-relevant evidence selectively and appropriately when choosing actions. This distinguishes what a model can represent or demonstrate from what actually guides its decisions.

- We introduce Capability Grounding Attribution (CGA), a controlled framework for testing such claims. CGA changes inputs and training supervision under specified controls, then examines the resulting actions to distinguish the proposed use of a capability from alternative explanations.
- Our experiments reveal that common indicators of capability use are not sufficient on their own. Large trajectory changes can coexist with sensitivity to task-irrelevant variations, while gains from added supervision need not correspond to improved capability-related behavior. Closed-loop results further show that capability-grounding analysis and direct driving evaluation provide complementary rather than interchangeable evidence.

2 Related Work

2.1 Driving Foundation Models and Capability-Augmented Planning

Driving foundation models increasingly augment end-to-end policies with semantic reasoning, spatial structure, and predictive representations. Reasoning-aware VLAs use pretrained multimodal knowledge, explicit rationales, reinforcement learning, or language–action alignment to produce trajectories (Zhou et al., 2025; Li et al., 2026b; Zhou et al., 2026; Wang et al., 2026b). Spatially enhanced policies inject 3D instances, geometry, or latent spatiotemporal experts into planning (Li et al., 2026c; Yu et al., 2026; Luo et al., 2026; Huang et al., 2026a). Predictive approaches learn from latent futures, future visual tokens, occupancy evolution, action-conditioned world models, or dynamics reasoning (Li et al., 2025a; 2026a; Huang et al., 2026b; Wang et al., 2026a; Shang et al., 2026; Zhong et al., 2026). These methods study how to supply richer information and whether it improves planning. A planning gain, however, does not identify which semantic, geometric, or predictive property produced it. We study this complementary question by focusing on how such information actually influences action.

2.2 Capability Evaluation and Action Attribution

One line of work evaluates whether driving models possess specific knowledge or reasoning abilities through question answering and related diagnostic tasks. DriveBench tests language-prior dependence using corrupted and vision-absent inputs, while VLADBench evaluates fine-grained driving knowledge and reasoning (Xie et al., 2025; Li et al., 2025b). NuScenes-SpatialQA and DriveSpatial use 3D scene graphs and multi-view temporal constraints to evaluate spatial and spatiotemporal reasoning (Tian et al., 2025; Vo et al., 2026). These benchmarks can show that a model recognizes or reasons about a traffic state or spatial relation, but the evaluated output is an answer rather than a planned trajectory. They therefore do not tell us whether the same information is actually used when the driving policy chooses an action.

A separate line of work moves closer to this question by directly testing what influences a model's action. VLADriveBench intervenes on generated reasoning to distinguish CoT–action agreement from causal influence (Nguyen et al., 2026). He et al. perturb visual channels and information structure to study how visual input affects driving behavior (He et al., 2026), while Counterfactual Vision Action Analysis removes individual road actors and measures the resulting changes in trajectories and representations (Panda et al., 2026). Related work in robotics attributes behavior to visual regions, modalities, or internal pathways using masking, attention knockout, and rollout probes (Shi et al., 2026; Zhang et al., 2026a; Hancock et al., 2024; Zhang et al., 2026b). In driving, planning-token probing further shows that semantic intent can be recoverable from an internal representation before that representation is compatible with the deployed planner (Babu et al., 2026). These methods identify which inputs or internal components influence action. Our work instead asks whether that influence can be attributed to the specific information associated with the claimed capability.

2.3 Information Use and Driving Performance

Planning performance alone does not tell us which information a policy relies on. Prior work has shown that strong open-loop results can be supported by ego-state or history shortcuts even when scene perception is weakly used (Zhai et al., 2023; Li et al., 2024). PlanTF further studies how imitation-based planners can exploit particular features while behaving differently in closed

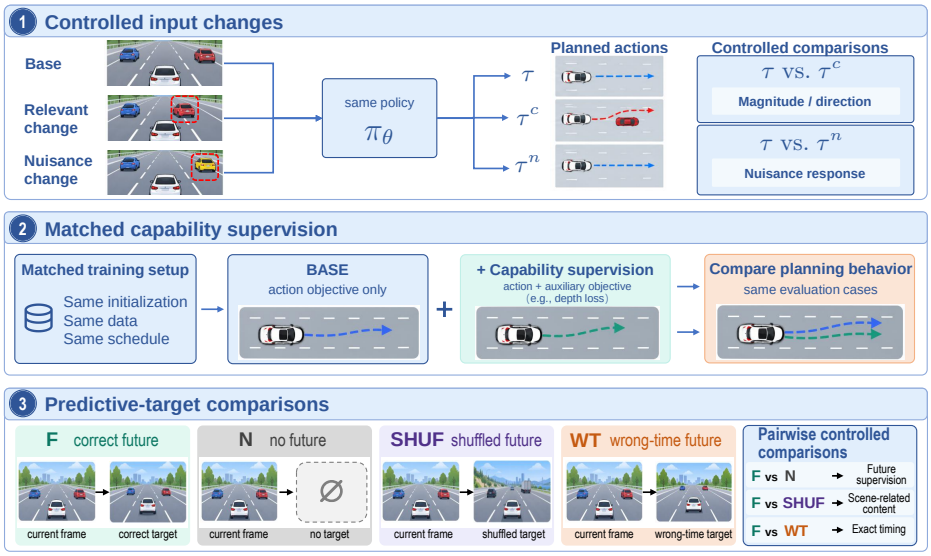

Figure 2: Overall CGA framework. (1) The same policy processes baseline, capability-relevant, and nuisance inputs; its trajectories are compared for response magnitude, direction, and nuisance sensitivity. (2) Matched training adds capability-specific supervision and compares behavior on the same cases. (3) Separate matched F–N, F–SHUF, and F–WT comparisons test objective presence, scene-related target content, and exact future-time correspondence.

loop (Cheng et al., 2024), while PGS, AdaptiveAD, and Drive-R1 explicitly address perception under-use, shortcut dependence, or reasoning–planning mismatch (Huang et al., 2025; Tang et al., 2026; Li et al., 2026d). These studies show that strong planning performance can coexist with shortcut dependence or weak use of scene information. Likewise, a planning gain after adding capability-related supervision does not reveal which information produced the improvement.

The reverse distinction is equally important: identifying what changes a planned trajectory does not by itself evaluate how the vehicle will perform once the policy is executed. Closed-loop evaluation introduces repeated replanning, control, and interaction with the evolving scene. Existing benchmarks expose errors and states that are absent from static evaluation (Jia et al., 2026), and cross-benchmark studies show that open-loop rankings need not agree with closed-loop outcomes (Wang et al., 2026c). We therefore treat information-use evaluation and driving-performance evaluation as complementary. CGA asks which information shapes the planned action, while closed-loop evaluation asks what happens when the complete policy is repeatedly executed.

3 Capability Grounding Attribution

3.1 Defining Capability Grounding

A driving policy may contain or benefit from information associated with a capability without necessarily using that information when choosing an action. We use *capability grounding* to describe whether such information actually guides the policy's action in the intended way. This requires the action to depend on the relevant information rather than arbitrary correlated changes, and to change in a direction that is appropriate for the driving situation.

Let c denote a claimed capability, x_i an evaluation input, and $\tau_i = \pi_\theta(x_i)$ the action or trajectory produced by policy π_θ. Observing τ_i, or even observing a change in τ_i, is not sufficient to determine whether information associated with c produced that behavior. **Capability Grounding Attribution (CGA)** addresses this ambiguity through controlled comparisons that alter the relevant input evidence, the supervision used to learn it, or the content of an auxiliary predictive target (Figure 2).

3.2 Operationalizing Capability Grounding

Different forms of evidence leave different attribution questions. We consider three common settings: a policy response to an input change, capability-specific supervision that reaches planning-related representations, and a planning gain produced by predictive supervision. Each requires a different control to determine what the observed effect should be attributed to.

Controlled input changes. We first separate the effect of changing capability-relevant evidence from the effect of changing an irrelevant property. Write T_c for the capability-relevant transformation and T_n for the nuisance transformation. Applying the policy to each changed input gives the outputs to compare with its baseline prediction:

$$\begin{aligned} x_i^c &= T_c(x_i), \quad \tau_i^c = \pi_\theta(x_i^c), \\ x_i^n &= T_n(x_i), \quad \tau_i^n = \pi_\theta(x_i^n). \end{aligned} \tag{1}$$

For example, moving a target into the ego lane changes its spatial relation to the ego vehicle, whereas changing its paint tests sensitivity to task-irrelevant appearance. These are separate experimental protocols, summarized in Section 4.2. To measure how much each change alters the trajectory, we compute the relevant and nuisance sensitivities $s_{c,i}$ and $s_{n,i}$:

$$s_{c,i} = d(\tau_i, \tau_i^c), \qquad s_{n,i} = d(\tau_i, \tau_i^n). \tag{2}$$

In our experiments, d is mean corresponding-waypoint displacement, as summarized in Section 4.3.

Sensitivity magnitude alone, however, does not tell us whether the policy changes its action in an appropriate direction. To evaluate response direction, we define a task-specific response statistic $r_c(\tau_i, \tau_i^c)$ and an acceptable response set $\mathcal{D}_c$. The statistic describes how the action changes in terms relevant to capability c, while $\mathcal{D}_c$ specifies which changes are considered task-appropriate. For a paired probe, $a_{c,i}$ records whether the response satisfies the rule:

$$a_{c,i} = \mathbf{1}\{r_c(\tau_i, \tau_i^c) \in \mathcal{D}_c\}, \tag{3}$$

where $\mathbf{1}\{\cdot\}$ equals one when the condition holds and zero otherwise. The response statistic and acceptable set are specified for each capability before evaluation; the experimental measures are summarized in Section 4.3.

Let $\mathcal{I}_c$ and $\mathcal{I}_n$ denote the valid cases for capability-relevant and nuisance changes, respectively. We summarize the corresponding responses as

$$\begin{aligned} S_c &= \frac{1}{|\mathcal{I}_c|} \sum_{i \in \mathcal{I}_c} s_{c,i}, \quad S_n = \frac{1}{|\mathcal{I}_n|} \sum_{i \in \mathcal{I}_n} s_{n,i}, \\ A_c &= \frac{1}{|\mathcal{I}_c|} \sum_{i \in \mathcal{I}_c} a_{c,i}. \end{aligned} \tag{4}$$

Here, S_c summarizes the response magnitude to capability-relevant changes, A_c summarizes how often those responses follow the expected direction, and S_n measures sensitivity to nuisance variation. We collect these three quantities in the grounding profile

$$G_c(\pi_\theta) = (S_c, A_c, S_n). \tag{5}$$

The components of the profile are interpreted jointly. A policy may respond strongly to relevant changes (large S_c) while responding in the wrong direction (low A_c) or remaining highly sensitive to irrelevant variation (large S_n).

Matched capability supervision. Capability use can also be examined by changing how capability-related information is introduced during training. If additional supervision is intended to strengthen a particular capability, a useful question is whether that supervision ultimately changes the policy's corresponding behavior.

We isolate this effect with a matched pair of training runs that differ only in the added capability-specific loss. Let θ_{init} be their common initialization, $\mathcal{D}$ the training dataset, $\mathcal{L}_{\text{action}}$ the action loss,

$\mathcal{L}_c$ the added capability loss, and λ_c its weight. Let Ω collect the intentionally matched data order, augmentations, optimizer and schedule, training budget, action head, and deployed inference graph. The training procedure Train returns parameters for the action-only baseline, θ_0, and supervised variant, θ_c:

$$\begin{aligned}\theta_0 &= \mathrm{Train}(\theta_{\mathrm{init}}, \mathcal{D}, \mathcal{L}_{\mathrm{action}}; \Omega),\\ \theta_c &= \mathrm{Train}(\theta_{\mathrm{init}}, \mathcal{D}, \mathcal{L}_{\mathrm{action}} + \lambda_c \mathcal{L}_c; \Omega).\end{aligned} \tag{6}$$

This matched comparison tests the added supervision while holding the surrounding protocol fixed. In our geometry experiment, $\mathcal{L}_c = \mathcal{L}_{\mathrm{depth}}$ is sparse metric-depth supervision; the instantiated comparison is summarized in Section 4.2.

Before attributing any behavioral difference to $\mathcal{L}_c$, we verify that this loss can influence representations used by planning. Let h_{shared} denote such a representation; the geometry check includes shared video-token states. We measure the gradient

$$\gamma_c = \left\| \frac{\partial \mathcal{L}_c}{\partial h_{\mathrm{shared}}} \right\|. \tag{7}$$

A nonzero γ_c indicates that the added supervision has a gradient path to states used by the planner. The parity and gradient checks are reported in Appendix A.2.

We then measure its behavioral effect using the grounding profile defined above:

$$\Delta G_c = G_c(\pi_{\theta_c}) - G_c(\pi_{\theta_0}). \tag{8}$$

The subtraction is componentwise, with sensitivity, response direction, and nuisance dependence interpreted separately. Together with the gradient check, this comparison distinguishes supervision access from changes in capability-related behavior.

Predictive target comparisons. When predictive supervision improves planning, the gain does not by itself reveal which properties of the prediction target are responsible. We use three matched contrasts to test the roles of the predictive objective, scene-related target content, and exact temporal correspondence.

In separate matched run pairs, we compare correct future supervision F against three controls. N removes future supervision, testing whether the predictive objective helps. $SHUF$ replaces the target with one from another log, breaking its correspondence to the current scene while retaining the predictive objective. WT uses a different time from the same log, preserving nearby scene context while breaking exact temporal correspondence. The comparison design is summarized in Section 4.2.

Let $J_F^{(b)}$ and $J_C^{(b)}$ denote the held-out planning performance of the future-supervised model and its matched control C in comparison b. We define

$$\begin{aligned}\Delta^{(b)}_{\mathrm{obj}} &= J_F^{(b)} - J_N^{(b)},\\ \Delta^{(b)}_{\mathrm{content}} &= J_F^{(b)} - J_{\mathrm{SHUF}}^{(b)},\\ \Delta^{(b)}_{\mathrm{time}} &= J_F^{(b)} - J_{\mathrm{WT}}^{(b)}.\end{aligned} \tag{9}$$

The three contrasts test objective presence, scene-related content, and exact timing within their respective matched comparisons. Each contrast is interpreted within its own matched comparison.

3.3 Capability Grounding and Executed Driving

CGA evaluates how capability-relevant information affects planned actions, while closed-loop evaluation measures the outcomes produced when a policy is repeatedly executed. To study the relationship between these two levels of behavior, we evaluate the same policies in closed loop and analyze the resulting driving outcomes separately from the grounding profile G_c.

Let κ be the fixed controller, e_i a scenario, and u a policy index with parameters θ_u. The simulator repeatedly queries the policy and executes its plans through κ, producing the state trajectory $\xi_{i,u}$. A

measurement q, such as time to collision (TTC), separation, or progress, gives outcome $y_{i,u}$:

$$\xi_{i,u} = \text{Simulate}(e_i; \pi_{\theta_u}, \kappa), \qquad y_{i,u} = q(\xi_{i,u}). \tag{10}$$

These outcome measures are used to examine how capability-grounding patterns relate to executed driving behavior. The execution protocol and measured outcomes are summarized in Section 4.

4 Experimental Setup

4.1 Models and Evaluation Overview

We instantiate the evaluations in Section 3 using four driving policies with complementary roles. DDV2 (Zou et al., 2025) serves as an end-to-end planning reference, while AutoVLA (Zhou et al., 2025) and ReCogDrive-Base-IL (hereafter ReCogDrive) (Li et al., 2026b) provide independently trained VLA policies for cross-model semantic and spatial response analysis. AutoVLA is additionally used for the matched geometry-supervision study, where two training variants differ in whether metric-depth supervision is added. DriveVLA-W0 (Li et al., 2026a) provides an explicit predictive objective whose future targets can be modified while retaining the surrounding planning setup. The three policies used for input-response analysis are also evaluated in closed loop to examine how their capability-grounding diagnostics relate to executed driving behavior.

4.2 Controlled Comparisons

The controlled-input scenes, matched geometry training, predictive-target comparisons, and closed-loop execution are specified in Appendices C.2, C.3, C.4, and B.1, respectively. Appendix B summarizes the controls used in each comparison.

4.3 Evaluation Metrics and Statistical Analysis

Trajectory sensitivity is measured by mean corresponding-waypoint displacement. For paired semantic and nuisance interventions, displacement is computed between the paired outputs; for ordered spatial interventions, it compares the lowest- and highest-risk outputs. Semantic correctness, RRA, MVR, URR, planning and execution outcomes, and experiment-specific statistical procedures are defined in Appendix C.5.

5 Experiments

5.1 Response Selectivity under Controlled Input Changes

We examine whether policy responses to controlled input changes are both selective and task-aligned. We consider two settings: semantic changes, where traffic meaning is contrasted with an appearance-only nuisance intervention, and spatial changes, where geometric risk is progressively increased.

For semantic interventions, the three policies differ not only in how strongly they respond, but also in how selectively they respond. AutoVLA shows substantial trajectory changes for traffic-light and emergency-role interventions, but its mean displacement under the color-only nuisance change is even larger than under either semantic change. On paired valid cases, its nuisance displacement exceeds DDV2 by 1.243 m (95% CI [0.159, 2.917]). This pattern indicates that a large response to a semantic change can coexist with strong sensitivity to task-irrelevant appearance. ReCogDrive shows a different profile: it has lower nuisance sensitivity and the highest traffic-light correctness, but its emergency-role correctness remains low. Individual traffic-light and emergency-light examples are shown in Appendix Figures 5 and 6.

The spatial interventions show the same separation between response magnitude and response direction. AutoVLA's lane trajectory sensitivity exceeds ReCogDrive's by 3.929 m (95% CI [1.508, 6.623]), yet ReCogDrive has a 0.083 higher safety-margin RRA (95% CI [0.017, 0.167]). A larger trajectory change therefore does not correspond to more consistent adaptation to increasing lane intrusion. Appendix Figure 4(c) shows the underlying lane responses: AutoVLA increasingly reduces

Table 1: Cross-model grounding profiles. Arrows indicate the intended diagnostic direction. Trajectory sensitivity and target-distance URR have no ranking arrow; their interpretation depends on response direction and consequence.

Capability	Metric	DDV2	AutoVLA	ReCogDrive
Semantic	Traffic-light correct response ↑	.30	.50	.55
Semantic	Emergency correct response ↑	.15	.35	.10
Nuisance	Color displacement ↓ (m)	.175	1.429	.105
Spatial–Lane	Traj. sensitivity (m)	.600	4.401	.472
Spatial–Lane	Safety-margin RRA ↑	.833	.917	1.000
Spatial–Lane	MVR ↓	.167	.083	.000
Spatial–Target distance	Safety-margin RRA ↑	.825	.875	.950
Spatial–Target distance	URR	.65	.50	.55

speed as intrusion grows, ReCogDrive exhibits a smaller final-level reduction, and DDV2 changes speed little. The target-distance intervention shows a related mismatch: as the target moves from 28 to 10 m, all three policies increase their planned mean speed, while their alignment and under-reaction diagnostics differ (Table 1). Together, these spatial results show that the magnitude of a response and whether it follows the expected risk direction are distinct properties.

Figure 3 illustrates this distinction in a single controlled scene: the three policies' trajectories are compared after lane intrusion and, separately, a vehicle-color change from the same baseline. Appendix Figure 7 shows one target-distance sequence.

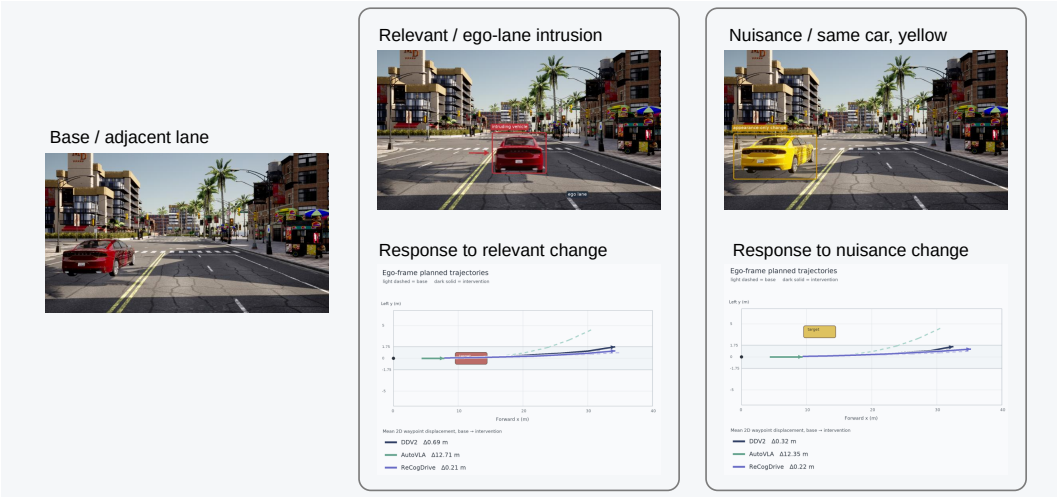

Figure 3: Qualitative responses to lane intrusion and vehicle color in one controlled scene. The adjacent-lane input is the shared baseline: the target is moved into the ego lane for the relevant change, or its color is changed locally at the baseline position for the appearance-only change. The lower panels show each policy's baseline (light dashed) and intervention (dark solid) ego-frame trajectories; Δ is their mean displacement over eight corresponding waypoints.

5.2 Effect of Geometry Supervision on Spatial Responses

The parity and gradient checks are reported in Appendix A.2.

At the fixed final training endpoint, GEOM does not show better directional responses to increasing spatial risk (Appendix Table A3). For lane intrusion, GEOM reduces trajectory sensitivity relative to BASE by 0.649 m (95% CI [0.017, 1.629] in magnitude), while safety-margin RRA changes by -0.0167 (95% CI $[-0.0833, 0.0500]$) and MVR by $+0.0167$ (95% CI $[-0.0500, 0.0833]$). For

target distance, trajectory sensitivity decreases by 0.688 m, while safety-margin RRA changes by -0.0500 and MVR by $+0.0500$; URR decreases only slightly. Color-nuisance displacement also changes little.

Overall, depth supervision reduces the magnitude of some spatial responses but shows no improvement in their directional consistency with increasing risk under this matched protocol.

5.3 Attributing Gains from Predictive Supervision

We evaluate whether gains from predictive supervision depend on the predictive objective itself, the scene-related content of its targets, or their exact temporal correspondence.

At the endpoint reached after 5% of the prescribed optimizer steps in each training stage, correct future supervision improves PDMS over removing future labels by $+0.05164$ (95% CI [$+0.03036$, $+0.07402$]; Appendix Table A5). The gain also depends on the content of the predictive targets. Replacing the correct targets with cross-log shuffled targets reduces performance in all three training seeds, with paired FUTURE–SHUF differences of $+0.07528$, $+0.04003$, and $+0.06422$, and all three confidence intervals remaining above zero. Future prediction is therefore useful in this setting, and its benefit is not reproduced by retaining the predictive objective with mismatched cross-log target content.

Cross-log shuffling changes both scene correspondence and temporal correspondence, so we further isolate the role of timing using same-log wrong-time targets. This control preserves nearby scene-related content while replacing the target time; mapping details are given in Appendix C.4. Under the matched FUTURE–WT comparison, the PDMS difference is only $+0.002269$ (95% CI [-0.02008, $+0.02645$]). We therefore find no detectable additional benefit from exact future-time correspondence under the tested training budget. The interval spans zero, so this experiment does not resolve an additional timing effect at the tested budget.

Inspection of the deployed inference path shows that predicted future states are not consumed by the planner. Together, the target comparisons and inference-path inspection are consistent with the gain arising during training rather than from online consumption of predicted future states.

5.4 Capability Grounding and Closed-Loop Outcomes

We compare the capability-grounding diagnostics of DDV2, AutoVLA, and ReCogDrive with closed-loop TTC, separation, progress, and speed on the lane and target-distance scenarios under the common controller described in Appendix B.1. The full associations are reported in Appendix Table A7.

Unlike the weak lane associations, the target-distance setting shows a different pattern. URR-marked episodes have higher TTC (+0.104 s, 95% CI [$+0.006$, $+0.199$]) and larger separation (+0.624 m, 95% CI [$+0.071$, $+1.119$]), while also making less progress at lower speed. Here, weaker planned responses to increasing geometric risk are associated with more conservative closed-loop execution.

Overall, the relationship between capability-grounding diagnostics and closed-loop outcomes varies across the tested scenarios: it is weak in the lane setting and changes direction toward more conservative execution in the target-distance setting.

6 Conclusion

We introduce CGA to test whether claimed semantic, geometric, and predictive capabilities guide a driving policy's actions. Controlled comparisons show that large responses can coexist with nuisance sensitivity, depth supervision can reach planning-shared states without improving spatial response alignment, and predictive gains depend on scene-related targets without a detectable additional benefit from exact future timing under the tested budget. Closed-loop associations vary across scenarios. Learning useful information, using it to plan, and driving well when executed therefore require distinct evidence.

Reproducibility Statement

Appendices B and C specify the intervention definitions, model interfaces, scene construction, matched training settings, metric definitions, and statistical procedures used in our experiments. They also describe the shared controller and simulator settings used for closed-loop evaluation.

AI Use Statement

We used generative AI tools to aid writing and polish the manuscript, to support literature retrieval and discovery, and to assist in creating and refining schematic illustration assets. The authors take full responsibility for the final content of this work, including its text, claims, and references.

References

Harisankar Babu, Benjamin Coors, Christopher Lang, Hendrik Berkemeyer, Tamim Asfour, and Simon Foell. Depth-wise probing and pruning of the planning token in a driving vision-language-action model. *arXiv preprint arXiv:2608.07361*, 2026. doi: 10.48550/arXiv.2608.07361. URL https://arxiv.org/abs/2608.07361. Accepted at the ECCV 2026 DriveX Workshop.

Jie Cheng, Yingbing Chen, Xiaodong Mei, Bowen Yang, Bo Li, and Ming Liu. Rethinking imitation-based planners for autonomous driving. In *2024 IEEE International Conference on Robotics and Automation*, pp. 14123–14130, 2024. doi: 10.1109/ICRA57147.2024.10611364. URL https://doi.org/10.1109/ICRA57147.2024.10611364.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. NAVSIM: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0902. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/32768f7faf1995026ef9821c696f3404-Abstract-Datasets_and_Benchmarks_Track.html.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pp. 1–16. PMLR, 2017. URL https://proceedings.mlr.press/v78/dosovitskiy17a.html.

Asher J. Hancock, Allen Z. Ren, and Anirudha Majumdar. Run-time observation interventions make vision-language-action models more visually robust. *arXiv preprint arXiv:2410.01971*, 2024. doi: 10.48550/arXiv.2410.01971. URL https://arxiv.org/abs/2410.01971.

Jingtao He, Hongliang Lu, Xiaoyun Qiu, Yixuan Wang, and Xinhu Zheng. Does visual information play a decisive role in vision-language-action model driving behavior? *arXiv preprint arXiv:2605.31041*, 2026. doi: 10.48550/arXiv.2605.31041. URL https://arxiv.org/abs/2605.31041.

Minqing Huang, Yujiao Xiang, Zihan Liang, Jiajie Huang, Jingqi Wang, Zhi Xu, Feiyang Tan, Hangning Zhou, Mu Yang, and Gong Che. CoWorld-VLA: Thinking in a multi-expert world model for autonomous driving. *arXiv preprint arXiv:2605.10426*, 2026a. doi: 10.48550/arXiv.2605.10426. URL https://arxiv.org/abs/2605.10426.

Yi Huang, Zhan Qu, Lihui Jiang, Bingbing Liu, and Hongbo Zhang. Prioritizing perception-guided self-supervision: A new paradigm for causal modeling in end-to-end autonomous driving. In *Advances in Neural Information Processing Systems*, volume 38, pp. 49956–49982. Curran Associates, Inc., 2025. doi: 10.52202/085713-1666. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/476c9ac41c967332dd28fb844e166b34-Paper-Conference.pdf.

Zhao Huang, Bowen Zhang, Zhongzhu Li, and Di Lin. OccDriver: Future occupancy guided dual-branch trajectory planner in autonomous driving. In *International Conference on Learning Representations*, pp. 141940–141962, 2026b. URL

https://proceedings.iclr.cc/paper_files/paper/2026/file/e59bf9a9077c361ed57180fc79c5b8d8-Paper-Conference.pdf.

Xiaosong Jia, Yuqian Shao, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2Drive-VL: Benchmarks for closed-loop autonomous driving with vision-language models. *arXiv preprint arXiv:2604.01259*, 2026. doi: 10.48550/arXiv.2604.01259. URL https://arxiv.org/abs/2604.01259.

Yingyan Li, Lue Fan, Jiawei He, Yuqi Wang, Yuntao Chen, Zhaoxiang Zhang, and Tieniu Tan. Enhancing end-to-end autonomous driving with latent world model. In *International Conference on Learning Representations*, pp. 42942–42959, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/6aa4967920e495e90aeeaa3acf18d019-Paper-Conference.pdf.

Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, Lu Hou, Lue Fan, and Zhaoxiang Zhang. DriveVLA-W0: World models amplify data scaling law in autonomous driving. In *International Conference on Learning Representations*, pp. 7890–7911, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/0d70423f59c5fdd24f0dd3fa52e34623-Paper-Conference.pdf.

Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, Wenyu Liu, and Xinggang Wang. ReCogDrive: A reinforced cognitive framework for end-to-end autonomous driving. In *International Conference on Learning Representations*, pp. 157518–157556, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/ff7bf6014f7826da531aa50f4538ee19-Paper-Conference.pdf.

Yongkang Li, Lijun Zhou, Sixu Yan, Bencheng Liao, Tianyi Yan, Kaixin Xiong, Long Chen, Hongwei Xie, Bing Wang, Guang Chen, Hangjun Ye, Wenyu Liu, Haiyang Sun, and Xinggang Wang. UniDriveVLA: Unifying understanding, perception, and action planning for autonomous driving. *arXiv preprint arXiv:2604.02190*, 2026c. doi: 10.48550/arXiv.2604.02190. URL https://arxiv.org/abs/2604.02190.

Yue Li, Meng Tian, Zhenyu Lin, Jiangtong Zhu, Dechang Zhu, Haiqiang Liu, Yueyi Zhang, Zhiwei Xiong, and Xinhai Zhao. Fine-grained evaluation of large vision-language models in autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9431–9442, October 2025b. URL https://openaccess.thecvf.com/content/ICCV2025/html/Li_Fine-Grained_Evaluation_of_Large_Vision-Language_Models_in_Autonomous_Driving_ICCV_2025_paper.html.

Yue Li, Meng Tian, Dechang Zhu, Jiangtong Zhu, Zhenyu Lin, Zhiwei Xiong, and Xinhai Zhao. Drive-R1: Bridging reasoning and planning in VLMs for autonomous driving with reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(8):6708–6716, 2026d. doi: 10.1609/aaai.v40i8.37602. URL https://ojs.aaai.org/index.php/AAAI/article/view/37602.

Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahan Li, Jan Kautz, Tong Lu, and Jose M. Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14864–14873, June 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Li_Is_Ego_Status_All_You_Need_for_Open-Loop_End-to-End_Autonomous_CVPR_2024_paper.html.

Yuechen Luo, Fang Li, Shaoqing Xu, Yang Ji, Zehan Zhang, Bing Wang, Yuannan Shen, Jianwei Cui, Long Chen, Guang Chen, Hangjun Ye, Zhi-Xin Yang, and Fuxi Wen. LaST-VLA: Thinking in latent spatio-temporal space for vision-language-action in autonomous driving. *arXiv preprint arXiv:2603.01928*, 2026. doi: 10.48550/arXiv.2603.01928. URL https://arxiv.org/abs/2603.01928.

Thach Nguyen, Danhua Guo, Tom Lampo, Fei Wu, and Burhan Yaman. VLADriveBench: Evaluating CoT-action relationship in VLA for autonomous driving. *arXiv preprint arXiv:2606.12706*, 2026. doi: 10.48550/arXiv.2606.12706. URL https://arxiv.org/abs/2606.12706.

Kalpana Panda, Wesley Maia, Vinti Agarwal, and Ross Greer. What do they see? interpreting complex road scenarios through the eyes of vision-language-action models for safe and trustworthy autonomous vehicle learning. *arXiv preprint arXiv:2607.16938*, 2026. doi: 10.48550/arXiv.2607.16938. URL https://arxiv.org/abs/2607.16938.

Shuyao Shang, Bing Zhan, Yunfei Yan, Yuqi Wang, Yingyan Li, Yasong An, Xiaoman Wang, Jierui Liu, Lu Hou, Lue Fan, Zhaoxiang Zhang, and Tieniu Tan. DynVLA: Learning world dynamics for action reasoning in autonomous driving. *arXiv preprint arXiv:2603.11041*, 2026. doi: 10.48550/arXiv.2603.11041. URL https://arxiv.org/abs/2603.11041.

Haoyuan Shi, Xiancong Ren, Yingji Zhang, Qinfan Zhang, Jiayu Hu, Haozhe Shan, Han Dong, Jinpeng Lu, Yinda Chen, Yi Zhang, Yong Dai, and Xiaozhu Ju. VLA-Trace: Diagnosing vision-language-action models through representation and behavior tracing. *arXiv preprint arXiv:2605.30117*, 2026. doi: 10.48550/arXiv.2605.30117. URL https://arxiv.org/abs/2605.30117.

Jiacheng Tang, Mingyue Feng, Jiachao Liu, Yaonong Wang, and Jian Pu. Decoupling scene perception and ego status: A multi-context fusion approach for enhanced generalization in end-to-end autonomous driving. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40 (11):9413–9420, 2026. doi: 10.1609/aaai.v40i11.37901. URL https://ojs.aaai.org/index.php/AAAI/article/view/37901.

Kexin Tian, Jingrui Mao, Yunlong Zhang, Jiwan Jiang, Yang Zhou, and Zhengzhong Tu. NuScenes-SpatialQA: A spatial understanding and reasoning benchmark for vision-language models in autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 4626–4635, October 2025. URL https://openaccess.thecvf.com/content/ICCV2025W/WDFM-AD/html/Tian_NuScenes-SpatialQA_A_Spatial_Understanding_and_Reasoning_Benchmark_for_Vision-Language_Models_ICCVW_2025_paper.html.

Hao Vo, Khoa Vo, Phu Loc Nguyen, Sieu Tran, Duc Minh Nguyen, Ngo Xuan Cuong, Gladys Gawugah, Sreevenkata Anjani Tishita Godavarthi, Chase Rainwater, Nghi D. Q. Bui, Anh Nguyen, Duy Minh Ho Nguyen, and Ngan Le. DRIVESPATIAL: A benchmark for spatiotemporal intelligence in VLMs for autonomous driving. *arXiv preprint arXiv:2605.23176*, 2026. doi: 10.48550/arXiv.2605.23176. URL https://arxiv.org/abs/2605.23176.

Guoqing Wang, Pin Tang, Xiangxuan Ren, Guodongfang Zhao, Bailan Feng, and Chao Ma. Learning vision-language-action world models for autonomous driving. *arXiv preprint arXiv:2604.09059*, 2026a. doi: 10.48550/arXiv.2604.09059. URL https://arxiv.org/abs/2604.09059.

Xinyang Wang, Qian Liu, Wenjie Ding, Zhao Yang, Wei Li, Chang Liu, Bailin Li, Kun Zhan, Xianpeng Lang, and Wei Chen. Unifying language-action understanding and generation for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25193–25203, June 2026b. URL https://openaccess.thecvf.com/content/CVPR2026/html/Wang_Unifying_Language-Action_Understanding_and_Generation_for_Autonomous_Driving_CVPR_2026_paper.html.

Yiru Wang, Anqing Jiang, Shuo Wang, Yuwen Heng, Hai Yang, Yang Chen, and Hao Sun. Do open-loop metrics predict closed-loop driving? a cross-benchmark correlation study of NAVSIM and Bench2Drive. *arXiv preprint arXiv:2605.00066*, 2026c. doi: 10.48550/arXiv.2605.00066. URL https://arxiv.org/abs/2605.00066.

Shaoyuan Xie, Lingdong Kong, Yuhao Dong, Chonghao Sima, Wenwei Zhang, Qi Alfred Chen, Ziwei Liu, and Liang Pan. Are VLMs ready for autonomous driving? an empirical study from the reliability, data and metric perspectives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6585–6597, October 2025.

URL https://openaccess.thecvf.com/content/ICCV2025/html/Xie_Are_VLMs_Ready_for_Autonomous_Driving_An_Empirical_Study_from_ICCV_2025_paper.html.

Jianguo Yu, Rukang Wang, Duanfeng Chu, Chen Wang, Renju Feng, and Liping Lu. S-squared-VLA: Decoupling semantic and spatial streams in vision-language-action models for autonomous driving. *arXiv preprint arXiv:2607.13926*, 2026. doi: 10.48550/arXiv.2607.13926. URL https://arxiv.org/abs/2607.13926.

Jiang-Tian Zhai, Ze Feng, Jinhao Du, Yongqiang Mao, Jiang-Jiang Liu, Zichang Tan, Yifu Zhang, Xiaoqing Ye, and Jingdong Wang. Rethinking the open-loop evaluation of end-to-end autonomous driving in nuScenes. *arXiv preprint arXiv:2305.10430*, 2023. doi: 10.48550/arXiv.2305.10430. URL https://arxiv.org/abs/2305.10430.

Hanxin Zhang, Mingshuo Xu, Abdulqader Dhafer, Shigang Yue, Hongbiao Dong, and Zhou Daniel Hao. Embodied interpretability: Linking causal understanding to generalization in vision-language-action models. In *International Conference on Machine Learning*, 2026a. URL https://arxiv.org/abs/2605.00321. Accepted paper.

Yichi Zhang, Weihao Yuan, Yizhuo Zhang, Xidong Zhang, and Jia Wan. FocusVLA: Focused visual utilization for vision-language-action models. *arXiv preprint arXiv:2603.28740*, 2026b. doi: 10.48550/arXiv.2603.28740. URL https://arxiv.org/abs/2603.28740.

Ruiguo Zhong, Benshan Ma, Xiaolong Chen, Lang Zhang, Mingyue Feng, Yaonong Wang, Pei Liu, and Jun Ma. DA-WAM: Decision-aligned future latents for driving world models. *arXiv preprint arXiv:2608.19085*, 2026. doi: 10.48550/arXiv.2608.19085. URL https://arxiv.org/abs/2608.19085.

Xingcheng Zhou, Xuyuan Han, Feng Yang, Yunpu Ma, Volker Tresp, and Alois Knoll. OpenDriveVLA: Towards end-to-end autonomous driving with large vision language action model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(16):13782–13790, 2026. doi: 10.1609/aaai.v40i16.38386. URL https://ojs.aaai.org/index.php/AAAI/article/view/38386.

Zewei Zhou, Tianhui Cai, Seth Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. AutoVLA: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 27920–27956. Curran Associates, Inc., 2025. doi: 10.52202/085713-0942. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/2843fccca5bedd369a4764848b9bd546-Paper-Conference.pdf.

Jialv Zou, Shaoyu Chen, Bencheng Liao, Zhiyu Zheng, Yuehao Song, Lefei Zhang, Qian Zhang, Wenyu Liu, and Xinggang Wang. DiffusionDriveV2: Reinforcement learning-constrained truncated diffusion modeling in end-to-end autonomous driving. *arXiv preprint arXiv:2512.07745*, 2025. doi: 10.48550/arXiv.2512.07745. URL https://arxiv.org/abs/2512.07745.

A Additional Experimental Results

This appendix provides additional results for the controlled-input, geometry-supervision, predictive-target, and closed-loop experiments, together with implementation checks used in the geometry and closed-loop analyses.

A.1 Additional Results for Controlled Input Changes

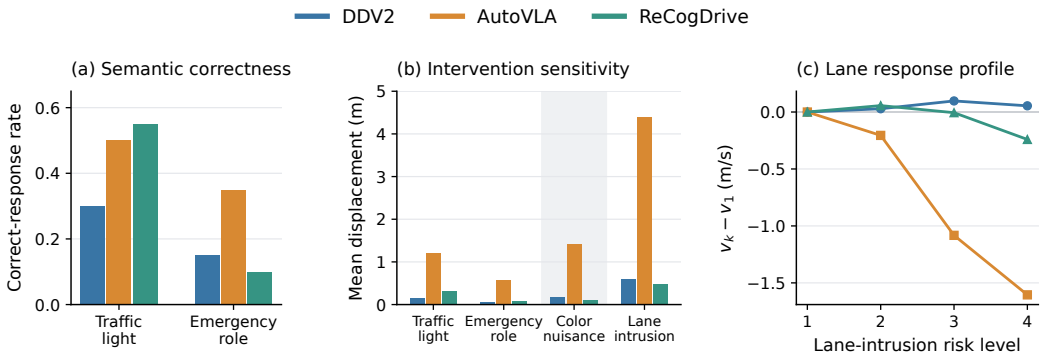

Figure 4: Relevant and nuisance responses across policies. (a) Semantic correct-response rates. (b) Mean waypoint displacement; the shaded color-nuisance category changes task-irrelevant appearance. Lane displacement compares levels 1 and 4. (c) Mean within-case speed change $v_k - v_1$ across adjacent lane (1), lane boundary (2), partial intrusion (3), and ego-lane occupancy (4). All groups contain 20 valid pairs/cases per model except AutoVLA color nuisance (18 valid pairs). Curves average the four-level responses of the same 20 cases; no endpoint interpolation is used. Table 1 provides the numerical profiles.

Table A1 reports response magnitudes and signed spatial changes that are not included in the main grounding-profile table. AutoVLA shows the largest semantic displacements and substantially larger signed lane responses than DDV2, while ReCogDrive shows the smallest speed increase in the target-distance intervention. Table A2 reports additional paired cross-model contrasts for the controlled-input evaluation. These additional measures are consistent with the main-text result that response magnitude alone does not determine whether the response follows the expected driving direction. Figures 5 and 6 illustrate individual semantic pairs; Figure 7 shows the ordered target-distance levels in one spatial case. The images and trajectories are examples, whereas the tables summarize the controlled-input evaluation across cases.

Table A1: Additional response measures and signed spatial changes. Lane changes compare ego-lane occupancy with the adjacent-lane condition; target-distance changes compare 10 m with 28 m.

Metric	DDV2	AutoVLA	ReCogDrive
Traffic-light displacement (m)	.157	1.201	.313
Emergency-role displacement (m)	.063	.564	.087
Target-distance mean-speed change (m/s)	+.572	+.462	+.091

Lane response	DDV2	AutoVLA
Mean-speed change (m/s)	+.054	-1.605
Progress change (m)	+.173	-6.382
Lateral-position change (m)	+.688	-.444

Table A2: Paired cross-model contrasts for the controlled-input evaluation. Effects retain the subtraction direction shown; intervals are 95% bootstrap confidence intervals.

Contrast	Estimate	95% CI
Traffic-light correctness: AutoVLA − DDV2	+.20	$[-.10, +.50]$
Emergency correctness: AutoVLA − DDV2	+.20	$[-.05, +.45]$
Color displacement: ReCogDrive − AutoVLA (m)	-1.327	$[-2.962, -.266]$
Lane sensitivity: AutoVLA − DDV2 (m)	+3.801	$[+1.384, +6.398]$

A.2 Depth Supervision in Shared Representations

A necessary condition for added metric-depth supervision to affect planning is that its training signal reaches representations used by the planner. We test this condition in matched AutoVLA BASE and GEOM variants. At zero depth-loss weight, the attached geometry branch leaves action logits, language-model loss, and selected shared gradients unchanged. With depth supervision enabled, its loss produces nonzero gradients in the geometry head, shared Qwen parameters, and shared video-token states. The loss can therefore update planning-shared representations without changing the deployed action path (Table A4).

Table A3: Matched geometry intervention at the pre-specified 2,767-step endpoint. Effects are GEOM minus BASE. Both policies share initialization, training protocol, action head, and inference graph. A dash denotes an unreported interval.

Setting	Diagnostic	Difference	95% CI
Lane	Safety-margin RRA	$-.0167$	$[-.0833, +.0500]$
Lane	MVR	$+.0167$	$[-.0500, +.0833]$
Lane	URR	.0000	—
Lane	Trajectory sensitivity (m)	$-.649$	$[-1.629, -.017]$
Target distance	Safety-margin RRA	$-.0500$	$[-.1250, .0000]$
Target distance	MVR	$+.0500$	$[.0000, +.1250]$
Target distance	URR	$-.0500$	$[-.1500, .0000]$
Target distance	Trajectory sensitivity (m)	$-.688$	$[-1.954, +.002]$
Color	Nuisance displacement (m)	$-.192$	$[-.880, +.206]$

Table A4: Depth-supervision parity and gradient measurements.

Measurement	Value
Zero-weight parity	
Maximum action-logit difference	0
LM-loss difference	0
Selected shared-gradient difference	0
Geometry gradient reachability	
Geometry-head gradient norm	68.7287
Shared Qwen parameter gradient norm	.234565
Shared video-token representation gradient norm	.025033

A.3 Additional Predictive-Target Results

Table A5: Matched predictive-target comparisons for DriveVLA-W0. Effects are FUTURE minus control; 95% confidence intervals use a log-clustered bootstrap.

Contrast	Control target	Seed	ΔPDMS	95% CI
$F - N$	No future labels	—	$+.05164$	$[+.03036, +.07402]$
$F - \mathrm{SHUF}$	Cross-log shuffle	1	$+.07528$	$[+.04779, +.10252]$
$F - \mathrm{SHUF}$	Cross-log shuffle	2	$+.04003$	$[+.01431, +.06724]$
$F - \mathrm{SHUF}$	Cross-log shuffle	3	$+.06422$	$[+.03686, +.09097]$
$F - \mathrm{WT}$	Same-log wrong time	—	$+.002269$	$[-.020081, +.026445]$

Table A6 complements the endpoint PDMS contrasts in the main text with absolute scores, trajectory-error reductions, leave-one-log-out robustness, and checkpoint consistency. FUTURE improves over NO FUTURE in both PDMS and trajectory error, and its PDMS gain remains positive in every leave-one-log-out evaluation. The F–SHUF differences are positive at 4/5, 2/5, and 3/5 aligned checkpoints across the three seeds, showing that the endpoint advantage is not uniformly present throughout training.

Table A6: Predictive endpoints and robustness summaries. Checkpoint counts refer to aligned checkpoints.

Quantity	Value
F–N matched comparison	
FUTURE PDMS	.70391
NO_FUTURE PDMS	.65227
ADE reduction (m)	.14848
FDE reduction (m)	.29573
Minimum leave-one-log-out PDMS gain	+.04716
Checkpoints with positive PDMS differences	5/5
F–SHUF checkpoint consistency	
Positive checkpoints: seed 1	4/5
Positive checkpoints: seed 2	2/5
Positive checkpoints: seed 3	3/5
F–WT matched comparison	
FUTURE PDMS	.681446
SAME-LOG WRONG-TIME PDMS	.679177

A.4 Additional Closed-Loop Results

Table A7: Associations between planned-response diagnostics and closed-loop outcomes. URR effects compare highest-risk units with URR$= 1$ minus URR$= 0$; correlation rows report sequence-level Spearman ρ. Lane URR intervals span zero; target-distance URR-marked units have higher TTC and larger separation.

Open-loop diagnostic	Outcome	Setting	Effect / ρ	95% CI
URR	TTC	Lane	$+.024$ s	$[-.046, +.105]$
URR	Separation	Lane	$+.184$ m	$[-.260, +.678]$
URR	TTC	Target distance	$+.104$ s	$[+.006, +.199]$
URR	Separation	Target distance	$+.624$ m	$[+.071, +1.119]$
MVR	Separation	Lane	$-.003$	$[-.210, +.219]$
MVR	Separation	Target distance	$+.025$	$[-.214, +.265]$
Traj. sensitivity	TTC	Lane	$-.277$	$[-.552, +.070]$

In the lane setting, the grounding diagnostics are only weakly associated with closed-loop margins (Table A7). Episodes marked by URR differ only slightly in TTC (+0.024 s, 95% CI $[-0.046, +0.105]$) and separation (+0.184 m, 95% CI $[-0.260, +0.678]$). MVR is similarly weakly associated with separation, and trajectory sensitivity has a Spearman correlation of $\rho = -0.277$ (95% CI $[-0.552, +0.070]$) with TTC.

Table A8 reports additional closed-loop associations and cross-model outcome contrasts not included in Table A7. In the lane setting, the additional associations with progress and separation remain small. In the target-distance setting, URR-marked episodes make less progress and travel at lower speed, consistent with the more conservative execution reported in Section 5.4. Collision is not informative for these comparisons because no target collisions occur in the valid highest-risk episodes.

Table A8: Additional closed-loop outcome contrasts and diagnostic associations. URR rows compare highest-risk URR$=1$ minus URR$=0$ units; model-comparison rows use ReCogDrive minus DDV2. Correlation rows use Spearman ρ; all intervals are 95% CIs.

Setting	Comparison / outcome	Estimate	95% CI
Lane	URR: progress (m)	-.192	$[-.669, +.240]$
Lane	Safety-margin RRA–separation ρ	+.003	$[-.218, +.204]$
Lane	Sensitivity–separation ρ	-.228	$[-.496, +.105]$
Target distance	URR: progress (m)	-.616	$[-1.115, -.077]$
Target distance	URR: mean speed (m/s)	-.159	$[-.303, -.001]$
Target distance	Safety-margin RRA–separation ρ	-.025	$[-.272, +.218]$
Target distance	ReCogDrive − DDV2: progress (m)	+.263	$[+.024, +.476]$
Target distance	ReCogDrive − DDV2: separation (m)	-.246	$[-.481, +.006]$

A.5 Illustrative Controlled Cases

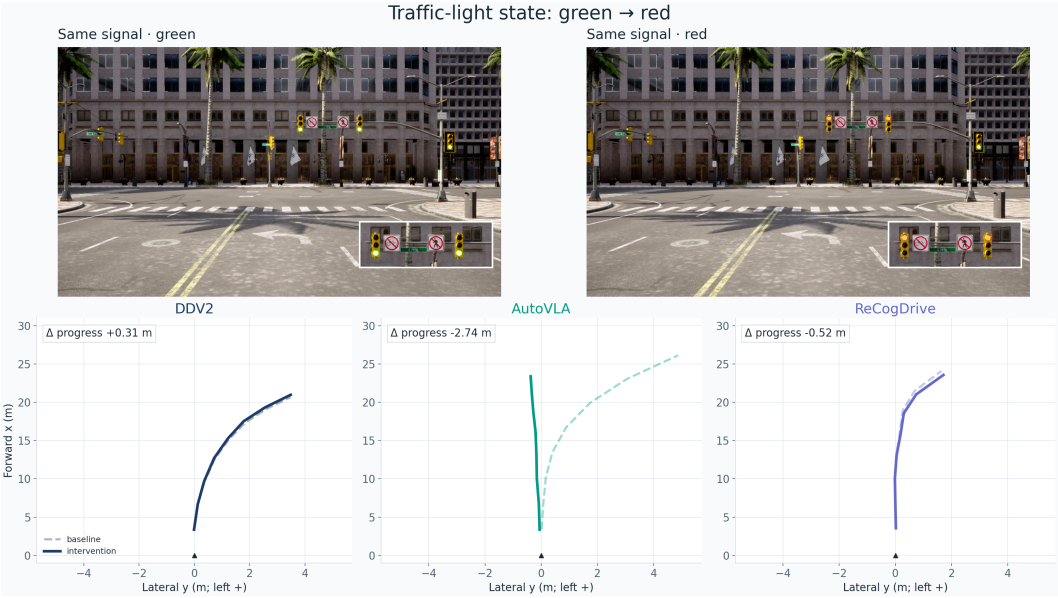

Figure 5: Illustrative traffic-light intervention. The same controlled scene is shown with the signal configured green (left) or red (right); the changed lights appear amber in the render, and the insets enlarge the cue. Below, light dashed and dark solid curves show each policy's baseline and changed-input eight-pose plan in the current ego frame over 4 s.

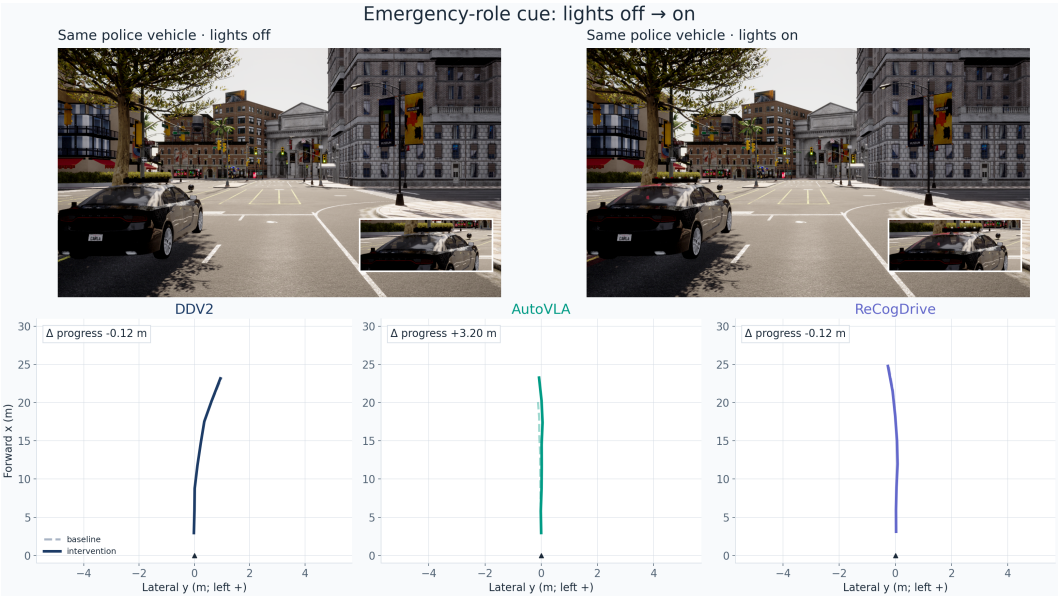

Figure 6: Illustrative emergency-role intervention. The same emergency-capable vehicle appears with its emergency lights disabled (left) or enabled (right); the insets enlarge the visual cue. Below, light dashed and dark solid curves show the baseline and changed-input eight-pose plans. Annotated progress differences are lights-on minus lights-off for this case; in particular, AutoVLA's forward progress increases here despite the prescribed yielding direction.

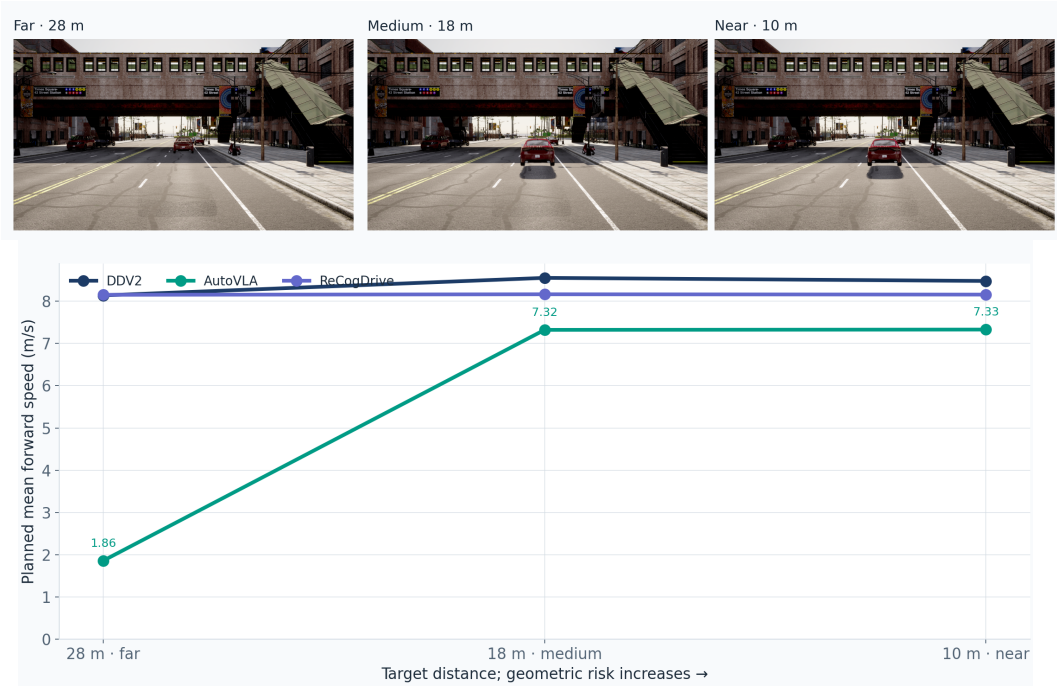

Figure 7: One controlled target-distance case at 28, 18, and 10 m: AutoVLA responds strongly but raises planned speed as the target moves closer, illustrating that response magnitude and direction can diverge.

B Control Design for CGA and Closed-Loop Evaluation

This appendix explains the controls behind the CGA interventions and the separate closed-loop evaluation. Table B1 summarizes what is changed, what remains matched, and the question each comparison addresses.

Table B1: Primary controls for CGA interventions and complementary closed-loop evaluation. Semantic/color and spatial inputs use separate scene-construction protocols; the predictive contrasts use separate matched run pairs.

Comparison	Manipulation	Main controls	What it tests
Semantic input	Traffic-light or vehicle-role state	Same actor; outside-mask RGB; paired LiDAR	Task-relevant semantic response
Color nuisance	Vehicle color	Blueprint; geometry; outside-mask RGB; paired LiDAR	Appearance sensitivity
Spatial input	Target lane position or distance	Ego state; target extent; common background	Response to geometric risk
BASE vs. GEOM	Added metric-depth loss	Initialization; data order; action path; training setup	Effect of depth supervision
Predictive targets	F–N, F–SHUF, F–WT	Pair-specific matched training	Objective, scene content, and timing
Predictive inference	Inference-path inspection	Reported deployed planner path	Online use of predicted future states
Closed loop	Repeated policy execution	Common controller; symmetric exclusion	Planned responses versus executed outcomes

B.1 Closed-Loop Execution Controls

DDV2, AutoVLA, and ReCogDrive are executed through the same pure-pursuit and speed PID controller at 20 Hz, with 2 Hz replanning, identical trajectory interpolation and actuator limits, a 0.05 s simulation step, a 4.0 s execution horizon, and no emergency override.

The evaluation initially contains 420 episodes across the lane and target-distance interventions. One unrelated collision occurs before planner action in a single base case; all 21 episodes associated with that base case across models and risk levels are excluded symmetrically. The reported analysis contains 399 episodes, with 133 episodes per model across 38 base-case clusters.

C Implementation and Reproducibility Details

This appendix provides the model interfaces, scene-construction parameters, training configurations, metric definitions, and statistical procedures used in the experiments.

C.1 Model Interfaces and Evaluation Settings

DDV2. DDV2 (Zou et al., 2025) uses the reinforcement-learning (RL) checkpoint and fixed trajectory candidate index zero, without NAVSIM scoring or ranking. Its latest left/front/right images are cropped by 28 pixels at the top and bottom; side views lose another 416 pixels at each horizontal edge. The combined image is resized to 1024×256 and normalized by 255. The input also includes a LiDAR histogram with 256 bins per horizontal axis over $[-32, 32]$ m; points satisfy $0.2 < z_{\mathrm{L}} < 100$ m, and bin counts are clipped at five and divided by five. An eight-entry status vector supplies the keep-forward command, longitudinal speed and acceleration, and zero lateral speed and acceleration. Torch is seeded to zero for each scene, and NumPy is seeded to zero before evaluation. The output is an 8×3 trajectory with planar waypoints at 0.5 s intervals in ego coordinates.

AutoVLA. For controlled-input evaluation, AutoVLA (Zhou et al., 2025) uses adaptive inference with one repeat and seed 20260811. The front/left/right cameras map to F0/L1/R1, alongside the ego state and command of the controlled scene. The matched geometry experiment initializes from the AutoVLA 3B PDMS-89 checkpoint. Evaluation uses eight decoded ego-frame poses at 0.5 s intervals.

ReCogDrive-Base-IL. ReCogDrive-Base-IL (Li et al., 2026b) uses the 2B vision-language model and IL diffusion planner with denoising diffusion implicit model (DDIM) sampling and inference seed 20260814. Its inputs are the latest front image and an eight-dimensional status vector with four command indicators, two velocity components, and two acceleration components. Because historical ego poses are unavailable in the evaluation inputs, we construct constant-velocity history at $-1.5, -1.0, -0.5, 0$ s from the observed speed and zero acceleration. The output is eight poses over four seconds in ego coordinates, with heading in radians. Invalid shapes or nonfinite outputs are excluded without a fallback trajectory.

DriveVLA-W0. DriveVLA-W0 (Li et al., 2026a) is the policy used for matched predictive-target comparisons. Its reported deployed action path takes observed camera tokens, navigation/history state, and flow noise through shared VLM/action layers, without consuming predicted future states online.

C.2 CARLA Intervention Construction

The semantic/color and spatial interventions use the CARLA (Dosovitskiy et al., 2017) `Town10HD_Opt` map with synchronous simulation at a 0.05 s fixed step, ClearNoon weather, zero wind, 1920×1080 cameras, a 64° field of view, postprocessing, and gamma 2.2. One captured observation is repeated four times as image history. The two intervention families use different scene-construction procedures.

Semantic and color pairs. Scene construction uses seed 20260811, the same traffic-manager seed, and sensor tick zero. Twenty traffic-light pairs switch the same light from green to red at a fixed location within each pair. The ego is placed nominally 12 or 16 m upstream along the lane from the road waypoint onto which the light's trigger volume is projected (13 and 7 pairs, respectively). Twenty emergency-role pairs place the same emergency-capable vehicle 10 m ahead and 3.2 m left of the ego vehicle; switching its emergency lights from off to on changes its visible role from ordinary to emergency, without siren audio. Twenty color pairs change only the target vehicle's paint by respawning the same vehicle model at the same geometry. The ego speed is 6–10 m/s with zero acceleration and a keep-forward command. The ego pose, camera setup, and surrounding scene are held fixed within each pair.

For formal comparison, target-local pixels from the changed render are copied into the baseline image. Outside-mask pixel parity therefore applies to this composite image. The baseline LiDAR scan is reused across each semantic or color pair. Vehicle masks use the projected target box dilated by 32 pixels; traffic-light masks use state-difference pixels in the central decision-relevant region dilated by 24 pixels. A valid pair requires at least 64 mask pixels, inside-mask difference at least 0.001, outside-mask difference at most 0.001, and outside-mask changed-pixel ratio at most 0.005, with an 8-bit changed-pixel threshold of four.

Spatial sequences. Construction uses seed 20260812, a common target-free RGB background, and common background LiDAR; target pixels and target LiDAR points change with target position. Lane cases keep the target 12 m ahead and move it laterally to 3.8, 2.2, 1.2, and 0 m in increasing risk order. Target-distance cases keep the lateral position at zero and place the target at 28, 18, and 10 m. Each axis has 20 cases, and target half-extents are approximately $(2.504, 0.941, 0.767)$ m. A valid scene requires at least 256 union-mask pixels, at least 16 target LiDAR points, outside-mask difference at most 0.001, and outside-mask changed-pixel ratio at most 0.005. The ego pose, target extent, and non-intervened coordinate remain fixed. Risk levels are ordered by simulator geometry before policy outputs are examined.

Spatial analyses require valid trajectories at every risk level. Two AutoVLA color pairs (indices 014 and 018) have invalid trajectory outputs and are excluded, leaving 18 valid color pairs for that model.

Cross-model contrasts use only case IDs valid for both models. The aggregate semantic/color and spatial evaluation sets are generated separately.

C.3 Matched Geometry Training

The matched BASE and GEOM runs initialize from the same AutoVLA (Zhou et al., 2025) 3B checkpoint and use seed 20260810 for training and target construction. The data configuration requires a route, four history frames, ten future frames, and frame interval four. The dataset contains 177,098 eligible samples from 1,310 logs; a deterministic global shuffle is applied before selecting the first 44,272 samples. Data order and augmentations are matched between the two variants. The comparison uses one training seed.

Training uses a global batch of 16 for 2,767 optimizer steps. Shared learning rate is 2×10^{-5}, geometry-head learning rate 2×10^{-4}, and weight decay 0.01. Training uses 500 warmup steps, learning-rate (LR) step frequency 2,000, LR multiplier 0.98, gradient clip value 1.0, bfloat16, and a LambdaLR schedule.

BASE uses direct-answer supervised cross-entropy for action prediction. GEOM adds masked SmoothL1 depth loss with weight 0.05. Targets derive from current and history NAVSIM (Dauner et al., 2024) LiDAR projected into F0, L1, and R1 using the AutoVLA camera calibration, with camera-axis depth z restricted to 1–80 m. Each camera's raw video grid is $2 \times 16 \times 30$ and merged grid $2 \times 8 \times 15$, yielding 720 target cells per sample across three cameras. A valid target is the median $\log(1 + z)$ of projected points within the corresponding temporal-pair/spatial cell. The final shared Qwen video-token hidden states have dimension 2,048; a training-only linear $2048 \to 1$ head predicts the targets. Raw visual parameters remain frozen while shared language-model states receive gradients. Depth targets are precomputed from LiDAR, which is not added to the AutoVLA action input. No semantic labels or future observations are used for the geometry targets.

The GEOM objective is

$$\mathcal{L}_{\mathrm{GEOM}} = \mathcal{L}_{\mathrm{action}} + \lambda_{\mathrm{geom}} \mathcal{L}_{\mathrm{depth}}, \qquad \lambda_{\mathrm{geom}} = 0.05. \tag{11}$$

Checkpoints are saved at steps 553, 1,107, 1,660, 2,214, and 2,767, with step 2,767 used as the fixed evaluation endpoint. The geometry head is removed when exporting the deployed model, leaving the action inference path unchanged.

C.4 Predictive-Target Comparisons and Inference Path

The predictive-target analysis uses three separate matched comparisons. FUTURE (F) uses correct future supervision; NO_FUTURE (N) removes future labels while preserving initialization, architecture, inputs, action labels, optimizer, schedule, sample order, capacity, and budget within its matched pair. SHUFFLED (SHUF) uses cross-log shuffled targets while preserving the future-token loss, supervised-token count, tensor shapes, masking, and compute. SAME-LOG WRONG-TIME (WT) uses targets from the same log but excludes the correct future instant.

Evaluation uses 703 NAVSIM tokens from 66 held-out logs. The F–SHUF study reports three independent training seeds, while the F–WT study is a strict same-seed pair. Both training stages use 5% of their respective maximum optimizer-step counts in the official DriveVLA-W0 schedule (Li et al., 2026a). Each matched comparison reports its fixed final endpoint, with aligned intermediate checkpoints used to examine consistency across training. Wrong-time targets are assigned within the same log for all 10,000 training samples, with zero correct-time overlap and a median offset of approximately 1.00 s. The evaluation count of 703 tokens is distinct from the training sample and future-token counts.

At inference, the deployed action path consumes observed camera tokens, navigation/history state, flow noise, and shared VLM/action layers. It does not consume a predicted-future tensor, world-model decoder, or future-state key/value cache.

C.5 Metrics and Statistical Procedures

Trajectory outputs and semantic rules. For this subsection we suppress the case index i. Let $k = 1, \ldots, K$ index the ordered intervention levels of a case and $j = 1, \ldots, H$ index its waypoints.

The planar waypoint $p_{k,j} = (X_{k,j}, Y_{k,j})$ has forward coordinate $X_{k,j}$ and leftward coordinate $Y_{k,j}$; the capital letters distinguish coordinates from the input x_i and outcome $y_{i,u}$. Here $H = 8$, $p_{k,0} = (0, 0)$, and the waypoint interval is $\Delta t = 0.5$ s. Let v_k be planned mean speed and P_k final forward progress:

$$v_k = \frac{1}{H} \sum_{j=1}^{H} \frac{\|p_{k,j} - p_{k,j-1}\|_2}{\Delta t}, \qquad P_k = X_{k,H}. \tag{12}$$

For two outputs τ and τ' with planar waypoints p_j and p'_j, respectively, the distance in equation 2 is

$$d(\tau, \tau') = \frac{1}{H} \sum_{j=1}^{H} \|p'_j - p_j\|_2. \tag{13}$$

Semantic and color pairs compare the intervention output with the corresponding baseline output, whereas spatial sensitivity compares the highest- and lowest-risk outputs. Maximum displacement takes the maximum over the same waypoint distances. For semantic pairs, define ΔP, Δv, and ΔY_H as intervention-minus-baseline changes in final progress, mean speed, and final lateral position. A traffic-light response is correct when $\Delta P \le -0.50$ m or $\Delta v \le -0.20$ m/s. Emergency correctness requires $\Delta P \le -0.50$ m or $-\Delta Y_H \ge 0.25$ m because the emergency vehicle is on the left. Color invariance requires mean displacement < 0.25 m and maximum displacement < 0.50 m. Rates average the indicators over valid pairs; these rules specify the task-dependent acceptable sets used in equation 3.

Risk and rectangular margin. The geometric-risk calculation uses a nominal ego half-length $a_e = 2.45$ m and half-width $b_e = 0.95$ m. Target half-length a_t and half-width b_t come from the target geometry. At waypoint coordinates (X, Y) and target center (X_t, Y_t), define axis gaps $g_X = |X - X_t| - (a_e + a_t)$ and $g_Y = |Y - Y_t| - (b_e + b_t)$. The signed axis-aligned rectangular margin m_{rect} is

$$m_{\text{rect}}(X, Y) = \begin{cases} \sqrt{\max(g_X, 0)^2 + \max(g_Y, 0)^2}, & g_X > 0 \text{ or } g_Y > 0, \\ \max(g_X, g_Y), & \text{otherwise.} \end{cases} \tag{14}$$

Write $m_{\text{rect},k}$ for this function evaluated against level-k target geometry. Define the raw minimum margin M_k and adaptation compensation C_k as

$$M_k = \min_j m_{\text{rect},k}(p_{k,j}), \qquad C_k = M_k - \min_j m_{\text{rect},k}(p_{1,j}). \tag{15}$$

The compensation reference reevaluates the lowest-risk trajectory against the current geometry. It is not the raw change $M_k - M_1$.

Let $\ell(I_1, I_2)$ denote the overlap length of intervals I_1 and I_2. The target's coordinate intervals are $\mathcal{B}_X = [X_t - a_t, X_t + a_t]$ and $\mathcal{B}_Y = [Y_t - b_t, Y_t + b_t]$. Define lane intrusion $L_{\text{lane}} = \ell(\mathcal{B}_Y, [-1.75, 1.75])/(2b_t)$. With ego speed v_{ego}, the collision corridor extends over $[-b_e, b_e]$ laterally and $[0, D_{\text{corr}}]$ longitudinally, where $D_{\text{corr}} = \max(10, 4v_{\text{ego}})$ m. Its normalized overlap O_{corr} is

$$O_{\text{corr}} = \frac{\ell(\mathcal{B}_Y, [-b_e, b_e])}{2b_t} \frac{\ell(\mathcal{B}_X, [0, D_{\text{corr}}])}{2a_t}. \tag{16}$$

Define geometric clearance from the ego origin, δ_{geo}, and the predefined geometric risk value, R_{geo}, by

$$\begin{aligned} \delta_{\text{geo}} &= \sqrt{\max(X_t - a_e - a_t, 0)^2 + \max(|Y_t| - b_e - b_t, 0)^2}, \\ R_{\text{geo}} &= 0.45 L_{\text{lane}} + 0.35 \exp(-\delta_{\text{geo}}/10) + 0.20 O_{\text{corr}}. \end{aligned} \tag{17}$$

R_{geo} is used to order and normalize the controlled spatial interventions independently of model output. Write $R_{\text{geo},k}$ for its value at level k.

RRA, MVR, URR, and signed gain. For valid cases $i \in \mathcal{I}_c$ with K_i ordered risk levels, let $a_{c,i,k}$ indicate whether the response from level k to $k + 1$ follows the prespecified direction. The within-case transition average is then averaged across cases:

$$\text{RRA} = \frac{1}{|\mathcal{I}_c|} \sum_{i \in \mathcal{I}_c} \frac{1}{K_i - 1} \sum_{k=1}^{K_i - 1} a_{c,i,k}. \tag{18}$$

For each adjacent risk increase, the speed condition is $v_{k+1} - v_k \leq 0.05\,\text{m/s}$ and the clearance condition is $C_{k+1} - C_k \geq -0.05\,\text{m}$. Speed RRA and clearance RRA are the respective fractions of satisfied conditions. Safety-margin RRA counts their logical OR; MVR is its complement. These transition indicators are averaged within each complete case before dataset-level aggregation. The case-level highest-risk underreaction flag U_i is

$$U_i = \mathbf{1}\{v_K - v_1 > -0.20,\ C_K < 0.25,\ M_K - M_1 < -0.25\}. \tag{19}$$

All three conditions must hold; speed is in m/s and margins in metres. URR averages U_i over valid cases. Signed speed, clearance, and progress gains are respectively

$$-\frac{v_K - v_1}{R_{\text{geo},K} - R_{\text{geo},1}}, \qquad \frac{C_K}{R_{\text{geo},K} - R_{\text{geo},1}}, \qquad -\frac{P_K - P_1}{R_{\text{geo},K} - R_{\text{geo},1}}. \tag{20}$$

Figure 4(c) instead plots the mean of per-case $v_k - v_1$ at each of the four lane levels, with the same complete-case mask throughout.

TTC, separation, and closed-loop progress. For spatial diagnostics, an open-loop time-to-conflict proxy is the first waypoint time with nonpositive rectangular margin, or 4.5 s if no waypoint conflicts. The open-loop separation proxy is $\min_j \big(\|p_{k,j} - (X_t, Y_t)\|_2 - \sqrt{(a_e + a_t)^2 + (b_e + b_t)^2}\big)$. These planned-trajectory quantities are distinct from the outcomes measured during execution.

For closed-loop TTC, at each 20-Hz simulation step t, let $P_{e,t}$ and $P_{a,t}$ be the planar convex hulls of the projected CARLA bounding-box vertices of the ego and target actors. Their footprint separation is $d_t = \text{dist}(P_{e,t}, P_{a,t})$. With planar actor centers $\mathbf{p}_{e,t}, \mathbf{p}_{a,t}$ and velocities $\mathbf{v}_{e,t}, \mathbf{v}_{a,t}$, define the radial closing speed along the ego-to-target centerline as

$$\hat{\mathbf{r}}_t = \frac{\mathbf{p}_{a,t} - \mathbf{p}_{e,t}}{\|\mathbf{p}_{a,t} - \mathbf{p}_{e,t}\|_2}, \qquad c_t = (\mathbf{v}_{e,t} - \mathbf{v}_{a,t})^\top \hat{\mathbf{r}}_t. \tag{21}$$

The target is stationary in these episodes. Instantaneous TTC and the episode-level analysis value are

$$\tau_t = \begin{cases} 0, & d_t \leq 0, \\ d_t / c_t, & c_t > 0.05\,\text{m/s}, \\ +\infty, & \text{otherwise}, \end{cases} \qquad \text{TTC}_{\text{analysis}} = \min\left(\min_{t \in \text{episode}} \tau_t,\ 4.0\,\text{s} \right). \tag{22}$$

Thus, the reported TTC is the minimum instantaneous value over the four-second execution, right-censored at 4 s for analysis. We also report minimum target separation, collision occurrence, executed forward progress, and mean speed.

Pairing, clustering, and bootstrap. Semantic and color analyses require pairs that pass the visual controls and have valid outputs for both members; spatial analyses require valid outputs at every risk level of a case. The matched geometry comparison subtracts BASE from GEOM at the same checkpoint. Cross-model contrasts use AutoVLA minus DDV2 or ReCogDrive minus the named baseline, as indicated for each comparison. Paired cross-model contrasts use the intersection of valid case IDs, so they need not equal differences between separately reported model means.

For the semantic/color, spatial, matched geometry, and ReCogDrive analyses, we use 10,000 cluster-bootstrap resamples. Controlled-input analyses cluster by base scene, while the matched geometry comparison clusters by driving log. Paired differences are formed before resampling, and the 2.5th and 97.5th percentiles define the 95% confidence interval.

For predictive-target comparisons, PDMS differences are computed per evaluation token as F minus its matched comparator and clustered by the original driving log. Closed-loop analyses cluster by base case after symmetric exclusion, compare highest-risk URR$= 1$ with URR$= 0$, and compute sequence-level Spearman correlations.

D Discussion

D.1 From Learning Information to Using It for Action

The geometry and predictive studies expose the same distinction from two different directions. Additional supervision can reach representations shared with planning without producing a corresponding improvement in capability-related behavior, while an auxiliary predictive objective can improve

planning even when not every property of its targets contributes equally to that gain. In the geometry study, metric-depth supervision reaches planning-shared states, yet the tested spatial responses do not become better aligned with increasing risk. In the predictive study, future supervision improves planning and benefits from scene-related target content, whereas exact future-time correspondence brings no additional PDMS gain under the tested budget.

These observations suggest that evaluating richer supervision requires more than verifying that the information is learned or that aggregate planning performance improves. A representation can be affected by a training signal without that information becoming behaviorally useful in the expected way, and a useful auxiliary objective need not owe its gain to every property that motivates its design. Capability grounding addresses this intermediate question by examining how the information associated with a capability actually changes the policy's action. Representation, behavioral use, and task performance are therefore related stages of evaluation rather than interchangeable evidence.

D.2 Capability Grounding and Driving Performance

Capability grounding and driving performance describe complementary properties of a policy. Capability grounding asks which information shapes the planned action and whether that response is selective and task-appropriate. Closed-loop evaluation instead measures the behavior that emerges when planning is repeatedly coupled with control and interaction. A policy may use a relevant cue appropriately yet still perform poorly because execution also depends on other capabilities, replanning, and control. Conversely, strong driving performance does not reveal which information was responsible for the action.

Our closed-loop experiments make this distinction concrete. The relationship between the grounding diagnostics and executed outcomes is weak in some settings and changes direction in others: the lane diagnostics show little correspondence with TTC or separation, while underreaction in the target-distance intervention is associated with slower, more conservative execution and larger margins. The mapping from an action-level diagnostic to a driving outcome is therefore neither direct nor monotonic in the tested scenarios. Rather than treating one as a proxy for the other, evaluating both reveals different aspects of policy behavior: CGA characterizes how capability-relevant information influences action, while closed-loop evaluation characterizes what happens when those actions are executed.

Scope. The present experiments instantiate capability grounding with semantic, spatial, and predictive information, but the evaluation principle is not tied to these particular capabilities. The same logic can be applied whenever capability-relevant information or supervision can be varied under controlled conditions and the expected behavioral response can be specified independently of the policy output. The concrete intervention, training control, and response metric will necessarily depend on the capability being studied. CGA is therefore best viewed as a way to structure evidence about capability use, rather than as a single universal diagnostic or scoring rule.