CAUSAL CONCEPT GATING FOR END-TO-END DRIVING: FAITHFUL CONCEPT ACTUATION IS NOT EFFECTIVE CONCEPT INTERVENTION, AND WHY

Anonymous authors
Paper under double-blind review

ABSTRACT

What does a named concept channel buy a competitive closed-loop driving planner, how much of it is free, and what does intervening on it do? We answer with a hybrid latent–concept planner and measurements on it. A multi-modal encoder predicts 25 named scene *factors*, a prior-initialized, end-to-end-learned graph $W \in \mathbb{R}^{25\times 8}$ maps them to 8 named action gates, and the gates *bias* each metric head of an anchor scorer that also reads the scene latent directly, so one can trace a factor's effect through a gate to a metric, and the channel is an auditable control path rather than a bottleneck. The planner is state-of-the-art-comparable on `navtest` (0.8985 EPDMS-v2; 0.9030 with an inference-time collision rerank independent of the channel), and a three-seed, same-budget ablation cannot distinguish the graph from a black-box gate MLP, so the findings concern a competitive policy. Three findings follow. *The channel has a free-amplitude boundary*. Its amplitude is an inference-time scalar, and a pre-registered rule raises it fourfold, to a tenth of the selection signal (half on time-to-collision), at no measurable cost, with the first measurable cost at sixfold. *It is a faithful actuator*. do-interventions on the gates move the emitted trajectory as the gate names prescribe at every amplitude (longitudinal Spearman ρ=1.00, lateral CIs excluding zero), factor interventions shift the anchor distribution in the prior's direction across 11 checkpoints (r=0.78) and 21–57$\times$ more strongly on factors active in the scene, and a permutation null with a second, non-additive read-out separates learned from architecturally forced alignment. *Faithful actuation is not effective intervention*, and this dissociation is our main finding. Under the standard test-time-intervention protocol, replacing all 25 factors by ground truth changes the score by less than 0.001, and a matched control shows that the gate edge a factor intervention pushes sets its effect, not whether the assertion is true. We locate the loss at the read-out, which maps both the additive coupling's scene-independent push and a multiplicative (FiLM) coupling's scene-specific perturbation to a single direction and flattens them. What the channel does support is auditing, reading back a missed factor on a real logged at-fault-collision scene, separating a mislabelled concept from a model error, and repairing a blind safety-factor detector at no detectable planning-score cost.

1 INTRODUCTION

End-to-end learned planners now dominate closed-loop driving benchmarks, because they optimize the driving objective directly (Hu et al., 2023; Dauner et al., 2024). Their strength is also their weakness. The mapping from sensors to trajectory is a single high-dimensional function, so when such a policy brakes, yields or changes lane it offers no account of *which* aspects of the scene drove that decision, and no handle for verification or targeted correction. Post-hoc explanations need not be faithful to the computation that produced the trajectory, which is the argument for models interpretable by construction (Rudin, 2019).

A natural remedy is to route the policy through human-interpretable intermediate variables, as in concept bottleneck models (CBMs) (Koh et al., 2020), which predict a set of named concepts and

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.

then predict the output from those concepts alone. A CBM is inspectable and *interventionable*, in that one can ask what the model would do under a counterfactual concept assignment. Two limits temper the appeal. Forcing predictions through a low-dimensional concept layer can reduce accuracy, which sustains the perception that interpretability and performance trade off. And where CBMs have reached driving (Echterhoff et al., 2024; Lai et al., 2024), they predict control offline rather than drive a closed-loop policy scored on its own rollout. Three questions therefore remain open. Can a *competitive* closed-loop driving policy carry a named concept channel *without* paying that cost? Is what the channel says *faithful* under intervention? And is it *actionable*, so that one can correct a wrong decision at the concept that caused it?

We answer all three with a planner that predicts 25 supervised, named binary *factors* (e.g. `front_pedestrian`, `cmd_left`) from its scene latent. A structured *factor-to-action* graph maps them to 8 named driving-action gates (e.g. `brake_hard`, `yield_stop`), and the gates bias a multi-metric, anchor-based trajectory scorer in the style of recent NAVSIM planners (Li et al., 2024) that *also reads the latent directly* (Eq. (2)). The policy is therefore not a concept bottleneck in the strict sense, since the black-box path carries most of the selection signal (Table 3). What the design guarantees instead is a named control channel whose contribution to every decision is explicit, measured and interventionable. Two choices separate it from a vanilla CBM. The concept→output map is a *structured concept→action graph*, seeded with a hand-built driving prior and then learned, which yields a two-level named hierarchy (scene factors → behavioural intents → trajectory). And the channel acts on trajectory *scoring* instead of emitting the trajectory, so the gates steer which candidates each metric favours rather than regressing geometry, which avoids the usual CBM accuracy penalty. Whether the channel is faithful, and how much it does, are then empirical questions (§5.3–§5.4), and their answers come apart. It is faithful under intervention, yet intervening on it does not reach the outcome.

Contributions. The planner is the instrument; the contributions are what it measures.

1. **A hybrid latent–concept planner with a free-amplitude boundary** (§3, §5.1): factors → gates → per-metric anchor-score biases next to a direct latent path, state-of-the-art-comparable, with a same-budget three-seed ablation that cannot separate the graph from a black-box gate MLP. The channel's share of the decision is an inference-time scalar, and a pre-registered sweep raises it fourfold at no measurable cost and no further.
2. **A verification suite that returns "faithful actuator"** (§5.3, App. E): a permutation null that separates architecturally forced from learned alignment, checked on a second, non-additive readout; a continuous-distribution do-estimator with a theorem on when a top-1 readout freezes or sign-flips a small effect; and a selectivity test. Gate interventions move the emitted trajectory as the names prescribe at every amplitude, and factor interventions shift the anchor distribution in the prior's direction across 11 checkpoints.
3. **Faithful actuation is not effective intervention, and why** (§5.4): under the standard test-time-intervention protocol the score effect of correcting all factors is below 0.001, and a matched control shows that what sets the effect is the gate edge a factor pushes rather than the truth of the assertion. A mechanism study with controls locates the loss at the metric read-out, for additive and multiplicative (FiLM) couplings alike, and states what a scene-aware coupling would need. This contradicts the working assumption behind concept intervention, that a faithful, interventionable concept layer is an effective correction handle. The channel still reads back a missed factor on a real logged at-fault-collision scene, separates a mislabelled concept from a model error, and lets us repair a blind safety-factor detector at no detectable planning-score cost. §5.5 adds an independent collision-aware rerank with a controlled negative result.

2 Related Work

Closed-loop end-to-end driving. UniAD (Hu et al., 2023), VAD (Jiang et al., 2023) and PARA-Drive (Weng et al., 2024) keep interpretable intermediate representations but optimize them jointly for planning, whereas sensor-fusion policies such as TransFuser (Chitta et al., 2023) map raw inputs to control end-to-end. nuPlan (Caesar et al., 2021) and the non-reactive NAVSIM (Dauner et al., 2024) score a planner's own rollout under a rule-based ego-simulator, and within NAVSIM the Hydra-MDP family (Li et al., 2024) scores a fixed anchor vocabulary against sub-metrics distilled

from a rule-based teacher. We adopt that backbone and add a named concept channel alongside its latent path.

Concept bottlenecks and structured layers. The concept channel descends from concept bottleneck models (Koh et al., 2020), whose concept layer can cost accuracy and whose later variants relax the dense-annotation requirement (Yuksekgonul et al., 2023) but keep the direct concept→output map. CBMs have reached driving in Concept Gridlock (Echterhoff et al., 2024), which predicts control from a bottleneck of named concepts on logged data, and DRIVE (Lai et al., 2024), which studies the stability of such explanations. What is new here is a named channel acting on closed-loop anchor *scoring* next to a direct latent path, a prior-seeded, learned factor-to-action hierarchy, a measured account of the channel's share and free amplitude, and intervention-based verification. The scoring placement is also why the accuracy cost is small, since the gates bias a scoring problem rather than constrain a regression. The graph connects the work to structured and causal layers (Schölkopf et al., 2021). Unlike NOTEARS-style structure discovery (Zheng et al., 2018) we do not learn a skeleton from scratch and claim no structural causal model, and being bipartite it is acyclic by construction. Our do-interventions (Pearl, 2009) are *model-internal*, replacing an activation inside the network and re-running what follows, which tests whether the model's own computation uses the named variable as claimed. No claim about real-world causation follows (§3.1). The rest is standard, FiLM modulation (Perez et al., 2018), straight-through Gumbel-Softmax (Jang et al., 2017; Maddison et al., 2017) and ConvNeXt (Liu et al., 2022).

Evaluating faithfulness. A plausible-looking explanation metric can be uninformative (Adebayo et al., 2018; Hewitt & Liang, 2019; Hooker et al., 2019). Our permuted-graph null is a structure-randomization check in that family, and the selectivity ratio is a control-condition test (Kim et al., 2018). Following the behavioural definition of faithfulness (Jacovi & Goldberg, 2020), our tests are intervention- rather than attribution-based (Vig et al., 2020; Elazar et al., 2021). Concept bottlenecks can leak so that concepts become unfaithful (Margeloiu et al., 2021; Mahinpei et al., 2021), and our suite tests for this. Test-time concept intervention (Koh et al., 2020; Shin et al., 2023) then becomes a per-concept audit handle whose limits we measure (§5.4; App. A).

3 METHOD

3.1 PROBLEM SETTING AND ARCHITECTURE

At each step the planner observes multi-modal input o (BEV rasters, a vectorized map, ego history, front-camera images, and during training a LiDAR voxel grid) and must output a trajectory $\tau \in \mathbb{R}^{T\times 3}$ of $T{=}8$ poses (x, y, θ) over a $4\,\mathrm{s}$ horizon, which a closed-loop simulator then scores. We learn $\pi_\phi(o) \to \tau$ so that a channel of 8 named gates modulates scoring, making the named-intent contribution to every decision explicit and auditable. Figure 1 shows the pipeline in four stages (tensor shapes in Appendix B). A multi-modal encoder first maps o to a single latent $z \in \mathbb{R}^{256}$, fusing tokenized cameras, the BEV raster, the vectorized map and ego history with a small transformer. LiDAR enters z through a zero-initialized projection *during training* only and is absent at evaluation, so every reported score is camera + map + ego-history (App. C).

From z, twenty-five independent linear heads predict *named scene factors* $f \in \mathbb{R}^{25}$, with activations $\hat{f} = \sigma(f) \in [0,1]^{25}$. These are binary attributes with hand-built geometric labels such as `front_pedestrian`, `intersection_10m` or `left_lane_clear`, plus the ego speed/acceleration/yaw band and the navigation command (taxonomy in Appendix F). Supervising the factors directly is what later makes the gates *nameable*, and a second, camera-only head predicts the same factors from images alone.

The heart of the model is a single linear map from the twenty-five factors to eight *action gates*,

$$a \;=\; \sigma\big(W^\top \hat{f} + b\big), \qquad W \in \mathbb{R}^{25\times 8},\; b \in \mathbb{R}^{8}, \tag{1}$$

whose eight outputs are `accelerate`, `cruise`, `brake_soft`, `brake_hard`, `steer_left`, `steer_right`, `lane_keep` and `yield_stop`, with an element-wise sigmoid rather than a softmax because intents co-occur. W does not start from scratch. We initialize it to a signed prior W_0 writing down uncontroversial driving knowledge one edge at a time (`front_pedestrian` → `yield_stop` at $+3$). Training then refines it end-to-end under one deliberately weak regularizer, the mean-reduced L_1 anchor $\lambda_1\big(\overline{|W{-}W_0|} + \overline{|b{-}b_0|}\big)$ with $\lambda_1{=}10^{-3}$ over the $25{\times}8$ weights and the

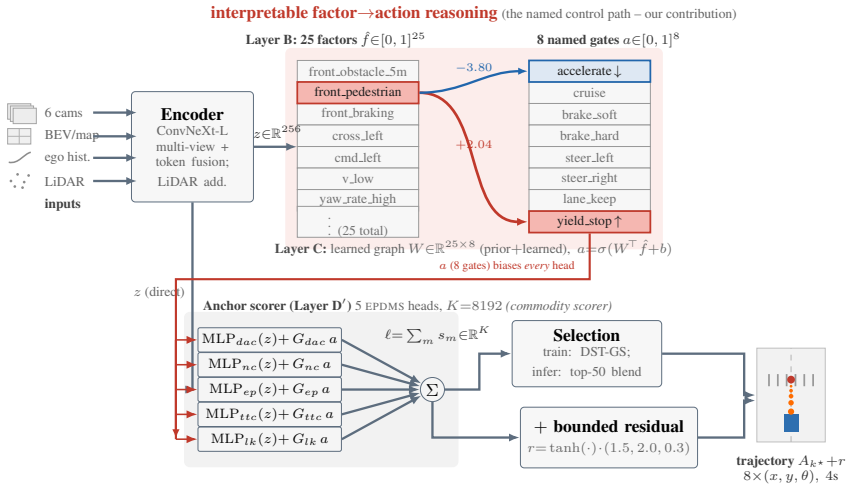


Figure 1: **The concept-gated planner.** A ConvNeXt-L encoder fuses tokenized multi-view cameras, rasterized BEV/map, ego history and (train-time only) LiDAR into a shared latent $z\in\mathbb{R}^{256}$. *Top row (accent, our contribution):* the interpretable named control path, i.e. 25 supervised factors (Layer B), the prior-initialized, end-to-end-learned graph $W\in\mathbb{R}^{25\times8}$ (Layer C), and 8 named action gates; bold edges mark one learned chain (Table 7). *Bottom row (grey, commodity):* the gates bias all five EPDMS metric heads, whose summed logit ℓ scores $K=8192$ anchors, from which a bounded residual yields the 4 s trajectory (§3).

8 biases b. The anchor keeps W near its readable prior, yet is weak enough that the task gradient adds, removes and reshapes edges extensively (§5.2).

These eight gates steer trajectory *selection* rather than regressing geometry, through a scorer that holds a fixed library of $K=8192$ trajectory anchors $\{A_k\}$, the k-means centroids of logged 4 s trajectories. For each of five EPDMS sub-metrics $m\in\{\mathrm{DAC},\mathrm{NC},\mathrm{EP},\mathrm{TTC},\mathrm{LK}\}$[1] a head reads z and emits a per-anchor logit, to which the gates add a metric-specific, per-anchor bias:

$$s_m \;=\; \mathrm{MLP}_m(z) \;+\; c\,G_m\,a, \quad G_m\in\mathbb{R}^{K\times8}, \quad \ell \;=\; \textstyle\sum_m s_m \;\in\mathbb{R}^K, \quad c{=}1 \text{ deployed}, \tag{2}$$

Equation (2) has two paths into every score. $\mathrm{MLP}_m(z)$ reads the latent directly and is a black box, while cG_ma is a learned, per-metric, per-anchor weighting of the eight named gates, so an active `yield_stop` gate can raise the NC and TTC logits of low-speed anchors and change which trajectory wins. The gates are a lever on selection, not a bottleneck through which the decision must pass, and how large a lever is a measured quantity (Table 3). Summing the five biased heads gives a selection score $\ell\in\mathbb{R}^K$. Training selects one anchor by a hard sample, made differentiable by a decoupled straight-through Gumbel-Softmax estimator (App. C.1); at inference the policy applies a drivable-area filter and emits a softmax-weighted blend of the top-k ($k=50$) anchors, each refined by a bounded residual r_k (caps 1.5 m, 2.0 m, 0.3 rad; App. B), $\tau=\sum_{k\in\text{top-}50}\mathrm{softmax}(\ell)_k(A_k+r_k)$. Inference is deterministic and every step from $\hat{f}$ onward is explicit, so one can read off the concept-channel contribution to an emitted trajectory in named quantities and intervene on it. Throughout, $\mathrm{do}(\hat{f}_k{=}v)$ denotes replacing the k-th factor activation inside the network by v and re-running the layers downstream, a model-internal intervention on the policy's computation, not on the world.

3.2 Training Objective

We train the deployed policy with a weighted sum of terms,

$$\begin{aligned}\mathcal{L} = {}& \lambda_{\text{anc}}\mathcal{L}_{\text{anc}} + \lambda_{xy}\mathcal{L}_{xy} + \lambda_h\mathcal{L}_h + \lambda_f\mathcal{L}_{\text{factor}} + \lambda_{\text{cam}}\mathcal{L}_{\text{cam}} + \lambda_{\text{off}}\mathcal{L}_{\text{off}} \\ & + \lambda_{\text{hyd}}\mathcal{L}_{\text{hyd}} + \lambda_j\mathcal{L}_j + \lambda_1\big(\overline{|W{-}W_0|} + \overline{|b{-}b_0|}\big) + \underbrace{\lambda_{cc}\mathcal{L}_{cc} + \lambda_{\text{dpo}}\mathcal{L}_{\text{dpo}} + \lambda_{\text{rl}}\mathcal{L}_{\text{rl}}}_{\text{deployed recipe; }=0\text{ in the ablations}},\end{aligned} \tag{3}$$

[1]The scorer models 5 of the 9 EPDMS-v2 sub-metrics as per-anchor heads; the omitted DDC, TLC, HC, TFC lack a clean single-frame per-anchor proxy, and two are near-saturated (DDC 0.991, TLC 0.998). Evaluation always uses the full 9-metric EPDMS-v2.

Table 1: Full navtest EPDMS-v2: EPDMS* = original release of the official v2 scorer, EPDMS = corrected release (human-behavior filter). Competitor numbers as reported; sources and re-scorings in App. C. "Concept channel": a named, intervention-tested one. The rerank row additionally reads the logged agent state at the current frame (§5.5).

Method	Backbone	Concept channel?	EPDMS* ↑	EPDMS ↑
Hydra-MDP++ (Li et al., 2025)	V2-99	✗	0.851	–
DiffusionDrive v2 (Zou et al., 2025)	ResNet-34	✗	0.855	0.875
DriveSuprim (Yao et al., 2026)	ViT-L	✗	0.871	–
SparseDrive v2 (Sun et al., 2026)	ResNet-34	✗	0.867	0.901
CLOVER (Ang et al., 2026)	DINOv2 ViT-S	✗	0.872	**0.904**
Ours, concept-gated (base)	ConvNeXt-L	✓	–	0.8985
Ours + collision rerank (§5.5)	ConvNeXt-L	✓	–	**0.9030**

which combine anchor classification, teacher-forced residual regression, factor supervision, an off-road penalty, a KL distillation of the 5 metric heads from a rule-based proxy teacher, a jerk term and the L_1 prior anchor. The deployed model adds three terms computed on the *training* split, a counterfactual-contrastive term $\mathcal{L}_{cc}$, a preference term $\mathcal{L}_{\text{dpo}}$ over anchor pairs ranked by the true EPDMS score, and a policy-gradient term $\mathcal{L}_{\text{rl}}$ (all defined in Appendix C). Two properties matter downstream. The deployed checkpoint is a short final finetune with $\lambda_{cc}{=}0$ whose factor-level intervention response is lower than the ablation recipe's (§5.3). The distillation is a softmax-over-anchors match, so each head learns a *ranking* of anchors per metric rather than a per-anchor probability, which is why the safety filter of §5.5 cannot threshold the NC head and uses a geometric test.

4 Experiments

We evaluate on NAVSIM navtest (Dauner et al., 2024) (12,146 scenes; 12,082 scored after sensor-availability filtering) with the official EPDMS-v2 metric (corrected release; App. C). Per scene it factorizes as a product of near-binary multiplicative gates times a weighted average of graded terms,

$$\text{EPDMS} \;=\; \underbrace{\text{NC}\cdot\text{DAC}\cdot\text{DDC}\cdot\text{TLC}}_{M\in[0,1]} \times \underbrace{\frac{5\,\text{EP}+5\,\text{TTC}+2\,\text{LK}+2\,\text{HC}+2\,\text{TFC}}{16}}_{Q\in[0,1]}, \tag{4}$$

with multiplicative gates NC (no at-fault collision), DAC (drivable area), DDC (driving direction), TLC (traffic light) and weighted terms EP (ego progress), TTC (time-to-collision), LK (lane keeping), HC, TFC (comfort), computed *per scene and then averaged*. Two properties of Eq. (4) drive our analysis. One gate failure zeroes a scene. And EP is normalized by the closed-loop expert's progress and clipped to $[0, 1]$, so "drive farther" is a saturating, risk-coupled lever. *Results are single-seed unless noted.*

The deployed model uses a ConvNeXt-large backbone with $K{=}8192$ k-means anchors, trained with AdamW under an EMA of the weights (decay 0.9995). Every reported number uses the inference of §3.1 (top-50 blend, drivable-area filter, no LiDAR; runtime in App. B). NAVSIM offers no held-out validation split, so we selected among six finetuning variants and 14 inference configurations by navtest score on a 267-scene subset, with all training objectives computed on navtrain only. The winner leads its runner-up by 0.15 subset standard errors and three unselected retrainings score 0.8965 ± 0.0003 (App. C), so 0.8985 and 0.9030 are benchmark results of the released artifact rather than held-out estimates.

4.1 Main result

Table 1 places the planner on the state-of-the-art line under the corrected scorer. The base model scores 0.8985, and the collision-aware reranking of §5.5 reaches 0.9030, above SparseDrive v2 and within 0.001 of CLOVER, the two systems reported under that scorer. DriveSuprim and Hydra-MDP++ have only original-release numbers, which understate the three methods reported under both by 2.0–3.4 points, so we claim no ordering against them. The rerank is a selection step NAVSIM planners routinely report, and it does not use the concept channel, so the gain says nothing about the gating. The base model trails both corrected-release systems, so Table 1 supports state-of-the-art-

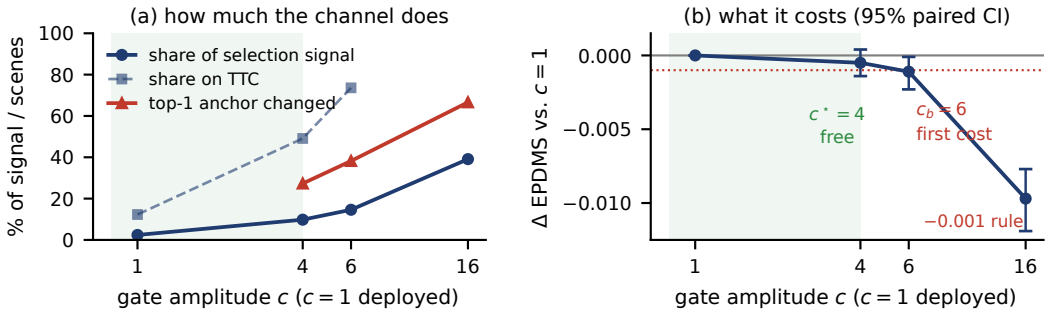


Figure 2: **The concept channel's influence and its cost.** c scales the gate term at inference (c=1 deployed; numbers in Table 3, App. C). (a) Share of the selection signal on the 5,964-scene probe sample, and scenes whose top-1 anchor changes. (b) Paired EPDMS-v2 change on full `navtest`; pre-registered, $c^\star$ is the largest c whose CI contains zero with $\Delta \geq -0.001$, c_b the first that fails.

comparable performance with an intervention-tested concept channel rather than a margin over the strongest system (breakdown in Table 10).

5 Analysis

5.1 Interpretability at no detectable cost

What does the concept channel cost? Under a same-budget recipe at the deployed scale we replace the factor-to-action graph with a black-box gate MLP that reads the latent z directly ($256 \to 64 \to 8$, GELU, 16,968 parameters against the graph's 208) and bypasses the factors, which stay supervised but no longer feed the gates. G_m, the scorer and every loss are unchanged (App. C). The two are indistinguishable over three seeds (graph 0.8860 ± 0.0021 against MLP 0.8858 ± 0.0016, paired Δ=$+0.0002 \pm 0.0017$, 95% CI $[-0.004, +0.004]$, sign-varying across seeds; Table 5), and parity also holds across four wirings of the gate pathway (absent, additive, strong counterfactual supervision, FiLM; three seeds each, means spanning 0.0006, ANOVA $F(3, 8)$=0.10; Table 6). With three seeds this excludes a cost above $\approx \pm\, 0.004$ and no more. Of the four wirings we adopt the light additive gate.

The channel is a real but low-amplitude lever. Score neutrality does not make it inert. Zeroing the gate term changes the top-1 anchor in one scene in ten and moves the emitted trajectory by centimetres, and per-scene gains and losses cancel (62.8% of scene scores move, aggregate +0.0001), which is the mechanism behind the parity above. The term carries 2.4% of the anchor-discriminating signal, but 12.3% on TTC and 6.0% on DAC against 1.3% on ego-progress, so its influence sits on the safety terms, where the reranking gain of §5.5 also lands.

Its free amplitude has a boundary. Because the gate projections are linear and bias-free, the amplitude of the gate term is an inference-time scalar c, so the share is a choice rather than a property, and Figure 2 sweeps c under a rule fixed in advance. The free region ends at four times the deployed amplitude. At c=4 the channel carries 9.8% of the signal (49.1% on TTC) and changes the selection in 27.3% of scenes at a cost whose CI contains zero, and the rerank headline of Table 1 is 0.9030 at c=4 as at c=1. At c=6 the cost first becomes measurable, narrowly (App. C), so we keep the deployed amplitude below it.

5.2 What the graph learns

How much of the deployed graph comes from the prior and how much from training? Set against its prior W_0 (Fig. 5a), the learned W departs substantially. The relative change is $\|W - W_0\| / \|W_0\| = 0.95$, and the 23 prior edges grow to 165 of magnitude $|W_{ij}| > 0.1$ and 93 above 0.5. It still retains the prior's sign structure (r=0.84), dense at low magnitude and sparse only at the strong-edge threshold ($|W_{ij}| > 1.4$) at which Fig. 5b draws it. Training refines the prior edges that receive gradient (`f_ped`$\to$`accel` moves $-3.0 \to -3.80$) and adds second-order edges the designer omitted.

Table 2: **Faithfulness measurements. (a)** Change in the emitted 4 s path under a gate intervention, latent fixed, deployed model and inference, 5,964 `navtest` scenes, c=1; longitudinal entries have the common mode removed (any raised gate shortens the path by 2.6 cm), lateral entries are endpoint displacements with 95% CIs. **(b)** Factor interventions: gate-level alignment of Δa with the prior W_0 against a permutation null; factor-level r between the shift in expected progress under $\mathrm{do}(\hat{f}_k)$ and the prior's longitudinal direction over 11 signed factors; and the absolute size of that shift.

(a) gate	`accelerate`	`brake_soft`	`brake_hard`	`yield_stop`	`steer_left`	`steer_right`
longitudinal Δpath (cm)	$+1.77$	$+0.84$	-0.54	-1.14	—	—
lateral Δendpoint (cm) [95% CI]	—	—	—	—	$+1.38$ [$+1.15$, $+1.62$]	-1.27 [-1.54, -1.01]

(b) readout	checkpoint, scenes	corr($W_0, \Delta a$)	perm. p	factor-level r [95% CI]	mean $\lvert\Delta\mathbb{E}$[progress]$\rvert$	EPDMS-v2
additive graph	deployed, 4,096 navtrain-val	0.819	$< 10^{-3}$	0.53 [0.28, 0.66], perm. p=0.11	1.3 mm (max 4.1)	0.8985
additive graph	deployed, 5,964 navtest	—	—	0.75 [0.72, 0.78]	3.6 mm (max 12.5)	0.8985
additive graph	repaired heads, same navtest scenes	—	—	0.76 [0.73, 0.78]	3.9 mm	0.8986
additive graph	11 checkpoints, 4,096 navtrain-val	0.815 ± 0.005	$< 10^{-3}$	0.78 [0.68, 0.88], s.d. 0.17	—	0.886–0.8985
2-layer MLP readout, no prior	seed 1, 4,096 navtrain-val	-0.009	0.57	0.14, perm. p=0.74	0.2 mm	0.8882

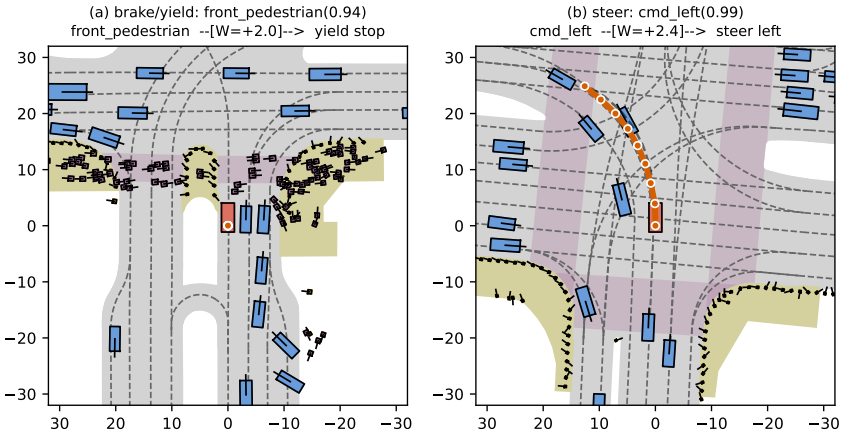


Figure 3: **Factor→gate→action chains read off real logged scenes.** Each panel is titled by the dominant active factor, its signed learned edge W to the chosen gate, and the resulting action (ego in orange, agents in blue, predicted trajectory in vermillion, ego-left on the left; two more scenes in App. D).

One can then *read* the channel's contribution to any decision in named quantities from W on real logged scenes. Figure 3 names, on held-out `navtest` scenes, the dominant active factor, its signed learned edge to a gate, and the action that follows. In the vulnerable-road-user panel `front_pedestrian` drives `accelerate` down (-3.80) and `yield_stop` up ($+2.04$) while `front_obstacle_5m` drives `brake_hard` up ($+3.01$), so the concept-channel contribution to the brake is attributable to two named factors, alongside whatever the latent path contributes. Table 9 profiles all 25.

5.3 Faithful interventions, and how to measure them

Interpretability is only useful if it is faithful, that is, if intervening on the channel changes the policy's output as the graph claims. We test this with model-internal do-interventions on the gates and on the factors, reporting direction, absolute size and how much of the deployed output changes. Two readouts carry the measurements, the *emitted trajectory* (the deployed top-50 blend) and the *expected progress* of the full anchor distribution, $\mathbb{E}_{k\sim\mathrm{softmax}(\ell)}[\mathrm{progress}(A_k)]$. The second is the untruncated readout, used for small factor interventions for the reason given below.

Training orders the gate→behaviour map semantically, so the channel is an actuator. Intervening directly on a gate, scene latent held fixed, moves the emitted trajectory as the gate's *name* prescribes (Table 2a). The four longitudinal gates order themselves exactly once the common mode is removed, and the two steering gates displace the endpoint in opposite, name-consistent directions with CIs excluding zero (on n=4 gates the CIs carry the weight, not the rank statistic). The additive layer fixes the gate logits rather than the geometry of the path, so this ordering comes from training rather than from the prior. It holds at every amplitude of Table 3 (ρ=1.00 from c=1 to 16; one ordering of

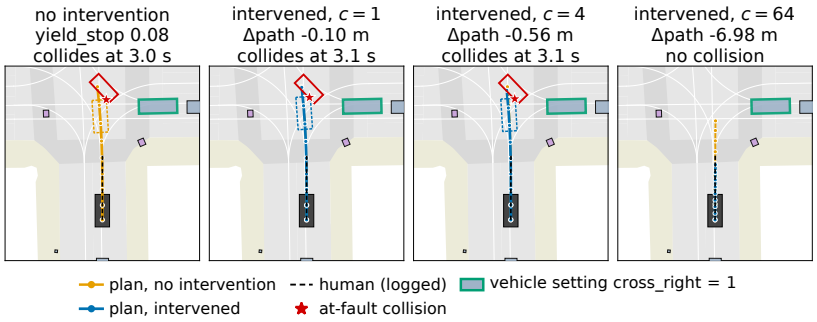


Figure 4: **Faithful actuation, ineffective intervention, on one logged scene.** The policy misses `cross_right` (green-outlined vehicle) and hits an oncoming vehicle (red); $\mathrm{do}(\texttt{cross_right}{=}1)$ raises `yield_stop` and shortens the plan at every amplitude, but the collision persists until $c{=}64$ (ego-left on the left; App. E).

four gates has exact permutation $p{=}0.042$, so the replication across six amplitudes is the evidence). At the free $c{=}4$ point the steering displacements are $+7.2$ $[6.0, 8.5]$ and -7.1 $[-8.4, -5.8]$ cm, so the centimetric scale is the deployed choice rather than a property of the channel. `cruise` is the exception, and its readout column explains why (App. E).

Factor→behaviour responses reproduce across checkpoints, and are millimetric. Propagating $\mathrm{do}(\hat{f}_k)$ to the shift in expected progress and correlating it with the prior's longitudinal direction over the 11 signed factors gives $r{=}0.78$, 95% CI $[0.68, 0.88]$, over 11 graph-bearing checkpoints (Table 4; direction agreement 0.79). The safety factors most strongly *reduce* expected progress. The deployed checkpoint is the weakest of the family ($r{=}0.53$ on the shared protocol, 0.75 on `navtest` scenes; Table 2b, App. C). The absolute effect is millimetric in expected progress (deployed mean 1.3 mm, max 4.1 mm; 3–4× larger at $c{=}4$), so the intervention shifts the *direction* of the distribution rather than the metres driven. The interventions are also *selective*, since zeroing a factor displaces the endpoint by 2.3 cm on average when it is active in the scene and by 0.1 cm when inactive ($n{=}1{,}024$; 21–57× per factor over a near-zero denominator, one factor at ≈1; App. E).

The permutation null certifies prior survival, and is auxiliary. Gate-level alignment is near-automatic for an additive layer, since $\mathrm{do}(\hat{f}_k)$ shifts gate j's logit by exactly W_{kj}. What permutation cannot reproduce is alignment with the *prior*, which the test certifies (0.815 against a null of $+0.005 \pm 0.072$; a non-additive readout with no prior returns the null, -0.009, $p{=}0.57$; App. E). All factor-level numbers are read off the continuous distribution $\mathrm{softmax}(\ell)$, since the top-1 anchor registers mostly quantization noise and inverts in sign ($r{=}+0.53$ against -0.54; App. E), and selected trajectories grouped by dominant gate match the gate names where the gate is well populated (Fig. 8).

5.4 Auditing the policy through named concepts, and where intervention loses the scene

Because the channel's contribution to every decision is explicit, one can *trace* a wrong decision to a wrong factor and correct the factor in place. Figure 4 follows one real logged `navtest` scene. The policy misses `cross_right` (probability 0.04 against label 1, set by a vehicle standing 18 m to the right) and drives on at 4.5 m/s into an official at-fault collision with an oncoming vehicle cutting across its lane, scoring EPDMS 0.000 where the logged human brakes to a stop. The audit reads back the missed factor, and $\mathrm{do}(\texttt{cross_right}{=}1)$ raises `yield_stop` (+0.26), flips the top-1 anchor and shortens the selected path by 0.10 m (0.56 m at $c{=}4$). Yet the collision only moves from 3.0 to 3.1 s, and only at $c{=}64$ does the plan stop short of it. Across the 1,598 probe-sample scenes with a missed safety factor, the correction moves the top-1 anchor in 6.1% of scenes (18% at $c{=}4$).

Faithful actuation is not effective intervention. Under the standard test-time-intervention protocol (Koh et al., 2020; Shin et al., 2023), replacing all 25 factors by ground truth in every `navtest` scene changes the score by -0.0001 $[-0.0005, +0.0003]$ against no intervention and by $+0.0004$ $[-0.0002, +0.0009]$ against a scene-permuted ground truth (minimum detectable effect 0.0008), and

by -0.0002 $[-0.0006, +0.0002]$ when the labels of App. F are repaired and the larger intervention they define is re-run (App. E). Nor is the response specific to the assertion being true. On the $1{,}598$ scenes with a missed safety factor, correcting it is indistinguishable from asserting the same multiset of factors where every assertion is false, on the score (-0.0009 $[-0.0038, +0.0018]$ at $c{=}4$) and on the path change (App. E).

The scene is lost at the read-out, and the reason is structural. In Eqs. (1)–(2) the truth of an assertion enters nowhere, since $\mathrm{do}(\hat{f}_k)$ shifts every gate logit by exactly W_{kj}, so an additive coupling pushes a scene-independent direction (per-scene cosine 0.99). A multiplicative FiLM coupling, three seeds against three additive seeds under the same control, is also null (all six paired CIs contain zero), and here the mechanism is the read-out. The FiLM term is scene-specific in latent space (direction consistency 0.28–0.30), but its multiplier is nearly constant across scenes, so it acts as a rescaling. The metric heads then map scaling-like perturbations to one direction (0.99) while random perturbations of the same norm keep theirs (0.03–0.12; App. E). A scene-aware intervention therefore needs a scene-varying, anisotropic modulation ahead of the read-out.

What survives the negative result is the audit, and the factors it reads can be repaired at no detectable planning-score cost. Auditing presumes trustworthy detectors, and initially ours were not. The seven safety factors had a macro recall of 0.15 on full `navtest`, and `front_obstacle_5m` was effectively blind (0.00). We traced this to a rear-axle labelling error and refit *only* the 25 factor heads on corrected labels, everything else frozen. Safety macro recall rises to 0.65 (`front_obstacle_5m` $0.00 \to 0.79$) and the all-factor AUC from 0.93 to 0.95, while the all-factor macro F_1 drops from 0.82 to 0.78 as the operating point moves toward recall. The planning score is unchanged (paired $\Delta{=}+8{\times}10^{-5}$, 95% CI containing 0) and so is the channel's faithfulness, re-measured on the repaired heads (Table 2b; App. F).

5.5 Per-scene EPDMS analysis and collision-aware reranking

This extension is independent of the concept channel. The NC gate fails most often (1.8% of scenes), has no loss on the final trajectory, and each failure zeroes a scene, so it is the steepest lever. Its head is rank-calibrated only (§3.2), so instead of thresholding it we demote any top-anchor trajectory that collides at fault against a constant-velocity prediction of the agent boxes and velocities NAVSIM logs at the current frame, reading no future frame. On full `navtest` this lifts the score from 0.8985 to 0.9030 ($+0.0045$; paired bootstrap 95% CI $[+0.0027, +0.0064]$), localized to the safety terms it targets (NC $+0.0058$, TTC $+0.0065$) with ego-progress unchanged. Per collision it removes 86 at-fault collisions and introduces 24 (net 62), and per scene NC flips $0 \to 1$ on 83 scenes. Only *injecting new information* helps, since inference-time reweighting is exhausted (Table 11). Finetuning the converged policy with a differentiable at-fault loss instead *reduces* the score (App. H), so the filter belongs at selection time rather than generation time.

6 Limitations

The scorer keeps a direct latent path, so the policy is not a concept bottleneck. The channel carries 2.4% of the selection signal at the deployed amplitude and at most $\approx 10\%$ for free (Table 3). It therefore provides a queryable reasoning chain over that share rather than behavioural control, and intervening on it is a diagnostic rather than a repair (§5.4). The graph is a prior-structured *gating* layer rather than a structural causal model, and every do here is model-internal (§3.1). Gate-level alignment certifies prior survival only, the factor-level response is weaker in the deployed checkpoint than in the ablation recipe (Table 2b), the suite has run on one task, and the headline model is a selected artifact whose recipe replicates at 0.8965 ± 0.0003 (App. C).

7 Conclusion

A competitive closed-loop planner can carry a named concept channel at no detectable cost, with a free-amplitude boundary at four times the deployed value. The channel is a faithful actuator at every amplitude, yet the read-out loses the scene-awareness an intervention would need. Auditability by construction and faithfulness under verification are a pairing worth building on.

REPRODUCIBILITY STATEMENT

Architecture and tensor shapes are in §3.1 and App. B; every loss term, weight and schedule in §3.2 and App. C; the model lineage, the checkpoint each table uses and the selection protocol in Table 4; the intervention protocol and estimators in App. E; the amplitude sweep, its pre-registered reading rule, the standard test-time intervention, the FiLM control and the simulator controls in App. C and App. E; the factor definitions and the label correction in App. F; and the rerank as Algorithm 1. All scores are means of the official devkit `score` column over the 12,082 valid `navtest` scenes. Code, the released and repaired checkpoints, the 267-scene selection subset (scene tokens) and the per-scene evaluation files will be released with the paper.

ETHICS STATEMENT

This work trains and evaluates a driving policy on NAVSIM (Dauner et al., 2024), an existing public dataset built on nuScenes, and compares against public checkpoints. It collects no new driving data and involves no human subjects. A planner that reports named reasons invites the reading that those reasons control the decision, and our central result is that on this policy they do not, since the channel is faithful under intervention yet intervening on it does not reach the outcome (§5.4). We state that limit directly because an auditable-looking interface is the kind of evidence a deployment argument reaches for. Nothing here is evidence of on-road safety. The ego is unrolled by a rule-based simulator whose other agents replay their logs, so the evaluation is non-reactive and does not test compounding interaction.

AI USE STATEMENT

We used generative AI tools throughout the preparation of this paper, and we disclose that use item by item rather than in summary. *Writing.* They drafted and rewrote sections, which the authors then revised and approved. No section reached its present form without an author rewriting it. *Code.* They acted as coding assistants for parts of the training, evaluation and analysis code and for the figure scripts. *Experiment design.* They proposed the gate-amplitude sweep of App. C, the FiLM control and the simulator test of factor correction in App. E, and the pre-registered rules by which each was read; the authors decided those rules before the runs. *Interpretation.* They assisted in reading the results, including the identification of the read-out as the point at which scene-specificity is lost. *Retrieval.* They were used to locate and check the cited literature, including the venue and page data in the bibliography. *Formal material.* They assisted with the $\arg\max$ margin condition of App. E and with the numerical verification reported beside it. We did not use generative AI to produce or impute any reported measurement, and we did not use it to generate training data. Every number in this paper comes from the training and scoring runs described in the text. We executed all AI-assisted code and cross-checked its outputs against independently computed values, and we re-checked the reported numbers against the stored per-scene arrays. An author checked every citation against the cited source. We have reviewed all AI-assisted work and take responsibility for the final content of this paper, including its text, claims and released artifacts.

REFERENCES

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Sining Ang, Yuguang Yang, Canyu Chen, and Yan Wang. CLOVER: Closed-loop value estimation and ranking for end-to-end autonomous driving planning. *arXiv preprint arXiv:2605.15120*, 2026.

Holger Caesar, Juraj Kabzan, et al. nuPlan: A closed-loop ML-based planning benchmark for autonomous vehicles. In *CVPR ADP3 Workshop*, 2021.

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 (TPAMI)*, 45(11):12878–12895, 2023.

Daniel Dauner, Marcel Hallgarten, Zhiyu Li, et al. NAVSIM: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024.

Jessica Maria Echterhoff, An Yan, Kyungtae Han, Amr Abdelraouf, Rohit Gupta, and Julian J. McAuley. Driving through the concept gridlock: Unraveling explainability bottlenecks in automated driving. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 7331–7340, 2024.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics (TACL)*, 2021.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, 2019.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

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

Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations (ICLR)*, 2017.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. VAD: Vectorized scene representation for efficient autonomous driving. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viégas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *International Conference on Machine Learning (ICML)*, 2018.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning (ICML)*, 2020.

Songning Lai, Tianlang Xue, Hongru Xiao, Lijie Hu, Jiemin Wu, Ninghui Feng, Runwei Guan, Haicheng Liao, Zhenning Li, and Yutao Yue. DRIVE: Dependable robust interpretable visionary ensemble framework in autonomous driving. *arXiv preprint arXiv:2409.10330*, 2024.

Kailin Li, Zhenxin Li, Shiyi Lan, Yuan Xie, Zhizhong Zhang, et al. Hydra-MDP++: Advancing end-to-end driving via expert-guided hydra-distillation. *arXiv preprint arXiv:2503.12820*, 2025.

Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, et al. Hydra-MDP: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*, 2024.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations (ICLR)*, 2017.

Anita Mahinpei, Justin Clark, Isaac Lage, Finale Doshi-Velez, and Weiwei Pan. Promises and pitfalls of black-box concept learning models. *arXiv preprint arXiv:2106.13314*, 2021.

Andrei Margeloiu, Matthew Ashman, Umang Bhatt, Yanzhi Chen, Mateja Jamnik, and Adrian Weller. Do concept bottleneck models learn as intended? *arXiv preprint arXiv:2105.04289*, 2021.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2nd edition, 2009.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *AAAI Conference on Artificial Intelligence*, 2018.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206–215, 2019.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 2021.

Sungbin Shin, Yohan Jo, Sungsoo Ahn, and Namhoon Lee. A closer look at the intervention procedure of concept bottleneck models. In *International Conference on Machine Learning (ICML)*, 2023. arXiv:2302.14260.

Wenchao Sun, Xuewu Lin, Keyu Chen, Zixiang Pei, Xiang Li, Yining Shi, and Sifa Zheng. SparseDriveV2: Scoring is all you need for end-to-end autonomous driving. *arXiv preprint arXiv:2603.29163*, 2026.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. PARA-Drive: Parallelized architecture for real-time autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Wenhao Yao, Zhenxin Li, Shiyi Lan, Zi Wang, Xinglong Sun, Jose M. Alvarez, and Zuxuan Wu. DriveSuprim: Towards precise trajectory selection for end-to-end planning. In *AAAI Conference on Artificial Intelligence*, 2026. arXiv:2506.06659.

Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *International Conference on Learning Representations (ICLR)*, 2023.

Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. DAGs with NO TEARS: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Jialv Zou, Shaoyu Chen, Bencheng Liao, et al. DiffusionDriveV2: Reinforcement learning-constrained truncated diffusion modeling in end-to-end autonomous driving. *arXiv preprint arXiv:2512.07745*, 2025.

Appendix overview. App. A positions the verification suite in the faithfulness-evaluation literature (§2); App. B gives encoder and head shapes and runtime (§3.1); App. C the loss terms, weights, training lineage, the §5.1 ablation recipes and the amplitude-sweep protocol; App. D the graph, chain and gate-signature visualizations behind §5.2–§5.3; App. E the intervention protocol, the $\arg\max$ theorem, the structural argument, the standard test-time intervention, the FiLM control and the simulator controls (§5.3–§5.4); App. F the factor taxonomy and detector repair (§5.4); App. G and App. H the rerank analysis and the training-time control (§5.5).

A Extended positioning

The faithfulness-evaluation literature supplies three cautionary results we build on directly. Saliency maps can be invariant to randomizing the model (Adebayo et al., 2018), so an explanation that looks structured need not depend on the model at all. Probe accuracy must be read against a control-task

baseline (Hewitt & Liang, 2019), so a high score is uninformative without a null. And remove-and-retrain protocols (Hooker et al., 2019) test whether the highlighted features actually carry the prediction. Our permuted-graph null instantiates the first two for a structured layer, and the active-vs-inactive selectivity ratio instantiates the control-condition idea (Kim et al., 2018). The $\arg\max$-inversion pitfall of App. E is, to our knowledge, a new addition to this list, specific to policies that select from a discrete vocabulary.

Concept bottlenecks for driving precede us. Concept Gridlock (Echterhoff et al., 2024) learns a bottleneck of named driving concepts on logged video and predicts steering and speed from it, reporting the explanations the bottleneck affords; DRIVE (Lai et al., 2024) studies the dependability and stability of such concept-based explanations across perturbations. Both operate on offline control prediction rather than on a closed-loop policy scored on its own rollout, do not place the concepts next to a direct latent path whose share of the decision is measured, and do not verify the concepts by intervention on behaviour or measure what intervening on them does to the outcome; those are the additions this paper makes, and its novelty should be read against that background rather than as the first driving CBM.

B Encoder and selection-head shapes

Encoder. Six scene tokens (a semantic-context token, a drivable-area raster token, an ego-history token, and ego/left/right lane-polyline tokens), a camera summary token, and 32 image patches from the ConvNeXt backbone together form 39 tokens of width 64, which a 4-layer transformer fuses into $z \in \mathbb{R}^{256}$. The train-time LiDAR branch encodes a $4{\times}128{\times}128$ voxel grid and adds it to z through a zero-initialized projection.

Runtime. On a single RTX PRO 6000 a planning step takes a median 22 ms (46 Hz), rising to 101 ms (9.9 Hz) with the constant-velocity collision rerank of §5.5, against NAVSIM's 2 Hz replanning cadence, so both leave substantial real-time headroom.

Residual head. The bounded residual is $r_k = \tanh(\cdot) \odot (1.5, 2.0, 0.3)$, factorized as $r_k = g(\mathrm{proj}(z) + \mathrm{emb}(k))$. The factorization avoids a $K{\cdot}T{\cdot}3$ regression head and adds only a few million parameters over the backbone.

C Loss weights, schedule, and training lineage

Terms of Eq. (3). $\mathcal{L}_{\text{anc}}$ is cross-entropy against the ground-truth nearest anchor; $\mathcal{L}_{xy}$ is a time-weighted squared error in x, y and $\mathcal{L}_h$ a $1-\cos(\Delta\theta)$ heading term, both teacher-forced on the residual at the ground-truth anchor; $\mathcal{L}_{\text{factor}}$ and $\mathcal{L}_{\text{cam}}$ are the binary factor supervision and its camera-only auxiliary; $\mathcal{L}_{\text{off}}$ is an off-road penalty on the *final* (anchor+residual) trajectory; $\mathcal{L}_{\text{hyd}}$ is the multi-target KL distillation of the 5 metric heads from a rule-based proxy teacher, $\mathrm{KL}\big(\mathrm{softmax}(\hat{s}_m/\tau) \,\|\, \mathrm{softmax}(s_m)\big)$; $\mathcal{L}_j$ is a jerk smoothness term. Of the three deployed-only terms, $\mathcal{L}_{cc}$ sharpens gate responses under factor interventions, $\mathcal{L}_{\text{dpo}}$ is a preference term over anchor pairs ranked by the true EPDMS score, and $\mathcal{L}_{\text{rl}}$ is a policy gradient whose reward is read from the same offline real-EPDMS cache, with a 0.05-weighted KL to the reference policy. The prior bias is $b_0{=}0$ except `cruise` and `lane_keep`, both 0.5.

Main recipe (deployed model). ConvNeXt-large backbone, $K{=}8192$ anchors, 15 epochs, batch 24, AdamW at lr $5{\times}10^{-5}$, EMA decay 0.9995, mixed precision, image backbone frozen during an initial warmup. Loss weights: $\lambda_{\text{anc}}{=}1$, $\lambda_{xy}{=}1$ (longitudinal 1, lateral 2), $\lambda_h{=}20$ ($1-\cos$ heading), $\lambda_f{=}0.25$, $\lambda_{\text{cam}}{=}0.25$, $\lambda_{\text{off}}{=}4$, $\lambda_{\text{hyd}}{=}1$ ($\tau{=}0.05$), $\lambda_j{=}0.5$, $\lambda_1{=}10^{-3}$, plus a counterfactual contrastive term ($\lambda_{cc}{=}0.3$, margin 0.3), a real-EPDMS preference term ($\lambda_{\text{dpo}}{=}0.5$), and a small policy-gradient term ($\lambda_{\text{rl}}{=}0.3$, 2-epoch warmup). The decoupled Gumbel temperatures (τ_f, τ_b) anneal through $(1.0, 1.0) \to (0.5, 1.0) \to (0.1, 0.5)$ toward a target final $(0.01, 0.5)$. *The checkpoint evaluated in Table 1 is a short $\lambda_{cc}{=}0$ fine-tune of this recipe; the multi-stage lineage is a limitation of the present implementation, and a clean single-stage retraining is left to future work.* The released 0.8985 sits $+0.0020$ above the 0.8965 ± 0.0003 replication of its recipe (three retrainings with a standardized finetune and no selection), which leaves the recipe's $+0.0105$ margin over the ablation unaffected.

Selection on the navtest subset, and how much it can inflate the headline. The deployed finetune

was chosen among six variants of the same parent (λ=0 control and five ego-progress-distillation targets) by EPDMS on a 267-scene navtest subset, where they scored 0.8912, 0.8944, 0.8942, 0.8993, 0.9011 (deployed) and 0.8979; the inference configuration was chosen from 14 candidates on the same subset, of which every pure re-weighting scored below the deployed top-50 blend with drivable filter (Table 11). The winner leads its runner-up by 0.0018, against a subset standard error of 0.0115 (0.15 SE), so the variants are not distinguishable there. The reported headline is measured on all 12,082 scenes, of which the 267 are 2.2%. If the six variants were equivalent, the expected maximum of six noisy subset means exceeds the truth by 1.27 SE, or 0.0146 on the subset. That enters the headline only through those 267 scenes and amounts to 0.0003, below the headline's own standard error of 0.0017. This is the optimism of the reported number for the selected model; it is not a bound on how the selected model would fare on a fresh benchmark, for which the unselected replication above is the right estimate.

LiDAR. The deployed agent declares no LiDAR sensor, so the LiDAR encoder and its projection never execute at evaluation and every reported score is camera + map + ego-history only; zeroing the branch at evaluation is therefore an identity. No modality dropout is used. A matched-seed retraining of the ablation recipe without the LiDAR cache scores 0.8882 against 0.8836 with it, a difference of 1.5 seed standard deviations (p≈0.12) that the training-time proxy contradicts in sign, so we report no measurable LiDAR contribution in either direction; LiDAR is present in 12.6% of training scenes.

Gate-amplitude sweep (Table 3). The gate term of every metric logit is $G_m(c\,a) = c\,G_m a$ because the projections are linear and bias-free, so c scales the channel exactly (verified to 0 numerical difference for every c used, and c=1 is bit-identical to the flag being absent). The reading rule was fixed before the sweep. $c^\star$ is the largest c whose paired 95% CI against c=1 contains zero with a point estimate $\Delta \geq -0.001$, and c_b the first c whose CI excludes zero with $\Delta < 0$; applied mechanically it gives $c^\star$=4 and c_b=6, and c=6 fails both criteria narrowly ($\Delta = -0.00114$ against -0.001; CI upper end -0.00006). The share column is computed on the 5,964-scene probe sample, whose rare-safety stratum is taken exhaustively, so it is not a navtest population mean (on its first 1,024 scenes the deployed share is 1.7%). Partitioning the per-scene score change, at c=4 the worst 1% of scenes carry 84% of the loss and 7,665 of the 7,673 changed scenes move only ego-progress; the aggregate is a few dozen binary-metric flips in both directions. At c=16 the concentration has fallen to 50% and 4,771 scenes worsen against 3,290 that improve. Removing the channel at $c^\star$ costs -0.0004 $[-0.0014, +0.0006]$. The new operating points were evaluated as log-partitioned shards, a layout shown to reproduce the published deployed run token by token with zero difference. Those runs also score the 64 scenes whose camera files were missing when the published runs were made, the same 64 in every published run. Every published figure is therefore a mean over 12,082 of 12,146 scenes, and a complete run of the deployed configuration gives 0.8985 either way. The interpretable gate changes the selected anchor on 2.3–2.7% of scenes against 2.9% for the black-box gate MLP, so it is as load-bearing as the black-box layer it replaces.

Scorer release, and the sources of Table 1. The official NAVSIM v2 scorer exists in two releases. The original omitted the human-behavior filter when aggregating EPDMS (a sub-metric the logged human trajectory itself fails is meant not to be penalized), and the corrected release applies it; the three methods reported under both gain 2.0–3.4 points from the fix (DiffusionDriveV2 85.5 → 87.5, SparseDriveV2 86.7 → 90.1, CLOVER 87.2 → 90.4; Sun et al., 2026; Ang et al., 2026). Every number in this paper is the corrected release's `score` column, which per scene equals Eq. (4) exactly wherever the two-frame term is defined and its 14-unit renormalization where it is not; the logged human replays at 0.9451 under it (§5.5) against the 0.903 reported under the original release (Yao et al., 2026). In Table 1, EPDMS* for Hydra-MDP++ (V2-99) is the 85.1 reported for it under the official v2 code by Yao et al. (2026) and listed again by Sun et al. (2026) and Ang et al. (2026); the Hydra-MDP++ paper itself reports 84.1 with its own implementation of the extended metrics. EPDMS for DiffusionDriveV2 is the 87.5 re-scored with the corrected release by Sun et al. (2026) and Ang et al. (2026); the remaining entries are the papers' own numbers. DriveSuprim (ViT-L) and Hydra-MDP++ have no corrected-release number, so Table 1 draws no ordering between them and ours.

Ablation recipe (Tables 5, 6). The ablations share a supervised-learning recipe warm-started from the same v5 checkpoint and differ *only* in the gate pathway: a black-box gate MLP, the light additive gate (ours, Eq. (2)), the additive gate under a strong counterfactual-contrastive loss (strong-cc: λ_{cc}=0.8, margin 1.5), or FiLM-style multiplicative conditioning. The black-box gate MLP replaces Eq. (1) by $a = \sigma(\mathrm{MLP}(z))$ with MLP of shape $256 \to 64 \to 8$ (GELU; 16,968 parameters against 208 for W and b). It ignores the factor activations, so no $\mathrm{do}(\hat{f}_k)$ can reach its gates, while the factor

Table 3: **The concept channel's influence and its cost, as a curve.** c scales the gate term at inference (c=1 deployed). Share of the anchor-discriminating signal (in brackets, on TTC) on the 5,964-scene probe sample (App. C); top-1 changed and median endpoint shift relative to c=1; EPDMS-v2 on full `navtest`, paired against c=1. Pre-registered rule, $c^\star$ is the largest c whose CI contains zero with $\Delta \geq -0.001$ and c_b the first that fails.

c	share (TTC)	top-1 changed	endpoint (cm)	EPDMS	Δ vs c=1 [95% CI]
1 (deployed)	2.4% (12.3%)	—	—	0.8985	—
4 ($c^\star$)	9.8% (49.1%)	27.3%	5.8	0.8980	-0.0005 $[-0.0014, +0.0004]$
6 (c_b)	14.6% (73.6%)	38.2%	9.7	0.8974	-0.0011 $[-0.0023, -0.0001]$
16	39.1%	66.6%	30.4	0.8888	-0.0097 $[-0.0119, -0.0077]$

heads and their supervision, the gate projections G_m, the scorer and all other losses are unchanged, and the L_1 prior anchor is dropped because there is no prior. The parity of Table 5 is therefore established on this shared supervised recipe and not at the deployed score. The deployed lineage passes through $\mathcal{L}_{cc}$, which sharpens gate responses under factor interventions and so has no counterpart in an arm that bypasses the factors, and we did not re-run that lineage for the MLP arm. Each configuration trains 15 epochs (effectively uncapped, ≈9.5k steps/epoch) at the deployed batch 24 and lr 5×10^{-5}, three seeds, and is scored at the final EMA epoch under the deployed top-k=50 drivable-area inference. The non-additive *MLP readout* of Table 2b is a different model. It keeps the factors as the gates' only input and replaces the linear map by a two-layer MLP $25\to8\to8$ (GELU; 280 parameters), with no prior, trained under this same recipe for one seed.

Table 4: **Model lineage: which model each result uses.** All models share the encoder architecture, K=8192 anchors and the inference of §3.1. "Deployed" is the released artifact v5 → iter90b → iter95 (full recipe with $\mathcal{L}_{cc}$, $\mathcal{L}_{\text{dpo}}$, $\mathcal{L}_{\text{rl}}$) → a two-epoch λ_{cc}=0 finetune selected among six variants on the 267-scene subset.

model	objective / provenance	factor heads	used in
deployed artifact (0.8985; 0.9030 with rerank)	full recipe, selected finetune	original	Tables 1, 3, 2a and the deployed rows of 2b; §5.2, §5.4, §5.5; Figs. 3–9
recipe replication, 3 seeds (0.8965 ± 0.0003)	full recipe, standardized finetune, no selection	original	§6; this appendix
SL-only ablation family, 4 wirings × 3 seeds	λ_{cc}=λ_{dpo}=λ_{rl}=0, warm-start v5	original	Tables 5, 6; 9 of the 11 checkpoints of Table 2b (the other two: deployed and its parent iter95); the FiLM and additive seeds are the two families of the App. E intervention control
deployed artifact at gate amplitude c=4	same weights, inference-time scalar	original	Table 3; the c=4 readings of §5.3–§5.4
MLP readout, 1 seed (0.8882)	SL-only recipe, $25\to8\to8$ gates	original	Table 2b, last row
repaired detector (0.8986)	deployed artifact with its 25 factor heads refit on corrected labels	repaired	repair paragraph of §5.4; third row of Table 2b; App. F

Faithfulness across the lineage. Behaviour-level r (Table 2b, shared protocol) spans 0.38–0.97 over 20 checkpoints measured this way; the SL-only recipe averages ≈0.84 and the deployed lineage ≈0.58, and within the deployed lineage the six finetune variants sit in a 0.54–0.63 band regardless of objective, so no single training stage is responsible for the difference.

C.1 Decoupled straight-through Gumbel-Softmax selection

Hard anchor selection is non-differentiable, and a plain softmax relaxation blends physically distinct trajectories. We use a decoupled straight-through estimator. With one Gumbel sample $g_k = -\log(-\log u_k)$, $u_k \sim \mathcal{U}(0,1)$, the forward pass takes a hard one-hot at a sharp temperature τ_f while the backward pass uses a soft distribution at a smoother temperature τ_b *computed from the same sample*:

$$y^{\text{hard}} = \text{onehot}\big(\arg\max_k \tfrac{\ell_k+g_k}{\tau_f}\big), \quad y^{\text{soft}} = \text{softmax}\big(\tfrac{\ell+g}{\tau_b}\big), \quad y = y^{\text{hard}} + \big(y^{\text{soft}} - \text{sg}[y^{\text{soft}}]\big), \tag{5}$$

Table 5: **Graph-vs-MLP ablation, three seeds at the full deployed scale.** The factor-to-action graph (ours, shaded) against a black-box gate MLP that reads the latent directly, under an identical same-budget recipe, both scored at the same final EMA epoch; Δ is the paired difference (t=0.17, df=2). The two are statistically indistinguishable, so we claim parity, not a gain.

Gate pathway	seed 1	seed 2	seed 3	mean $\pm$ s.d.
black-box gate MLP ($z \to a$, no factors)	0.8851	0.8877	0.8847	0.8858 ± 0.0016
factor-to-action graph (ours)	0.8872	0.8872	0.8836	0.8860 ± 0.0021
Δ (graph $-$ MLP)	$+0.0021$	-0.0005	-0.0011	$+0.0002 \pm 0.0017$

Table 6: **Gate-pathway ablation at the full deployed scale.** Four ways of coupling the gate pathway to the scorer (rows; *mechanism* column), scored by EPDMS-v2 under an identical same-budget recipe on full `navtest`, three seeds (mean $\pm$ s.d.); the bottom row reports a one-way ANOVA across the four means. Coupling strength does not move the score.

Gate pathway	mechanism	EPDMS-v2 (3 seeds) ↑
none	black-box gate MLP ($z \to a$)	0.8858 ± 0.0016
light additive (ours)	per-metric gate bias, Eq. (2)	0.8860 ± 0.0021
strong (cc)	heavy counterfactual supervision	0.8863 ± 0.0013
strong (FiLM)	multiplicative conditioning	0.8865 ± 0.0012
one-way ANOVA		$F(3, 8)$=0.10, n.s.

with $\mathrm{sg}[\cdot]$ the stop-gradient. Decoupling τ_f from τ_b gives a hard, low-variance forward selection (so training matches the argmax used at evaluation) while preserving a smooth, well-conditioned gradient through τ_b; both temperatures are annealed over training. At evaluation no sample is drawn. After the drivable-area filter the policy blends the top-50 anchors with weights $\mathrm{softmax}(\ell)$ restricted to that set (§3.1), so the emitted trajectory is a deterministic, continuous function of ℓ; the top-1 anchor $\arg\max_k \ell_k$ is reported only as a summary statistic.

D Additional interpretability visualizations

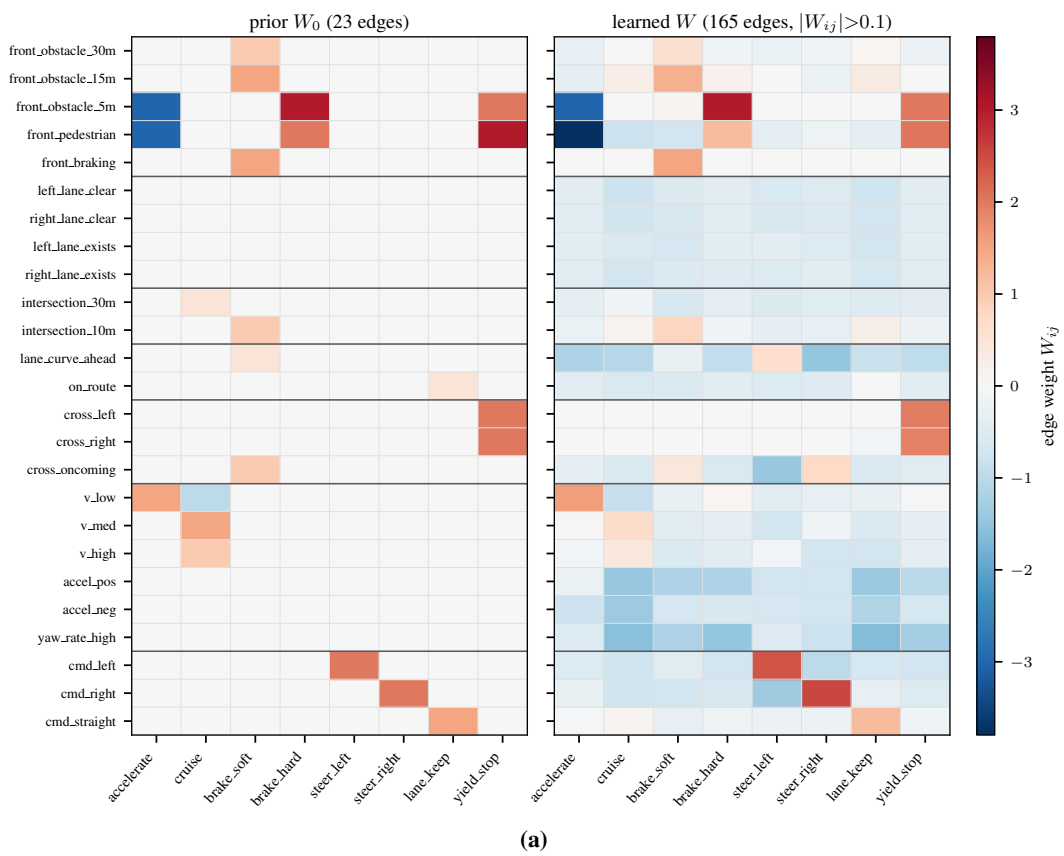
Figure 5 contrasts the hand-coded prior W_0 with the learned graph W as signed heat maps (a) and draws the strongest learned edges as a directed graph (b); Table 7 tabulates representative perception→intent→behaviour chains read off the deployed W, and Figure 3 reads them off real held-out scenes (all four in Figure 6).

The strongest *de novo* edge is `yaw_rate_high`, which suppresses `lane_keep` (-1.65), `cruise` (-1.59) and even `brake_hard` (-1.48); read together, mid-turn the policy holds neither a straight lane nor the brakes.

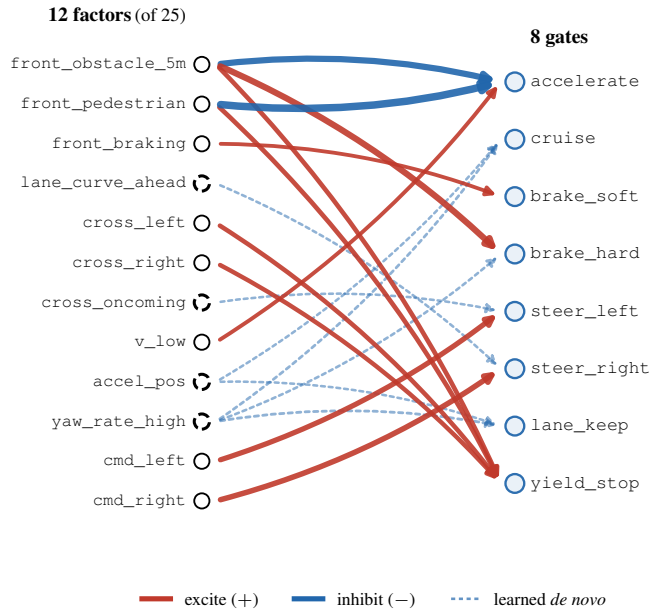
Figure 8 shows the behavioral signature of each named gate on a 3,000-scene held-out set, without any supervision on trajectory shape. `accelerate` and `cruise` select long, straight reaches, `brake_soft` and `brake_hard` short ones, the two steering gates cleanly mirror-imaged curving families, `lane_keep` centred straight paths, and `yield_stop` short, fanned paths consistent with a hold; `brake_hard` (5 scenes) and `yield_stop` (73) are rarely dominant, so their names rest on the interventions of §5.3 rather than on this grouping.

E Intervention protocol and the $\arg\max$ artifact

Why the additive gate-level correlation is near-automatic. Because $\mathrm{do}(\hat{f}_k)$ shifts gate j's logit by exactly W_{kj}, the response's sign equals $\mathrm{sign}\, W_{kj}$ in every scene, so a permuted graph (a random permutation of W's entries at matched marginal, 1000 permutations) already reproduces $\mathrm{corr}(W, \Delta a)$=0.83. Figure 7a draws that gate-level response against the prior edge weight, and Figure 7b the behaviour-level test beside it. Alignment with the *prior* is what survives permutation, and the learned graph stays close to it ($\mathrm{corr}(W_0, W)$=0.844 ± 0.001 across the 11 graph-bearing checkpoints, set by the L_1 anchor). On the non-additive MLP readout the same intervention leaves



(a)



(b)

Figure 5: **The factor-to-action graph. (a)** Signed heat maps of the hand-coded prior W_0 (left) and the end-to-end-learned W (right); red excites, blue inhibits, rows are the 25 factors and columns the 8 action gates. The model refines the sparse prior into a denser, data-optimal graph while preserving its interpretable sign pattern. **(b)** Strongest edges ($|W_{ij}|>1.4$) of the deployed W, drawn as a directed graph from factors (left) to action gates (right); line width $\propto |W_{ij}|$, solid edges are in the hand-built prior, dashed edges are learned *de novo*.

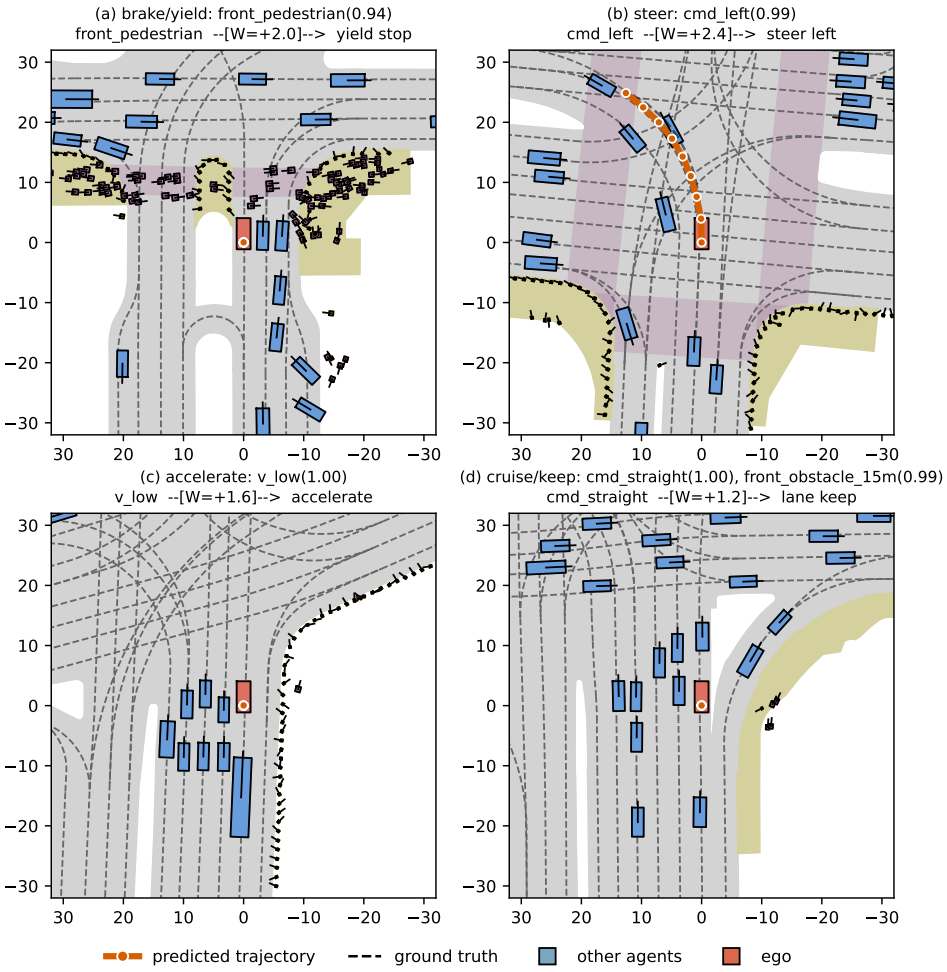


Figure 6: **All four read-out scenes** of Figure 3, same conventions.

per-factor coupling weaker by 4–17× and alignment at ≈ 0, indistinguishable from its own permutation null.

For factor k we form the counterfactual policy by overriding the factor probability, $\mathrm{do}(\hat{f}_k{=}v)$ for $v \in \{0, 1\}$, and propagating through Eqs. (1)–(2). We measure (i) the gate response $\Delta a = a^{(1)} - a^{(0)}$, correlated with the prior column $W_0[k, \cdot]$, and (ii) the behavioral response as the change in expected longitudinal progress of the *continuous* anchor distribution $\mathrm{softmax}(\ell)$. Significance is a bootstrap 95% CI over signed factors plus a permutation test. The continuous measurement is essential, because an estimator that intervenes and then reads the $\arg\max$-selected trajectory measures a quantized signal that inverts the apparent sign of small interventions (on our model it produced a spurious *negative* correlation), whereas the continuous estimator yields the positive effect of §5.3.

When the $\arg\max$ estimator inverts: a margin condition. This concerns any readout that reports the top-1 anchor; the deployed policy emits a top-50 softmax blend (§3.1), which responds continuously, and the top-1 flip rates quoted in the paper are summaries of that blend, not its output. Write the pre-intervention anchor scores as $z \in \mathbb{R}^K$ with top index $t = \arg\max_k z_k$, and let an intervention perturb them to $z + \varepsilon u$ for a direction u and magnitude $\varepsilon > 0$. The selected anchor changes iff ε exceeds a threshold set by the runner-up gaps,

$$\varepsilon^\star = \min_{k:\, u_k > u_t} \frac{z_t - z_k}{u_k - u_t},$$

and a sufficient condition for the $\arg\max$ to stay *frozen* ($\Delta{=}0$) is $\varepsilon \left(\max_k u_k - \min_k u_k\right) < z_t - z_{(2)}$, with $z_{(2)}$ the second-largest score. We verified both directions numerically over 26,400 randomized configurations with zero violations. This partitions a small intervention into three regimes.

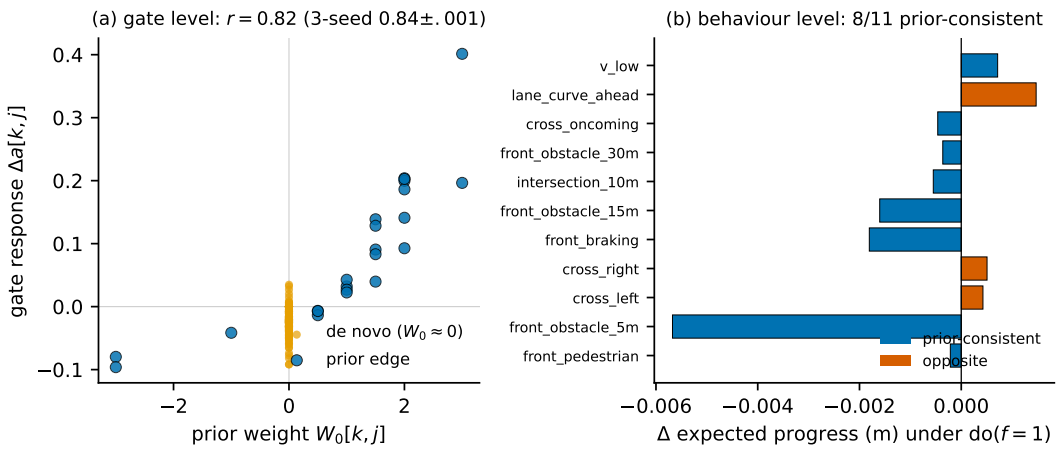


Figure 7: do-interventions at two levels. **(a) Gate-level prior-alignment** (deployed model): prior graph weight $W_0[k, j]$ vs. the measured gate response $\Delta a[k, j]$ (mean change in gate j under $\mathrm{do}(\hat{f}_k{=}1)$ vs. $\mathrm{do}(\hat{f}_k{=}0)$) over all $25{\times}8$ edges; prior edges (blue) move in the prior-consistent direction, de-novo edges (orange, $W_0{\approx}0$) cluster near zero (Pearson $r{=}0.82$; the in-panel three-seed figure of 0.84 is the earlier estimate, superseded by 0.815 ± 0.005 over 11 checkpoints in Table 2b). **(b) Behaviour level:** change in expected forward progress of the continuous anchor distribution under $\mathrm{do}(f_k{=}1)$, per signed-longitudinal factor. Panel (a) certifies *prior survival*, not faithfulness; panel (b) is the non-tautological test (§5.3).

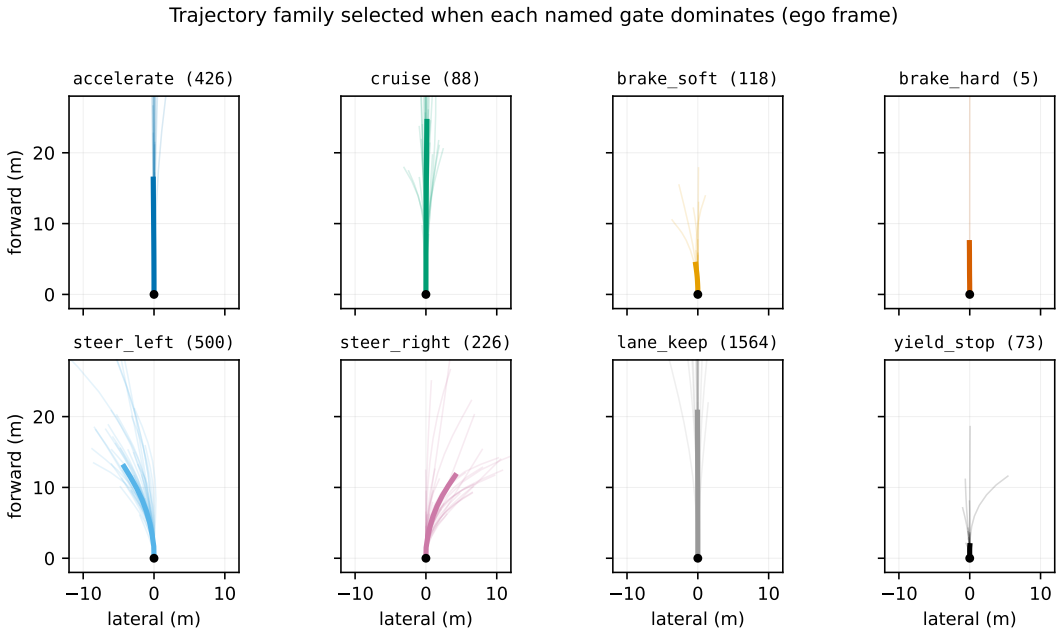


Figure 8: Behavioral signature of each named gate. For $3{,}000$ held-out scenes we group the deployed model's *selected* trajectory by its dominant gate $\arg\max_j G_j$ and plot the ego-frame family (40 sampled paths, bold mean; count in parentheses; `brake_hard` is rare in nominal driving, $n{=}5$).

A *frozen* regime ($\varepsilon < \varepsilon^\star$) is one where the $\arg\max$ estimator reports exactly zero. In a *transition* regime the selected anchor flips and the measured sign is set by whichever coordinate crosses first, so a sub-threshold, gate-mediated effect can register with the *wrong* sign (the inversion rate peaks near 0.42 under weak coupling). In a *drift* regime where the sign is correct but the magnitude is inflated 5.8–$7.7\times$. The continuous-distribution estimator, the expectation over $\mathrm{softmax}(\ell)$, is sign-correct in all $26{,}400$ cases with no inflation, which is why we adopt it. At our deployed operating point,

Table 7: **Representative perception→intent→behaviour chains read directly off the deployed factor-to-action graph W.** Each row gives an active factor, the gate(s) it most strongly drives (signed learned W_{ij}), and the induced behaviour.

Scenario	active factor	dominant gate(s) (W_{ij})	induced behaviour
Longitudinal / safety (prior-seeded)			
pedestrian ahead	`front_pedestrian`	`accelerate` (-3.80), `yield_stop` ($+2.04$)	decelerate & yield
close obstacle	`front_obstacle_5m`	`brake_hard` ($+3.01$), `accelerate` (-3.00)	hard brake
lead vehicle braking	`front_braking`	`brake_soft` ($+1.50$)	ease off throttle
stopped, lane clear	`v_low`	`accelerate` ($+1.58$)	resume motion
Lateral / command (prior-seeded)			
crossing agent (L/R)	`cross_left` / `cross_right`	`yield_stop` ($+1.97$/$+1.90$)	yield at junction
turn commanded	`cmd_right` / `cmd_left`	`steer_right` ($+2.55$)/`steer_left` ($+2.38$)	execute turn
Ego-state (learned de novo, absent from prior W_0)			
ego mid-turn	`yaw_rate_high`	`lane_keep` (-1.65), `cruise` (-1.59), `brake_hard` (-1.48)	commit to turn; do not hold lane or brake

single-factor corrections flip the top anchor in only 3.8–16.7% of scenes (per factor; the pooled rate is 6.1%, §5.4), i.e. 83–96% of scenes sit in the frozen regime, which both explains the spurious *negative* $\arg\max$ correlation we first observed and confirms that the continuous estimator's millimetre-scale readings are the correct estimate, not an absence of effect. Raising the gate weight at inference, which is free, amplifies the concept-level correction of §5.4 from a few centimetres to 0.24–0.56 m of endpoint displacement across the missed-factor set.

Selectivity. Setting a factor to zero (n=1,024 scenes) displaces the trajectory endpoint by 2.27 cm on average when the factor is active in the scene and by 0.11 cm when it is inactive, a ratio of 21× overall and 21–57× per factor; the symmetric $\mathrm{do}(1)$-vs-$\mathrm{do}(0)$ contrast moves the endpoint 6.8 cm on average and 17 cm for the strongest factor. Four rarely active factors have natural effects near zero, and `right_lane_clear` moves the trajectory whether active or not (ratio ≈1). The ratios are point estimates over a near-zero denominator and carry no per-factor CI.

Why each gate has the sign it has, including `cruise`. To first order, raising gate j by a step s_j changes the expected path length by $s_j \,\mathrm{Cov}_p\big(\mathrm{pathlen}, \sum_m G_m[:, j]\big)$, the covariance taken under the softmax p over the top-50 filtered anchors, i.e. among the candidates the model actually chooses between. All eight gates sit at low base values (0.03–0.08), so the steps are nearly equal; the covariances are all negative, which is the common mode by which raising any gate shortens the plan, and `cruise`'s is the most negative (-0.032 against -0.011 to -0.027), which is why it slows the car most. Over the eight gates the first-order prediction matches the measured lateral effects at Pearson 0.86–0.88 (p≈0.004–0.006) at every amplitude tested. The longitudinal agreement is marginal at c=1 (0.71, p=0.049) and not significant at c=4, 6, 8 (0.63, 0.53, 0.47), so the model explains the lateral gates and the specific `cruise` anomaly, not all eight longitudinal effects. A first attempt that correlated the readout columns with path length over all K anchors predicted the ordering at only Spearman 0.45 and put `cruise` on the wrong side; it fails because never-competitive anchors carry no weight.

Why concept interventions are actuators here: a structural argument. Write $\ell = W^\top \hat{f} + b$ for the gate logits of a scene. Under $\mathrm{do}(\hat{f}_k{=}1)$ against $\mathrm{do}(\hat{f}_k{=}0)$, $\Delta\ell_j = W_{kj}$ exactly, for every scene, so $\Delta a_j = \sigma(\ell_j + W_{kj}) - \sigma(\ell_j)$ has the sign of W_{kj} in every scene, and the perturbation of every metric logit is $G_m \Delta a$, a fixed linear image of an 8-vector whose direction is set by $W_{k,\cdot}$ and whose only scene dependence is the eight sigmoid slopes $\sigma'(\ell_j)$. Nothing in this chain depends on whether factor k is true of the scene; the pre-intervention $\hat{f}_k$ does not enter the $\mathrm{do}(1)$-vs-$\mathrm{do}(0)$ contrast

at all, and for the correction of §5.4, $\mathrm{do}(\hat{f}_k{=}1)$ against the natural $\hat{f}_k$, a missed true factor and a predicted-false factor both have $\hat{f}_k{\approx}0$. Numerically, on the $4{,}096$ scenes of the shared protocol with the deployed gates as ℓ, the per-scene Δa has median cosine 0.954 to its scene-mean direction over all 25 factors and 0.996 (5th percentile 0.80) over the seven rare safety factors, while its norm varies with the slopes (median coefficient of variation 0.57; slope 5th–95th percentiles 0.004–0.18, median 0.038). The intervention thus pushes a scene-independent direction with a scene-dependent gain; the top-50 softmax and the drivable filter can move the emitted trajectory by different amounts per scene, but cannot make the push depend on the truth of the assertion. An independent computation at the anchor-logit level (the exact two-arm difference of ℓ over $K{=}8192$ anchors, $1{,}024$ scenes) gives median cosines of 0.993, 0.994 and 0.993 for the three additive ablation seeds. This predicts the matched-false null below and the 52% pooled correct-direction rate that no amplitude improves (`data/actuator_check_20260905.json` in the supplement).

Standard test-time intervention, and its power. The protocol of the concept-bottleneck literature replaces the predicted concept vector by ground truth and scores against no intervention. Done for all 25 factors on all `navtest` scenes with the deployed model (1.4 factors flipped per scene on average, 72% of scenes touched; $12{,}082$ paired scenes), it changes EPDMS by -0.00008 $[-0.00045, +0.00030]$ and removes 3 while introducing 2 of 234 at-fault collisions. A control that assigns each scene another scene's ground-truth vector changes it by -0.00051 $[-0.00110, +0.00008]$ (removes 2, introduces 8), and true minus permuted is $+0.00035$ $[-0.00021, +0.00093]$ (collision verdicts better under truth in 8 scenes, under the permutation in 2, $p{=}0.11$), with a minimum detectable effect of 0.0008 at 80% power. Split by direction, turning off false positives is the marginally favourable half and turning on misses the marginally unfavourable one. **The same protocol on the repaired detector.** The run above uses the original factor heads and the original labels, so a natural objection is that it injects assertions that do not mean what their names say (App. F). Re-running all three arms on the repaired detector with the corrected labels answers it. The corrected labels are a substantially larger intervention, flipping 1.8 factors per scene against 1.4 and touching 82% of scenes against 72%, and they halve the missed-safety-factor scenes from $1{,}598$ to 796; `front_obstacle_5m` goes from 35 positives to 859, of which 93.8% are triggered by a vehicle rather than by the cones and roadside objects the original definition admitted. The score still does not move. Over the $12{,}082$ paired scenes, ground-truth override changes EPDMS by -0.0002 $[-0.0006, +0.0002]$ against no intervention (minimum detectable effect 0.0006), the scene-permuted control by -0.0003 $[-0.0010, +0.0004]$, and true minus permuted is $+0.0001$ $[-0.0005, +0.0008]$; at $c{=}4$ the override reads -0.0007 $[-0.0016, +0.0002]$. A no-op override reproduces the base run to the digit on every scene, which is the harness control. The matched-false test below was not re-run on the repaired detector. For the matched-false test of §5.4 at $c{=}4$, the paired standard error of 0.0014 gives a minimum detectable effect of 0.0039, about six extra repaired collisions; on the path change the paired difference between correcting the true factor and asserting a matched false one is $+0.04$ mm $[-2.7, +2.5]$ (true arm -2.9 mm, 53.1% decelerating; false arm -3.0 mm, 52.4%).

The multiplicative control: FiLM does not restore the scene, and the read-out is why. Six checkpoints of the SL-only ablation recipe, three FiLM and three additive, were evaluated through the official simulator on the same $1{,}598$ missed-factor scenes with the same true assertions and the same matched-false assignment, under a rule fixed before the runs (H1 iff the FiLM family's true-minus-matched-false CI excludes zero and is positive, or two of three seeds do, with all additive seeds containing zero). Per seed, true minus matched-false is $+0.0007$ $[-0.0021, +0.0038]$, -0.0010 $[-0.0039, +0.0018]$, -0.0010 $[-0.0037, +0.0017]$ for FiLM and -0.0013, -0.0002, -0.0004 for additive, all CIs containing zero, with 0–2 collision flips per arm: H0. Phase-0 checks passed on all six (the FiLM path is live, toggling it moves the anchor logits by 6–8, and on additive checkpoints the flag is a no-op to the last bit); the FiLM family's base score on this subset is 0.0045 above the additive family's (1.7 pooled seed s.d., $p{\approx}0.11$), which cancels in the paired contrast. The mechanism carries a control at each step. The multiplicative path carries 62–75% of the anchor-logit perturbation, against 0 on additive checkpoints, and inside it the scale term γ dominates (96%). That term is scene-specific in latent space (direction consistency across scenes 0.28–0.30) but scene-generic after the metric heads (0.99 for six of seven factors). The heads do not flatten everything, since a random latent perturbation of the same norm keeps consistency 0.03–0.12, but they map a pure rescaling of the latent to one direction (0.993), and the FiLM multiplier is nearly

constant across scenes (0.90–0.96), so the term acts as a rescaling. `front_pedestrian` is the one factor whose FiLM term stays scene-specific at the logit level, and not reproducibly (0.43, 0.98, 0.89 across seeds). The behavioural split by ground-truth value (E4 estimator) shows no family separation either. The amended prescription is that multiplicative coupling is necessary but not sufficient, and the modulation must vary across scenes and be anisotropic, or the scene-specificity it creates is removed at the read-out.

The simulator test of factor correction, and its controls. On the 1,598 probe-sample scenes in which the deployed detector missed a ground-truth safety factor (1,769 factor–scene pairs), the corrected trajectory was scored by the official simulator against the uncorrected one, both through the identical agent, at c=4 and c=64. Four override arms were run: the true correction; the same factor sets given to the wrong scenes (shuffled); the same multiset, per-scene count and mean gate edge assigned so that every assertion is false (matched-false); and any false factor (unmatched-false), which is confounded because restricting to false factors selects edges 2.7× stronger and is reported only for that reason. At c=4: 53 base collisions, none removed and 2 introduced by the true correction, 1 removed and 2 introduced by the shuffled arm, none either way by the matched-false arm. At c=64: 126 base collisions, 40 removed and 35 introduced (true), 32/31 (shuffled), 37/34 (matched-false). Every properly matched paired contrast contains zero (true minus matched-false: -0.0009 $[-0.0038, +0.0018]$ at c=4, $+0.0029$ $[-0.0107, +0.0163]$ at c=64; McNemar p=0.50 and 0.92). On a scene subset the devkit drops TFC where the previous frame is absent, which affects both arms identically, so only paired deltas are quoted. Pooled over the 1,769 pairs the correct-direction rate is 52% at every amplitude, because three of the seven rare factors (`cross_oncoming`, `front_obstacle_15m`, `front_braking`, 65% of the pairs) have no decelerating edge in the learned graph. Stratified by the pre-intervention edge strength, a split defined after seeing the pooled number, the rate is 49.0% (n=502, no edge), 51.8% (846), 54.1% (307) and 59.6% (114, strong) at c=1, with Spearman $\rho \approx 0.08$ ($p \approx 10^{-3}$) between edge strength and path shortening, so no edge means no effect. The case of §5.4 clears its collision only at c=64, where the base plan has already drifted more than a metre from the deployed one and the common mode of a gate intervention exceeds two metres, so that point is not a deployment.

F Factor taxonomy

The 25 supervised factors (Layer B), each a binary label computed geometrically from the logged scene; thresholds in ego-frame metres unless noted. The three `front_obstacle_{5,15,30}m` labels measure the clearance from the ego *front bumper* to a blocking object's rotated footprint. This corrects an originally-released definition that tested the object centre against the *rear-axle* frame ($0<x<D, |y|<2$). At D=5 m that rectangle sat mostly inside the ego body, so `front_obstacle_5m` fired only on cones and roadside objects (never a lead vehicle) and the 15/30 m variants were mis-named ($\approx$11/26 m ahead of the bumper). §5.4 reports the factor-accuracy repair this correction enables. A second, independent fault compounded it. An unused positive-class loss weight left the factor head under-trained on the 0.2–3%-prevalence safety classes, and restoring that weight is part of the repair of §5.4. That repair moves the macro AUC of the seven safety factors from 0.85 to 0.92 and the all-25 AUC from 0.93 to 0.95; recall stays imperfect for two rarer factors and the operating point shifts slightly toward recall (macro F_1 0.82 → 0.78). The `brake_hard` gate-share of `front_obstacle_5m` lifts 1.16× → 2.46×. Across the fitted graph (Table 9), the safety factors (`front_*`, `cross_*`) keep their strong prior-seeded edges, whereas the lane-, speed- and ego-state factors acquire their dominant influence *de novo*.

G Reranking and selection: algorithm, sweeps, and the finetuning control

What the rerank does per scene. The demotion test is an oriented-bounding-box check that replicates the simulator's collision typing. Across full `navtest` it flips NC 0 → 1 on 83 scenes, with the largest per-scene gains reaching ΔEPDMS +0.8, and every demotion is inspectable per scene (Table 10). On the reweighting side, all six head-reweightings we tried lower the score and vetoing the uncalibrated NC head is far worse, which is why only new information helps. The structural

Table 8: The 25 named scene factors.

#	name	definition	# / name / definition
0	front_obstacle_30m	blocking obj., bumper gap$\leq$30	13 cross_left: veh. $0<x<20$, $0<y<12$, $\Delta\theta\approx\frac{\pi}{2}$
1	front_obstacle_15m	bumper gap$\leq$15	14 cross_right: mirror of 13
2	front_obstacle_5m	bumper gap$\leq$5	15 cross_oncoming: $0\leq x\leq 30$, $\lvert y\rvert<5$, $\Delta\theta\approx\pi$
3	front_pedestrian	ped. $0<x<20$, $\lvert y\rvert<3$	16 v_low: ego speed $<$ 1 m/s
4	front_braking	lead veh. $<$20m, $v_x<-1$	17 v_med: $1\leq v<8$
5	left_lane_clear	no agent in left-lane box	18 v_high: $v\geq 8$
6	right_lane_clear	no agent in right-lane box	19 accel_pos: $a_{\text{fwd}}>0.5$
7	left_lane_exists	left lane valid	20 accel_neg: $a_{\text{fwd}}<-0.5$
8	right_lane_exists	right lane valid	21 yaw_rate_high: $\lvert\dot{\theta}\rvert>0.1$ rad/s
9	intersection_30m	roadblock connector in next 30m	22 cmd_left: nav. command
10	intersection_10m	next 10m	23 cmd_right
11	lane_curve_ahead	ego lane $\lvert\Delta\theta\rvert>0.2$ in 20m	24 cmd_straight
12	on_route	ego lane on route	

Table 9: **Quantitative profile of all 25 factors on the deployed model.** Columns: prevalence on full navtest; the *dominant* learned edge (action gate with the largest $|W_{ij}|$, signed; + excite, − inhibit); whether that edge was seeded in the prior W_0 or learned *de novo* (type); and the number of strong edges ($|W_{ij}|>0.5$) the factor drives.

factor	prev. %	dominant gate (W_{ij})	type	#strong
front_obstacle_30m	43.7	brake_soft (+0.61)	prior	1
front_obstacle_15m	17.2	brake_soft (+1.36)	prior	1
front_obstacle_5m	0.3	brake_hard (+3.01)	prior	3
front_pedestrian	2.9	accelerate (−3.80)	prior	5
front_braking	0.2	brake_soft (+1.50)	prior	1
left_lane_clear	85.8	cruise (−0.79)	de novo	5
right_lane_clear	84.8	cruise (−0.77)	de novo	5
left_lane_exists	39.7	lane_keep (−0.69)	de novo	4
right_lane_exists	40.7	cruise (−0.69)	de novo	4
intersection_30m	71.9	brake_soft (−0.60)	de novo	2
intersection_10m	53.6	brake_soft (+0.81)	prior	1
lane_curve_ahead	23.2	steer_right (−1.48)	de novo	7
on_route	94.0	cruise (−0.57)	de novo	3
cross_left	1.1	yield_stop (+1.97)	prior	1
cross_right	1.2	yield_stop (+1.90)	prior	1
cross_oncoming	7.3	steer_left (−1.43)	de novo	5
v_low	11.8	accelerate (+1.58)	prior	2
v_med	66.8	steer_left (−0.71)	de novo	3
v_high	21.3	lane_keep (−0.71)	de novo	3
accel_pos	22.6	cruise (−1.43)	de novo	7
accel_neg	25.7	cruise (−1.38)	de novo	8
yaw_rate_high	21.3	lane_keep (−1.65)	de novo	7
cmd_left	20.6	steer_left (+2.38)	prior	7
cmd_right	13.0	steer_right (+2.55)	prior	6
cmd_straight	66.4	lane_keep (+1.20)	prior	1

ceiling follows from EP being normalized against a privileged planner, which places the attainable range in the low-0.9s.

Selection-variant sweep and oracle comparison. Algorithm 1 gives the rerank in full; Table 11 then reads the selection policy as a Pareto probe. The deployed equal-weight metric-logit sum already sits at a local optimum of the reweighting family, and only injecting new information (the geometric predictor) helps. On a held-out subset the constant-velocity filter recovers 90% of an oracle that uses ground-truth agent futures (Table 11, right), confirming that most relevant collisions are predictable over the 4 s horizon.

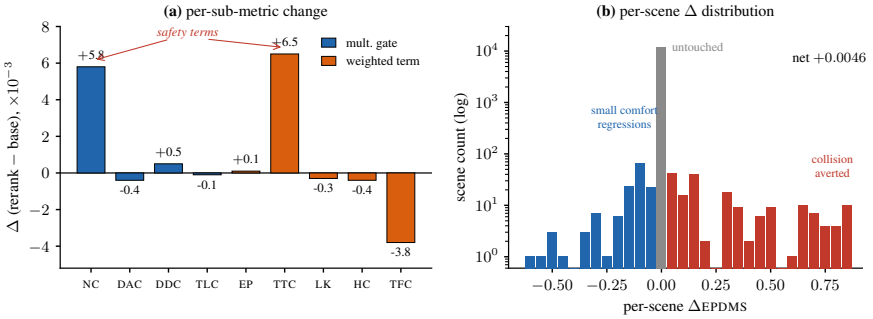
Rescue case studies. Fig. 9 shows three of the 83 full-navtest scenes where NC flips $0 \rightarrow 1$ under the legitimate rerank. The base trajectory drives the ego footprint into a crossing or stopped agent, while the rerank, selecting a different anchor from the *same* forward pass, keeps it clear, a genuine spatio-temporal avoidance that stays auditable (the demoted candidates and the geometric reason are inspectable per scene).

Algorithm 1 Inference-time at-fault collision rerank

Forward pass → anchor logits $\ell \in \mathbb{R}^K$, residuals r; take the top-N=200 candidates $\mathcal{C}$ by ℓ
Predict each logged agent over the 4 s horizon by constant velocity from the current frame
for $k \in \mathcal{C}$ **do**
    if trajectory $(A_k + r_k)$ has an at-fault collision (stopped-track or active-front) **then**
        $\ell_k \leftarrow -\infty$ ▷ demote
    end if
end for
if all $\mathcal{C}$ demoted **then** restore ℓ ▷ safety fallback
end if
return top-k softmax-blended trajectory under updated ℓ

Table 10: **Per-sub-metric effect of the collision-aware rerank** (full `navtest`). Multiplicative gates (NC,DAC,DDC,TLC) × weighted terms (EP,TTC,LK,HC,TFC); the gain is concentrated in the safety terms with no progress cost. Below, the same deltas as a chart ($\times 10^{-3}$) alongside the per-scene ΔEPDMS histogram (log count). Many small comfort regressions are traded for a few large safety gains. The table's $+.0045$ and the chart's $+0.0046$ round the same full-precision net of $+0.00456$.

	multiplicative gates ↑					weighted terms ↑				
	NC	DAC	DDC	TLC	EP	TTC	LK	HC	TFC	EPDMS↑
base	.9817	**.9899**	.9913	**.9977**	.8633	.9753	**.9641**	**.9812**	**.8716**	.8985
+rerank	**.9874**	.9895	**.9918**	.9976	**.8634**	**.9817**	.9637	.9808	.8678	**.9030**
Δ	**+.0058**	−.0004	+.0005	−.0001	+.0001	**+.0065**	−.0003	−.0004	−.0038	**+.0045**



Why selection, not generation: the finetuning control. Finetuning the converged model with a differentiable at-fault loss (App. H) *reduces* the subset score, and a control isolates why. Three finetunes, (i) with the loss, (ii) loss disabled and (iii) loss + EMA shadow + comfort, degrade by -0.015, -0.007, and -0.0045 (Table 12). That the loss-disabled control *also* degrades (-0.007) shows roughly half the loss is independent of the objective and attributable to finetuning a converged, EMA-averaged checkpoint at all, because the training loss and closed-loop EPDMS are misaligned near $\theta^\star$, so a step that lowers the loss can worsen the score. The EMA-shadow run, staying nearer $\theta^\star$, recovers most of it (subset 0.9145, highest NC 0.9863 vs. 0.9827) but does not turn it positive.

H Differentiable At-Fault Collision Loss

For a candidate trajectory (anchor+residual), at step t the ego footprint centre is $c = (x, y) + d_{\text{rac}}(\cos\theta, \sin\theta)$. For agent m at p_m we rotate the relative position into the ego frame to obtain (lon, lat) and form an elliptical safety region with semi-axes $a = \frac{1}{2}(L_{\text{ego}}+L_m)+\delta$, $b = \frac{1}{2}(W_{\text{ego}}+W_m)+\delta$:

$$e = \left(\frac{\text{lon}}{a}\right)^2 + \left(\frac{\text{lat}}{b}\right)^2, \qquad \phi = \text{relu}(1-e)^2 \in [0,1]. \tag{6}$$

The ellipse encodes the vehicle aspect ratio, so a laterally offset agent does not trigger the penalty as a circular proxy would; summing the two extents assumes the agent is axis-aligned with the ego, which makes this a bounding approximation rather than an exact footprint test. An at-fault gate $w \in \{0, 1\}$

Table 11: Inference-time selection variants (held-out subset, base = 0.9190). Reweighting the learned metric logits, or vetoing by the *uncalibrated* NC head, does not help; only the geometric at-fault rerank improves the score, consistent with the heads being rank- not probability-calibrated (§3.2).

selection variant	EPDMS (subset) ↑
base (equal-weight metric-logit sum)	0.9190
reweight EP × 1.5 / 2 / 3	0.9172 / 0.9155 / 0.9101
reweight TTC × 2	0.9155
reweight NC × 2	0.9133
reweight EP × 2 + TTC × 2	0.9162
hard veto by σ(NC-logit)<0.5 (uncalibrated)	0.8542
geometric at-fault collision rerank (ours)	**0.9249**

Configuration	EPDMS (subset) ↑
base	0.9190
+ rerank (const.-velocity; legit.)	**0.9249**
+ rerank (oracle; GT futures)	0.9256

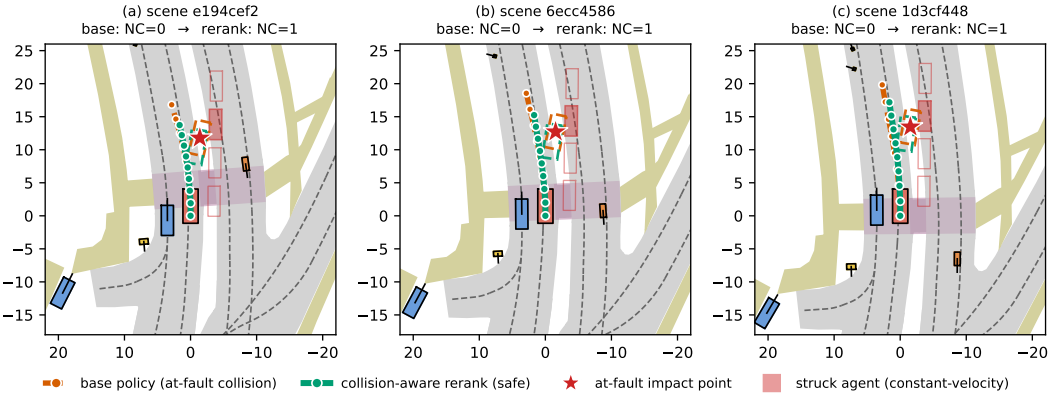


Figure 9: **Collision-rescue case studies** (full `navtest`). Base trajectory (vermillion dashed) reaches the at-fault impact (⋆); the rerank trajectory (green) avoids it; dashed rectangles are the ego footprint at impact (navsim convention: ego-left on the left). In each scene the rerank flips the at-fault indicator from NC=0 (base collides) to NC=1 (safety recovered).

Table 12: Finetuning the *converged* checkpoint with the collision loss degrades the subset score; a loss-disabled control also degrades, so checkpoint perturbation and the objective each account for roughly half; EMA-shadow recovers most of the loss but does not turn it positive.

finetune of the converged model	Δ EPDMS (subset)
collision loss (λ=1)	−0.0151
loss disabled (control)	−0.0071
collision loss + EMA shadow + comfort	−0.0045

(a detached weight) replicates the simulator's collision typing: w=1 for stopped-track or active-front contacts with a moving ego, w=0 for rear contacts or an at-rest ego. The per-candidate penalty $P = \sum_{t,m} w\,\phi$ is combined over anchors with the selection softmax, $\mathcal{L}_{\text{coll}} = \sum_k \text{softmax}(\ell)_k\, P_k$, mirroring the off-road loss. Its gradient $\partial\phi/\partial c \propto -\text{relu}(1-e)(c-p_m)$ moves the footprint away from the agent, and ϕ=0 outside the ellipse gives *zero* gradient where there is no conflict, so progress and comfort are preserved away from collisions. We verified the gradient sign (a step reduces the penalty and increases clearance), the locality, and the at-fault gating in unit tests, and confirmed that the soft penalty ranks true at-fault collisions on held-out scenes.