TRIAGE BEFORE FINE-TUNING: FEASIBILITY-GATED SELECTION AND MONITORED ADAPTATION OF EMBODIED FOUNDATION MODELS FOR A FIXED PHYSICAL PLATFORM

Anonymous authors
Paper under double-blind review

ABSTRACT

Embodied foundation models are increasingly deployed by selection: a practitioner picks a public checkpoint by benchmark rank and fine-tunes it on local data. The platform, however, is fixed before the model is chosen: its sensors, onboard compute and operating site are set by the task, and at the start there is little or no local data. We ask what such a checkpoint loses on a fixed physical platform and which part of the loss local data can buy back. Our answer is to split the gap by *what can repair it*. A sensing gap, information the sensors do not capture, cannot be closed with their data; a latency gap, output arriving after the control loop needs it, belongs to the loop, not the weights; only a distribution gap is fine-tuning's to repair. We turn this into TRIAGE, a procedure for *feasibility-gated selection* and *monitored adaptation* in five stages: gates on the sensor contract and on latency that no score can override, ten-minute local probes that place candidates in tiers, a target-relevant pick within the top tier, and fine-tuning monitored for tracking error, perturbation sensitivity and reliance on the image. In Bench2Drive, a 500 ms deadline below its latency makes the closed-loop champion collide on 80 to 90% of routes; on NAVSIM real logs, losing field of view moves plans 2.2 to 5.7 times more than losing resolution at equal pixels (1.6 to 2.4 on Bridge V2 manipulation), 20k local frames close 75% of a stability gap but raise perturbation sensitivity up to 3.4 times, and picking by score or by champion violates a gate in two and four of four deployment profiles. In a field test on a full-scale vehicle, the factory camera failed the sensing gate, the prescribed added camera repaired the view, laboratory latencies did not carry over on board, and in closed loop plans rested on images over two seconds old, the one stale-command takeover coming under command hold. More broadly, it lets organizations that cannot train foundation models from scratch find out, before buying data or hardware, whether a public checkpoint can run on the robot or vehicle they own, and what to fix if not.

1 INTRODUCTION

Public checkpoints of embodied foundation models, released by research groups and industry teams, now span end-to-end driving systems (Chitta et al., 2023; Liao et al., 2025), vision-language-action policies (Renz et al., 2025; Zhou et al., 2026; Li et al., 2026b; Kim et al., 2024), video world models (Liu et al., 2026) and generalist robot policies (Open X-Embodiment Collaboration, 2024; Octo Model Team et al., 2024; Black et al., 2024), and public benchmarks built on recorded datasets (Caesar et al., 2021; Dauner et al., 2024; Walke et al., 2023) and simulated environments (Dosovitskiy et al., 2017; Jia et al., 2024; Li et al., 2025) evaluate and compare them. It is increasingly practical to begin a new deployment with an existing model: select a pretrained checkpoint and adapt it.

A benchmark evaluates candidates under its own observation, environment, and execution conditions, whereas a deployed policy acts through a particular platform's sensors, onboard compute, and controller, and its actions change the state from which its next observations come. The order of these decisions matters. A deployer, such as a campus shuttle operator, a mining or construction fleet, or a hospital delivery robot, chooses the platform for the task: payload, ruggedness, cost, and sensing are

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.

fixed long before any model is considered and are not changed merely to suit a checkpoint. Such a deployer does not train a foundation model from scratch and starts with little or no local data, since collecting it on the target platform is the expensive step the model is meant to make worthwhile. Benchmark scores are thus evidence of capability, not of compatibility with the platform. In our experiments the current closed-loop champion of a standard driving benchmark consumes a single 110° camera that neither the 64° front camera of a nuPlan-style monocular camera setup nor its surround ring reproduces as one view, and under an inference deadline shorter than its own latency, a regime it was not designed for, it collides on 80 to 90% of routes; the failure belongs to the deployment loop, not to the model. Rank alone is therefore not a sufficient criterion; the question is which model is a viable starting point for the platform and what must change to make it work: the model, the sensing hardware, the execution architecture, or the training data.

Fine-tuning is the natural response to such differences, but the first question is not how to fine-tune; it is which part of the gap between a public checkpoint and the platform fine-tuning can reach at all: more data cannot recover information the observations do not preserve, and lower open-loop error does not resolve delayed execution. **Our insight** is that deployment gaps should therefore be diagnosed by *repairability*, by whether their dominant cause is missing information, late outputs, or distribution shift, since these call for interventions that are not interchangeable (Figure 1a–c). A *sensing gap* arises when task-relevant distinctions cannot be recovered from the deployed observations (Blackwell, 1953); the repair is a different sensing configuration or a model whose requirements match the platform. A *latency gap* arises when model outputs arrive too late for the control loop; its consequences depend on whether the controller replays stale commands or tracks a delayed plan with fresh state (Katsikopoulos & Engelbrecht, 2003; Ramstedt & Pal, 2019), and the repair is faster computation or a different output-consumption architecture, not more data. A *distribution gap* arises from differences in appearance, scene structure, or behavior, the mechanism target-domain adaptation is designed to address (Ben-David et al., 2010).

Building on this diagnosis, **we introduce TRIAGE** (Figure 1d), a staged diagnostic framework for the feasibility-gated selection and monitored adaptation of embodied foundation models. It checks each candidate's sensing requirements against the platform and measures its inference latency on the deployment hardware against the intended control interface. Candidates that violate either requirement are excluded, and a higher score cannot bring them back: that takes a change to sensing, compute, model implementation, or the execution interface. For the remaining candidates, passive probes on short local logs characterize responses to deployment-relevant perturbations, and target-relevant evaluation supports selection; adaptation is evaluated on tracking performance, perturbation sensitivity (Kumar et al., 2022), and reliance on the image. When no candidate satisfies the requirements, TRIAGE reports the limiting constraints rather than forcing a choice. We say up front that its measured value lies in exclusion and monitoring, not in fine-grained ranking: the probes place candidates into tiers but do not separate models of the same generation, where following the benchmark score remains the best rule we tested. Its purpose is to determine when benchmark evidence applies to a deployment and where effort should go, not to produce another universal ranking.

Contributions. (i) We formulate the deployment of pretrained embodied foundation models on a fixed platform as a problem of *repairability* rather than benchmark ranking, distinguishing sensing, latency, and distribution gaps by their causes and interventions, and turn it into TRIAGE, a staged framework that returns either a viable starting point or the constraint preventing deployment. (ii) We evaluate TRIAGE on 1,085 closed-loop runs in CARLA/Bench2Drive, thirteen driving configurations on four-city nuPlan logs (eleven scored under the official NAVSIM evaluator), and six open-weight manipulation models on Bridge V2. Restricting field of view costs 2.2–$5.7\times$ more than reducing resolution at a matched pixel budget in driving and 1.6–$2.4\times$ in manipulation, and whether fine-tuning compensates a sensory restriction depends on the model. (iii) We run TRIAGE end to end on a full-scale vehicle (Section 5 and Appendix A): its factory camera fails the sensing gate, and for all five candidates plans made from it depart from the repaired view's more on turns than on straight road; laboratory latencies did not carry over to the onboard GPU; and in closed loop plan tracking drew three takeovers and command hold four, one of which was rated stale-command.

2 RELATED WORK

Selecting and adapting checkpoints. Transferability estimators such as LEEP (Nguyen et al., 2020) and LogME (You et al., 2021) estimate transfer performance from labeled target predictions

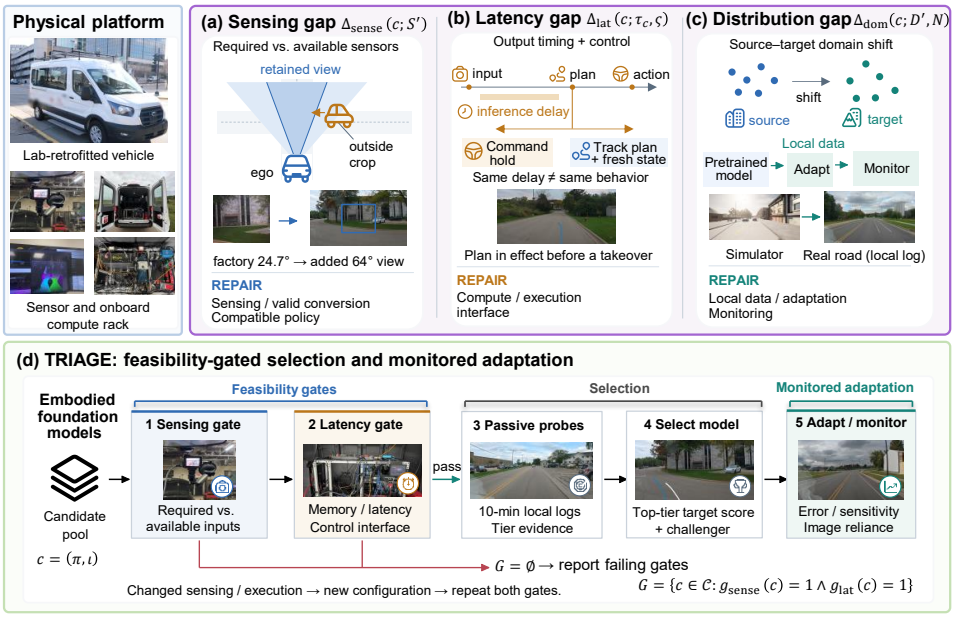

Figure 1: The repairability principle. (a) Sensing gap: no local data closes it. (b) Latency gap: damage depends on how stale outputs are consumed. (c) Distribution gap: local data repairs it. (d) TRIAGE over a pool of embodied foundation models: feasibility gates (Stages 1–2), selection (3–4) and monitored adaptation (5).

or features, and offline policy selection (Konyushkova et al., 2021), simulation-based evaluation (Li et al., 2025), and runtime monitoring (Agia et al., 2024) estimate performance or detect failures; none is designed to check sensing or control compatibility with the platform before deployment. Driving benchmarks (Jia et al., 2024; Dauner et al., 2024) address limitations of open-loop evaluation (Li et al., 2024), and counterfactual image edits show that leaderboard scores do not reveal whether end-to-end and vision-language-action driving policies respond to safety-critical scene content under domain shift (Yang et al., 2026); we ask the prior question of whether a platform can run the checkpoint at all. Surgical fine-tuning (Lee et al., 2023) selects which weights to update for a given shift; we extend intervention selection to sensing, compute, and execution.

Diagnosing the deployment gap. Sim-to-real surveys organize the reality gap by system component (Aljalbout et al., 2026); our taxonomy cuts across these components, classifying mismatches by the interventions they need, and rests on Blackwell's comparison of experiments (Blackwell, 1953), delayed decision processes (Katsikopoulos & Engelbrecht, 2003; Ramstedt & Pal, 2019), and domain-adaptation theory (Ben-David et al., 2010). Delay-aware learning (Chen et al., 2021) and real-time chunking (Black et al., 2025) address latency through state augmentation or asynchronous execution, interventions beyond ordinary target-domain fine-tuning, and cross-embodiment studies (Open X-Embodiment Collaboration, 2024; Park et al., 2025) show that data and interface design can accommodate some differences across robots. We therefore diagnose individual gap components rather than treating an entire embodiment as transferable or irreparable.

3 The TRIAGE Framework

3.1 Problem Formulation

A deployer holds a platform whose sensors S', compute T' and operating domain D' are fixed before any model is chosen, and a budget of N local frames for adaptation; the candidates are a pool $\mathcal{C}$ of public checkpoints, each trained under its own sensors, platform and data (S_i, T_i, D_i). What is deployed is a checkpoint π with the interface ι it runs through (cameras consumed, pre- and post-processing, controller, hardware), a *deployment configuration* $c = (\pi, \iota)$, and what matters is its risk on D', $R(c) = \mathbb{E}_{\omega \sim P^c_{D'}}[L(\omega)]$, the expected task loss (collisions, tracking error) over rollouts whose law depends on policy and interface alike. Benchmark scores and local probes are evidence about $R(c)$, not measurements of it. *Post-training* is any adaptation on observations from the target sensors and their labels; changing architecture or interface creates a new configuration.

The framework rests on one structural claim: the gap between $R(c)$ and the checkpoint's risk on its own platform has three sources, and the frame budget acts on only one,

$$\Delta(c) = \Big(\underbrace{\Delta_{\mathrm{sense}}(c; S')}_{\substack{\text{sensing term:} \\ \text{invariant to } N}}, \underbrace{\Delta_{\mathrm{lat}}(c; \tau_c, \varsigma)}_{\substack{\text{latency term:} \\ \text{invariant to } N}}, \underbrace{\Delta_{\mathrm{dom}}(c; D', N)}_{\substack{\text{distribution term:} \\ \text{reduced by } N}} \Big), \tag{1}$$

with τ_c the measured latency and ς the way the control loop consumes stale outputs. The terms are listed, not summed, and not commensurable (Appendix B); each has a distinct repair, and TRIAGE determines which term a candidate's gap occupies before any score is compared.

3.2 THREE GAP CLASSES

Sensing gaps. A sensing gap $\Delta_{\mathrm{sense}}(c; S')$ exists when the decision depends on observations the target sensors S' do not carry, as when a checkpoint trained with three cameras, LiDAR or six surround cameras must decide from one. By Blackwell's comparison of experiments (Blackwell, 1953), no policy computed from a coarser channel, including one fine-tuned through it, can reach a lower risk than the best policy on the richer channel, and it stays strictly above whenever no optimal decision can do without what was removed (Prop. 1, Appendix B.2). What matters is whether the missing information is *inferable* from what remains: monocular depth is (Kwon et al., 2025), cross-traffic outside the frustum is not. The deficit concentrates where the decision needs the missing view: on turning frames, the position two seconds ahead lies outside a moderately cropped image 79.7% of the time (Figure 14). The repair is a sensor or a model the platform's sensors can feed; how much of a moderate deficit is non-inferable is empirical.

Latency gaps. A latency gap $\Delta_{\mathrm{lat}}(c; \tau_c, \varsigma)$ exists when the policy's output arrives later than the control loop consumes it. With latency τ_c, control period δ and $k = \lceil \tau_c / \delta \rceil$ pending outputs, the closed loop is a delayed decision process (Katsikopoulos & Engelbrecht, 2003; Ramstedt & Pal, 2019) that combines the newest output, k control steps old, with current proprioception p_t through a consumption rule ς. Fine-tuning changes neither k nor ς: it can learn to extrapolate what is predictable over the delay, but the policy still acts on k-step-old observations, so the least attainable risk is unchanged (Prop. 2, Appendix B.3). The consumption rule decides the failure mode: holding the last command discards p_t and accumulates open-loop error, whereas tracking the last plan with fresh state loses, if the tracked loop is contracting, at most a constant times the plan's drift during the delay. The repair is faster compute or an architecture that tolerates stale plans.

Distribution gaps. A distribution gap $\Delta_{\mathrm{dom}}(c; D', N)$ exists when the appearance, layout or behavior the checkpoint was trained on differs from the target domain D' with sensors and control loop unchanged, as when a checkpoint trained in a simulator or another city meets unfamiliar streets: covariate shift, or concept shift where local driving behavior differs. It is the only class local data can reach and repair: in the source-and-target bound of Ben-David et al. (2010) (Appendix B.4), a hypothesis trained on m source and N target samples pays the divergence $d_{\mathcal{H}\Delta\mathcal{H}}(D, D')$ with a weight the optimal mixture drives to zero as N grows, at the price of an estimation term.

3.3 GATES BEFORE RANKING

Sensing and latency compatibility are checked by operational *gates*, not by estimating the theoretical risk gaps of Equation (1). A score cannot override a failed requirement; selection is restricted to configurations passing both checks and ranks them by a proxy $\widehat{\Delta}_{\mathrm{dom}}$ from probe-supported tiers and target-relevant scores (Section 3.4):

$$G = \{\, c \in \mathcal{C} :\ g_{\mathrm{sense}}(c) = 1 \ \wedge\ g_{\mathrm{lat}}(c) = 1 \,\}, \qquad \hat{c} = \arg\min_{c \in G} \ \widehat{\Delta}_{\mathrm{dom}}(c; D', N). \tag{2}$$

$g_{\mathrm{sense}} = 1$ means the configured inputs meet the sensor contract; $g_{\mathrm{lat}} = 1$ means the model fits the hardware and its measured latency meets a preset, interface-specific deadline d': the control period δ under command hold, and under plan tracking a separate replanning deadline whose configuration must be validated in closed loop, not inferred from stale-image tolerance. Remark 1 (Appendix B.5) gives a sufficient condition for gating first under a common risk measure, which collision rates and PDMS regret do not numerically verify. An empty G reports the failing gates, and changed sensing or execution interfaces re-enter both gates. Passing the gates is not a safety guarantee.

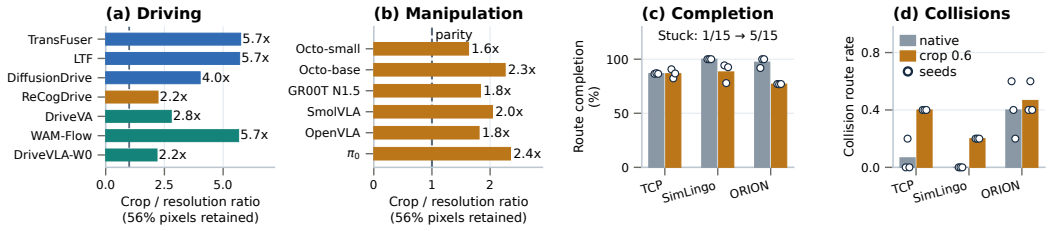

Figure 2: The sensing gate (RQ1, RQ3, RQ9). (a, b) At equal pixels a crop costs more than a resolution drop in every model. (c, d) A moderate crop raises TCP's and SimLingo's collisions and costs ORION completion.

3.4 Workflow

Figure 1d shows the five stages, and Algorithm 1 (Appendix C) states them formally with their costs. *Stage 1* evaluates g_{sense} by checking each sensor contract against S' on paper; a structural deficit excludes, a convertible one re-enters as its own configuration. *Stage 2* evaluates g_{lat} by checking memory fit and measuring τ_c on the deployment hardware in its operating state; the candidates that pass form G. *Stage 3* runs each remaining candidate offline on ten minutes of recorded local logs, recording temporal self-consistency and paired responses to gamma, JPEG, field-of-view and resolution degradations, with two sanity checks (planned turns agree in sign with the driver; blind anchors, the image-free baselines of Table 1 (the rest are *sighted*), respond exactly zero); the result is the candidate's *probe readings*. *Stage 4* estimates Δ_{dom} from two proxies: the remaining candidates fall into architectural tiers, each placement confirmed or vetoed by the probe readings, and within the top tier (VLAs and world models) the candidate with the strongest target-relevant score is selected, a target-domain training set breaking ties, with the runner-up kept as a challenger. *Stage 5* spends N on fine-tuning the selection while watching tracking error, perturbation sensitivity and the plan's reliance on the image, and tests any converted interface in closed loop.

4 Experiments

The subsections cover the two gates, the distribution gap and its repair, selection, and transfer to manipulation; full results are in Appendix D and metric definitions in Appendix E.

Setup. Closed loop: CARLA/ Bench2Drive (Dosovitskiy et al., 2017; Jia et al., 2024) with a version-fixed evaluator, 1,085 accepted runs of TCP, SimLingo, UniAD and VAD, with ORION admitted mid-study, over six town–weather environments plus a deadline axis and the occlusion and frame-drop axes of Bench2Drive-Robust (Zhang et al., 2026). Open loop: the official NAVSIM evaluator at a fixed version (Dauner et al., 2024) (PDMS, a 0 to 100 composite of safety, comfort and progress terms) over nuPlan logs (Caesar et al., 2021) from Singapore, Boston, Las Vegas and Pittsburgh, 12,146 test tokens. The pool (Table 1) holds thirteen driving configurations in four tiers; every configuration with a published score entered after reproducing it under the version-fixed evaluator (one exception, ORION, in Appendix F). A candidate's sensor contract is the set of sensors its inference code consumes (Table 1).

Table 1: The driving pool. Official PDMS (per-city best in bold), sensors as consumed, and p50 latency on an idle RTX A6000.

Configuration	Venue	Sensors	ms	SIN	BOS	LVG	PIT	Glob.
Blind anchors								
Const.-velocity	anchor	none	0.01	17.6	20.6	21.1	22.3	20.7
EgoStatusMLP	NeurIPS'24	state	0.12	52.9	63.8	78.9	55.8	65.5
Traditional stacks								
TCP	NeurIPS'22	3 cam, stitched	63	–	–	–	–	–
TransFuser	TPAMI'23	3 cam + LiDAR	69	76.8	80.8	91.0	79.9	83.4
LTF	TPAMI'23	3 cam	69	74.8	82.6	91.2	82.1	84.1
DiffusionDrive	CVPR'25	3 cam + LiDAR	85	82.0	87.0	93.0	86.6	88.1
VLAs								
ReCogDrive-2B	ICLR'26	mono	513	85.7	89.3	**94.7**	88.8	90.4
AutoVLA-3B	NeurIPS'25	6 cam	1,249	82.5	90.1	92.4	88.7	89.4
Curious-VLA-3B	CVPR'26	mono	4,067	85.9	88.5	93.6	88.2	89.8
Alpamayo-1.5-10B	release'26	flexible	959	–	–	–	–	–
World models								
DriveVLA-W0-8B	ICLR'26	mono	485	86.2	89.5	93.7	89.0	90.3
WAM-Flow-1.5B	CVPR'26	mono	781	85.5	90.1	93.7	88.8	90.3
DriveVA-5B	ECCV'26	mono video	947	**87.6**	**91.1**	94.0	**89.6**	**91.2**

4.1 The sensing gate

RQ1: At equal pixel budgets, does structure loss damage more than detail loss? Open loop on nuPlan frames from four cities and seven configurations, central crops keeping 90, 75 and 60%

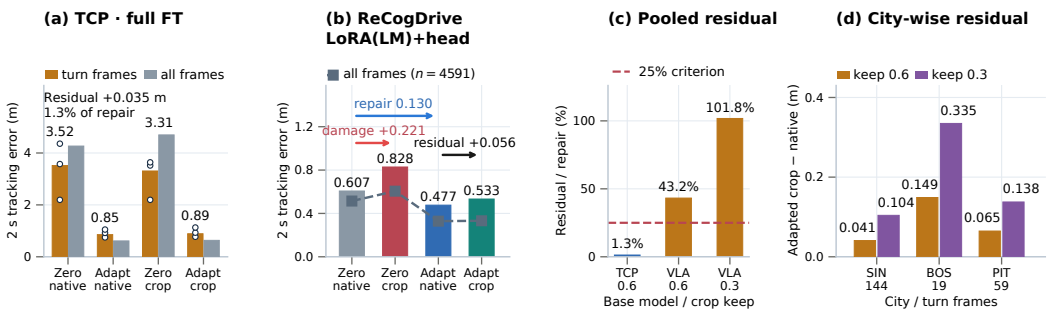

Figure 3: Fine-tuning under a moderate crop (RQ2). Turn-frame tracking error, native/crop × zero-shot/adapted: (a) TCP, (b) ReCogDrive; (c) residual as a share of repair; (d) ReCogDrive's residual by city.

of each image dimension remove structure, and downscaling each dimension to 75, 50 and 33% removes detail, both read as mean plan displacement, the average distance between the plans on the degraded and original frames. Every sighted model displaces further as the view narrows; resolution curves are shallower, and blind anchors sit at exactly zero on both axes (curves in Appendix D.1). At a matched 56% of pixels kept (Figure 2a), losing them at the border costs 2.2–5.7× more than losing them everywhere, and six of seven configurations respond less to $1/9$ of native resolution than to a crop that removes 19% of pixels. The ratio does not track the sensor suite, though the largest absolute response is monocular (DriveVLA-W0, 1.2 m). Losing view costs several times more than losing sharpness, which is why a narrower view is gated and a lower resolution treated as a format conversion.

RQ2: Can fine-tuning compensate a moderate sensory deficit? Open loop on nuPlan frames, we fine-tune TCP under a crop keeping 60% of each dimension and evaluate under it against a matched-budget uncropped run, with turning frames reported separately; ReCogDrive repeats it with a recipe that leaves its vision encoder frozen (LoRA(LM)+head, Appendix D.1). We set the criterion that a sensing gap leaves at least 25% of the uncropped repair as residual. TCP repairs 73–76% of its turn error with and without the crop and leaves 1.3%, so the criterion is not met (Figure 3a); ReCogDrive leaves 43% and meets it (Figure 3b), repairing straight driving to within a millimeter but not turns, although the frozen encoder could explain this equally well. With a crop keeping 30% the residual is 101.8% of the same repair (Figure 3c), and the compensated fraction (the share of the crop's zero-shot cost that adaptation removes) falls from 0.746 to 0.642: the larger the gap, the less local data buys back. City-wise residuals are positive in all three cities with turns (Figure 3d).

RQ3: Does a moderate sensory deficit harm safety even when capability metrics look intact? Closed loop on Bench2Drive, two probes run on released checkpoints: a random rectangle blacked out on every camera at severities 50 and 80, and the crop keeping 60% applied zero-shot to TCP and SimLingo over 30 runs. At severity 50 the forward-facing configurations' collision route rates rise by +0.3 while the six-camera configurations hold or fall; at 80 every configuration fails, so redundant cameras help only up to moderate severity. The crop (Figure 2c, d) separates capability from safety: route completion stays within seed noise, while collision route rate rises beyond it with all three seeds on the same value (0.067 to 0.400 for TCP, 0.000 to 0.200 for SimLingo). ORION is damaged in a different channel: completion falls from 0.973 to 0.770 and stuck routes rise from $1/15$ to $5/15$. RQ2 measures fine-tuned models open loop and RQ3 zero-shot models closed loop, so together they show only that open-loop repair does not certify closed-loop safety.

4.2 The latency gate

RQ4: Is a latency violation a safety event, what decides its failure mode, and can wrappers or fine-tuning repair it? We time every model per frame with no other process on the GPU, then enforce deadlines closed loop on Bench2Drive, holding the last command whenever a plan is late, over three towns, five routes and two or three seeds. Latency spans five decades (Figure 4a): 63–85 ms for the traditional stacks of Table 1, 0.49–4.1 s for every VLA and world model. At a 500 ms deadline SimLingo still covers 48 to 58% of route length but collides on 0.8–0.9 of routes in every town (Figure 4b). Freezing the camera input in random bursts of up to 1 or 3 s, while the model keeps replanning on fresh speed and position, keeps collision rates near baseline (Figure 4c; the largest

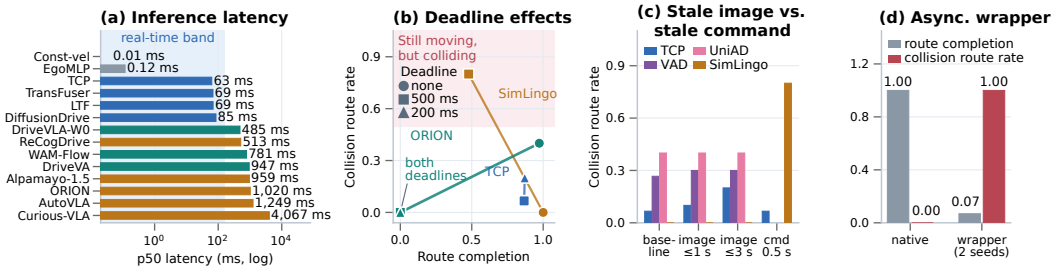

Figure 4: The latency gate (RQ4). (a) Latency spans five decades. (b) At 500 ms the champion moves but collides; ORION stalls. (c) Stale images are tolerated, stale commands not. (d) The wrapper always collides.

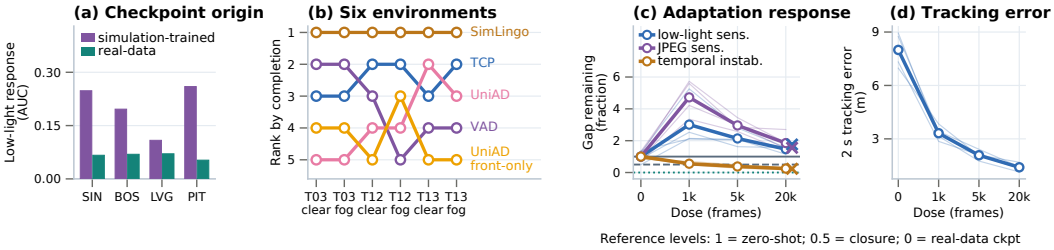

Figure 5: The distribution gap (RQ5, RQ6). (a) The simulation-trained checkpoint is 1.5–4.9× more sensitive. (b) Environments reorder the pool. (c) Stability closes; sensitivity spikes at 1k. (d) Tracking error falls.

rise, TCP's, is from 0.07 to 0.20): a stale image is tolerated when the loop acts on fresh state, a stale command is not. Of three asynchronous-planning wrappers two failed before completing a route, and the filtered-pose one completes 0.07 of route length and collides on every route (Figure 4d). Fine-tuning does not move the clock: the training architecture's latency changed by +0.18%, and ReCogDrive's adapted model, its adapters mounted rather than merged, is 13.3% slower. ORION (1,020 ms) misses every scheduled call and never leaves the start; latency therefore belongs in a gate, and its repair lies in the consumer, not in fine-tuning or a wrapper.

4.3 The distribution gap

RQ5: How large and how structured is the gap that training data and environment create? We contrast public checkpoints of the same TransFuser architecture (Appendix G), one trained in CARLA and one on real data, open loop under an identical probe code on nuPlan frames. The simulation-trained checkpoint reacts 1.5–4.9× more strongly to darkened and compressed frames in all four cities (Figure 5a), and its plans disagree with their own previous frame 8.6–10.3× more (Appendix D.3). Closed loop, six environments reorder the pool: 13 of 15 environment pairs change some pairwise order among the native configurations (Figure 5b), with each configuration's spread across environments dwarfing its seed noise. A UniAD pair narrows the claim to 1.13×, and because UniAD holds state across calls, its run-to-run noise exceeds its own perturbation readings on the simulation-trained side, which are therefore not attributable.

RQ6: What does fine-tuning repair, and at what cost? We behavior-clone the simulation-trained TransFuser on target-city frames at 1k, 5k and 20k, measuring each gap against the real-data checkpoint, with a second seed at 20k and an LP-FT variant (Kumar et al., 2022). Stability closes 75% of the zero-shot gap by 20k (Figure 5c), but after 1k frames sensitivity to darkened and compressed inputs rises 2.4× and 3.4×, leaving a gap 3.0 to 4.7× its zero-shot size, so these gaps do not close; the spike appears in every city, a second seed changes no conclusion, and LP-FT raises the 1k response 1.6 to 2.0 times. Tracking error still falls from 8.0 to 1.4 m (Figure 5d). On ReCogDrive (LoRA(LM)+head) no spike appears (Figure 21): the three responses fall to 0.36, 0.33 and 0.55 of zero-shot by 20k, though the smallest dose is worse than zero-shot on turns (1.15). Since falling sensitivity also fits a model that attends less to the image, we measure the plan's reliance on the image as its response to a constant gray frame (Appendix E): it falls below the range seen when only the training seed changes, at every dose (1k marginally), so part of the fall is exactly that. Fine-tuning repairs its assigned class, but the spike depends on the recipe and the base model, so Algorithm 1's monitored step must watch the plan's reliance on the image beside sensitivity.

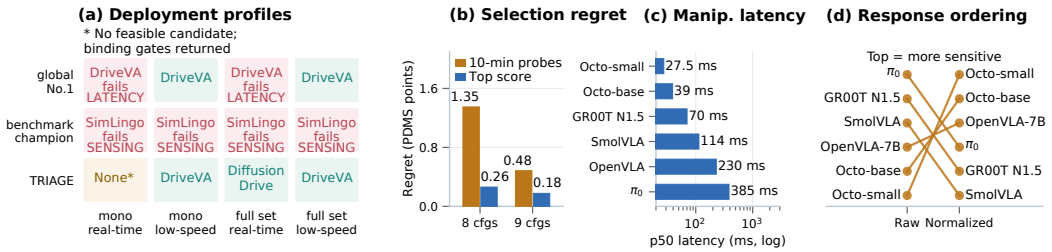

Figure 6: Selection (RQ7; c, d from RQ9). (a) Picks per profile (red: violates a gate; amber: empty set). (b) Probes lose to the top score (8 cfgs: without DriveVA). (c) Manip. latency. (d) Raw vs. normalized ranks.

4.4 Selection with the full framework

RQ7: Does TRIAGE select better than standard practice, and where does cheap evidence stop? Open loop under the NAVSIM evaluator, across four deployment profiles (Figure 6a), score-first violates the latency gate in 2/4 and champion-first the sensing gate in 4/4; TRIAGE agrees with the scoreboard wherever all gates pass and, for monocular real-time deployment, returns an empty set with the gates that fail. Figure 6b tests the probes: nine configurations in four cities give 36 leave-one-city-out cases with ten fixed features and four predictor families; regret is the PDMS the pick gives up against the best candidate in the held-out city. The probes never pick below the top tier, but within it they lose to picking the top-scoring candidate (0.48 against 0.18 points of mean regret), and adding latency as a feature worsens it to 1.29. Raw sensitivities are not comparable across models (Figure 19a): normalizing each by the model's response to a constant gray frame leaves no stateless stack in its rank. Cheap probes place tiers but cannot separate models of the same generation; latency belongs in a gate, not a selection feature.

RQ8: Does the framework absorb new state-of-the-art candidates mid-study? Four NAVSIM candidates joined mid-study (Table 2; full results in Appendix D.4). Alpamayo 2 Super fails both gates on paper: its documented 70 GiB footprint (for an 80 GB H100) and six cameras do not fit a monocular 48 GB platform. Alpamayo-1.5 was admitted (959 ms); its front-only ablation leaves plan self-consistency intact but lowers agreement with the driver's turn direction (0.915 against 0.954), placing the model card's caution in turn correctness rather than plan stability. WAM-Flow and DriveVLA-W0 reproduced their published scores and RQ1's asymmetry.

Table 2: Mid-study admissions (RQ8). Green: admitted, score reproduced; red: excluded.

Candidate	Sensor contract	p50 (ms)	PDMS (pub. / ours)
Alpamayo 2 Super	6 cams, 70 GiB	–	excluded on paper
Alpamayo-1.5	flexible	959	probe level only
WAM-Flow	mono	781	90.3 / 90.32
DriveVLA-W0	mono	485	90.2 / 90.30

4.5 Transfer to manipulation

RQ9: Do the instruments and the sensory asymmetry transfer to a second embodiment? The same sensor-contract and latency checks and the same probes run unmodified on six manipulation models (Octo-small, Octo-base, GR00T N1.5, SmolVLA, OpenVLA-7B and π_0; sources in Table 13) in offline inference on recorded episodes of an official Bridge V2 shard (Walke et al., 2023). Latency spans 27.5 ms (Octo-small) to 385 ms (π_0), a 14× spread (Figure 6c), narrower than the five decades of driving. The field-of-view asymmetry holds in all six: cropping the view moves the actions 1.6–2.4× more than downscaling it at the same pixel budget (Figure 2b), with the constant-action anchor at exactly zero. The probe's limitation transfers too (Figure 6d): responses to losing the image span 31.8×, and ranking by raw versus normalized response gives a near-reversal; as in driving, the probe resolves several-fold differences but not small ones.

5 Real-world field study on a laboratory vehicle

We run TRIAGE on a full-scale laboratory vehicle (Figure 7a; Appendix A) to test the offline checks on a real platform; its onboard GPU is the RTX A6000 model behind every latency in Table 1.

The sensing gate fails, and the prescribed repair works. The 24.7° factory front camera is narrower than every sighted sensor contract (64 to 110°), so no sighted candidate passes Stage 1, a

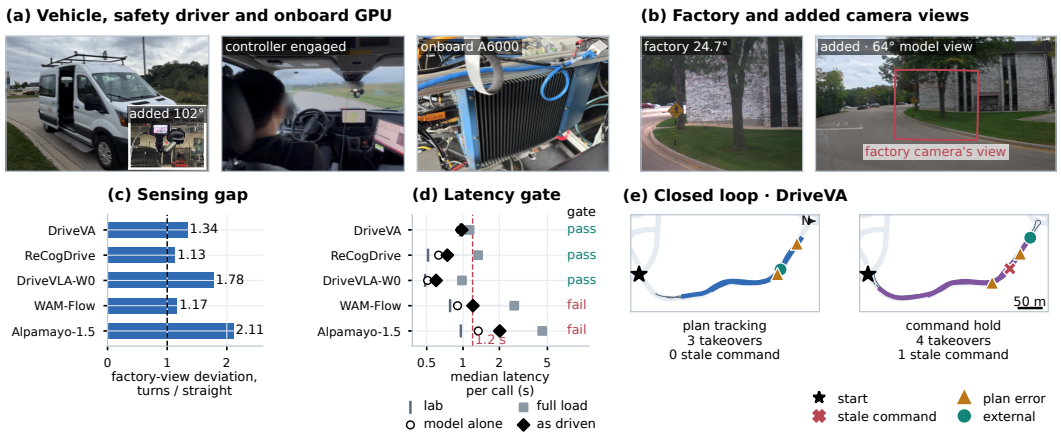

Figure 7: The field study. (a) Inset: added camera. (b) One turn in both views (red: factory view). (c) Factory-view deviation, turns over straight. (d) Median latency; gate judged as driven. (e) Closed-loop takeovers.

physical instance of the crop of RQ1–RQ3. Following Algorithm 1, we add a wide-angle front camera (102°) and reproject its frames into the 64° view the monocular candidates consume. On a hand-driven lap with the models planning but not in control, all five candidates' plans from the factory view departed from those from the 64° view more on turns than on straight road, by 1.13 to 2.11× (Figure 7b, c), and four of five erred more against the driver on turns, the turn-specific damage of RQ2; on the added view a crop again moved plans 2.4–6.4× more than an equal-pixel resolution cut.

Latency must be measured on the host that will run it. Six NAVSIM candidates remain, none fast enough for real-time control, so the vehicle tracks plans under a 1.2 s deadline, which excludes Curious-VLA (4.1 s); the top-scoring DriveVA (PDMS 91.2) is the default choice and the runner-up, ReCogDrive, the challenger (Stage 4). With the same GPU model, weights and inputs, onboard latency differed from the laboratory by over 10% for three of five candidates, and under standard load three of five crossed the deadline (Figure 7d). As driven (visualization closed), the gate kept DriveVA, ReCogDrive and DriveVLA-W0 and removed WAM-Flow (by 7 ms) and Alpamayo-1.5.

The selection returned no model. On 24 min of local driving the probes separated every sighted candidate from the blind anchor, but ReCogDrive's readings left their four-city ranges by more than the allowed twice its run-to-run noise, so, under the range check fixed before the drive (Appendix A), none was deployed. We still drove the default choice, two runs each at 5.4 km/h (Figure 7e), under plan tracking and, as the RQ4 contrast, under command hold, whose deadline its latency fails. Plan tracking drew three takeovers in 6.1 min and command hold four in 7.4 min; a reviewer blind to the controller attributed one, under command hold, to stale commands, the direction RQ4 found but too few to separate the controllers. Under both, plans arrived every 1.5 s at the median (1.03 s of it inference), so every cycle overran the 1.2 s deadline by about 0.3 s and the image behind the plan in use was 2.26 s old at the median; this is why Section 3.3 requires plan-tracking deadlines to be validated in closed loop. Local adaptation (Stage 5) was not run.

6 Conclusion and Limitations

Choosing an embodied foundation model for a fixed platform is not a leaderboard lookup: the checkpoint–platform gap splits by what can repair it, and only its distribution part is fine-tuning's problem. Triage turns this into feasibility-gated selection and monitored adaptation, valuable for exclusion and monitoring, not ranking. In an end-to-end field test on a full-scale laboratory vehicle, the sensing gate and its turn-specific damage transferred, while laboratory latency, even on the same GPU model, and a selection rule calibrated offline did not; both must be measured on-platform. A benchmark score informs deployment only with its sensor contract and latency.

Limitations. Closed-loop evidence is simulated, real-data evidence open loop, apart from a low-speed field study (two runs per controller, one reviewer, no adaptation). Repair covers behavior cloning at modest budgets on three architectures (VLA vision encoder frozen); manipulation is offline. Few architectures publish both simulation and real-data checkpoints, so RQ5 uses two pairs.

Reproducibility statement

All evaluations use public checkpoints and public benchmarks at fixed versions: CARLA 0.9.15 with Bench2Drive, and the NAVSIM v1.1 evaluator on the `navtest` split. A configuration with a published score entered the pool only after that score was reproduced under the same evaluator, with one documented exception; closed-loop and open-loop results are never pooled, missing measurements are never imputed, and fine-tuning data are log-level disjoint from every evaluation window (Appendix F). Every reported quantity is defined in Appendix E, the formal statements with their assumptions and proof sketches are in Appendix B, and Algorithm 1 with its costs is in Appendix C. Appendix D gives the complete results of every research question, including per-city and per-seed values; Appendix G documents the search for checkpoint pairs; and Appendix A describes the vehicle, its calibration and the protocol and results of each field experiment. Code, probe definitions and per-run outputs will be released under an open-source license.

Ethics statement

This work evaluates publicly released driving and manipulation checkpoints on public benchmark logs, in simulation, and in a field study on a laboratory vehicle. The field study is not a study of human participants: it was carried out by members of our laboratory, a safety driver who had completed the institution's Institutional Review Board (IRB) certification training and an operator, and its recordings of public roads are used only for the measurements reported here. Closed-loop safety outcomes were measured in simulation, apart from the closed loop of the field study, driven with speed commands capped at $1.5\,\mathrm{m/s}$ and a safety driver ready to take over. The field study ran on low-traffic business-park and residential roads posted at $25\,\mathrm{mph}$ or less, under safety abort rules fixed before the drive. Redistribution of third-party weights follows their licenses; version identifiers and download scripts are released, not the weights.

AI use statement

We used generative AI tools (large language model assistants) to implement methods, by writing, debugging and running evaluation, analysis and figure scripts on archived outputs; to discuss design options for the methodology and experiments; and to check the formal statements and proofs of Appendix B. We have not used generative AI tools to generate or label data: every takeover in the field study was rated by a human reviewer. Additionally, we used generative AI tools to draft and edit paper text, including the abstract, the conclusion and figure captions; to search for and summarize related work and public checkpoints; and to check the numbers in the text against the archived outputs. We have reviewed all AI-assisted work: every cited work was checked against its source and every reported number against the archived outputs. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

References

Christopher Agia, Rohan Sinha, Jingyun Yang, Zi ang Cao, Rika Antonova, Marco Pavone, and Jeannette Bohg. Unpacking failure modes of generative policies: runtime monitoring of consistency and progress. In *Conference on Robot Learning*, 2024.

Elie Aljalbout, Jiaxu Xing, Angel Romero, Iretiayo Akinola, Caelan Garrett, Eric Heiden, et al. The reality gap in robotics: challenges, solutions, and best practices. *Annual Review of Control, Robotics, and Autonomous Systems*, 9:403–432, 2026.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79:151–175, 2010.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Kevin Black, Manuel Y. Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. *arXiv preprint arXiv:2506.07339*, 2025.

David Blackwell. Equivalent comparisons of experiments. *Annals of Mathematical Statistics*, 24 (2):265–272, 1953.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: a multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Holger Caesar, Juraj Kabzan, Kok Seang Tan, et al. nuPlan: a closed-loop ML-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021.

Baiming Chen, Mengdi Xu, Liang Li, and Ding Zhao. Delay-aware model-based reinforcement learning for continuous control. *Neurocomputing*, 450:119–128, 2021.

Canyu Chen, Yuguang Yang, Zhewen Tan, Yizhi Wang, Ruiyi Zhan, Haiyan Liu, Xuanyao Mao, Jason Bao, Xinyue Tang, Linlin Yang, Bingchuan Sun, Yan Wang, and Baochang Zhang. Devil is in narrow policy: Unleashing exploration in driving VLA models. In *Findings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1062–1072, 2026.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. TransFuser: imitation with transformer-based sensor fusion for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

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*, 2024.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: an open urban driving simulator. In *Conference on Robot Learning*, 2017.

Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, et al. ORION: a holistic end-to-end autonomous driving framework by vision-language instructed action generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Haoyu Fu, Diankun Zhang, Zongchuang Zhao, et al. MindDrive: a vision-language-action model for autonomous driving via online reinforcement learning. In *European Conference on Computer Vision*, 2026.

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.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Feiyang Jia, Lin Liu, Ziying Song, et al. DriveWorld-VLA: unified latent-space world modeling with vision-language-action for autonomous driving. *arXiv preprint arXiv:2602.06521*, 2026.

Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2Drive: towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2024.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, et al. VAD: vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Konstantinos V. Katsikopoulos and Sascha E. Engelbrecht. Markov decision processes with delays and asynchronous cost collection. *IEEE Transactions on Automatic Control*, 48(4):568–574, 2003.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning*, 2024.

Ksenia Konyushkova, Yutian Chen, Thomas Paine, Caglar Gulcehre, Cosmin Paduraru, Daniel J Mankowitz, Misha Denil, and Nando de Freitas. Active offline policy selection. *Advances in Neural Information Processing Systems*, 34:24631–24644, 2021.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022.

Donghyeon Kwon, Youngseok Yoon, Hyeongseok Son, and Suha Kwak. MemDistill: Distilling LiDAR knowledge into memory for camera-only 3D object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6828–6838. IEEE, 2025.

Lucien Le Cam. Sufficiency and approximate sufficiency. *Annals of Mathematical Statistics*, 35(4): 1419–1455, 1964.

Yoonho Lee, Annie S. Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. Surgical fine-tuning improves adaptation to distribution shifts. In *International Conference on Learning Representations*, 2023.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Oier Mees, Karl Pertsch, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. In *Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 3705–3728. PMLR, 2025.

Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, et al. DriveVLA-W0: world models amplify data scaling law in autonomous driving. In *International Conference on Learning Representations*, 2026a.

Yongkang Li, Kaixin Xiong, Xiangyu Guo, et al. ReCogDrive: a reinforced cognitive framework for end-to-end autonomous driving. In *International Conference on Learning Representations*, 2026b.

Yongkang Li, Lijun Zhou, Sixu Yan, Bencheng Liao, Tianyi Yan, Kaixin Xiong, Long Chen, Hongwei Xie, Bing Wang, Guang Chen, et al. UniDriveVLA: Unifying understanding, perception, and action planning for autonomous driving. *arXiv preprint arXiv:2604.02190*, 2026c.

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*, 2024.

Bencheng Liao, Shaoyu Chen, Haoran Yin, et al. DiffusionDrive: truncated diffusion model for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Mengmeng Liu, Diankun Zhang, Jiuming Liu, Jianfeng Cui, Hongwei Xie, Guang Chen, Hangjun Ye, Michael Ying Yang, Francesco Nex, and Hao Cheng. DriveVA: Video action models are zero-shot drivers. In *European Conference on Computer Vision*, pp. 315–335. Springer, 2026.

Cuong V. Nguyen, Tal Hassner, Matthias Seeger, and Cédric Archambeau. LEEP: A new measure to evaluate transferability of learned representations. In *International Conference on Machine Learning*, 2020.

Long Nguyen, Micha Fauth, Bernhard Jaeger, Daniel Dauner, Maximilian Igl, Andreas Geiger, and Kashyap Chitta. LEAD: minimizing learner-expert asymmetry in end-to-end driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

NVIDIA. Alpamayo 1.5: an open 10B reasoning VLA model for autonomous vehicles. Hugging Face `nvidia/Alpamayo-1.5-10B`, 2026a.

NVIDIA. Alpamayo 2 Super: model card and weights (OpenMDW-1.1). Hugging Face `nvidia/Alpamayo2-Super`, 2026b.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Robotics: Science and Systems*, 2024.

Open X-Embodiment Collaboration. Open X-Embodiment: robotic learning datasets and RT-X models. In *IEEE International Conference on Robotics and Automation*, 2024.

Junsung Park, Hogun Kee, and Songhwai Oh. Modality-augmented fine-tuning of foundation robot policies for cross-embodiment manipulation on GR1 and G1. *arXiv preprint arXiv:2512.01358*, 2025.

Simon Ramstedt and Christopher Pal. Real-time reinforcement learning. In *Advances in Neural Information Processing Systems*, 2019.

Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. SimLingo: vision-only closed-loop autonomous driving with language-action alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. SmolVLA: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

Wenchao Sun, Xuewu Lin, Keyu Chen, Zixiang Pei, et al. SparseDriveV2: scoring is all you need for end-to-end autonomous driving. In *European Conference on Computer Vision*, 2026.

Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. BridgeData V2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pp. 1723–1736. PMLR, 2023.

Jingke Wang, Zhenru Zhao, Shuangming Lei, et al. DriveStack-VLA: render-teacher alignment for BEV-based DeepStack vision-language-action model. *arXiv preprint arXiv:2606.24051*, 2026.

Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. *Advances in Neural Information Processing Systems*, 35:6119–6132, 2022.

Yifang Xu, Jiahao Cui, Zhihao Zhu, Hanlin Shang, Shan Luan, Mingwang Xu, Feipeng Cai, Neng Zhang, Yaoyi Li, Jia Cai, et al. WAM-Flow: Parallel coarse-to-fine motion planning via discrete flow matching for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24918–24928, 2026.

Ruolin Yang, Zilin Huang, Boyue Wang, Zhengyang Wan, Yuhao Luo, Zihao Sheng, and Sikai Chen. Beyond the leaderboard: Counterfactual diagnosis of end-to-end and VLA driving policies under domain shift. *arXiv preprint arXiv:2609.22582*, 2026.

Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. LogME: Practical assessment of pre-trained models for transfer learning. In *International Conference on Machine Learning*, 2021.

Zhiyuan Zhang, Zhenghao Jin, Yanlun Peng, et al. Bench2Drive-Robust: benchmarking closed-loop autonomous driving under deployment perturbations. *arXiv preprint arXiv:2605.18059*, 2026.

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. *Advances in Neural Information Processing Systems*, 38: 27920–27956, 2026.

A Details of the real-world field study

This appendix details the real-world field study of Section 5, which applies TRIAGE to a **fixed physical platform** (a full-scale vehicle retrofitted for autonomous driving). We ran four field experiments in September 2026, each testing one part of the TRIAGE workflow (Figure 1d) against its results on Bench2Drive (Jia et al., 2024) and NAVSIM (Dauner et al., 2024). Appendix A.1 describes the vehicle and its cameras, lists which candidates pass the sensing and latency gates, and gives the route and how the data were collected. Field experiment I (Appendix A.2) re-measures the latency of candidates on the onboard GPU (the **latency gate**); Field experiment II (Appendix A.3) runs the probes on a local recording and makes the selection; Field experiment III (Appendix A.4) measures the **sensing gate** on vehicle frames, comparing the factory camera with the added one; and Field experiment IV (Appendix A.5) drives DriveVA (Liu et al., 2026) in closed loop under command hold and plan tracking (the latency gate's consumption rule).



Figure 8: The fixed physical platform. (a) Vehicle, compute rack, added camera, safety driver and operator. (b) Fields of view from above: factory front (24.7°, red) and corner (87.8°) cameras, added front camera (102°, blue) and the 64° sensor contract (dashed). (c) All eight cameras at one instant. (d) Factory and added front cameras on the recording drive and on a hand-driven lap.

A.1 Platform, candidates, route and data collection

Fixed physical platform. The test vehicle is a full-size electric van retrofitted with drive-by-wire, an inertial navigation system and an onboard RTX A6000, the same GPU model as the laboratory workstation on which every latency in Table 1 was measured (Figure 8a). Its factory sensors are a forward camera (24.7° horizontal, 2448 × 2048, mounted 2.45 m high), four corner cameras (87.8° each, yawed about ±44° and ±135°), three lidars and a front radar (Figure 8b, c); these factory cameras are the fixed camera setup that every candidate is first checked against. Because they fail the sensing gate (below), we added three DJI wide-angle cameras, front, left and right (1920 × 1080, 100

to 102° horizontal measured with lens distortion; Figure 8c), of which only the front one, mounted 1.83 m high and pitched 8.9° upward, was used in the experiments. The front added camera was read at 29.9 frames per second over USB by the inference process and at 10 Hz as republished on the vehicle network for recording; its frames are published without undistortion, which our reprojection performs. Figure 8d shows the factory and added front cameras at the same instant.

Table 3: Sensing and latency gates for the candidates considered on the vehicle. *Factory*, *Added*: the sensing gate without and with the added camera; latency gate: laboratory p50 against the 1.2 s deadline. Blue: passes sensing; green: passes latency; red: reason for exclusion; –: not reached. On-vehicle latencies later removed WAM-Flow and Alpamayo-1.5 (Appendix A.2). Sources: constant velocity (Dauner et al., 2024), EgoStatusMLP (Li et al., 2024), DriveVA (Liu et al., 2026), ReCogDrive (Li et al., 2026b), DriveVLA-W0 (Li et al., 2026a), WAM-Flow (Xu et al., 2026), Alpamayo-1.5 (NVIDIA, 2026a), Curious-VLA (Chen et al., 2026), AutoVLA (Zhou et al., 2026), TransFuser (Chitta et al., 2023), DiffusionDrive (Liao et al., 2025), SimLingo (Renz et al., 2025), UniAD (Hu et al., 2023), VAD (Jiang et al., 2023), ORION (Fu et al., 2025), TCP (Wu et al., 2022).

Candidate	Sensor contract	Factory	Added	Lab p50 (ms)	Latency gate	Role
Blind anchors						
Constant velocity	none	pass	pass	0.01	pass	reference
EgoStatusMLP	ego state only	pass	pass	0.12	pass	reference
NAVSIM candidates						
DriveVA	front 64°, video	fail	pass	947	pass	default choice
ReCogDrive	front 64°	fail	pass	513	pass	challenger
DriveVLA-W0	front 64°	fail	pass	485	pass	outscored by DriveVA in every city
WAM-Flow	front 64°	fail	pass	781	pass	outscored by DriveVA in every city
Alpamayo-1.5	flexible (fed 64°)	fail	pass	959	pass	no driving score; latency and probes only
Curious-VLA	front 64°	fail	pass	4,067	fail	
AutoVLA	six surround cameras	fail	fail	1,249	–	
TransFuser	three cameras + LiDAR	not evaluated		69	–	LiDAR differs in model and mounting
DiffusionDrive	three cameras + LiDAR	not evaluated		85	–	as TransFuser
Bench2Drive candidates						
SimLingo	front 110°	fail	fail	598	–	$102° < 110°$
UniAD front-only	front 70°	fail	pass	722	pass	trained in simulation, no real-data score; never selected
UniAD	six cameras ($5 \times 70°$, rear 110°)	fail	fail	722	–	no rear camera
VAD	six cameras	fail	fail	567	–	no rear camera
ORION	six cameras	fail	fail	1,020	–	no rear camera
TCP	three 70° front cameras, stitched	fail	fail	63	–	no matching three-camera set

Sensing gate: factory versus added cameras. The gate is checked on paper, from the cameras each candidate consumes (Table 3). On the factory cameras every sighted sensor contract fails: the monocular candidates need a 64° to 110° front view, the six-camera ones also a rear view the vehicle lacks, and TCP three 70° front cameras at 0° and ±55°, stitched into one image; stitching cannot widen the view, since the cameras share no mounting point, and LiDAR sensor contracts are not evaluated, the vehicle's LiDAR differing in model and mounting. With the added camera, the six monocular NAVSIM candidates pass through the 64° virtual view, and of these Curious-VLA then fails the latency gate at 4,067 ms.

Camera calibration and virtual views. The added front camera's intrinsics and distortion come from a checkerboard calibration with the same field-of-view mode and stabilization setting, and



Figure 9: Route and data collection. (a) Recording drive (blue; Experiments II and III) and test loop (green; Experiments III and IV); numbers mark the places in (c), the star the start of the loop. (b) The safety driver. (c) The added front camera at each place.

its extrinsics from a lidar-to-camera calibration of the vehicle; the camera was not moved between calibration and the experiments. The virtual view reprojects its frames into the NAVSIM front camera's geometry ($f_x = 1545$ px at 1920×1080 with nuPlan distortion) at the median orientation of the nuPlan front camera. Fields of view are measured horizontal values, not the lens's nominal diagonal. The added camera is mounted higher than the nuPlan capture vehicle's front camera (1.83 against 1.52 m), a viewpoint shift reported with Experiment III rather than corrected.

Route. All driving took place on business-park and residential roads in the target city, posted at 25 mph or less (Figure 9a, with the added camera's view at seven places in Figure 9c). Experiments II and III use a 24-minute recording driven over 7.7 km through places 1 to 6 of Figure 9. The hand-driven laps of Experiment III and the closed-loop runs of Experiment IV use a 1.66 km test loop with six turns (place 7 in Figure 9), driven counterclockwise from a fixed start at an intersection.

Data collection and safety. A safety driver who had completed the institution's Institutional Review Board (IRB) certification training drove the recording and the hand-driven laps in normal traffic, with an operator on board running the recording and inference processes (Figure 9b). In closed loop the safety driver stayed at the controls: the controller was engaged only with both feet off the pedals, speed was capped at 1.5 m/s, the controller stopped the vehicle whenever no new plan arrived within 2.2 s, and any brake, throttle or steering input by the driver disengaged it at once. Such a disengagement, read from the controller's own enable flag together with the drive-by-wire override flags, counts as a takeover; each takeover was classified afterwards by a reviewer blind to the controller (Appendix A.5).

A.2 Field experiment I: latency on the vehicle

Protocol. Five candidates were timed on the onboard A6000 with no other process on the GPU: DriveVA, ReCogDrive, DriveVLA-W0, WAM-Flow and Alpamayo-1.5. Alpamayo-1.5, the model the vehicle's existing stack runs, has no driving score to rank by, so it is timed here but takes no part

in the selection. Each ran 20 warm-up and 200 timed calls per condition, recording p50, p95, failed calls, GPU power and clocks, and concurrent processes, in three conditions: the model process alone (*alone*); the vehicle's standard start-up with cameras, localization, logging and its three display tools (the ROS visualizer RViz, a live view of all eight cameras and a camera-exposure panel, all running on the same computer as the model) running, the model not controlling the vehicle (*full load*); and the same load with the display tools and the added camera's network node closed, the state in which Experiment IV was driven (*as driven*). The latency gate is judged on the as-driven condition.

Table 4: Experiment I: latency on the vehicle (ms). *Alone*: model process only; *full load*: standard start-up with display tools; *as driven*: without display tools, as in Experiment IV. *Over* 1.2 *s*: share of as-driven calls above the deadline; the gate uses the as-driven median.

Candidate	Lab p50	Alone p50	Alone p95	Full load p50	Full load p95	As driven p50	As driven p95	Over 1.2 s (%)	Gate
DriveVA	947	947	949	1133	1174	976	981	0.0	pass
ReCogDrive-2B	513	627	643	1334	1497	741	840	0.0	pass
DriveVLA-W0	485	509	515	980	1051	598	606	0.0	pass
WAM-Flow	781	900	935	2657	3010	1207	1262	59.5	fail
Alpamayo-1.5	959	1337	2289	4531	7490	2014	3452	100.0	fail

Results, model alone. Before any reading, the vehicle's environments reproduced the laboratory outputs on fixed test inputs: ReCogDrive to within 2×10^{-6}, DriveVA within its own bf16 run-to-run noise (0.75 against 0.78 m). Alpamayo-1.5, whose reasoning length depends on the image, was timed on the 200 windows of its laboratory measurement with the same settings, every rebuilt model input identical to the laboratory's. With the same GPU model, weights and inputs, DriveVA kept its laboratory latency (947 ms, Table 4) and DriveVLA-W0 rose by 5%, but WAM-Flow rose by 15%, ReCogDrive by 22% and Alpamayo-1.5 by 39% (959 to 1337 ms, 174 of 200 calls over 1.2 s). What the measurement shows is that a latency measured on one host did not carry over to another host with the same GPU model for three of five candidates, which is why Experiment I re-measures before any later stage.

Results, full load. Under full load, with the vehicle's standard start-up running, the sixteen host cores were saturated (load average 16 to 19); the largest consumers were the three display tools and the five factory camera compressors. Median latency rose to 1133 ms for DriveVA, 980 ms for DriveVLA-W0, 1334 ms for ReCogDrive, 2657 ms for WAM-Flow and 4531 ms for Alpamayo-1.5, so three of five cross the deadline under full load. The 1.2 s deadline was not changed: the procedure changes the deployment configuration, not the gate.

Results, as driven. A second session under load, run that evening, closed the display tools and the added cameras' ROS nodes, which is the state in which Experiment IV was driven: factory cameras, localization, logging and control services running, vehicle parked, load average 2 to 10 (Table 4, *as driven*). Median latency was then 976 ms for DriveVA, 598 ms for DriveVLA-W0, 741 ms for ReCogDrive, 1207 ms for WAM-Flow and 2014 ms for Alpamayo-1.5, between 3% and 51% above the readings with the model alone, so the display tools, not the sensors, account for most of the slowdown under full load. The latency gate is judged on this session because it matches the deployment: DriveVA, ReCogDrive and DriveVLA-W0 meet the 1.2 s deadline; WAM-Flow misses it by 7 ms and is excluded, the gate not being widened; Alpamayo-1.5 misses it with either setting. The default choice (DriveVA) and the challenger (ReCogDrive) both pass. With the settings the vehicle's existing stack uses (32 generated tokens instead of 256, five diffusion steps instead of ten, greedy decoding), Alpamayo-1.5 was faster, 1132 ms alone, 2559 ms under full load and 1688 ms as driven, but still over the deadline in 197 of 200 as-driven calls, so, under this deployment's 1.2 s deadline, it does not pass the latency gate.

Summary. Laboratory latency did not carry over to the vehicle: with the model alone, three of five candidates ran more than 10% slower than in the laboratory on the same GPU model, and under full load three of five crossed the 1.2 s deadline, mostly because of the display tools. As driven, the latency gate keeps DriveVA, ReCogDrive and DriveVLA-W0 and removes WAM-Flow and Alpamayo-1.5; the default choice and the challenger both pass.

A.3 Field experiment II: probes on a local recording and the selection

Table 5: Experiment II: local probe readings and the selection. Ranges from the four-city offline study; other columns from the vehicle recording. *TI*: temporal instability (m); *noise*: run-to-run noise. Colored: the six readings the rule compares, green within the range or twice the noise, red outside.

Candidate	TI, four cities	TI	A_γ	A_{jpeg}	D_{gray}	Noise
DriveVA	0.37–0.51	0.70	0.245	0.182	3.15	0.208
ReCogDrive-2B	0.63–0.82	0.84	0.179	0.135	2.11	0.000
DriveVLA-W0	1.08–1.28	1.39	0.891	0.680	5.72	0.000
WAM-Flow	0.45–0.55	0.71	0.280	0.139	1.34	0.068

Selection: no model.

Protocol. The safety driver drove the recording route (Figure 9) for 24 minutes with the factory and added front cameras recording. On the virtual view of this recording, the probe readings of Appendix E (temporal instability; the darkening and compression responses A_γ and A_{jpeg}; the response to a gray frame, D_{gray}; and run-to-run noise, from six repeated calls on one frame) were computed after the drive for DriveVA, ReCogDrive, DriveVLA-W0, WAM-Flow and the constant-velocity anchor. A candidate's readings count only if its sign diagnostic, the share of turning frames on which the plan turns the way the driver did, is at least 0.8. The probes should separate every sighted candidate from the blind anchor: the anchor responds exactly zero, and each sighted candidate's D_{gray} exceeds twice its run-to-run noise. The selection rule, set from the offline values, compares the local temporal instability, A_γ and A_{jpeg} of the default choice (DriveVA) and the challenger (ReCogDrive) with their four-city ranges (Table 5): if any reading leaves its range by more than twice the locally measured run-to-run noise (for A_γ and A_{jpeg}, converted to their units), with healthy sign diagnostics, no model is deployed and more data is to be collected; otherwise the challenger is deployed if the six readings, standardized over the four cities, lie nearest to Las Vegas, the one city where the challenger outscores the default choice in Table 1, and the default choice if not.

Results and the selection. The recording lasted 23.9 min (18.3 min moving, 7.7 km, mean moving speed 25 km/h) and yields 2842 five-frame windows; both front cameras were recorded at 10 Hz and the navigation system at 100 Hz. Replacing every frame with gray moved DriveVA's plan by 3.15 m and ReCogDrive's by 2.11 m, far above their run-to-run noise (0.208 m and 0), and it moved DriveVLA-W0 by 5.72 m and WAM-Flow by 1.34 m (run-to-run noise 0 and 0.068 m), so all four candidates are separated from the anchor (Table 5). Sign diagnostics were healthy (DriveVA 0.968 and ReCogDrive 0.974, on over 1000 turning windows each). The selection rule then chose no model: ReCogDrive's temporal instability (0.84) lies above, and its darkening and compression responses (0.179 and 0.135, against four-city ranges of 0.206 to 0.235 and 0.173 to 0.195) below, their four-city ranges; DriveVA's readings stayed within their allowances.

Reading the no-deploy decision. The decision does not mean that either model fails locally; it means that the offline evidence cannot rank the pair here. DriveVA left its range by more than ReCogDrive did (temporal instability 0.184 above the four-city maximum, against 0.022) but stayed within its allowance, whereas ReCogDrive's planner is deterministic, so its run-to-run noise and hence its allowance are zero: the rule tolerates no excursion for a deterministic model, a limitation of the rule. On its own, DriveVA showed no sign against deployment: it met the deadline in all three conditions of Experiment I, responded to a gray frame far above its run-to-run noise, and planned turns in the direction the driver took (0.968). Had the range test passed, the rule would have kept DriveVA, since the local readings lie nearest Pittsburgh, not Las Vegas. They lie farther from it (6.88 in standardized units) than any two cities lie from each other (2.0 to 4.9), and every candidate's temporal instability exceeds its four-city maximum; a shift shared by four architectures points to a common input path, such as the reprojected virtual camera, rather than to one model. Why ReCogDrive's readings moved is not identified; candidate causes, none tested, are the reprojected image, its acceleration input (differenced from navigation velocities and noisier than in its training logs), a turn-signal command in place of a route command, and the wide suburban roads.

Summary. On the local recording the probes separated every sighted candidate from the blind anchor, but ReCogDrive's readings left their four-city ranges, so the selection rule chose no model and none was deployed.

A.4 FIELD EXPERIMENT III: THE SENSING GATE ON VEHICLE FRAMES

Protocol. On the virtual view of every tenth window of the local recording, the response curves of RQ1 (center crops keeping 90, 75 and 60% of each image dimension; resolution reduced to 75, 50 and 33%) were repeated for the four candidates with an offline ratio (DriveVA, ReCogDrive, DriveVLA-W0, WAM-Flow), and for Alpamayo-1.5, which has no offline ratio. Offline, the crop-over-resolution ratio at the matched budget exceeded one in every driving and manipulation model. The turn/straight comparison measures the factory camera's sensing gap on synchronous frames: each of the five models plans on the factory view, fed unchanged as its front camera, and on the 64° virtual view, frames paired within 0.1 s, and windows are split into turning (heading change of at least 15° over the next 2 s) and straight. A sensing gap shows as the 2 s plan points of the two views lying farther apart on turning than on straight windows, the state-conditional signature RQ2 measured under a synthetic crop. We also compare the factory view's error against the driver's path at 2 s on turning and straight windows, and its departure from the virtual view with the offline displacement at the strongest crop.

Table 6: Experiment III: crop against resolution on vehicle frames. Plan displacement at the matched budget (56% of pixels) on every tenth window of the local recording (285 windows); offline ratios from RQ1. Green: ratio above one.

Candidate	Crop (m)	Resolution (m)	Ratio	Offline ratio
DriveVA	0.94	0.22	4.26	2.8
ReCogDrive-2B	0.42	0.08	5.00	2.2
DriveVLA-W0	1.61	0.68	2.37	2.2
WAM-Flow	0.61	0.10	6.36	5.7
Alpamayo-1.5	0.73	0.22	3.31	–

Results, crop and resolution. At the matched budget of 56% of the pixels the crop moved plans by 0.42 to 1.61 m and the resolution cut by 0.08 to 0.68 m (Table 6), so the ratio was 4.26 for DriveVA, 5.00 for ReCogDrive, 2.37 for DriveVLA-W0 and 6.36 for WAM-Flow: above one for every candidate and at or above each offline value; Alpamayo-1.5 gives 3.31.

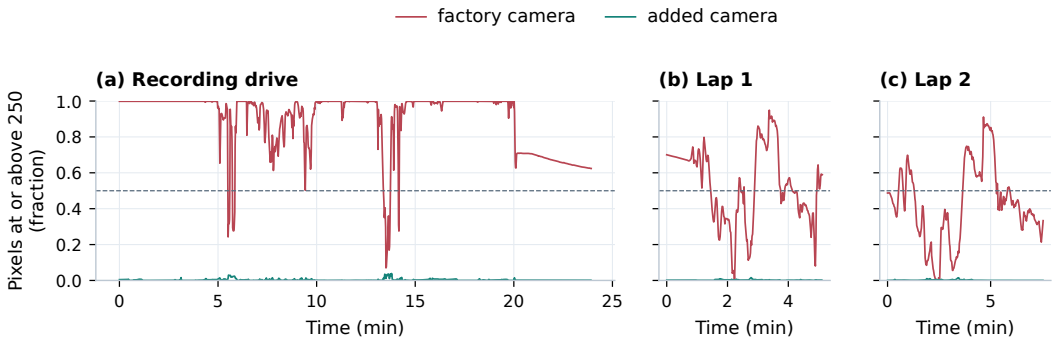

Figure 10: Experiment III: factory-camera overexposure. Share of pixels at or above 250, one frame per second, for the factory (red) and added (green) cameras, on the recording drive (a) and the two hand-driven laps (b, c); dashed: half the pixels, the overexposure threshold. The vehicle is parked after minute 20 of the drive.

Results, factory camera. The factory camera was overexposed throughout the local recording (median brightness 255 of 255; 97% of frames more than half saturated, in one frame per second), while the added camera was not (median 114; Figure 10a, Figure 8d), so the turn/straight comparison uses the second hand-driven lap of the test loop (32% of frames more than half saturated, against 56% on the first; Figure 10c and b), where 22 of 889 usable windows were turning. On this lap all five models show the signature: the factory camera's plans departed from the virtual view's more on turning than on straight windows, by 1.13 to 2.11 times (Table 7). With 22 turning windows, a factory camera still partly saturated and the two cameras mounted at different heights, this is one observation in the direction RQ2 found, not a measurement of its size. Against the driver's path at 2 s, the factory view's plans erred more than the virtual view's on turns for four of the five models (DriveVA 2.60 against 0.97 m), ReCogDrive being the exception; and over all windows they

departed from the virtual view's by more than the offline displacement at the strongest crop for each of the four models with an offline result (DriveVA 1.64 against 0.91 m), consistent with a 24.7° view spanning about a third of the image. Figure 7b shows one turning window of the hand-driven lap.

Table 7: Experiment III: factory view against virtual view. Second hand-driven lap, 22 turning and 867 straight windows; meters. *Deviation*: distance between the two views' 2 s plan points; *error to driver*: virtual / factory view; *offline crop*: offline displacement at the 60% crop. Green: ratio above one.

	Deviation, factory vs virtual			Error to driver		All windows	
Model	Turning	Straight	Ratio	Turning	Straight	Deviation	Offline crop
DriveVA	2.18	1.63	1.34	0.97 / 2.60	0.73 / 1.67	1.64	0.91
ReCogDrive-2B	0.88	0.78	1.13	0.84 / 0.56	0.58 / 0.75	0.78	0.59
DriveVLA-W0	5.40	3.04	1.78	1.93 / 4.19	2.07 / 2.22	3.09	1.71
WAM-Flow	0.92	0.79	1.17	1.00 / 1.28	0.59 / 1.09	0.79	0.54
Alpamayo-1.5	1.47	0.70	2.11	1.36 / 1.87	1.21 / 1.29	0.72	–

Summary. On vehicle frames a field-of-view crop again moved plans 2.4 to 6.4 times more than a resolution cut at the same pixel budget, at or above each offline value. On the 22 turning windows of the hand-driven lap, the plans all five models made from the factory view departed from those made from the virtual view more on turns than on straight road (1.13 to 2.11 times), one observation on one lap.

A.5 Field experiment IV: the latency gate in closed loop

Table 8: Experiment IV: the two controllers. DriveVA, two runs each, plan tracking first. *Stale command*: takeovers so rated by the blinded reviewer; *deadline misses*: share of engaged control steps with no new plan for over 1.2 s.

Controller	Uses each plan	Runs	Engaged	Takeovers	Stale command	Deadline misses
Plan tracking	tracked every 10 ms	2	6.1 min, 511 m	3	0	20%
Command hold	held until the next	2	7.4 min, 595 m	4	1	20%

Protocol. Although the selection returned no model, Experiment IV tests the latency gate's consumption rule on the vehicle: DriveVA, the default choice, drives the test loop under two controllers fed identical plans, about one every 1.5 s. *Plan tracking*, the vehicle's existing controller, tracks the newest plan with the current pose at every 10 ms control step, compensating for the plan's age; *command hold*, the same controller changed in one place, computes speed and curvature once per plan and holds them until the next, as in RQ4. The question is whether plan tracking removes the stale-command takeovers RQ4 found under command hold in simulation. The inference process attaches to each plan the pose nearest its newest frame (within 0.1 s). Plan age is the time from that frame, as received (the camera gives no exposure timestamp), to the control step that uses the plan; a deadline miss is an engaged control step on which no new plan has arrived for more than the 1.2 s deadline. Per run we record engaged time and distance, takeovers and the share of deadline misses. Each controller drove two runs, plan tracking first, with steering limited to a 10 m turning radius, a minimum look-ahead of 6 m and speed capped at 1.5 m/s (5.4 km/h), the setting of the vehicle's existing stack, adopted after the first plan-tracking run weaved under the default limits (Table 8). Takeovers are counted after the first stretch of each run, which was spent adjusting the equipment; disengagements there are not takeovers.

Results, plans and driving. The inference process sent every plan, attached to each a pose within 55 ms of its newest frame, and took 1.03 s per plan at the median in every run; plans arrived every 1.5 s at the median, so every cycle overran the 1.2 s deadline by about 0.3 s. Plan tracking was engaged for 6.1 min over 511 m with three takeovers, command hold for 7.4 min over 595 m with four; the longest engaged stretches were 176 s (219 m) and 238 s (309 m). Both controllers consumed the same plans: 20% of engaged control steps missed the deadline under each, and the image behind the plan in effect was 2.26 s old at the median (2.93 s at the 95th percentile), because each plan took about 1.5 s from frame to arrival and then stayed in use until the next arrived. At the end of the

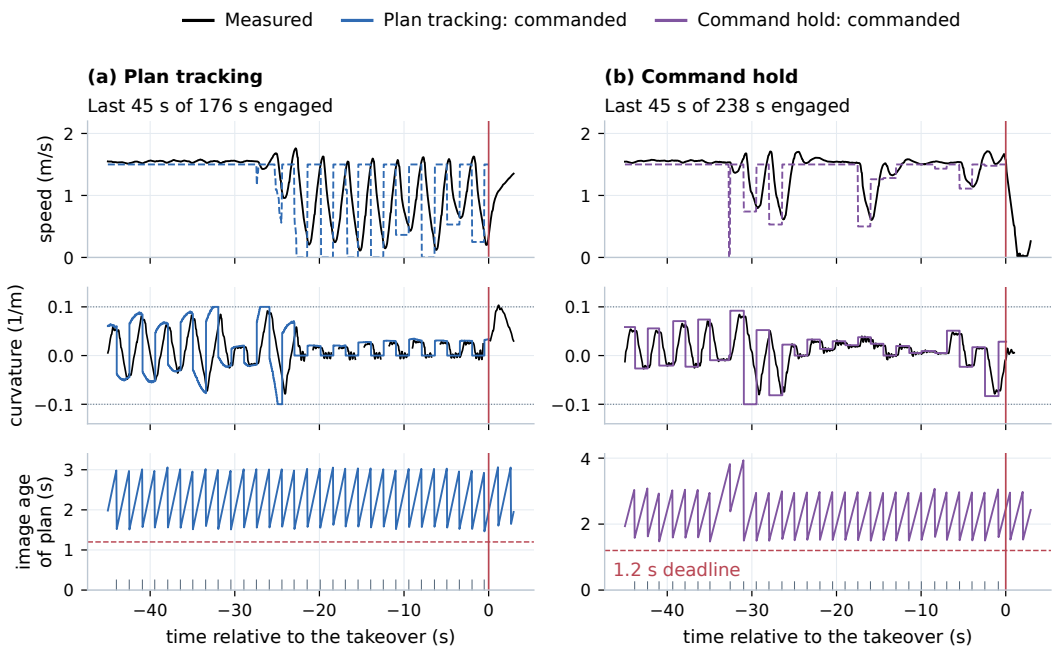

Figure 11: Experiment IV: the last 45 s before a takeover, for takeovers 3 (a) and 4 (b) of Table 9. Top: speed; middle: curvature, with the steering limit dotted; both measured (black) and commanded (color). Bottom: age of the image behind the plan in effect, ticks at plan arrivals; dashed: the 1.2 s deadline, which applies to plan arrival, not image age. Red line: the takeover.

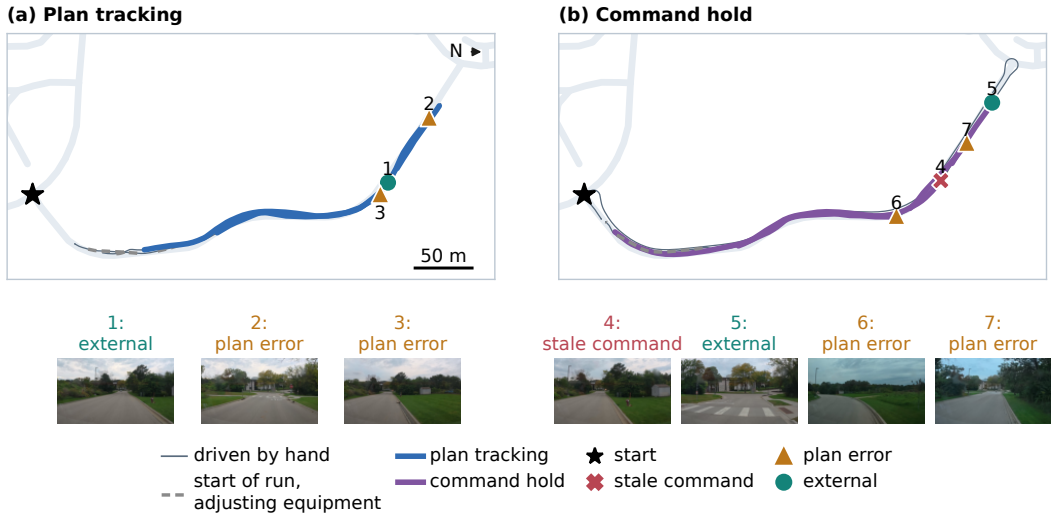

Figure 12: Experiment IV: closed-loop runs and takeovers. Top: track of each controller's two runs, north to the right, by hand (gray), during equipment adjustment (dashed) and under control (color), with takeovers numbered as in Table 9. Bottom: the image the model received at each takeover; two building signs are blurred.

longest stretch of each controller, plan tracking fell into a stop-and-go cycle (Figure 11a): every second plan, made after the vehicle had slowed below 0.7 m/s, advanced less than 1.3 m in 4 s, so the speed command fell to zero every two to three seconds; under command hold (Figure 11b) both commands moved in steps at each plan, and a 2.4 s gap between plans exceeded the controller's 2.2 s timeout and triggered its stop. All engaged driving stayed within about 370 m of the start, on the first street of the loop (Figure 12); at 5.4 km/h the rest of the loop was not reached under control.

Results, takeovers. Each takeover was summarized without its controller, run or clock time (the images the model was fed, the path against the newest plans, speed, yaw rate, plan age and offset from the newest plan) and classified by a reviewer given only these summaries and the category

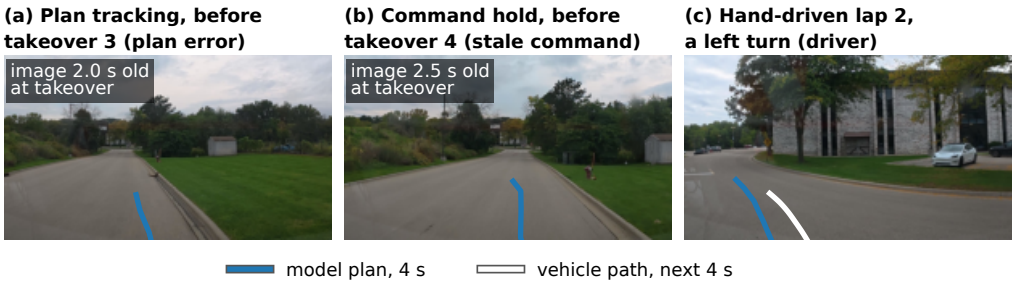

Figure 13: Experiment IV: plans in the image. Plan in effect (blue) before takeovers 3 and 4 (a, b) and on a hand-driven left turn (c), with the path driven over the next 4 s (white), projected on flat ground. Before the takeovers the vehicle moved less than 5 m, all below the image. Labels: age of the plan's image.

definitions (Table 9, Figure 12, Figure 13). The reviewer rated one takeover as stale-command (under command hold; Figure 13b), four as plan errors (two under each controller; one in Figure 13a) and two as external (an oncoming car under plan tracking; a crosswalk after a stop sign under command hold). Three of the four plan errors show the model slowing to near-stops or producing plans only a few meters long where the road was clear; Figure 13c shows a hand-driven turn for comparison.

Table 9: Experiment IV: every takeover while engaged, in time order, with the blinded rating. *By*: how the driver took over; *engaged before*: length of the engaged segment; *plan age*: age of the image behind the plan; *offset*: distance from the newest plan's path, left positive.

No.	Run	Controller	Engaged before (s)	By	Speed (m/s)	Plan age (s)	Offset (m)	Rating
1	1	plan tracking	108	brake	1.37	2.62	-0.36	external
2	1	plan tracking	39	brake, steering	1.45	1.63	-0.29	plan error
3	2	plan tracking	176	steering	0.33	1.96	-0.11	plan error
4	3	command hold	238	brake	1.57	2.45	-0.35	stale command
5	3	command hold	43	brake	1.01	2.38	+0.75	external
6	4	command hold	103	steering	1.60	1.65	-1.08	plan error
7	4	command hold	62	steering	0.65	2.47	-2.23	plan error

Summary. The reviewer attributed none of the three takeovers under plan tracking and one of the four under command hold to stale commands, the direction RQ4 found in simulation. With two short runs per controller and a single stale-command takeover, this is one observation, reported beside RQ4 and not pooled with it.

B Formal statements and proofs

B.1 Setting

This appendix states formally what Section 3 states in words, with the same symbols: configurations $c = (\pi, \iota)$, candidates trained under (S_i, T_i, D_i), the target (S', T', D') with N local frames, and the risk $R(c)$ of Section 3.1; lower is better. When L is a sum of per-step losses, $R(c)$ is proportional to $\mathbb{E}_{x\sim d^c}[\ell(x, u_c(x))]$, with d^c the normalized closed-loop state distribution. We write $R^*(\cdot)$ for the least risk attainable by any policy under the constraint in parentheses. *Post-training* is the operator

$$\mathcal{T}_N : \pi_\theta \longmapsto \pi_{\theta'}, \qquad \theta' = \mathcal{A}\big(\theta, \widehat{D}'_N\big), \tag{3}$$

where $\mathcal{A}$ is any training algorithm initialized at θ whose data $\widehat{D}'_N$ are N observations from the target sensors and their labels; architecture and interface are unchanged by definition, and altering either creates a new configuration.

The three terms of Equation (1) are defined below as differences of least attainable risks: Δ_{sense} in Appendix B.2, Δ_{lat} in Appendix B.3, and Δ_{dom} in Appendix B.4. The one property the framework uses is that $\mathcal{T}_N$ cannot reduce the first two for any N (Propositions 1 and 2), while only the third carries a coefficient that N drives down. They are listed, not summed: nothing here claims that $R(c)$ is their sum, or ranks them against each other.

B.2 SENSING GAPS

Write S_c for the observations the sensor contract of c consumes and S' for those the platform provides, and suppose the platform's observations can be generated from the sensor contract's by a stochastic map M, $S' = M \circ S_c$ (a lossy transformation: a cropped, lower-resolution or missing view). The sensing term is

$$\Delta_{\text{sense}}(c; S') \;=\; R^*(S') - R^*(S_c) \;\geq\; 0, \tag{4}$$

the risk that no policy observing S' can avoid relative to one observing S_c. The operational gate g_{sense} in Equation (2) checks whether the configured inputs satisfy the checkpoint's sensor contract. It does not evaluate Equation (4) or identify its value from a sensor list.

Proposition 1 (Post-training cannot undo a lossy transformation)**.** *For every policy π' that is a function of observations from the target sensors, including any post-processing of S' and any $\mathcal{T}_N$-fine-tune of π on data collected through S', $R(\pi') \geq R^*(S') \geq R^*(S_c)$. The second inequality is strict if and only if no optimal decision rule under S_c can be implemented through M. Hence no post-training budget closes Δ_{sense}; only changing the sensors or the requirement can.*

Proof sketch. Any rule $\phi'(a \mid o')$ for S' is reproduced from S_c by

$$\phi(a \mid o) \;=\; \int \phi'(a \mid o')\, M(do' \mid o), \tag{5}$$

with identical risk in every decision problem (the easy direction of Blackwell's comparison of experiments (Blackwell, 1953)), so the risks attainable under S' are a subset of those attainable under S_c. Applying M independently at every step extends this to history-dependent policies and hence to closed-loop risk (Le Cam, 1964). Equality holds exactly when some optimal rule under S_c can be written in the form of Equation (5). □

Where the deficit sits. In a one-step decision problem with state prior p, let ϕ^*_c and $\phi^*_{S'}$ be optimal rules under S_c and S'. Then

$$\Delta_{\text{sense}}(c; S') = \mathbb{E}_{x\sim p}\big[r(x)\big], \qquad r(x) = \mathbb{E}\big[\ell(x, \phi^*_{S'}(o')) \mid x\big] - \mathbb{E}\big[\ell(x, \phi^*_c(o)) \mid x\big], \tag{6}$$

and $r(x) = 0$ on states where the two rules take the same action; $r(x)$ may be negative on some states, but its average is not. In the sequential case the two optimal policies visit different states, so Equation (6) locates the deficit rather than decomposing it; the experiments read it as a prediction about which states carry the residual (turns, for a narrowed view), not as a quantity to be summed.

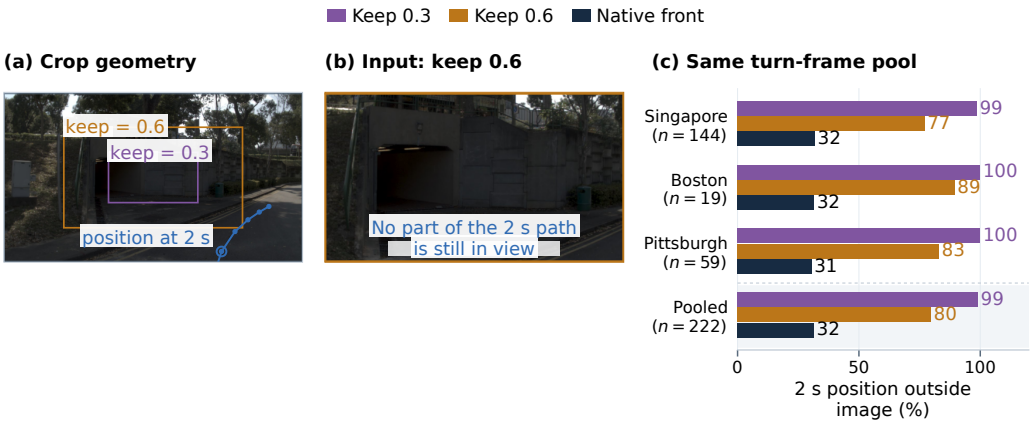

Figure 14: On turns, the crop removes the evidence the decision needs. (a) A turning frame with the logged 2 s path and both crop boundaries. **(b)** The frame as delivered at `keep` $= 0.6$. **(c)** Share of the 222 turning frames whose 2 s position leaves the image.

This is visible directly in the data: on turning frames the position the vehicle occupies two seconds later leaves a moderately cropped image 79.7% of the time (Figure 14c; one such frame before and after the crop in a and b).

Two refinements are used in the main text. The criterion is measured-*or-inferable*: statistically redundant information, such as monocular depth, is recoverable by data (Kwon et al., 2025), which does not contradict Proposition 1, since the proposition only bars recovering what the observations never carried. And a policy with temporal memory may recover in one frame what another frame lost, which shrinks the practical deficit; the deficit must therefore be established by experiment, not by comparing sensor lists.

B.3 LATENCY GAPS

The control period δ is the interval at which the controller needs a new command. The deadline d' of the gate is set by the controller and fixed before any τ_c is measured: $d' = \delta$ when a missed control step replays the last command, and d' is the replanning deadline when the controller tracks the last plan with fresh state.

τ_c is the median per-frame latency measured on the deployment GPU; the 99th percentile is recorded and does not enter the gate. Write $R(c; \tau, \varsigma)$ for the risk of c when its outputs arrive with latency τ and are consumed under rule ς, and $R^*(S'; \tau, \varsigma)$ for the least such risk over all policies observing S'. The latency term is

$$\Delta_{\mathrm{lat}}(c; \tau_c, \varsigma) \;=\; R^*(S'; \tau_c, \varsigma) - R^*(S'; 0, \varsigma) \;\geq\; 0. \tag{7}$$

Proposition 2 (Latency-gap structure)**.** *With $k = \lceil \tau_c/\delta \rceil$ pending outputs, the closed loop is the delayed decision process*

$$\tilde{x}_t = \big(x_t,\ \xi_{t-1}, \ldots, \xi_{t-k}\big), \qquad u_t = \varsigma\big(\xi_{t-k},\ p_t\big), \tag{8}$$

with ξ the model outputs, p_t fresh proprioception and ς the rule by which the loop consumes stale outputs (Katsikopoulos & Engelbrecht, 2003; Ramstedt & Pal, 2019). (i) $R(c; \tau, \varsigma)$ is a property of the loop, not of π alone. (ii) $\mathcal{T}_N$ updates θ but neither k, which architecture and compute fix, nor ς, so every post-trained policy still computes ξ_{t-k} from observations at least k control steps old, and for every N

$$R(\mathcal{T}_N c; \tau_c, \varsigma) \;\geq\; R^*(S'; \tau_c, \varsigma) \;\geq\; R^*(S'; 0, \varsigma). \tag{9}$$

The second inequality is Proposition 1 applied to the delayed history, a lossy transformation of the fresh one, and is strict whenever the optimal action depends on disturbances realized within the delay window. Post-training can therefore recover the part of π's own excess risk that is predictable from delayed observations, but not Δ_{lat}, and it does not change the failure mode that k and ς determine. (iii) Under command hold, $\varsigma_{\mathrm{cmd}}(\xi_{t-k}, p_t) = u(\xi_{t-k})$ ignores p_t and the error grows open loop over the delay. Under plan tracking, $\varsigma_{\mathrm{plan}}(\xi_{t-k}, p_t) = \Gamma(\xi_{t-k}, p_t)$ with a feedback tracker Γ; if the tracked loop is contracting, then

$$R(c; \tau_c, \varsigma_{\mathrm{plan}}) - R(c; 0, \varsigma_{\mathrm{plan}}) \;\leq\; K \cdot \mathbb{E}\big[\, \|\xi_t - \xi_{t-k}\|_\infty \big], \tag{10}$$

with K a constant that absorbs the loss's Lipschitz constant and the loop's contraction rate.

Two remarks on (ii) and (iii). The recoverable part in (ii) is real: if the state drifts at a known constant rate, learning to extrapolate removes the delay's error entirely, which is why the latency term is defined through the best delayed policy rather than through π. In (iii), under command hold a double integrator whose held acceleration command is off by Δa is displaced by $\frac{1}{2}\|\Delta a\|(k\delta)^2$ after the delay; without a contracting tracker, errors compound across steps and Equation (10) does not hold. The drift on the right of Equation (10) is measurable on recorded plans without running anything: on the turning frames of the model being fine-tuned it is 1.36 m at a staleness of 500 ms (maximum over the eight plan waypoints of the per-waypoint Euclidean distance, frame-weighted over 144, 19 and 59 turning frames in the three cities that have them). We report it without converting it into a bound, since K is not measured here. The repairs for this class are therefore a faster compute tier or an architecture on the consuming side; a wrapper changes ς only if the downstream consumer accepts stale plans (Figure 4d shows one failing where the consumer rejects them). Delay-aware retraining (Chen et al., 2021) changes the policy's input interface, so in this taxonomy it is an architectural repair, not post-training Equation (3).

B.4 DISTRIBUTION GAPS

This is the term $\Delta_{\text{dom}}(c; D', N)$ of Equation (1): appearance, layout and behavior shift between the training domain D and the target D', covariate shift and, where local behavior differs, concept shift. The domain-adaptation bounds of Ben-David et al. (2010) are stated for binary classification under 0-1 loss, not for trajectory losses, so we use them as a structural analogy. For a hypothesis π trained on the source alone,

$$R_{D'}(\pi) \;\le\; R_D(\pi) \;+\; \tfrac{1}{2}\, d_{\mathcal{H}\Delta\mathcal{H}}(D, D') \;+\; \lambda, \tag{11}$$

with λ the combined error of the best hypothesis on both domains, which absorbs concept shift. Post-training does not shrink $d_{\mathcal{H}\Delta\mathcal{H}}(D, D')$, a property of the two domains; it changes what the model is trained on. The relevant statement is the paper's Theorem 3: if $\hat{\pi}$ minimizes an α-weighted empirical risk over m source and N target labelled samples, and $\pi^*_{D'}$ is the best hypothesis on the target, then with high probability

$$R_{D'}(\hat{\pi}) \;\le\; R_{D'}(\pi^*_{D'}) \;+\; \epsilon_{\text{est}}(\alpha, N, m) \;+\; 2(1-\alpha)\big(\tfrac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(D, D') + \lambda\big), \tag{12}$$

where the estimation term ϵ_{est} shrinks as N grows and grows with the weight α placed on the target. As N grows the best α moves toward 1, the divergence enters with a vanishing coefficient, and what remains is target estimation error; this is the sense in which N reduces Δ_{dom}, while the divergence itself stays as it is. No choice of α or N enters Equation (4) or Equation (7), so this is the only gap class at whose cause post-training is aimed; Figure 5 measures the rate, and the robustness cost (Kumar et al., 2022) that accompanies it.

B.5 WHY THE GATES COME BEFORE RANKING

The two gates of Equation (2) are operational checks, not measurements of the theoretical risk gaps (Section 3.3); passing them certifies neither a zero gap nor safe deployment.

Why this order? A deployer who picks by benchmark score takes the top-scoring candidate c^{score} whether or not it fits the sensors. TRIAGE can lose to that deployer only if skipping the gates costs less than a ranking mistake.

Remark 1 (When gating first is better)**.** *Suppose any candidate that fails a gate has risk at least R_{gate}, and the pick $\hat{c}$ has risk at most ϵ_{rank} above the best feasible candidate, $R(\hat{c}) \le \min_{c\in G} R(c) + \epsilon_{\text{rank}}$. If c^{score} fails a gate and $R_{\text{gate}} > \min_{c\in G} R(c) + \epsilon_{\text{rank}}$, then*

$$R(\hat{c}) \;\le\; \min_{c\in G} R(c) + \epsilon_{\text{rank}} \;<\; R_{\text{gate}} \;\le\; R\big(c^{\text{score}}\big). \tag{13}$$

If no candidate fails a gate, $G = \mathcal{C}$, and $\hat{c}$ is the score-first pick whenever $\widehat{\Delta}_{\text{dom}}$ orders candidates as the score does.

In words: when the scoreboard's favourite does not fit the sensors, gating first is safer as long as running an unfit candidate costs more than a ranking mistake among fit ones; when every candidate fits, gating first changes nothing. The remark is elementary, and its content is the premise $R_{\text{gate}} > \min_{c\in G} R(c) + \epsilon_{\text{rank}}$, the one value judgment the framework makes. Our measurements cannot check it as a number, because its two sides are in different units: a gate violation shows up in closed loop as collisions on 80 to 90% of routes (Figure 4), a ranking mistake in open loop as 0.18 to 1.35 PDMS points of regret (Figure 6b; regret is the PDMS a pick gives up against the best candidate in the held-out city). A deployer who would accept a collision-class outcome in exchange for a few points of score should not gate first.

C ALGORITHM AND COSTS

This appendix collects the formal statement of the procedure and its cost accounting. Stages 1 and 2 evaluate the operational checks g_{sense} and g_{lat}, not the theoretical risk gaps of Equation (1); Stages 3 and 4 estimate the ranked term Δ_{dom}; Stage 5 spends the budget N on it (Figure 1d).

What the procedure costs. Cheap evidence runs first and cuts the pool before expensive evidence is owed. On the thirteen-configuration pool of this paper (two of them blind anchors), Stages 1–2 cost one person-afternoon of sensor-contract reading plus a few minutes of timing per candidate with no

Algorithm 1 TRIAGE: repairability-ordered selection

Require: pool $\mathcal{C}$; locked target (S', T', D'); deadline d'; local logs ($\sim$10 min); data budget N

1: **Read on paper:** for each $c \in \mathcal{C}$, record the sensor contract and the documented memory footprint
2: **Sensing gate (g_{sense}):** drop c if its required channels are neither contained in S' nor closable by a semantics-preserving conversion; log conversions as named interface interventions that re-enter as their own configurations
3: **Latency gate (g_{lat}):** for each candidate that passes the sensing gate, measure the median latency τ_c on the deployment hardware in its operating state; drop c if it does not fit in the GPU's memory or if $\tau_c > d'$, where d' is the control period under command hold and a preset replanning deadline when the controller tracks the last plan with fresh state. Validate that execution configuration in closed loop before release; stale-image tolerance does not validate stale-plan tolerance
4: **If no candidate survives: return** the failing gate per candidate (recorded as pass, fail or unevaluated); no ranking is defined. Either commission the repair that the failing gate's class names, or adopt plan tracking as a new configuration with a deadline specified independently of measured candidate latencies and validated before release; re-enter both gates. Never widen a gate to obtain a selection
5: **Probe:** run remaining candidates offline on recorded local logs; verify sign diagnostics and blind anchors; normalize each response by the model's response to a constant gray frame; report readings below the model's run-to-run noise as not attributable; compute probe readings
6: **Rank (estimate Δ_{dom}):** place remaining candidates into their architectural tiers (traditional stacks; VLAs and world models), the probe readings confirming or vetoing each placement; within the top tier, select by the strongest target-relevant score, preferring models trained on target-domain data at equal scores; designate the runner-up as challenger. Probe readings place a tier and do not rank within one
7: **Allocate and verify:** spend N on post-training the selection while monitoring tracking error, perturbation sensitivity and reliance on the image together; a dose whose sensitivity exceeds its zero-shot level is not deployable until it recovers; test any converted interface or moderate sensory deficit in closed loop before release; buy hardware only for sensing gaps; commission architecture only for latency gaps
8: **return** feasible set, selection + challenger, budget allocation

other process on the GPU; on a monocular camera setup they excluded the six-camera and LiDAR sensor contracts, and one candidate on its own documented 72,115 MiB memory footprint, before a single frame was processed. Stage 3 adds ten minutes of offline inference on recorded logs per remaining candidate. Only Stage 4 requires the expensive artifact, a reproduction of each remaining candidate's published score under the version-fixed evaluator, up to a GPU-day per model (RQ8).

The probe readings, their sanity checks and the two checks that bound what they license are defined in Appendix E.

D EXTENDED RESULTS

This appendix gives the complete account of every research question, with the setup, all results and the per-city readings that the main text summarizes. Every frame and trajectory drawn here is drawn from the recorded per-frame measurements, and example frames are medians under a stated ranking, not extremes. Driving models are cited in Table 3 and manipulation models in Table 13.

D.1 THE SENSING GATE: MEASURE, ATTEMPT REPAIR, TEST SAFETY

This section validates Stage 1 of Figure 1d and the closed-loop step of Stage 5 by following one deficit through the framework's arc. We measure how much a sensory loss damages behavior (RQ1), test whether fine-tuning can compensate it (RQ2), and ask what open-loop compensation is worth in closed loop (RQ3).

RQ1: At equal pixel budgets, does structure loss damage more than detail loss?

Setup. Open loop, real frames, seven configurations, four cities. Center crops that keep 90, 75 and 60% of each image dimension, resampled to native size, remove structure; downscaling each dimension to 75, 50 and 33% and back removes detail. Both are read as mean plan displacement on identical frames; blind anchors are included as sanity checks that must respond exactly zero.

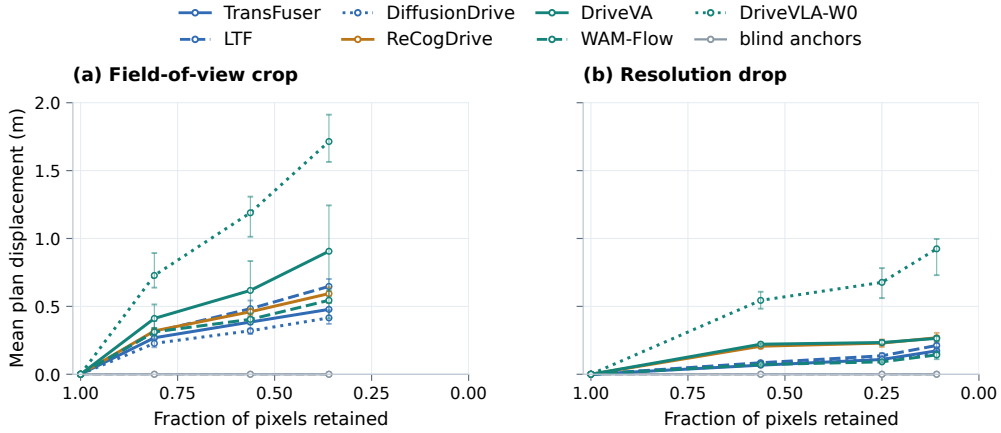

Figure 15: RQ1: losing view vs. losing sharpness. (a) Response to field-of-view crop (lines: four-city mean; bars: range across cities). **(b)** Response to resolution reduction, same axes. Blind anchors read zero on both.

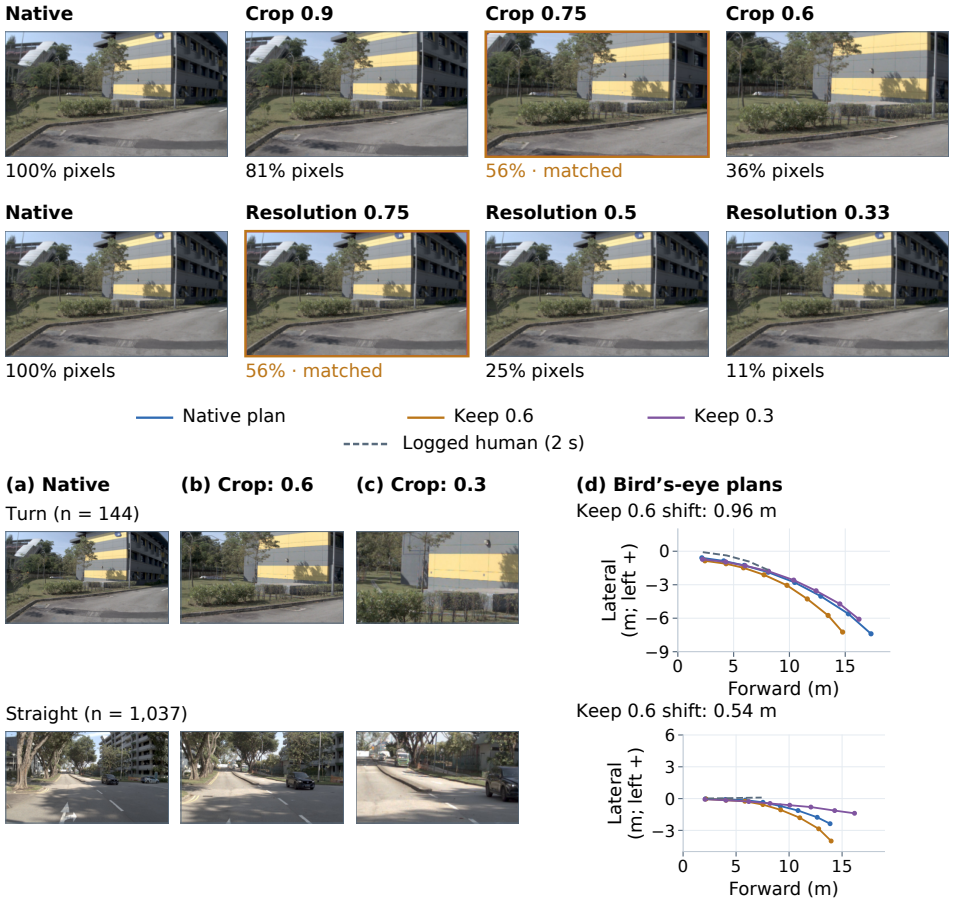

Figure 16: The matched pixel budget, and the same crop on a turn and on a straight road. *Top:* crop and resolution ladders on one Singapore turning frame; both keep $\texttt{keep}^2$ of the pixels and meet at $\texttt{keep} = 0.75$ (outlined), where Figure 2a is read. **(a–c)** Median Singapore turning and straight frames, native and cropped. **(d)** Their plans in the ego frame; dashed: logged human path at 2 s.

Results. **Losing view costs 2.2–$5.7\times$ more than losing sharpness.** Every sighted model's plans displace further as the view narrows (Figure 15a): DriveVLA-W0 climbs steepest, from 0.73 m at the mildest crop to 1.71 m at the strongest, and the two LiDAR-carrying stacks, TransFuser and DiffusionDrive, climb least (0.21 and 0.19 m). Resolution curves (Figure 15b) are shallower on the same axes, and blind anchors sit at exactly zero in both panels. With 56% of pixels kept on both axes (Figure 2a), the two inputs at the top of Figure 16, losing them at the border costs 2.2–$5.7\times$ more than losing them everywhere, for every sighted configuration and in every city (the smallest city-level ratio is $1.9\times$). Six of the seven respond less to $1/9$ of native resolution than to the mildest crop, which removes only 19% of pixels; DriveVLA-W0 is the exception.

The ratio does not track the sensor suite; the absolute stake does. The three largest ratios belong to the tri-camera-plus-LiDAR TransFuser, its camera-only sibling LTF and the monocular WAM-Flow (all $5.7\times$), the smallest to the monocular ReCogDrive and DriveVLA-W0 ($2.2\times$). The closest comparison in the pool, TransFuser against LTF, two separately trained checkpoints of one architecture with and without LiDAR input, shows LiDAR lowering the crop response by 20% and the resolution response by 21%, which leaves the ratio unchanged. The absolute response differs more: at the matched budget the two largest are monocular, DriveVLA-W0 (1.19 m) and DriveVA (0.62 m), while the other two monocular candidates (0.46 and 0.40 m) sit among the traditional stacks.

Takeaway. At equal pixel budgets, losing view costs several times more than losing sharpness for every configuration, whatever its sensors: this is why the framework treats a narrower view as a gate and a lower resolution as a format conversion. When forced to trade sensing, give up resolution before field of view. The ratio says how uneven the trade-off is and the absolute response how much is at stake; the two rank the pool differently.

RQ2: Can fine-tuning compensate a moderate sensory deficit?

Setup. Open loop, real frames. We fine-tune TCP under the 0.6 field-of-view crop on each of its three stitched cameras, evaluated under the crop, against a matched-budget uncropped run with identical data, hyperparameters, and epochs. TCP is used because a paired design needs a public, exactly replicable training recipe; RQ3 covers the other tiers in closed loop. We report turning frames separately, where the missing lateral view should matter most (252 turning frames across three cities; the fourth contributes none in this window). We repeat the design on ReCogDrive, a vision-language-action model, with a LoRA (Hu et al., 2022) recipe we denote LoRA(LM)+head, used for every adapted ReCogDrive-2B: its vision-tower adapters did not train, so the vision encoder stays frozen; its set of turning frames is a different 222 frames. The criterion: a sensing gap should leave a residual of at least 25% of the uncropped repair.

Results. **Whether the residual survives depends on the base model.** Before any adaptation the damage sits on turns: on ReCogDrive the crop keeping 60% raises tracking error against the human driver by 0.221 m on turning frames and 0.091 m over all frames, $2.4\times$ concentrated. Figure 16 shows the same crop on one turning and one straight frame (a–c) and the plans it moves (d). On TCP both runs recover 73–76% of their zero-shot turn error (Figure 3a), and the residual, though of the predicted sign, is 1.3% of the repair: the criterion fails by a factor of twenty. On ReCogDrive the residual on turning frames is 0.056 m against a repair of 0.130 m, or 43% (Figure 3b), and sixteen times the all-frame residual (0.0035 m; about 0.8 mm on straight frames): straight driving is repaired, turns are not, and the criterion is met. Two caveats apply: the fine-tuning recipe left the vision encoder frozen, a rival explanation for the residual that an encoder-adapting run would separate, and a third criterion, the residual against the crop's own zero-shot damage, passes by under a millimetre, carried by Boston's 19 frames, and fails in Singapore, so we do not rely on it.

The larger the gap, the less local data buys back. Keeping 30% raises turn damage to 0.371 m and all-frame damage to 0.251 m, so the damage is no longer turn-specific ($1.5\times$). The residual is then 0.133 m, 101.8% of the same repair (Figure 3c); the repair-normalized criterion passes on the pooled turning frames and in Singapore and Pittsburgh, and the crop-damage criterion in every city with turns (0.577, 0.671, 0.705 compensated). From $\texttt{keep} = 0.6$ to 0.3 the zero-shot turn damage grows $1.67\times$ while the residual grows $2.36\times$, so the compensated fraction falls from 0.746 to 0.642. The severe residual is only $5.1\times$ concentrated on turns ($16\times$ at the moderate crop): the moderate crop locates the surviving deficit, the severe crop shows that one survives in every city. The residual is positive in all three cities with turns ($+0.041$, $+0.149$, $+0.065$ m at $\texttt{keep} = 0.6$; Figure 3d),

while the residual-to-repair ratio is undefined in Boston, where uncropped adaptation raises turn error from 0.793 to 0.874 m. Per frame, the damage, repair and residual agree in sign with their pooled values on 73%, 70% and 62% of turning frames (all three on 31%), so we judge the pooled turning frames.

Takeaway. Whether a moderate sensory deficit can be adapted away depends on the base model, the fine-tuning recipe and the dose as well as on the deficit. The traditional stack, retrained end to end, relearns to drive from the narrower view (criterion not met), consistent with the measured-or-inferable criterion of Section 3.2. The vision-language-action model under LoRA(LM)+head, its vision encoder frozen, keeps a residual on the frames the theory predicts, the turns (criterion met). Neither result says a moderate crop is safe. Do not conclude from open-loop metrics that a sensor downgrade has been trained away; the next question is always the closed-loop one.

RQ3: Does a moderate sensory deficit harm safety even when capability metrics look intact?



Figure 17: Closed-loop inputs for RQ3. (a–d) The crop keeping 60%: captured frame with the kept region outlined, and the frame as delivered, for TCP (one of its three cameras) and SimLingo. **(e–g)** One delivered frame without occlusion and at severities 50 and 80. **(h–j)** Pixels black in all 300 consecutive frames: none at severity 50, the central 36% at severity 80.

Setup. Closed loop, simulation, where ground-truth safety is measurable. First probe, occlusion: a black rectangle is applied to every camera the configuration declares, redrawn at an independent random position per camera and frame, at the upstream protocol's severities 50 and 80, per-dimension ratios that black out 25% and 64% of each frame. Second probe: the crop keeping 60% of RQ1 and RQ2, introduced zero-shot on two models spanning the pool's tiers, TCP (a traditional stack) and SimLingo (the closed-loop champion VLA), over 30 runs (two configurations, five routes, three seeds) against native baselines (inputs in Figure 17a–d); ORION, admitted mid-study, repeats the crop run on the same routes and seeds.

Results. **Occlusion: redundant cameras help only up to moderate severity.** At severity 50 the two forward-facing configurations (TCP, three stitched cameras; SimLingo, one wide camera) see collision route rates rise by 0.3, while the two six-camera configurations hold or fall (UniAD 0.40 to 0.40, VAD 0.27 to 0.20; Table 10). At severity 80 every configuration fails regardless of camera count. Figure 17e–j shows why the effect changes abruptly: essentially no pixel is black

Table 10: Occlusion (RQ3). Collision route rate / route completion, Town13, seeds pooled. Severities 50 and 80 black out 25% and 64% of every camera frame.

Configuration	no occlusion	severity 50 (area 25%)	severity 80 (area 64%)
TCP (three forward cameras, stitched)	0.07 / 0.87	0.40 / 0.88	0.80 / 0.71
SimLingo (single wide camera)	0.00 / 1.00	0.30 / 1.00	1.00 / 0.66
UniAD front-only (masked)	0.40 / 0.77	0.40 / 0.76	1.00 / 0.63
UniAD (six-camera)	0.40 / 0.91	0.40 / 0.80	1.00 / 0.75
VAD (six-camera)	0.27 / 0.80	0.20 / 0.85	1.00 / 0.69

in every frame at severity 50, whereas the central 36% is black at severity 80. Since every camera is occluded independently, no view stays intact, and six overlapping, partially occluded fields carry more recoverable scene than one or three forward fields. The protection does not track the number of views reaching the model, however: UniAD with five of its six views zeroed does not degrade either (0.40 to 0.40), though its unperturbed rate is already the highest in the table, a confound this design cannot separate. The rectangles come from an unseeded generator, so their sizes replicate and their positions do not.

The crop damages safety, in a channel that differs by model. For TCP and SimLingo, route completion stays within the seed-noise bound (TCP -0.1 and SimLingo -11.7 points, against bounds of 8.4 and 18.0; Figure 2c), while collision route rate rises beyond it, all three seeds landing on the same elevated value (0.067 to 0.400 for TCP, 0.000 to 0.200 for SimLingo; panel d). SimLingo also acquires a failure its native runs never show: 4 of 15 cropped routes end in a scenario timeout, running past the 200 s limit. ORION is damaged in a different channel. Its route completion falls from 0.973 to 0.770 with a seed range of 0.0026 (native 0.081), and the routes abandoned with the vehicle stuck rise from 1 to 5 of 15; its collision route rate swings by up to 0.4 across seeds and is uninformative. In both patterns the damaged channel has near-zero seed variance while the intact channel swings widely, so a single-seed study could misread either model.

Takeaway. Open loop, adaptation removes most of the crop's damage on one base model and leaves a residual on another; closed loop, the same crop on released checkpoints damages safety through a channel that differs by model. Because RQ2 and RQ3 use different checkpoints, this shows only that open-loop repair does not certify closed-loop safety; it does not show that the adapted residual is what collides. Moderate deficits are therefore tested in closed loop, on several channels and seeds, and structural ones gate outright; occlusion is priced as occluded area on every camera.

D.2 The latency gate: a safety property of the loop

This section validates Stage 2 of Figure 1d.

RQ4: Is a latency violation a safety event, what decides its failure mode, and can wrappers or fine-tuning repair it? *Setup.* Closed loop in simulation; latency measured on real hardware. We time every model per frame with no other process on the GPU, then enforce inference deadlines in closed loop on five routes per town: 500 ms in Town03 and Town13 (clear weather, three seeds) and Town12 (fog, two seeds), and 200 ms in Town03. Under a deadline, a missed control step replays the previous actuation command (command *hold*). To separate staleness from how stale outputs are consumed, a second axis freezes each camera's input in random bursts of up to 1 or 3 s while the model keeps replanning on fresh speed and position and the controller tracks with fresh vehicle state (Appendix E). Finally we attempt two repairs. One is an asynchronous-planning wrapper delivering stale plans to the unchanged controller, implemented three ways (coordinate-frozen, raw-GPS pose, filtered pose); the other is fine-tuning, whose effect on latency we measure directly.

Results. **A missed deadline is a safety event whose form depends on the model.** Latency spans five orders of magnitude on one GPU (Figure 4a): 0.01 and 0.12 ms for the blind anchors, 63–85 ms for the traditional stacks of Table 1 and 0.49–4.1 s for every VLA and world model, so of these only the traditional stacks fit a real-time budget as shipped. In Town13, where SimLingo completes every route without a collision when no deadline is enforced (Figure 4b), a 500 ms deadline leaves it still moving but colliding: it covers 48% of route length and collides on 0.8 of routes, and the pattern repeats in Town03 (48%, 0.8) and in fog in Town12 (58%, 0.9). At 200 ms (Town03) it does not move and so does not collide. TCP, inside its budget, keeps its completion at both deadlines.

ORION, a second modern VLA admitted mid-study, stops rather than collides, as expected when latency exceeds the deadline: its 1,020 ms latency (p99 1,032 ms) exceeds both deadlines, so all 36,000 of its scheduled calls miss and over thirty routes it never leaves the start (completion and collision route rate both 0.00, against 97% completion without a deadline).

Stale images are tolerated; stale commands are not. Freezing the camera input in bursts of up to 1 or 3 s while the loop keeps acting on fresh vehicle state (Figure 4c; two seeds) leaves collision route rates near baseline: SimLingo stays at 0.00, UniAD at 0.40, VAD moves from 0.27 to 0.30 and TCP from 0.07 to 0.10 and 0.20. Replaying the stale command at 500 ms instead raises SimLingo's rate to 0.80. Planning from an image up to a few seconds old is tolerated; holding an old steering angle is not.

Neither a wrapper nor fine-tuning moves a candidate across the gate. None of the three asynchronous-planning wrappers works: the coordinate-frozen one had a coordinate-frame error and was not run, the raw-GPS one stalled at 4% of its first route, and the filtered-pose one reproduced that route bit-identically and, even without a deadline, completes 0.07 of route length with a collision on every route (two seeds; Figure 4d). The mechanism is shared: the unchanged controller assumes a plan starts at the vehicle's current position, so a stale plan's early waypoints fall behind it and the interpolated trajectory folds back on itself. Fine-tuning does not move the clock either: it changed TransFuser's forward pass by +0.18% (18.46 to 18.49 ms). The adapted ReCogDrive, timed back-to-back with the zero-shot model in one process, is 13.3% slower, which makes the gate harder to clear. Its adapters are mounted rather than merged, adding computation at every adapted module; without a merged-adapter timing this cause is a candidate.

Takeaway. A missed deadline changes the failure mode as well as the score: one modern model keeps driving and collides, another never moves, so there is no single degradation curve beyond the gate and latency belongs in a gate rather than a ranking feature. The repair lies in the consumer, fresh-state tracking of low-frequency plans; neither a wrapper nor fine-tuning moves a candidate across the gate. Measure latency on the deployment GPU and treat a violation as an exclusion, not a discount.

D.3 The distribution gap: how large, how structured, and what repair costs

This section sizes the gap that Stages 3–5 of Figure 1d exist to measure and shrink (RQ5), then prices its repair (RQ6).

RQ5: How large and how structured is the gap that training data and environment create? *Setup.* Open loop, real frames: we contrast public checkpoints of the *same* TransFuser architecture, one trained in CARLA and one trained on real data, under an identical probe code, cities, budget, and output horizon. Appendix G documents the pool-wide search for such pairs; its one vision-language-action candidate differs architecturally across its two releases at the tensor level and is excluded. To test whether the effect is architecture-specific, we repeat the contrast on the two ecosystem releases of UniAD, which pass the same comparison. Closed loop, simulation: a six-environment matrix (five configurations × five routes × three seeds per environment) including a masked intervention, `front-zero`, a six-camera model consuming only its front view.

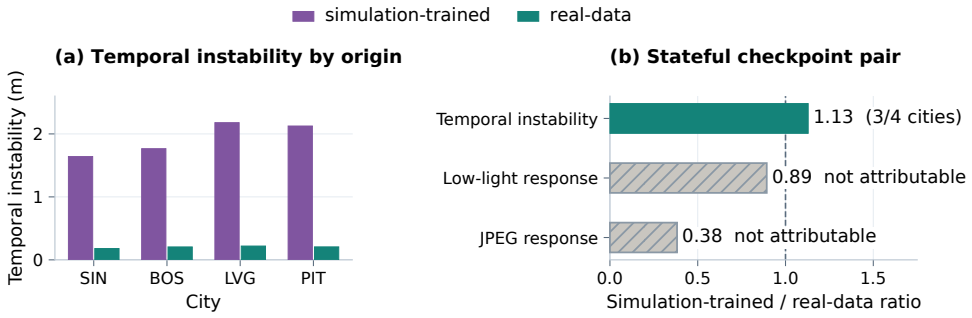

Figure 18: RQ5: temporal instability and cross-architecture attribution. (a) Stateless pair (TransFuser): temporal instability on real frames by training source. **(b)** Stateful pair (UniAD): only the temporal measure transfers; hatched measures fall below the run-to-run noise.

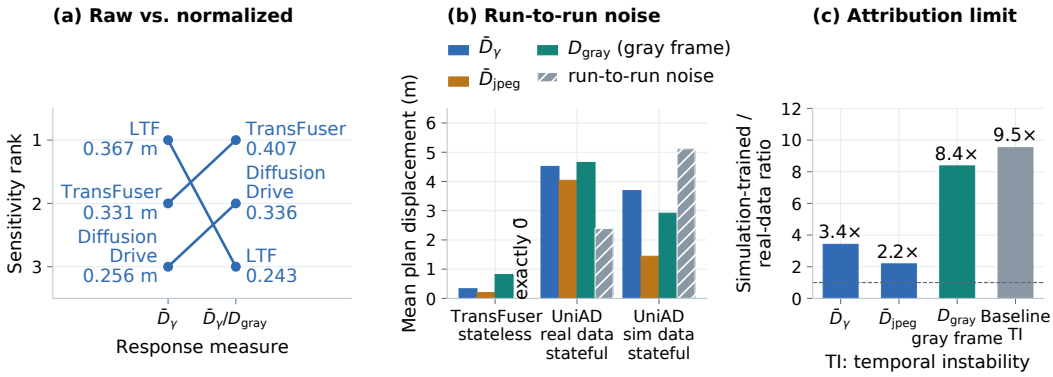

Figure 19: Raw probe sensitivity is not comparable across models. **(a)** Ranks by raw and by normalized response. **(b)** Run-to-run noise: zero on the stateless architecture, above every measure on the stateful simulation-trained checkpoint. **(c)** On RQ5's pair the gray-frame gap is matched by the instability gap.

Results. **Training data separates identical architectures.** The simulation-trained TransFuser reacts 1.5–4.9× more strongly to darkened and compressed real frames in all four cities (low-light response in Figure 5a), and its plans disagree with their own previous frame 8.6–10.3× more, again in four of four cities (Figure 18a). This architecture computes each frame independently: six consecutive calls on identical input return exactly zero displacement (checked on 120 Singapore frames; Figure 19b), so run-to-run drift contaminates neither measure. The response to a constant gray frame (RQ7) is inconclusive on this pair and leaves the gap standing: the simulation-trained side responds 8.4× more to losing the image, but its baseline instability is 9.5× larger, and dividing one by the other leaves the two sides within 12% (Figure 19c).

Table 11: Six closed-loop environments (RQ5). Route completion (mean ± seed s.d., three seeds) and collision route rate (CRR), without perturbations or deadlines. Ranks are plotted in Figure 5b.

	T03 clear		T03 fog		T12 clear	
Configuration	RC	CRR	RC	CRR	RC	CRR
SimLingo	84.2±5.2	0.07	98.0±3.4	0.13	95.5±0.4	0.33
TCP	57.6±0.5	0.60	58.4±0.0	0.60	80.6±0.0	0.60
VAD	66.9±9.1	0.53	63.0±1.5	0.47	49.3±0.3	0.33
UniAD	33.6±5.1	0.33	40.9±7.1	0.47	45.7±10.1	0.53
UniAD front-only	47.3±9.4	0.33	53.3±15.5	0.27	43.7±3.0	0.53

	T12 fog		T13 clear		T13 fog	
Configuration	RC	CRR	RC	CRR	RC	CRR
SimLingo	88.0±6.9	0.27	100.0±0.0	0.00	100.0±0.0	0.07
TCP	80.6±0.0	0.60	86.7±0.3	0.07	88.2±1.1	0.40
VAD	40.7±6.4	0.20	79.7±8.2	0.27	79.4±8.3	0.33
UniAD	54.4±4.9	0.67	90.6±16.2	0.40	84.8±14.2	0.47
UniAD front-only	54.5±4.3	0.80	76.7±4.9	0.40	71.9±0.0	0.40

Environment reorders a closed-loop pool beyond noise. Across the six environments (Figure 5b, Table 11), 13 of 15 environment pairs change some pairwise order among the native configurations (14 of 15 with the masked line), and each configuration's spread across environments (15.8–57.0 completion points) dwarfs its seed noise (0.2–8.6 points). The masked intervention beats its own parent in 3 of 6 environments, by 12–14 points in Town03, a gain that clears seed noise in clear weather and not in fog, and it never overtakes a native front-view model.

A second architecture narrows the claim. The simulation-trained UniAD is again less steady on real frames, in the predicted direction in three of four cities, but by 1.13× rather than 8.6–10.3×. Its two perturbation measures are not readable: UniAD holds state across calls, and six consecutive calls on identical input (120 frames per city) move its plan by 5.12 m on the simulation-trained side and 2.39 m on the real-data side, a floor that on the simulation-trained side exceeds its own gamma and compression readings (3.69 and 1.44 m; Figure 18b). Those ratios are reported as not attributable to the perturbations, not as a reversal. The temporal-instability measure comes from the baseline pass,

is unaffected, and is the one measure that transfers. One further observation is descriptive only: in the three cities with enough turning frames, the simulation-trained UniAD agrees with the human driver's turn direction less often (0.62–0.72 against 0.85–0.92).

Takeaway. The distribution gap is large and structured: training data alone separates identical architectures by up to an order of magnitude on real frames, and environment alone reorders a closed-loop pool beyond noise, so rankings do not transfer across environments. On a second architecture the training-data effect is real but only one of three measures transfers, the other two being unreadable on a stateful architecture rather than negative. A masked intervention can beat its parent but does not overtake native sensor contracts. Prefer the checkpoint trained closest to the domain, measure its gap on local frames above the candidate's own run-to-run noise, and re-check a masked or re-wired model as a new candidate.

RQ6: What does fine-tuning repair, and at what cost? *Setup.* Open loop, real frames. We behavior-clone the simulation-trained TransFuser on real target-city frames, log-level disjoint from all evaluation windows, at nested doses of 1k, 5k and 20k frames. We read probe sensitivities, temporal instability, and tracking error against the real-data checkpoint, which gives the level a dose curve should approach. We replicate the top dose with an independent seed and run an LP-FT variant (Kumar et al., 2022), the standard remedy for fine-tuning's robustness cost in classification.

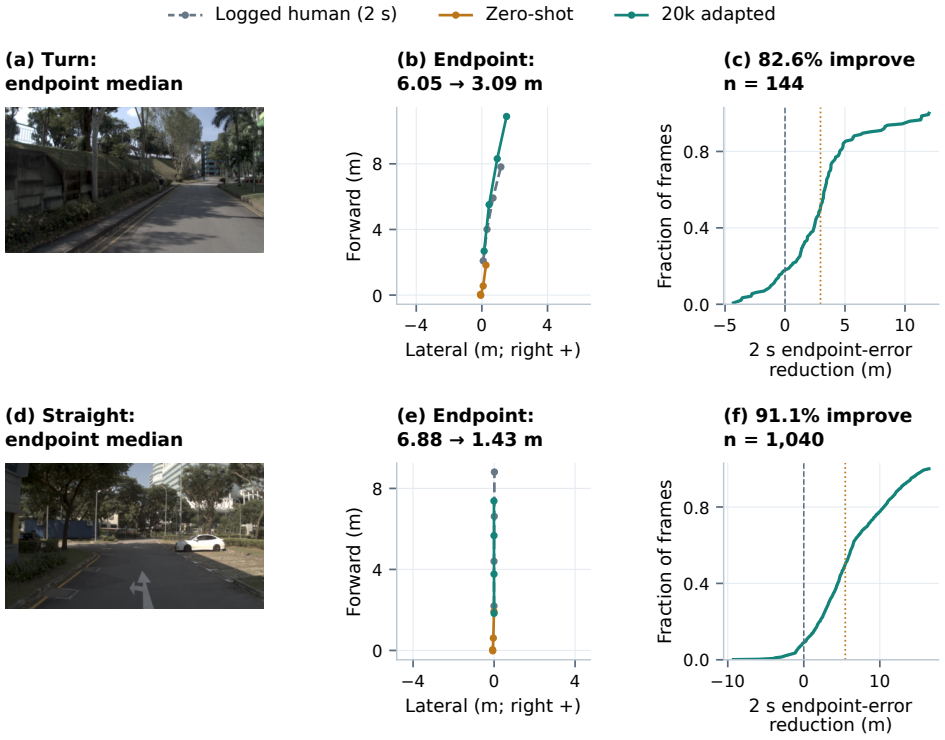

Figure 20: Adaptation measured by 2 s endpoint error. Simulation-trained TransFuser, Singapore. **(a,b,d,e)** Median turning and straight frames by the zero-shot-to-20k error reduction. **(c,f)** The full distributions; 82.6% and 91.1% of frames improve.

Results. **On TransFuser the repair costs robustness at small doses.** Stability behaves as a deployer would hope (Figure 5c): the sim-versus-real instability gap falls monotonically to 25% of its zero-shot size (75% closure) by 20k frames, and an independent seed, run at 20k only, lands within 4% of it and leaves every conclusion unchanged. The two sensitivities do the opposite. After 1k frames the model reacts 2.4× more strongly to darkened and 3.4× more strongly to compressed inputs than before fine-tuning, in every city (1.4–3.3× and 3.0–3.6×), leaving a gap to the real-data checkpoint 3.0 to 4.7× its zero-shot size; at 20k both are still above their zero-shot values, so the closure criterion is not met (1/3 components). LP-FT enlarges the small-dose spike: at 1k its responses are 1.96× (darkening) and 1.60× (compression) those of plain fine-tuning, higher in all four cities. Tracking error against the human driver nonetheless falls monotonically from 8.0 to 1.4 m (Figure 5d), and frame by frame the gain is broad: 82.6% of turning and 91.1% of straight frames

improve (Figure 20c, f; median frames in a, b, d and e), and mean trajectory error improves on 86.8% of turning and 93.5% of straight frames, though monotonically across doses on only 28.5% and 27.4%, so the repair works even while robustness dips.

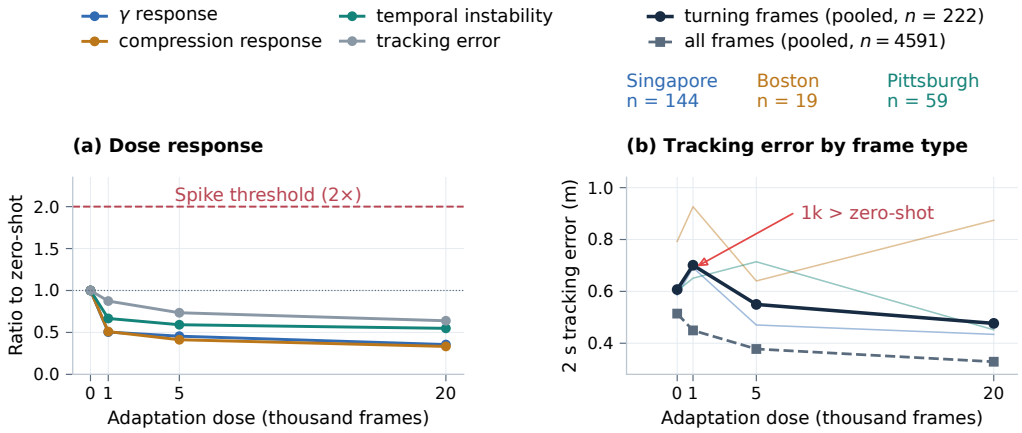

Figure 21: The dose series on ReCogDrive (LoRA(LM)+head). (a) Every measure falls with dose, far below the spike threshold. **(b)** Tracking error on turning frames (pooled) and over all frames; thin lines: cities. On turns the smallest dose is worse than zero-shot.

On ReCogDrive no spike appears, but a small dose moves turns backwards. Under the LoRA(LM)+head recipe at the same budgets (Figure 21a), the response to darkening falls to 0.50, 0.46 and 0.36 of zero-shot, the response to compression to 0.51, 0.41 and 0.33, and temporal instability to 0.67, 0.59 and 0.55; the largest darkening or compression ratio, 0.51, is a quarter of the spike threshold of 2.0, which applies to those two responses. Tracking error improves to 0.87, 0.73 and 0.64 of zero-shot over all frames, but on turning frames the smallest dose is worse than no fine-tuning (1.15) before improving to 0.91 and 0.79 (Figure 21b). Boston, 19 of the 222 pooled turning frames, is the only city whose largest dose regresses; weighting cities equally would halve the apparent improvement.

Part of that fall is the model attending less to the image. Falling sensitivity fits a more robust model and equally one that leans more on ego state and its own priors. The response to a constant gray frame (RQ7) separates the two, against a null control that repeats the smallest dose with only the training seed changed: reseeding alone moves the response to losing the image to 0.868 of the original, and combining that seed half-width with the pool's city dispersion gives a band of chance variation of $[0.709, 1.160]$ (Appendix E). Every variant tested falls below it: 0.705, 0.526 and 0.473 at the three doses, 0.438 and 0.346 on the two cropped runs. The last four clear the lower edge by 0.18 to 0.36 against a half-width of 0.16; the smallest dose clears it by only 0.004 and is itself part of the calibration, so we do not rely on it. The direction is settled and the magnitude is not: dividing by instability, itself down to 0.55 at 20k, would leave 0.86, but that divisor itself changes with fine-tuning (a policy that follows the image less also varies less), so the image-specific fall at 20k lies between 14% and 53%. The fine-tuning recipe never reached the vision encoder (Appendix D.1), so none of this speaks to encoder-adapting repairs of this model.

Takeaway. Fine-tuning repairs the class the framework assigns it, but at a cost. On TransFuser it raises perturbation sensitivity at small doses, and LP-FT makes this worse; on ReCogDrive every response falls, partly because the model attends less to the image, which a sensitivity measure alone would score as an improvement. The spike is thus a property of the fine-tuning recipe and the base model, not of post-training, and Algorithm 1's monitored fine-tuning step must watch reliance on the image beside sensitivity. Monitor both at every dose against a run that changes only the training seed, budget past the small doses, and check separately the frames the deployment depends on.

D.4 Selection with the full framework

We now run the composed workflow end to end (RQ7) and test whether it admits new candidates mid-study (RQ8).

RQ7: Does TRIAGE select better than standard practice, and where does cheap evidence stop? *Setup.* We compare score-first selection, champion-first selection, and Algorithm 1 across four deployment profiles (monocular or full camera set; real-time, a 150 ms deadline, or low-speed, a 1.2 s replanning deadline under plan tracking). Fixed predictor families then select within a tier from probe features under leave-one-city-out, to locate where cheap evidence stops resolving.

Results. **The gates do the selection work.** Score-first selection violates the latency gate in both real-time profiles, and champion-first selection violates the sensing gate in all four, because neither profile's cameras provide the champion's single 110° view (Figure 6). TRIAGE is feasible in three profiles and agrees with the scoreboard wherever every gate passes. In the monocular real-time profile it has nothing to return: every sighted candidate inside the 150 ms budget needs three cameras and every monocular one exceeds it, so it returns each candidate's failing gate instead (Algorithm 1, line 4). This empty pool admits no public checkpoint until the deployer adopts plan tracking or changes the sensors; a higher score does not change that.

Cheap probes place tiers but cannot rank within one. In the leave-one-city-out test (Figure 6b), nine configurations in four cities give 36 cases under leave-one-city-out, with ten fixed features (three environment offsets, seven probe measures) and four predictor families chosen within each fold. The probes never pick below the top tier, but within it they lose to picking the top-scoring candidate: 0.48 against 0.18 PDMS points of mean regret (1.35 against 0.26 on the eight-configuration candidate list before DriveVA joined). Adding latency as a feature makes selection worse ($0.48 \rightarrow 1.29$), since a constant across cities cannot help decide between them. This test uses none of the per-candidate information we record (sensor contract, latency, license), so feasibility is decided by the gates alone, and AutoVLA's two photometric measures are front-view perturbations of a six-camera model, unlike its neighbors'.

Raw probe readings compare within a model, not across models. One photometric perturbation level is large for a model that barely uses the image and small for a model that leans on it (Figure 19a). Normalizing each response by the model's response to a constant gray frame reorders the three stateless traditional stacks from LTF, TransFuser, DiffusionDrive to TransFuser, DiffusionDrive, LTF: no model keeps its rank, though the permutation is a rotation, not a reversal. LTF looks the most sensitive and is the least, relative to how much it uses the image. On the stateful architecture of RQ5, whose run-to-run noise lies above its measures (Figure 19b), the normalizer is not established, since the response to a gray frame there is confounded with how well the model falls back on its own history.

Why not transferability estimators. Estimators such as LEEP (Nguyen et al., 2020) and LogME (You et al., 2021) score how well a source model's predictions or features fit labeled target data. In TRIAGE they would be candidates for the Stage 4 proxy $\widehat{\Delta}_{\mathrm{dom}}$, not replacements for the gates: they assume the candidate can already consume the target inputs and answer in time, which is what the sensing and latency gates test, and here the gates alone decide feasibility in every profile (Figure 6a). They also need a feature or label space shared across candidates, which per-waypoint regressors, token-decoding vision-language-action models and video world models do not expose in the same form, and labeled target data, of which a deployer has little at the start. Within the top tier, where such a proxy would act, ten probe features and four predictor families already lose to the top score (Figure 6b); whether a transferability estimator does better there is left to future work.

Takeaway. The gates do measurable selection work, and latency's value lies entirely in the gate. Cheap probes place tiers but cannot separate models of the same generation, so Algorithm 1 scopes them to tier placement with picking the top-scoring candidate as the within-tier default. Normalize probe readings by each candidate's response to complete input removal before comparing candidates, and check that they exceed the model's run-to-run noise.

RQ8: Does the framework absorb new state-of-the-art candidates mid-study? *Setup.* Four NAVSIM candidates joined mid-study. Two members of NVIDIA's Alpamayo family had their sensor contracts checked, and the one admitted was probed with a camera-count ablation. The ablation tests the model card's note that with a front camera only the model is "expected to struggle to see cross-traffic in a right turn". Two world models, WAM-Flow and DriveVLA-W0, went through the full pipeline: version-fixed weights, score reproduction on all 12,146 test tokens, four-city probes, both perturbation axes, and latency with no other process on the GPU.

Table 12: Camera-count ablation on Alpamayo-1.5 (RQ8). Front-only against three cameras on identical frames, four cities pooled; turn frames: heading change above 2° over 2 s.

Measure	front-only	three-camera
Sign-diagnostic agreement, turn frames	0.915	0.954
Temporal instability, turning frames (m)	1.431	1.544
Temporal instability, straight frames (m)	0.921	0.934
Low-light response, turning frames (γ=2.6)	0.792	1.079
Latency p50 (ms, GPU not shared)	959	
Probe readings (4 cities), frame failures	8/8 sets, 0 failures	

Results. **Sensor and memory requirements decide the Alpamayo family before any GPU work.** Alpamayo 2 Super is excluded on paper, consistent with its model card (NVIDIA, 2026b): its reported 72,115 MiB footprint (70 GiB), documented for an 80 GB H100, does not fit the 48 GB (49,140 MiB) of the RTX A6000 deployment GPU, and its six-camera sensor contract fails on a monocular camera setup. Alpamayo-1.5 (10B, 24 GB, flexible cameras) was admitted: eight city×configuration sets of probe readings, zero frame failures, and a p50 latency of 959 ms, the VLA tier. Its camera-count ablation (Table 12) refines the model card's caution. The front-only variant is not less self-consistent; it is slightly smoother on both turning and straight frames (1.43 against 1.54 m on turns, 0.92 against 0.93 m on straight roads). Its agreement with the human driver's turn direction is lower, 0.915 against 0.954 (the sign diagnostic is defined on turning frames only; 771 of 843 and 809 of 848 frames). The caution is thus consistent with a cost in turn correctness, while plan stability is not degraded. The two channels are worth separating, and the probe readings do so.

The world models enter through the full pipeline. Both reproduced their published scores before entering the pool (WAM-Flow 90.32 against 90.3; DriveVLA-W0 90.30 against 90.2), both landed in the VLA latency tier (781 and 485 ms), and both replicated RQ1's asymmetry. The DriveVLA-W0 check also matched the published score to its checkpoint: the paper's 90.2 corresponds to the query-based checkpoint with anchors, the one admitted here, while another released checkpoint, labeled with a PDMS of 87.2, is a flow-matching variant. Appendix G records a sharper case of the same question: two releases of one model name that differ by an occupancy branch, an ego-history input pathway, twelve extra decoder layers and a different sampling geometry in the ego branch, differences we found only by comparing the checkpoints' parameter names and shapes.

Takeaway. A new candidate costs a sensor-contract check plus at most a GPU-day, and infeasible ones are excluded without any GPU. Probes on recorded logs add detail that complements a model card, here separating turn correctness from plan stability. Checking sensor and memory requirements before downloading weights is the cheapest step in the procedure.

D.5 Transfer to manipulation

We transplant the sensor-contract and latency checks and the probes unchanged to a second embodiment, to test whether the driving findings of Section 4 are properties of the framework or of driving.

RQ9: Do the instruments and the sensory asymmetry transfer to a second embodiment? *Setup.* The same checks and probes run unmodified (same scripts, the same check that no other process uses the GPU, and fixed perturbations) on six manipulation models (Octo-small/base, GR00T N1.5, SmolVLA, OpenVLA-7B, π_0; sensor contracts in Table 13), under offline inference on recorded real robot frames (Figure 22). We read action displacement in place of plan displacement, as within-model ratios so heterogeneous action spaces compare validly; a constant-action policy is the zero-response anchor. We test whether two driving findings carry over: a latency spread of at least two orders of magnitude, and the field-of-view asymmetry in most sighted models.

Results. **The sensory asymmetry transfers; the latency span does not.** The six latency measurements (Figure 6c) span 27.5 ms (Octo-small) to 385 ms (π_0), a 14× spread from a ∼36 Hz replanner to a ∼2.6 Hz one. That falls short of two orders of magnitude, so we narrow the claim to what was measured; driving's five-decade spread came from blind anchors and multi-second reasoning VLAs absent here. The asymmetry holds in all six models: cropping the camera view moves the actions 1.6–2.4× more than downscaling it at the same pixel budget (Figure 2b), milder than driving's 2.2–5.7× but unanimous. The constant-action anchor responds exactly zero at every perturbation level,

Table 13: Sensor contracts of the manipulation pool (RQ9). Octo is release 1.5, whose closed-loop success in SIMPLER (Li et al., 2025) is low; no closed-loop claim is made for it. Sources: OpenVLA (Kim et al., 2024), π_0 (Black et al., 2024), SmolVLA (Shukor et al., 2025), GR00T N1.5 (NVIDIA et al., 2025), Octo (Octo Model Team et al., 2024).

Model	Venue	Cameras	Wrist cam	Action space
OpenVLA-7B	CoRL'24	1 (third-person)	no	7-DoF EEF delta, 256-bin discretized tokens, per-robot de-normalization
π_0	arXiv'24	3	yes	continuous actions padded to 32-d, flow-matching chunk; base checkpoint ships no de-normalization statistics
SmolVLA	arXiv'25	3 (generic slots)	platform-dependent	6-d SO-10x joint-space targets, checkpoint-stat de-normalized, flow-matching chunk
GR00T N1.5	arXiv'25	embodiment-dependent (oxe_droid: 2 ext. + 1 wrist)	head-dependent	embodiment-indexed continuous chunk ($\leq$32-d), flow matching
Octo-small/base	RSS'24	1–2 (primary + optional wrist)	optional slot	7-d [dxyz, drpy, gripper] continuous, diffusion head, per-dataset unnormalization (incl. bridge)

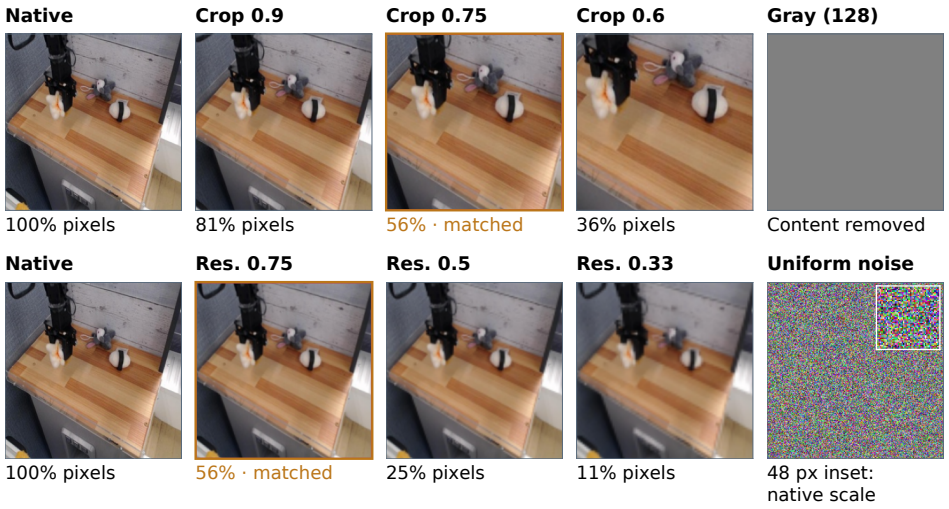

Figure 22: The same interventions on a manipulation frame (Bridge V2, median-index frame of the 120-frame probe subset). The ladders meet at `keep` $= 0.75$ (outlined), where Figure 2b is read.

every model's run-to-run displacement is exactly zero, these being stateless policies, and all paired comparisons reproduce bit-identically.

The probe's cross-model limitation transfers too. Responses to losing the image span $31.8\times$, from 0.056 to 1.777 (Figure 23a), so a fixed perturbation level is a very different fraction of each model's signal. Ranking the six by raw field-of-view response and by the same response normalized to each model's own ablation gives near-opposite orders (Figure 23b): π_0 drops from first to fourth and Octo-small rises from sixth to first. This limitation is a property of the probe design, not of driving, and so is its repair.

The probe resolves several-fold differences, not small ones. Frame by frame (stimulus in Figure 24a), every pair of models whose pooled crop responses differ by at least $4.4\times$ agrees with the pooled order on 94–100% of the 120 frames (the widest pair in Figure 24b), and every pair within $1.3\times$ on only 49–66%, close to a coin flip (Figure 24c); the full six-model raw ordering appears on 14 frames and the normalized one on none. This is the manipulation counterpart of RQ7, where probes place tiers but cannot separate models of the same generation.

Takeaway. The instruments and the central sensory asymmetry are not driving artifacts: the field-of-view asymmetry holds in all six models, and the latency spread falls short and is narrowed to

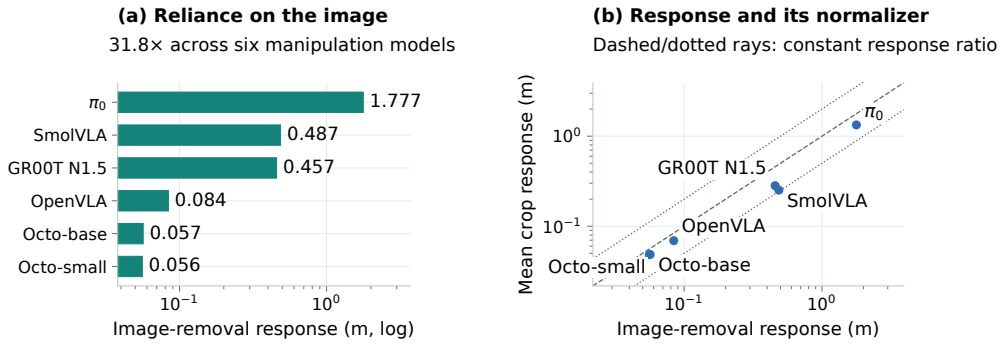

Figure 23: Reliance on the image and normalized responses (RQ9). **(a)** Response to image removal, which spans $31.8\times$ across the six models; the constant-action policy reads zero. **(b)** Crop response against image-removal response: crop responses, averaged over the three crop levels, span $27.6\times$ across the six models, and normalized by the image-removal response they span $1.67\times$.

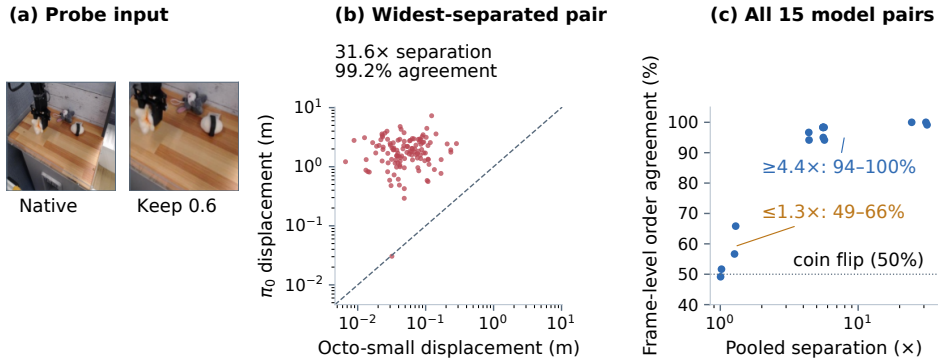

Figure 24: What the probe resolves frame by frame (RQ9). **(a)** The stimulus. **(b)** The widest pair, whose mean crop responses at the strongest crop (keeping 60%) differ $31.6\times$, agrees with that order on 99.2% of frames. **(c)** All 15 pairs: pooled separation against frame-level agreement.

what was measured. Transplant the instruments unchanged and validate the zero-response anchors before trusting any reading. Normalize before comparing candidates: normalization rearranged the ordering in both embodiments and nearly reversed it in manipulation.

E Measurement definitions and checks

This appendix defines every quantity reported for the driving and manipulation experiments.

Frames and plans. Open-loop measures use nuPlan logs at $2\,\mathrm{Hz}$. Each city contributes a ten-minute window of $1{,}200$ frames, drawn with a fixed seed, and each model receives a four-frame history. A plan ξ_t is eight waypoints at $0.5\,\mathrm{s}$ spacing (0.5 to $4\,\mathrm{s}$) in the ego frame (x forward, y left); the simulation-trained TransFuser outputs four (to $2\,\mathrm{s}$), so its real-data counterpart is truncated to four whenever the two are compared. Baseline plans, temporal instability, the sign diagnostic and tracking error use every frame of the window; perturbations and the gray and noise inputs use every tenth frame, about 120 per city. The *probe readings* are the measures below, computed on one city's window by offline inference on the recorded logs. A four-city value is the mean of the four city means unless stated otherwise. Frames a model cannot process are kept with their reason and never imputed: of ReCogDrive's $4{,}800$ window frames, 146 are missing because the logs carry a command class its deployment interface does not map (134 of them in one city) and 30 because fewer than four frames of history precede them at a log's start.

Benchmark scores. PDMS (NAVSIM v1.1) is computed per token as $\mathrm{NC} \times \mathrm{DAC} \times (5\,\mathrm{EP} + 5\,\mathrm{TTC} + 2\,\mathrm{C})/12$ over a $4\,\mathrm{s}$ horizon, where no-collision (NC; 0.5 for a static object) and drivable-area compliance (DAC) act as multiplicative gates on ego progress (EP), time to collision (TTC) and comfort (C). A configuration's PDMS is the mean over tokens that returned a plan; tokens with

no plan are excluded, not scored zero. *Regret* (RQ7) is computed leave one city out: in each of the four folds, the PDMS of the best candidate in the held-out city minus the PDMS of the candidate the selector picks, averaged over folds. The candidate set is the whole candidate list, blind anchors included, and picking the top-scoring candidate means taking the configuration with the highest global PDMS.

Plan displacement and probe responses. For a perturbation P_ρ at strength ρ applied to image I_t, the displacement is the waypoint-averaged distance between the plan on the perturbed frame and the plan on the same frame unperturbed,

$$D(\rho) = \operatorname*{mean}_t \frac{1}{H} \sum_{j=1}^{H} \left\| \xi_{t,j}\big(P_\rho(I_t)\big) - \xi_{t,j}(I_t) \right\|_2. \tag{14}$$

Every camera the model consumes is perturbed, except that AutoVLA's probes perturb its front view only. The field-of-view crop keeps 90, 75 and 60% of each image dimension around the centre and resizes back; the resolution drop scales each dimension to 75, 50 and 33% and back; darkening applies gamma 1.8 and 2.6 with Gaussian noise of standard deviation 4 grey levels; compression uses JPEG quality 50 and 20. The matched-budget ratio is $\kappa = D_{\text{fov}}(0.75)/D_{\text{res}}(0.75)$, the two perturbation levels that keep 56% of the pixels, taken as a ratio of four-city means (Figure 2a); no city ratio falls below 1.9. For darkening and compression the reported *response* is the area under the displacement-versus-severity curve, with severity scaled to $[0, 1]$ from the unperturbed reference (gamma 1, JPEG quality 90): $A_\gamma = \frac{1}{2} D(1.8) + \frac{1}{4} D(2.6)$ and $A_{\text{jpeg}} = \frac{1}{2} D(50) + \frac{3}{14} D(20)$. The plain average of the responses at the two severities, $\bar{D}$, is used for normalization (below) and for comparison with the run-to-run noise. In manipulation, ξ is the first step of the action chunk over the pose dimensions, gripper excluded, in each model's native units, averaged over 120 evenly spaced Bridge frames; the constant-action policy returns a fixed action and so reads zero.

Temporal instability. For consecutive frames of one log, 0.5 s apart, the earlier plan is moved into the later ego frame with the logged ego motion $E_{t\to t+1}$, and waypoint $j{+}1$ of the earlier plan is compared with waypoint j of the later one, since both refer to the same future instant:

$$\text{TI} = \operatorname*{mean}_t \frac{1}{H-1} \sum_{j=1}^{H-1} \left\| E_{t\to t+1}\, \xi_{t,j+1} - \xi_{t+1,j} \right\|_2. \tag{15}$$

RQ5's 8.6–10.3× is the per-city ratio of TI between the simulation-trained and real-data TransFuser, and 1.5–4.9× is the range of the per-city ratios of the darkening and compression responses.

Tracking error and turning frames. Tracking error is the distance, in the ego frame, between the plan's waypoint at 2 s and the logged ego position 2 s later, over every window frame with a logged future. A turning frame is one whose logged heading changes by at least 15° over the next 2 s. RQ2's values pool frames across cities; RQ6's TransFuser curve (8.0 to 1.4 m) is the mean of city means, while ReCogDrive's tracking ratios pool frames. Mean trajectory error in RQ6 averages over the waypoints to 2 s instead of taking the 2 s endpoint. Turning frames are unevenly spread: 144, 62, 46 and 0 in Singapore, Pittsburgh, Boston and Las Vegas for TCP's windows, and 144, 59, 19 and 0 for ReCogDrive's, so no per-city turn reading is made for Las Vegas.

Repair and closure. RQ2 uses four settings, zero-shot (zs) or fine-tuned (ft) and native (nat) or cropped (crop), each scored by turn-frame tracking error e: damage $= e_{\text{zs,crop}} - e_{\text{zs,nat}}$, repair $= e_{\text{zs,nat}} - e_{\text{ft,nat}}$, residual $= e_{\text{ft,crop}} - e_{\text{ft,nat}}$, and compensated fraction $= 1 - \text{residual}/\text{damage}$; the share of the zero-shot error of a setting s that fine-tuning removes (the 73–76% of RQ2) is $\eta_s = (e_{\text{zs}} - e_{\text{ft}})/e_{\text{zs}}$. The criterion is a positive residual of at least 0.25× the repair. For RQ6, with zero-shot value $m(0)$, dose-N value $m(N)$ and real-data reference m_{ref}, all four-city means,

$$\text{cl}(N) = \frac{m(0) - m(N)}{m(0) - m_{\text{ref}}}. \tag{16}$$

The criterion requires at least two of the three measures (darkening response, compression response, TI) to fall monotonically from a positive gap and reach $\text{cl}(20\text{k}) \geq 0.5$; only TI did (0.753). A *spike* is a response at a dose divided by its zero-shot value; on ReCogDrive the spike threshold is a darkening or compression ratio of at least 2.0 at 1k or 5k frames.

Response to a gray frame and run-to-run noise. To measure how much a plan relies on the image, we replace every camera view with constant gray, or with seeded uniform noise (rightmost column of Figure 22), and measure the displacement D_{gray} with the same metric and frames; LiDAR is left intact. Dividing a response by it makes responses comparable across models: RQ7's normalized sensitivity is $\bar{D}_\gamma / D_{\text{gray}}$ per city, averaged over cities, and RQ6's *reliance on the image* at a dose is D_{gray} at that dose over D_{gray} at zero shot, as four-city means. *Run-to-run noise* is the mean displacement of calls two to six from the first, over six consecutive calls on the same unperturbed input (four in manipulation); a model that keeps internal state across calls shows non-zero run-to-run noise, and a perturbation reading below it cannot be attributed to the perturbation (RQ5). The gray-frame response behaves monotonically on models that plan each frame independently, where removing the whole image moves the plan further than any perturbation level; on a model that keeps state it does not, because such a model falls back on its own history when the image carries nothing.

Band of chance variation for reliance on the image. The null control repeats the smallest dose with only the training seed changed, giving per-city ratios of 0.795, 0.892, 0.941 and 0.846 (mean $m = 0.868$). The seed half-width is twice their standard error, 0.063; the city half-width is twice the zero-shot coefficient of variation of D_{gray} across cities divided by $\sqrt{4}$, 0.147; the two combine in quadrature to $h = 0.160$. The band is $[\min(1, m) - h,\ \max(1, m) + h] = [0.709, 1.160]$, anchored at 1 on the upper side. It is applied to four-city means; per-city values are descriptive only.

Sign diagnostic and blind anchors. The sign diagnostic takes the window frames whose logged heading changes by more than 2° over the next 2 s and whose planned final lateral offset exceeds 0.05 m, and reports the share on which the plan turns the same way as the logged heading. It is recorded with every set of probe readings and enforced as a gate at 0.8 on the ground-truth labels before every fine-tuning run and on Alpamayo's readings; the turn-frame rows of Table 12 use this 2° definition. Blind anchors (constant velocity and the ego-status MLP) must respond exactly zero to every image perturbation, which verifies that a response measures reliance on the image rather than pipeline noise.

What is recorded per candidate, and latency. For each candidate we record its training ecosystem, sensor contract (the cameras its inference code consumes and the preprocessing it applies), license, weight hashes and per-frame latency. Latency is measured on the deployment GPU under a guard that refuses to measure if any other compute process is present: five warm-up calls, then 200 consecutive frames, each timed with CUDA synchronization from agent input to returned plan (for models that read images from disk, decoding falls inside); p50 and p99 are percentiles of these samples. Image acquisition, transport, control and actuation lie outside the timed boundary. Two latency measurements differ: TCP's uses 279 samples after 20 warm-ups and includes its JPEG preprocessing, and ORION's 300 samples after 20 warm-ups and includes its controller. For manipulation models the latency measurement times the whole action chunk as one call.

Closed-loop metrics. In Bench2Drive, route completion is the share of the route driven; the collision route rate is the fraction of route-and-seed runs with at least one collision with a vehicle, pedestrian or static object; the driving score is route completion times Bench2Drive's infraction penalty, averaged over routes. A route is *stuck* when the vehicle stays below 0.1 m/s for 60 s, and ends in a *scenario timeout* when it runs past $4{,}000$ simulation steps (200 s). RQ3's seed-noise bound is twice the larger of the native and cropped runs' seed standard deviation of five-route mean completion (8.4 points for TCP, 18.0 for SimLingo). In RQ5 a configuration's spread is the range of its seed-mean completion across the six environments, its seed noise the median over environments of the per-environment seed standard deviation, and an environment pair changes the order if the completion ranking of the four native configurations differs. An occlusion of severity s blacks out a rectangle of $s\%$ of each image dimension, so s^2 of the area, at a new random position for every camera and frame.

Deadlines and stale inputs (RQ4). Under a deadline, each planning call is charged the next latency sample from the model's latency record, in order; if it exceeds the deadline, the new control is discarded and the last on-time control is applied again (hold), and before any control has arrived the vehicle brakes. On the frame-drop axis each camera independently starts a burst with probability 0.01 per step, lasting a uniformly drawn 2 to 20 or 2 to 60 steps (up to 1 or 3 s at 20 Hz); during a

burst the camera returns its last frame, while the model keeps planning every step with fresh speed and position and the controller tracks with fresh vehicle state. The asynchronous wrapper delivers each plan $\lceil \tau / 50\,\mathrm{ms} \rceil$ steps after it was computed and always uses the newest plan.

F Reproducibility protocols

Fixed versions and admission. Closed loop runs on CARLA 0.9.15 with Bench2Drive at a fixed commit, with every metric definition fixed before the first run. Open loop runs the NAVSIM v1.1 evaluator at a fixed commit on the `navtest` split (136 logs, 12,146 tokens), with metric caches rebuilt under that evaluator, fixed probe definitions and a recorded sampling seed. A configuration with a published score was admitted only after that score was reproduced under the version-fixed evaluator (for example EgoStatusMLP 65.5165 against the published 65.6, bit-identical across three runs). ORION's reproduction on Bench2Drive's ten-route Dev10 subset (seed 0) came out four points below the published mean (71.19 against 75.19), beyond the two-point admission threshold; the whole gap is one collision on route 25381, whose reruns with new seeds return 60 and 100, so ORION was admitted by exception.

Data handling. Closed-loop and open-loop measurements are never pooled; published scores never stand in for missing measurements; missing measurements are never imputed; and fine-tuned checkpoints never receive official benchmark scores, because their fine-tuning pool shares the split. Fine-tuning pools are log-level disjoint from all evaluation windows, checked automatically before training, ground-truth waypoints pass a 20-frame sign diagnostic before any gradient step, and every dose starts from zero-shot weights.

G Checkpoint pairs that differ only in training data

Table 14: Public checkpoints of one architecture in both ecosystems are rare. The open driving pool by publisher, 2026-08. "Sim": trained in CARLA/Bench2Drive; "real": trained on nuScenes (Caesar et al., 2020) or nuPlan. Other candidates: MindDrive (Fu et al., 2026), DriveWorld-VLA (Jia et al., 2026), SparseDriveV2 (Sun et al., 2026).

Candidate	Family	Sim	Real	Status for a training-data contrast
UniDriveVLA	VLA	✓	✓	both sides public; architectures differ (below); not used
DriveStack-VLA	VLA	–	–	scores on both; weights not public
ORION, MindDrive	VLA	✓		simulation side only
ReCogDrive	VLA		✓	real side only (sim release announced)
AutoVLA, Curious-VLA	VLA		✓	real side only
DriveVLA-W0, WAM-Flow	world model		✓	real side only
DriveWorld-VLA	VLA		✓	real side only
SparseDriveV2	vision only	✓	✓	both sides public (Bench2Drive weights 2026-05); not a VLA; not checked
LEAD / TFv6	traditional	✓	✓	both sides public; modality symmetry unresolved, and one description indicates real panoramic data in the simulation side's pre-training, so the separation of training data could not be confirmed; not used
TransFuser	traditional	✓	✓	controlled pair used in RQ5
UniAD	traditional	✓	✓	controlled pair used in RQ5

RQ5 measures how the training data changes a model. That needs one architecture with public checkpoints trained in simulation and on real data and otherwise alike. We searched the open driving pool by publisher, on code hosts and weight hubs, rather than by model name (Table 14). Five architectures publish both sides. TransFuser and UniAD are the controlled pairs used in RQ5. LEAD/TFv6 (Nguyen et al., 2026) is not used, because one of its descriptions indicates that the simulation side's pre-training included real panoramic data, so we could not confirm that the two sides differ in training data. SparseDriveV2 also publishes both sides but is not a vision-language-action

model, and whether its two checkpoints share one architecture was not checked. UniDriveVLA is the only vision-language-action model with both sides public. DriveStack-VLA (Wang et al., 2026) reports scores on both benchmarks; its weights are not public, so it could not be included.

UniDriveVLA's two checkpoints differ in architecture. Its two public checkpoints (Li et al., 2026c) share sensor contracts, preprocessing, model class, planning head, planning objective and vision-language co-training corpus, but not an architecture: comparing parameter names and shapes, 260 tensors exist on one side only and 52 differ in shape (2,809 shared), adding up to an occupancy branch, an ego-history input pathway, twelve extra decoder layers and a different sampling geometry in the ego branch. The UniAD pair, by contrast, has no tensor on one side only and differs only in the counts and widths of task-head outputs (35 shape mismatches in 2,457 shared tensors), so it is used as a controlled pair. We report these differences so that UniDriveVLA's two checkpoints are not used as a controlled pair.