Plans Are Not Interventions: Measuring and Repairing Plan Leakage in Driving World Models

Sahil Apage
Veermata Jijabai Technological Institute
sahilvapage@gmail.com

Medhansh Abhilash
Veermata Jijabai Technological Institute
mmabhilash_b25@el.vjti.ac.in

Abstract: Self-driving systems increasingly use learned world models to judge candidate plans and train policies. This requires predicting what happens if the ego executes a plan, yet a model trained on driving logs learns what happened when drivers chose that plan. These differ because drivers react to traffic: slowing down in the logs usually means the car ahead slowed down, so the model learns to infer other cars' motion from the plan. We call this the *plan leak* and measure it by changing only the plan and checking whether cars that cannot react move. In OmniNWM, a 4D driving world model, a plan that drives into the car ahead, even one stopped at a red light, moves that car away by as much as the ego advances, while parked cars barely move, so the crash never appears. Policies trained in a leaky model crash up to 2.8 times as often, and large leaks make planners brake unnecessarily. The cause is how the model sees the plan, not missing data: training on counterfactual plans barely helps, letting each step see only the ego's past removes the leak, and averaging over past-only plans repairs frozen models, bringing OmniNWM's crash back in 16 of 18 scenes.

Keywords: Autonomous Driving, World Models, Causal Inference, Planning, Evaluation

1 Introduction

World models increasingly support driving decisions: they score candidate plans, supply rewards, and serve as simulators for training policies [1, 2, 3, 4, 5, 6]. Whether a generated world is useful for decisions is an open question [7, 8]. It can fail in a way no image, depth or occupancy metric detects, because the failure lies in the question the model answers. A planner needs to know what happens *if it executes* a plan, $p(\text{future} \mid \text{history}, \text{do}(\text{plan}))$. A model trained on logs learns $p(\text{future} \mid \text{history}, \text{plan})$. The two differ whenever the plan carries information about the rest of the world, and in driving it always does: a driver slows down because the car ahead slows down. So when a planner proposes to speed up, a plan-conditioned model moves the car ahead out of the way (Fig. 1).

This is known as causal confusion in imitation learning [9] and as self-delusion in sequence models [10, 11]; in motion forecasting, conditioning other agents on the ego's full future [12] is acausal: MotionLM shows a lead car predicted to slow because the car behind it brakes [13], and IBP proposes interventional prediction [14]. Yet driving world models condition on the whole future plan or on the ego pose of every generated frame [15, 16, 6, 17], some so that other agents react to it [18]; OmniNWM treats them as "reactive latent variables" whose yielding is the most likely response in the logs [6]. AD-R1 observed that world models imagine safe outcomes for unsafe plans and blamed missing crash data [2]. We find a different cause and cure:

- **A leak test and a finding.** OmniNWM moves the car ahead metre for metre with the plan (§3).
- **The cause.** Where the plan enters the model, not the training data (§4).
- **The cost.** Missed crashes, riskier trained policies, unnecessary braking (§5).
- **Repairs.** Causal conditioning, g-computation, and past-only queries (§6).

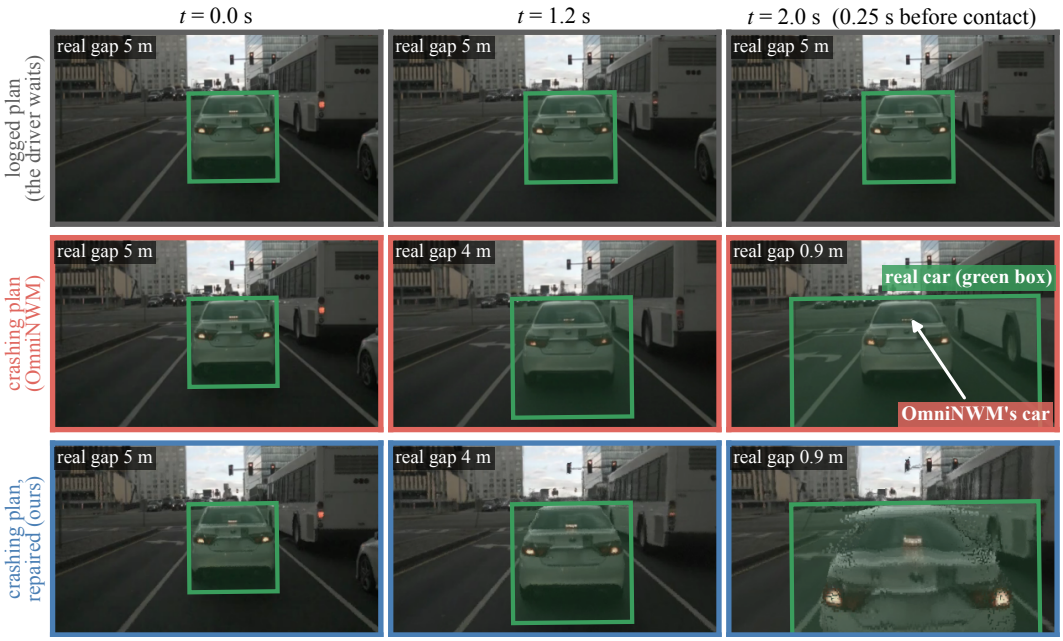


Figure 1: **OmniNWM moves the car ahead out of the ego's way; a repair puts it back.** Generated front camera, one held-out nuScenes scene and seed; green: the real car's box. Logged plan: the car sits in the box. Crashing plan: just before the ego reaches the real car, OmniNWM's car is metres ahead, while the bus beside it is passed. Repaired: OmniNWM's world for a plan that reveals only the past, so no car can respond to the crash, seen from the crashing plan's cameras (speckle: re-rendering holes); the car fills the box (§6).

2 The plan leak and how to measure it

Why logs leak. Let D_e and D_a be the future displacements of the ego and of another agent. A model fitted to logs learns, to first order, the slope $\beta = \mathrm{Cov}(D_a, D_e \mid h)/\mathrm{Var}(D_e \mid h)$ given the history h. When the ego follows a car, its motion is caused by the lead's, so $\beta > 0$, although within seconds the lead does not react to the car behind it: the true, interventional slope is zero. This is the time-varying confounding of Robins [19]. How much a model absorbs depends on which plan steps it sees (Fig. 2).

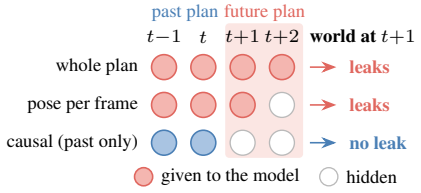


Figure 2: **What the model is given when it generates the next moment.** Future plan steps encode how the driver will react to the world, so a model that sees them can read the world off the plan. A causal model sees only the past plan.

The leak test. We give the model the logged plan and a counterfactual plan that differs only in the ego's motion, and measure the *plan leak* β: how far the car ahead moves in the generated future per metre that the plan moves the ego further, read when the counterfactual ego truly reaches the real car. At $\beta = 0$ the ego reaches the car, as in reality; at $\beta = 1$ the gap never closes. Parked cars, which no plan can affect, are the control.

3 A 4D video world model leaks the plan

Setup. OmniNWM [6] generates panoramic video with metric depth and semantics, conditioned on the ego's camera pose in every frame. We use the released weights on held-out nuScenes [20] scenes with a real car up to 25 m ahead in the ego lane (first 18 in a fixed random order). The counterfactual plan adds the smallest constant acceleration that drives the ego at least 1 m into the real car. We read the generated car from OmniNWM's own depth and semantics; all tests were preregistered (App. F).

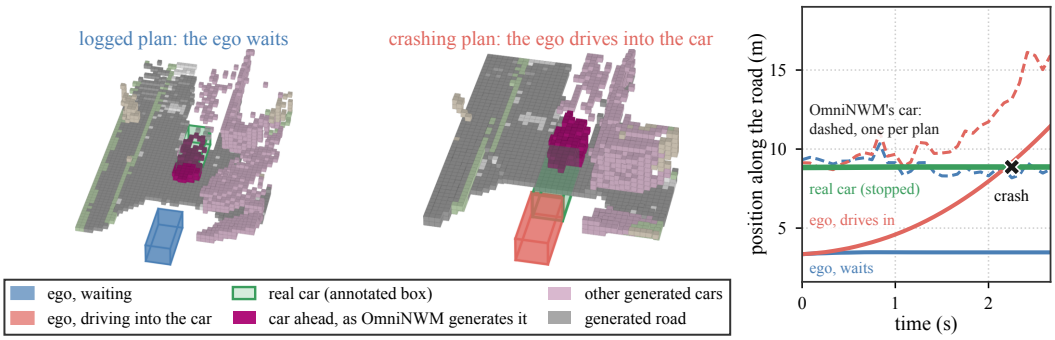


Figure 3: **Inside OmniNWM's generated world** at the real crash (Fig. 1's scene in 3D, from its bias-corrected depth and semantics; App. A): the crashing ego sits in the real car's box, while OmniNWM's car stays ahead (left) and pulls away (right, dashed).

The car ahead moves with the plan, and only the car ahead. Across scenes $\beta = 1.02$ (95% CI 0.98 to 1.08); an independent re-implementation gives 1.00 (App. D). In 3D, the crashing ego occupies the real car's position on an empty road (Fig. 3). The generated ego does move: visual odometry on the generated frames recovers 86% of the commanded motion. Parked cars move 0.23 m per metre of plan change, about what this imperfect execution alone produces, against a full metre for the car ahead (Fig. 4a). The leak is linear (slope 0.94 over braking, mild and crashing plans) and works both ways: braking, well within the training distribution, pulls the car ahead towards the ego in all eight scenes where the ego moves. A real car ahead reacts to neither plan.

Its crash check misses the collisions. A check for generated vehicles inside the ego's footprint misses all 18 crashes; OmniNWM's released collision rule alarms on half of the safe logged plans.

The pattern is general (Fig. 4a). Vista [17] also keeps the gap ($\beta = 1.04$, 30 scenes), but executes under half of the plan change (supporting evidence only). Occupancy models leak too: SWTC [15] and DOME [16] near 1, I^2-World [21] about half; parked cars stay below 0.1.

4 Where the leak comes from

Where the plan enters decides the leak. Object world models trained on nuScenes vehicle tracks share network, data and budget and differ only in which plan steps they see (Fig. 4b; App. Fig. 11a). The whole-plan model leaks 0.64, matching the coupling in the training logs (0.67) with nothing fitted; seeing only the next 0.5 s, as per-frame poses do, leaks as much. The causal model, whose world at $t{+}1$ sees the ego only up to t, does not leak ($|\beta| \leq 0.13$ under two plan families) yet keeps the real reaction of a car following the ego; in a simulator with known effects it recovers the true follower effect. Accuracy rewards the leak: logged-plan error is lowest for the whole-plan model. Occupancy models leak each kind of car as strongly as the logs couple it (App. Fig. 10).

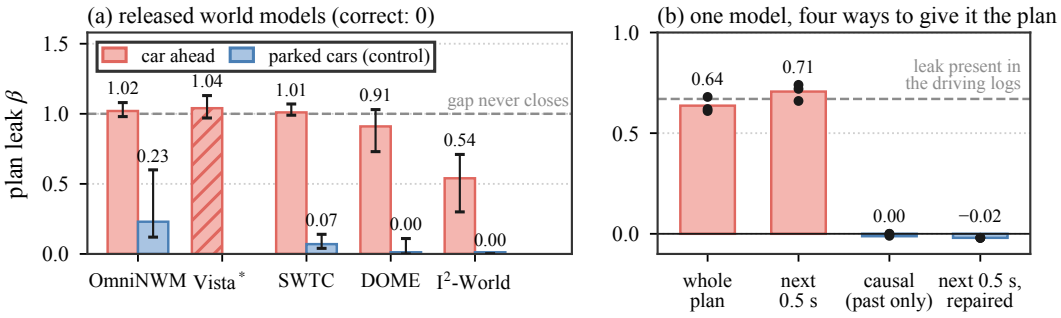


Figure 4: **The plan leak** β: metres the car ahead moves per metre the plan moves the ego. (a) Released models; parked cars are the control (95% CI; *Vista executes under half the plan change). (b) Object models identical except for the plan input; repaired: g-computation (dots: seeds).

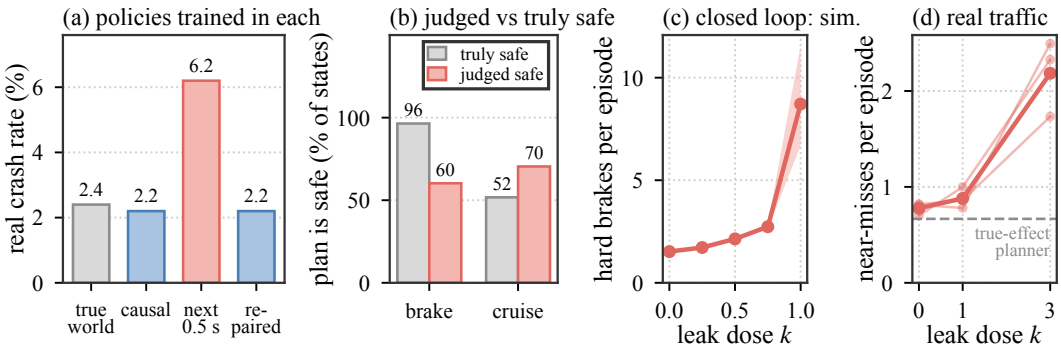


Figure 5: **What the leak costs.** (a) Real crash rate of policies trained in each model (repaired: g-computation; 3×3 seeds). (b) Braking is truly safer than cruising, yet the whole-plan model judges it less safe. (c, d) Closed loop at leak dose k (0: causal, 1: whole plan); thin lines: seeds.

More counterfactual data does not remove it. Fine-tuning SWTC on counterfactual plans, as suggested by [2, 22], barely changes its leak: 0.83 to 0.77, or 0.73 with fully random plans.

5 What the leak costs

Crash checks. Extra ego motion closes the imagined gap by only $1 - \beta$, so near $\beta = 1$ every plan looks safe. On 653 held-out decisions, trusting SWTC's or DOME's verdict does not beat ignoring it; physics is best (App. B).

Policy training. Policies trained inside the next-0.5 s model crash $2.8\times$ as often on held-out real logs as those trained inside the causal one, at equal progress (6.2% vs 2.2%; Fig. 5a): they learn that speeding up makes the car ahead drive away. SWTC as reward does the same.

Closed-loop planning. A planner replans every 0.5 s, keeping plans whose predicted gap to every car ahead stays above a margin, in the simulator and on real nuScenes traffic (log replay, 150 scenes). To vary only the leak, it judges with the causal prediction plus k times (whole-plan − causal). Harm grows with k (Fig. 5c,d). In the simulator, the whole-plan model brakes hard four to eight times as often as the causal one, and 72% of these brakes are unnecessary: the leak inverts the order of plans, judging braking less safe than cruising (Fig. 5b), as braking would pull the car ahead back. On real traffic, near-misses (time to collision below 1.5 s) rise 1.7- to 3.3-fold at $k = 3$ for all seeds, and at the learned leak ($k = 1$) for one seed of three.

6 Repairs

Frozen models can be repaired at test time. g-computation steps a one-step model forward, showing the candidate plan only up to the present and averaging over past-only continuations; it removes the leak and the policy cost of the next-0.5 s model (Fig. 4b, 5a). A whole-plan video model can be repaired for detection only: OmniNWM, queried with plans that reveal only the past and viewed from the candidate's cameras, shows the crash (Fig. 1, bottom) and its check catches 16 of 18 crashes, but a planner built on such queries under-reacts to braking cars (App. C).

Recommendations. (1) Vary only the plan; report β with a parked-car control (App. E). (2) Condition causally: video models should place other agents from the ego's past, then render frame $t{+}1$. (3) Select models by closed-loop decisions against physics and a plan-blind baseline [23, 24], not by logged-plan accuracy. **Limitations.** nuScenes only; one validated video model (18 scenes); physics remains the best judge; no trained video-model fix.

References

[1] Y. Li, Y. Wang, Y. Liu, J. He, L. Fan, and Z. Zhang. End-to-end driving with online trajectory evaluation via BEV world model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[2] T. Yan, T. Tang, X. Gui, Y. Li, J. Zheng, W. Huang, L. Kong, W. Han, X. Zhou, X. Zhang, Y. Zhan, K. Zhan, C.-z. Xu, and J. Shen. AD-R1: Closed-loop reinforcement learning for end-to-end autonomous driving with impartial world models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. arXiv:2511.20325.

[3] H. Lin, C. Zheng, Y. Yang, et al. RoXDrive: Closed-loop reinforcement learning for end-to-end autonomous driving via action-faithful rollouts. *arXiv preprint arXiv:2609.36851*, 2026.

[4] Z. Yang, X. Jia, Q. Li, X. Yang, M. Yao, and J. Yan. Raw2Drive: Reinforcement learning with aligned world models for end-to-end autonomous driving (in CARLA v2). *arXiv preprint arXiv:2505.16394*, 2025.

[5] J. Pei, M. Lu, S. Ang, et al. World4Scorer: Outcome-grounded world modeling for autonomous driving. *arXiv preprint arXiv:2609.36438*, 2026.

[6] B. Li, Z. Ma, D. Du, B. Peng, Z. Liang, Z. Liu, X. Guo, Z. Zhu, C. Ma, Y. Jin, X. Jin, H. Zhao, and W. Zeng. OmniNWM: Omniscient driving navigation world models. In *European Conference on Computer Vision (ECCV)*, 2026.

[7] C. Oefinger, F. R. Schäfer, K. Moller, M. Piccinini, and J. Betz. Validate the dream before you trust its verdict: Admissibility for world-model simulators. *arXiv preprint arXiv:2607.07196*, 2026.

[8] J. Zhang, M. Jiang, N. Dai, T. Lu, et al. World-in-world: World models in a closed-loop world. In *International Conference on Learning Representations (ICLR)*, 2026.

[9] P. de Haan, D. Jayaraman, and S. Levine. Causal confusion in imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[10] P. A. Ortega, M. Kunesch, G. Delétang, T. Genewein, J. Grau-Moya, J. Veness, et al. Shaking the foundations: Delusions in sequence models for interaction and control. *arXiv preprint arXiv:2110.10819*, 2021.

[11] R. Douglas, J. Karwowski, C. Bae, A. Draguns, and V. Krakovna. Limitations of agents simulated by predictive models. *arXiv preprint arXiv:2402.05829*, 2024.

[12] E. Tolstaya, R. Mahjourian, C. Downey, B. Varadarajan, B. Sapp, and D. Anguelov. Identifying driver interactions via conditional behavior prediction. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[13] A. Seff, B. Cera, D. Chen, M. Ng, A. Zhou, N. Nayakanti, K. S. Refaat, R. Al-Rfou, and B. Sapp. MotionLM: Multi-agent motion forecasting as language modeling. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[14] C. Tang, W. Zhan, and M. Tomizuka. Interventional behavior prediction: Avoiding overly confident anticipation in interactive prediction. *arXiv preprint arXiv:2204.08665*, 2022.

[15] J. Du, Y. Zhao, Z. Guo, Y. Pan, W. Hou, Z. Hao, K. Zhan, and Q. Chen. SparseWorld-TC: Trajectory-conditioned sparse occupancy world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

[16] S. Gu, W. Yin, B. Jin, X. Guo, J. Wang, H. Li, Q. Zhang, and X. Long. DOME: Taming diffusion model into high-fidelity controllable occupancy world model. *arXiv preprint arXiv:2410.10429*, 2024.

[17] S. Gao, J. Yang, L. Chen, K. Chitta, Y. Qiu, A. Geiger, J. Zhang, and H. Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[18] T. Yan, H. Zheng, D. Chen, M. Qu, Y. Shen, L. Zhou, M. Tu, B. Wang, G. Chen, H. Ye, H. Sun, C.-z. Xu, and J. Shen. CausalDrive: Real-time causal world models for autonomous driving. *arXiv preprint arXiv:2606.15341*, 2026.

[19] J. Robins. A new approach to causal inference in mortality studies with a sustained exposure period: Application to control of the healthy worker survivor effect. *Mathematical Modelling*, 7:1393–1512, 1986.

[20] H. Caesar, V. Bankiti, A. H. Lang, et al. nuScenes: A multimodal dataset for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[21] Z. Liao, P. Wei, R. Zhang, S. Chen, H. Wang, and Z. Ren. I^2-World: Intra-inter tokenization for efficient dynamic 4D scene forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[22] J. Yang, K. Chitta, S. Gao, L. Chen, Y. Shao, X. Jia, H. Li, A. Geiger, X. Yue, and L. Chen. ReSim: Reliable world simulation for autonomous driving. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[23] D. Dauner, M. Hallgarten, A. Geiger, and K. Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *Conference on Robot Learning (CoRL)*, 2023.

[24] Z. Li, Z. Yu, S. Lan, J. Li, J. Kautz, T. Lu, and J. M. Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[25] Y. Shi, K. Jiang, Q. Meng, K. Wang, J. Wang, W. Sun, T. Wen, M. Yang, and D. Yang. COME: Adding scene-centric forecasting control to occupancy world model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[26] W. Sun, X. Lin, Y. Shi, C. Zhang, H. Wu, and S. Zheng. SparseDrive: End-to-end autonomous driving via sparse scene representation. *arXiv preprint arXiv:2405.19620*, 2024.

A More evidence from OmniNWM

All results in this section use OmniNWM's released weights on held-out nuScenes, one seed per plan, and read the generated car from OmniNWM's own generated depth and semantics. Figure 6 repeats the 3D occupancy view of Fig. 3 for a second scene.

Reaction or leak? OmniNWM restricts explicit control to the ego and treats the other agents as reactive latent variables, so that, for example, a truck yields to an ego that cuts in [6]. Cars that can see the ego and respond to it, such as a car behind or one being cut off, may truly react, and the causal model of §4 keeps such reactions. The car ahead in our test cannot respond within seconds, so its motion under the counterfactual plan is read from the plan, not caused by it.

OmniNWM's depth reads long. Its generated images are geometrically faithful: the real car's annotated box, projected into the generated camera, frames the generated car at the start of the rollouts (Fig. 1). Its generated depth, however, overestimates distance. On the real car's pixels in all 18 scenes (three frames each, logged plan), the nearest-surface depth fits $1.09\times$ the true distance $+0.9$ m, about 1.5 m too far at 7 m. A constant bias cancels in β, which compares the two plans at the same moment while the car keeps its distance: in the independent re-implementation (App. D), undoing the fitted bias leaves β at 1.00 (0.87 to 1.19), the dose slope at 0.94 and the crash check missing 18 of 18; a pure 24% scale correction also gives 1.00. The positions in Fig. 3 and Fig. 6 use the corrected depth.

Vista. Vista [17] keeps the car's gap as well ($\beta = 1.04$, 95% CI 0.97 to 1.13, 30 scenes; Fig. 7), but its camera realizes only about 0.44 of the commanded extra advance (0.48 at doubled guidance), so part of its leak may be an execution failure. Relative to its realized motion the car ahead still moves about 1.1 m per metre. Vista's authors show that it predicts the consequences of abnormal actions, such as a crossing car that waits for an ego forced through a junction [17]; the car ahead in our test, which cannot react, instead keeps its gap. We use Vista as supporting evidence only.

More scenes. Figure 8 shows four more scenes at the moment the crashing plan reaches the real car, including the scene with the weakest leak. Figure 9 shows every scene at once and the full dose response in one scene, and Table 1 lists every scene.

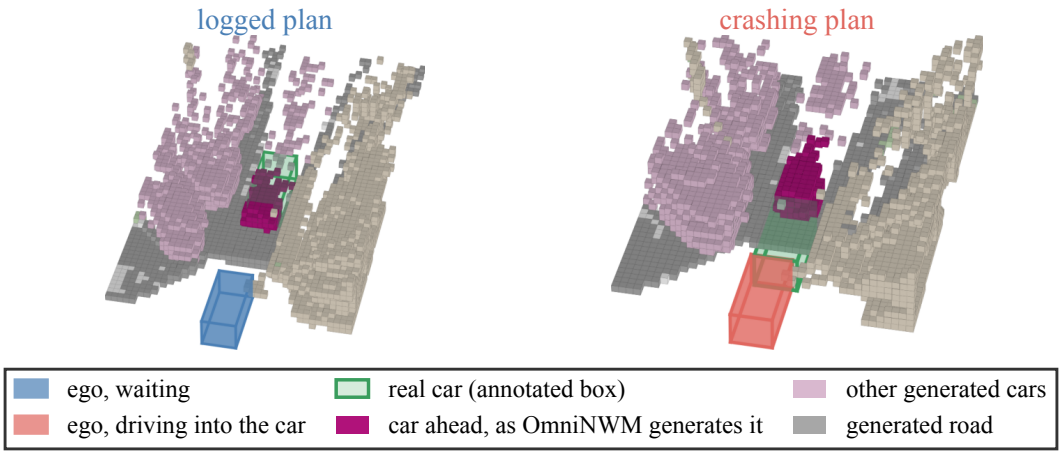


Figure 6: **3D occupancy, second scene** (scene-0097, a car stopped in traffic, at the real contact). As in Fig. 3, the generated car fills the real car's box under the logged plan and has moved ahead under the crashing plan, leaving the real car's position empty.



Figure 7: **Vista** at +2.0 s, logged against crashing plan, two scenes. The car ahead keeps its size under the crashing plan, but the camera also advances less than commanded.



Figure 8: **Four more scenes at the real contact.** Left: logged plan; right: crashing plan, same seed and same moment. Green: the real car's annotated box projected into the generated camera. In the first three scenes the generated car stays where it was under the logged plan, far ahead of the green box. The last row is the scene with the weakest leak ($\beta = 0.42$): there OmniNWM does render the bus close up, although the gap is still too large.

Table 1: **OmniNWM, every confirmatory scene.** a^*: extra acceleration of the crashing plan (m/s^2); t^*: real contact (s); Δs: extra ego advance at t^* (m); gap: generated car minus ego front at t^* (m; positive means the car moved away); ego moves: realized share of the commanded advance (visual odometry); self and plan-marginal: whether that check flags the crash; –: no valid readout. Scene-0100 holds two test windows.

scene	car ahead	a^*	t^*	Δs	β	gap	ego moves	self	plan-marg.
scene-0014	stopped	2.00	2.07	4.3	0.42	+2.3	0.65	miss	hit
scene-0095	stopped	2.50	2.25	6.3	0.74	+6.4	0.63	miss	hit
scene-0097	stopped	2.50	2.22	6.1	0.87	+5.7	0.64	miss	hit
scene-0100	stopped	2.00	2.28	5.2	1.94	+3.5	0.93	miss	hit
scene-0100	stopped	2.00	2.35	5.5	1.03	−1.3	0.97	miss	hit
scene-0556	moving	3.75	2.42	11.0	1.08	+18.2	0.58	miss	miss
scene-0563	moving	2.50	2.33	6.8	–	–	0.16	miss	hit
scene-0626	stopped	2.25	2.25	5.7	–	–	0.82	miss	hit
scene-0780	stopped	2.25	2.25	5.7	1.08	+3.8	0.94	miss	hit
scene-0781	stopped	2.25	2.25	5.7	0.99	−1.7	0.91	miss	hit
scene-0794	stopped	1.25	2.17	2.9	1.53	+4.8	1.10	miss	hit
scene-0797	moving	3.25	2.45	9.8	1.05	+19.3	0.76	miss	miss
scene-0798	moving	3.25	2.30	8.6	0.90	+2.6	0.87	miss	hit
scene-0907	stopped	1.50	2.22	3.7	1.42	+5.6	1.28	miss	hit
scene-1067	moving	2.50	2.33	6.8	1.24	+10.9	–	miss	hit
scene-1068	moving	2.25	2.17	5.3	1.01	+6.2	0.86	miss	hit
scene-1070	moving	4.00	2.42	11.7	0.98	+14.0	–	miss	hit
scene-1071	moving	3.00	2.42	8.8	1.00	+11.2	–	miss	hit

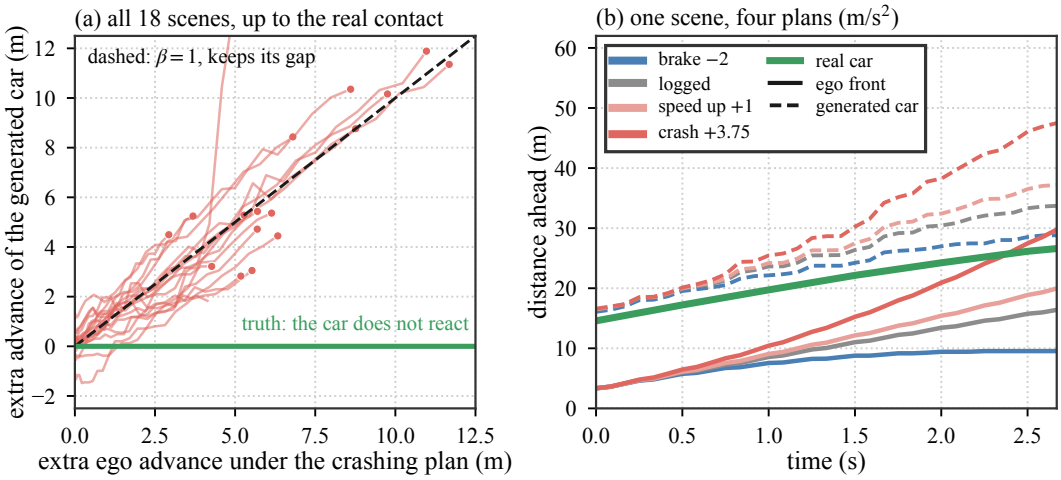


Figure 9: **Every scene, and every plan in one scene.** (a) For each of the 18 scenes, the extra distance the generated car advances under the crashing plan, against the extra distance the ego advances, from the start until the ego truly reaches the real car (dots). A model that respects the real world would stay on the green line; OmniNWM follows the dashed line, on which the gap never closes. One scene leaves the axes because of a readout jump. (b) A moving scene rendered under four plans. The generated car (dashed) moves with the ego (solid) in both directions: braking pulls it back towards the ego, accelerating pushes it away. The real car (green) is the same in every plan.

B Cause and cost in more detail

Leak by relation. Figure 10 measures the leak of three released occupancy models for every kind of nearby car (exploratory; 9 to 16 scenes per model) against the coupling of that car's motion with the ego's in the nuScenes training logs. Where the logs couple a car to the ego, the models leak; parked cars, which the logs do not couple, stay put. Consistent with this, though not proof of it, SWTC's own ablation gains 4 mIoU points when the logged rather than a planned future trajectory is given [15]. The leaks of the car behind and of oncoming cars depend on the estimator (below); the contrast between the car ahead and parked cars does not.

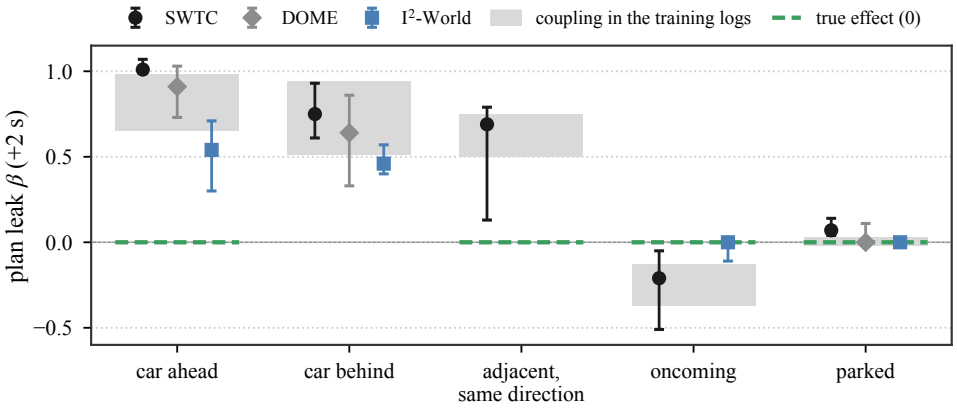


Figure 10: **The leak follows the coupling in the training logs.** Plan leak at +2 s of three occupancy world models by the relation of the car to the ego (median, 95% scene-bootstrap CI), against the band of slopes of that car's motion on the ego's in the training logs (with and without conditioning on the history). Dashed: the true effect of the plan within 2 s; a car behind may truly react, so its true effect is not zero.

Decision benchmark (653 held-out decisions, App. G). Reduction in crash rate against a plan-blind planner at matched progress: trusting SWTC's verdict on its own plan $+0.018$ (-0.006 to $+0.043$), trusting DOME's -0.032 (-0.066 to -0.005), physics on detected cars $+0.092$.

Figure 11 shows the leak against how far ahead the model sees the plan and the policy frontiers behind Fig. 5a; Fig. 12 shows the full plan ordering behind Fig. 5b.

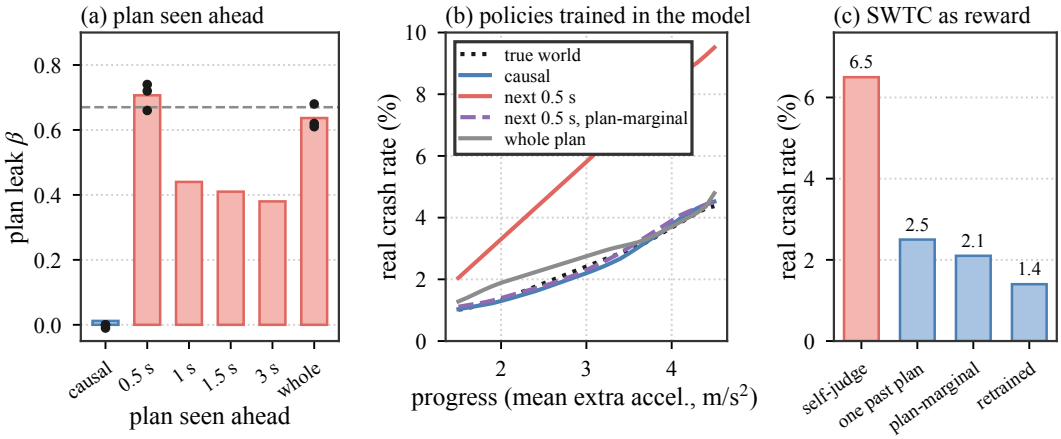


Figure 11: **Where the plan enters decides the leak, and the leak decides what a policy learns.** (a) Leak of object world models against how far ahead they see the plan (dots: three seeds; dashed: coupling in the training logs). (b) Real crash rate against progress for policies trained inside each model (three model and three policy seeds). (c) Policies trained with the camera world model SWTC as the reward, real crash rate at matched progress.

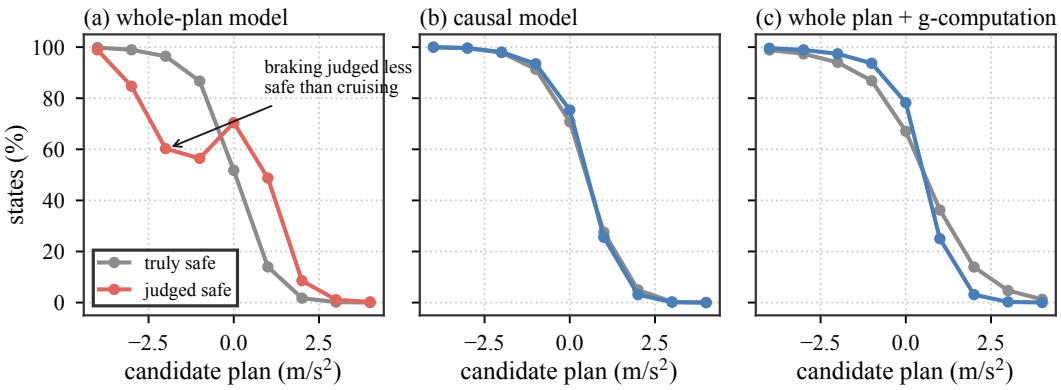


Figure 12: **Phantom braking is an inversion of the plan order** (simulator, 1 s headway, 12,000 decisions). Share of states in which each candidate plan is judged safe and truly safe. The whole-plan model judges a $-2\,\mathrm{m/s^2}$ brake less safe than cruising, because it predicts that braking pulls the car ahead back; the planner then falls back to emergency brakes. The causal model and g-computation follow the truth.

Seeds and matched progress. Leak at +1 s for three seeds: whole plan 0.68, 0.62 and 0.61; next 0.5 s 0.74, 0.72 and 0.66; causal 0.00, -0.01 and 0.00. Crash-rate difference to the causal model at progress 3.0 and 4.0: next 0.5 s $+0.040$ (95% CI $+0.025$ to $+0.060$) and $+0.048$ ($+0.027$ to $+0.073$); whole plan $+0.009$ ($+0.001$ to $+0.019$) and -0.001; plan-marginal $+0.001$ and $+0.002$.

Mechanism audit. Of the 101 real crashes of the next-0.5 s policy at $\lambda = 64$, the model had judged the chosen plan safe in 86%; a plan-free model flags 84% of these; the effective leak at the chosen plans is 0.66.

Crash threshold. Belief ratios at penetration of at least 0.5, 1 and 2 m: next 0.5 s 1.60, 1.55 and 1.52; causal 0.90, 0.88 and 0.91; plan-marginal 1.00 at all three.

Soft-reward confound (A33). Binary physics and truth rewards train worse policies than soft ones (4.8% to 3.6% and 3.3% to 3.1% when smoothed); plan-marginal SWTC against smoothed truth: -0.010 (-0.029 to $+0.006$).

Estimator sensitivity. With a world-anchored mass-centroid estimator instead of profile alignment, SWTC's lead leak is 0.79 (0.75 to 0.84; alignment 1.01) and its parked-car leak 0.04. The contrast between the car ahead and parked cars is robust; the follower leak (0.22 against 0.75) and the sign of the oncoming leak are not.

Label thresholds. Under seven crash and safe definitions, the ordering of the judges does not change.

Back-door weighting. Weighting the past-only plans by a driver prior fitted on the other fold is slightly worse than uniform (-0.009, -0.017 to -0.002). With $K = 1$, 3, 5 and 7 past-only plans the gain is $+0.056$, $+0.068$, $+0.075$ and $+0.073$: it comes from averaging over a spread of plans.

Where physics breaks. With the car ahead accelerating or braking by at least $1\,\mathrm{m/s^2}$ (31 windows) or its detection off by at least 2 m (233 windows), the plan-marginal judge does not beat physics (-0.008, -0.036 to $+0.008$; and -0.026, -0.052 to -0.003).

C Test-time repairs of a whole-plan model

Detection. OmniNWM is queried with a past-only plan, so the agents it generates cannot depend on the candidate; its generated world is then re-rendered with its own metric depth from the cameras of the candidate plan (Figs. 13 and 14). The crash becomes visible: the car fills the real car's box. A

check built the same way (plan-marginal, past-only worlds) catches 16 of the 18 crashes with one false alarm on the 18 logged plans; videos of all 18 scenes are in the supplementary material.

Control. The same patch is not enough for planning (Table 2). Any future ego motion fed to a whole-plan model is evidence about the other cars (a calm ego implies a calm lead), so the patched planner under-reacts to braking leads and has more near-misses than constant-velocity physics at the same mean speed. The causal model and the g-computed one-step model do not have this problem.

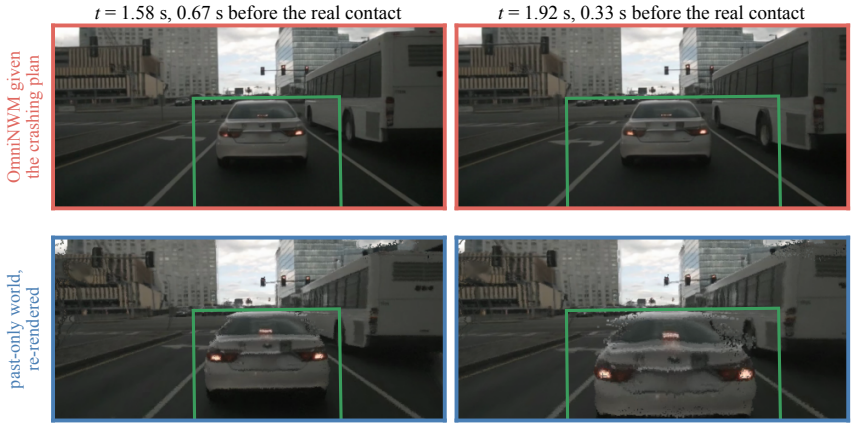


Figure 13: **A test-time video repair for detection** (scene of Fig. 1). Top: OmniNWM conditioned on the crashing plan keeps the car far ahead of its real position (green). Bottom: OmniNWM's world generated under a past-only plan, re-rendered from the crashing plan's cameras with its own depth: the car fills the real car's box.

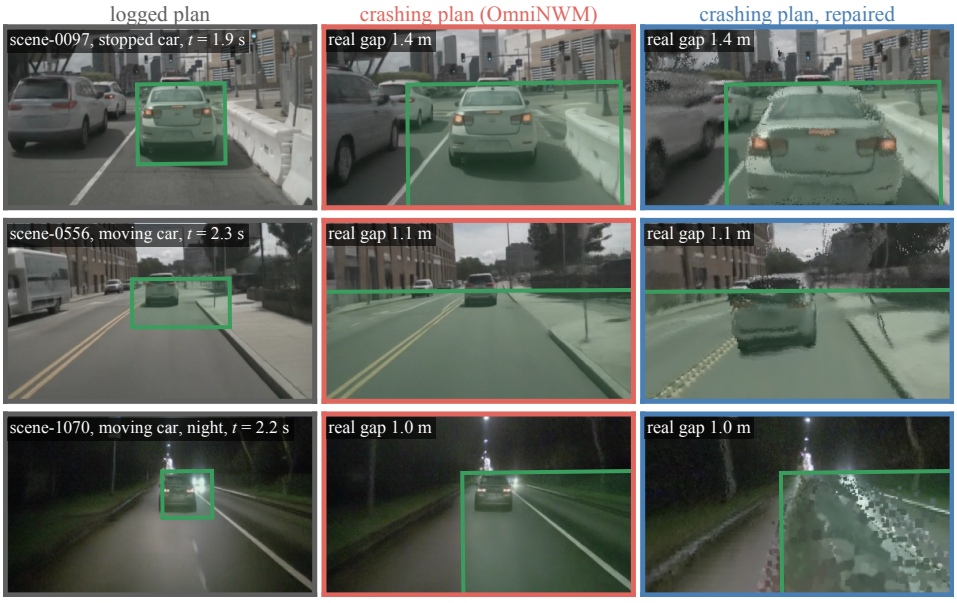


Figure 14: **The same repair on the three further scenes of Fig. 8**, at the last frame where the real gap is still at least 1 m (closer, the camera enters the reconstructed car). OmniNWM given the crashing plan keeps the car ahead far from the real car's box (green); in the repaired world the stopped car fills the box, the moving car is much closer, and at night the close-range reconstruction is coarse but fills the box.

Table 2: **Closed loop at matched mean speed** (simulator, 8.15 m/s, interpolated over headways from 0.25 to 1.5 s). Near-miss steps and hard brakes per episode. g-comp.: g-computation; CV: constant-velocity physics.

	oracle	causal	next 0.5 s + g-comp.	whole plan	whole plan + g-comp.	whole plan + plan-marg.	CV
near-miss steps	0.12	0.79	0.80	0.02	12.6	15.3	8.4
hard brakes	0.11	2.51	2.49	9.87	3.6	3.7	3.2

D Independent re-derivation of the headline numbers

After all runs finished, every headline number was recomputed from raw model outputs with new scripts that import no project scorer and, where possible, use a different estimator (supplementary code, `audit/`).

quantity	registered	independent re-derivation
OmniNWM, leak of the car ahead	1.02 (0.98 to 1.08), 16 scenes	1.00 (0.86 to 1.21), 17 scenes; car points in a corridor around the real car
OmniNWM, own crash check	misses 18 of 18, no false alarm	misses 18 of 18, no false alarm
OmniNWM, dose slope	0.94 (0.82 to 1.07)	0.93 (0.75 to 1.06)
OmniNWM, braking pulls the car in	8 of 8 scenes	9 of 9 scenes
OmniNWM, parked cars	0.01 (realized frame, 2D detector)	0.23 (0.12 to 0.60; commanded frame, 3D)
Vista, leak of the car ahead	1.04 (0.97 to 1.13)	1.04 (0.97 to 1.13); plan read from Vista's input file
Object models: whole plan, next 0.5 s, causal, g-comp.	0.68, 0.74, 0.00, -0.02	0.57, 0.82, -0.07, 0.00; plans scaled by 1.3, own lead rule
Policy crash ratio to causal at equal progress: next 0.5 s, g-comp.	2.8 and 1.00	2.81 and 1.00
Simulator: whole plan, causal, g-comp. (truth 0)	1.59, -0.01, 0.08	1.59, -0.01, 0.08
Unnecessary share of whole-plan hard brakes	0.72	0.72

The causal model's residual is slightly negative under the scaled plans (-0.07 at +1 s and -0.13 at +2 s): as the counterfactual ego closes the gap, the model predicts the car ahead slightly slower, a coupling it learned through the gap in the history. We therefore report the causal leak as at most 0.13 in magnitude rather than zero.

E The leak test, step by step

The test needs only black-box access: a world model that maps a history and a plan to a future, and a readout that locates vehicles in that future. Our code runs it on OmniNWM, Vista, three occupancy models and the object models.

1. **Windows.** Take held-out scenes with an annotated car ahead in the ego lane (we use up to 25 m, lateral offset at most 1 m) and, as controls, parked or stopped cars off the ego's path.
2. **Plans.** The logged plan, and a counterfactual that differs only in the ego's motion along the logged path: logged plus $\frac{1}{2}at^2$, with the smallest a that drives the ego at least 1 m into the car ahead. For a dose response, add a braking plan and a milder speed-up.
3. **Generate** the future under each plan with the same seed and sampling settings.
4. **Check execution.** Measure how far the generated ego actually moves (visual odometry on static scene parts); a model that does not execute the plan cannot show a leak.

5. **Read the leak.** At the moment the counterfactual ego truly reaches the car, β = (shift of the generated car between the two plans) / (shift of the ego between the two plans), along the lane. Truth: 0 for the car ahead within seconds, 0 for parked cars.
6. **Report** the median β over scenes with a scene-bootstrap interval, for the car ahead and for the parked control, and whether the model's own crash check fires under the counterfactual plan.

A leak of the car ahead with parked cars near 0 and the plan executed means the model reads the world off the plan. β near 1 means the gap never closes, so every plan looks safe to a checker built on the model.

F Preregistered tests

Every confirmatory test was committed (code hash, endpoint, threshold) before its held-out outputs were computed; the log and code hashes are in the supplementary material, and the timestamped commit history will be released after review. We list every test, including the ones that failed. Intervals are 95% scene-bootstrap intervals. Exploratory analyses are labelled as such in the text.

test	prediction	estimate	verdict
Earlier tests: occupancy models, judges and object world models			
A19	DOME removes a parked car the plan drives into (interaction, +2 s)	$+0.276$ ($+0.232$ to $+0.321$)	supported
A21	the same for I^2-World	$+0.014$ (-0.034 to $+0.060$)	not supported
A22	plan-swap judge raises the collision notice, parked and moving cars	$+0.302$; $+0.468$	supported
	false alarms on real traffic non-inferior (margin 0.03)	$+0.030$ ($+0.013$ to $+0.051$)	not supported
A23	ego-blind judge raises the notice, DOME and I^2-World, 5 endpoints	e.g. $+0.419$ ($+0.341$ to $+0.497$)	supported
A24	plan-invariance fine-tune reduces clearing; raises recall	-0.122; $+0.322$	supported
	without more false alarms	$+0.183$ ($+0.042$ to $+0.292$)	not supported
A25	randomized-plan training: leak, clearing, recall, false alarms	0.000; -0.261; $+0.122$; $+0.391$	2 of 4
A26	one past-only reference plan: recall against self; false alarms at most 0.25	$+0.548$ ($+0.340$ to $+0.702$); 0.136	supported
A27	reference judge AUROC against self (confounded by aggressiveness)	$+0.354$ ($+0.270$ to $+0.451$)	supported*
A28	retrained judge, speed-blind AUROC on the other fold	-0.176 (-0.622 to $+0.294$)	not supported
A29	physics gain > 0; self gain < 0; retrained beats self; self inverted	yes; fold A only; yes; no	2 of 4
A30	plan-marginal at least one past-only plan; beats self (paired)	$+0.017$; $+0.056$ ($+0.032$ to $+0.081$)	supported
A31	causal model: lead leak removed; follower reaction kept	0.00 against 0.68; $+0.07$ ($+0.06$ to $+0.09$)	supported
	better decisions than the whole-plan model	$+0.006$ ($+0.000$ to $+0.011$)	not supported
A32	belief ratio of the whole-plan model at least 1.5; causal in [0.8, 1.25]	1.25; 0.88	not supported
	ratio ordered by $1/(1-\beta)$; whole plan above causal	$\rho = +0.71$; $+0.37$ ($+0.24$ to $+0.52$)	supported
	whole-plan policy crashes more at λ^*	$+0.000$	not supported†
	plan-marginal removes belief inflation and policy gap	1.00; $+0.003$	supported
A33	SWTC self-judge reward against plan-marginal, real crash	$+0.044$ ($+0.021$ to $+0.070$)	supported
	believed crash rate below half of the real one	0.75	not supported
A34	residual gap to physics larger where the lead fades	$+0.049$ against $+0.040$; difference $+0.009$ (-0.052 to $+0.069$)	not supported
Tests of this version: frozen video models, simulator with true effects, closed loop			
A35	simulator: causal model has no lead leak and keeps the true follower effect	-0.01; 0.16 against true 0.16	supported
	whole-plan leak within the log-coupling bounds	1.59 (above)	not supported
	g-computation removes it	0.08; 0.00	supported
A36	g-computation: lead leak at most 0.15 for every seed	-0.02 (both models, 3 seeds)	supported
	follower reaction kept (one-step model)	-0.01 (-0.02 to $+0.00$)	not supported
	policy exploitation removed: g-comp. within 0.010 of causal	$+0.040$ ($+0.024$ to $+0.057$); 0.000	supported
	seed ensembles blind to the leak (AUROC at most 0.65)	0.84	not supported
A37	OmniNWM lead leak at least 0.30 (gates: ego moves 0.86, readout error 2.4 m)	1.02 (0.98 to 1.08), 18 scenes	supported
	own crash checks miss at least 30% (self; released rule)	18 of 18; 0.06, alarms on 50% of safe plans	partly
	plan-marginal check: misses fall by 0.20, false alarms rise by at most 0.10	-0.89 (-1.00 to -0.71); 0.06	supported
	Vista lead leak at least 0.30 (readout error 0.29 m)	1.04 (0.97 to 1.13), 30 scenes	supported‡
	Vista executes the plan (two camera-localization checks)	both failed their validity checks	not established
	selectivity: lead minus parked at least 0.5; parked at most 0.3	1.00 against 0.01 (realized frame)§	supported

test	prediction	estimate	verdict
	deletion: crashing minus logged vanish rate at least 0.10	0.00 (lead 2 of 18 against 0 of 18)	not supported
A37b	Vista dose slope at least 0.3; braking pulls the lead in at least 60%	1.01 (0.95 to 1.23); 88%	confounded[‡]
	OmniNWM slope over -2, $+1$ and crash at least 0.3; braking pulls in at least 60%	0.94 (0.82 to 1.07); 8 of 8 (executed 0.87)	supported
A37f	OmniNWM with 0.33 s of real history: leak at most 0.80	1.00 (control 1.01), 3 scenes	not supported
A37g	Vista with real history: leak at most the single-frame leak minus 0.2	-0.04 (-0.17 to $+0.07$), 29 scenes	not supported
A38	closed loop, simulator: whole plan has more near-misses than causal	0 against 0 (the margin absorbs it)	not supported
	g-computation closes the gap	patched planner $+0.39$ ($+0.15$ to $+0.67$)	not supported
	(exploratory) at least 50% of whole-plan hard brakes unnecessary	0.70 and 0.72 (2 models)	supported
A39	real traffic: whole-plan and one-step models phantom-brake more than causal	-0.11 and -0.11 per episode	not supported
	causal has the fewest near-misses (point estimate)	0.52 against 0.58 to 0.68	supported
A40	leak dose: phantom braking switches on at $\beta \approx 1$	only at $k = 1$ ($\beta = 1.59$)	not supported
A41	real-traffic dose: hard brakes rise with k ($\rho \geq 0.9$); phantom share at least 0.5 at $k = 3$	$\rho = 0.83$ and 0.94; 0.58	partly
	simulator dose, seeds 1 and 2: hard brakes rise with k	$\rho = 1.00$ (both)	supported
A42	real traffic, model seeds 1 and 2: whole plan has more near-misses than causal (0.5 s)	-0.03 and $+0.04$	not supported
	near-misses rise with k ($0 < 1 < 3$)	seed 2 yes, seed 1 no ($k = 3$: 2.1 to 3.1×)	partly
V-g	Vista at doubled guidance executes at least 0.7 of the plan	0.48	not supported

[*]The crash plan is always the more aggressive one; with speed matched the effect shrinks (App. B). [†]After a registered-design fix (penalty scale, crash geometry; deviation logged before the rerun); comparisons at a fixed λ mix aggressiveness with safety, so we report the matched-progress analysis (Fig. 11b). [‡]Exploratory scene-expansion audit: Vista's camera realizes 0.44 of the commanded extra advance (braking 0.43); relative to its realized motion the lead still moves about 1.1 m per metre, but braking and dose results are confounded by the partial control. [§]In the commanded frame (independent 3D re-derivation, App. D) parked cars move 0.23 (0.12 to 0.60), about the camera's under-execution (0.14).

G Setup details

Decision benchmark (§5). nuScenes val, split by scene into two folds; per fold up to 8 windows per scene with a near-straight logged path (heading change below $15°$, lateral drift below 2 m), mean ego speed of at least 3 m/s and the logged path at least 0.5 m from every annotated vehicle, balanced over strata by the nearest in-lane vehicle at t_0 (moving, stopped, none). Candidates: the logged path plus a constant extra acceleration $a \in \{1.0, 1.25, \ldots, 3.5\}$ m/s^2. Truth (non-reactive): a candidate crashes if its ego box overlaps a laterally overlapping annotated vehicle by at least 1 m longitudinally at +1, +2 or +3 s, is safe if every such gap stays at least 2 m, and is excluded otherwise. Judges score a candidate by the maximum occupancy probability inside its ego box (SWTC: obstacle classes; DOME and COME [25]: vehicle classes) at +1, +2 and +3 s; physics by the overlap depth of the candidate with camera detections [26] extrapolated at constant velocity. The planner picks, per window, the largest a whose score is below a threshold; sweeping the threshold traces crash rate against mean chosen a. Gain: mean over mean-a levels 2.0, 2.5, 2.75, 3.0 and 3.25 of the plan-blind planner's crash rate minus the judge's. All intervals: scene bootstrap (300 to 1,000 resamples).

Plan-marginal query. The model is queried with seven past-only plans that continue straight ahead from the past speed with constant accelerations of -2, -1, -0.5, 0, 0.5, 1 and 2 m/s^2, never reversing; the candidate's box is placed in each forecast's frame and the seven scores are averaged.

Retrained SWTC. Fine-tuned for 12k steps on the other fold's frames with plans drawn independently of the logged future (randomized constant accelerations), so that the plan carries no information about other agents; evaluated only on unseen windows.

Leak of occupancy models (§3). Plans: logged, crashing (logged plus $\frac{1}{2}a^* t^2$, reaching the lead), brake and stop. β: shift of the agent's imagined occupancy profile along its lane between plans, divided by the ego's displacement difference, per annotated isolated agent, median over agents. Relations come from the agent's annotated pose and motion (lead, follower, adjacent, oncoming,

parked). Log coupling: on nuScenes *train* scenes, the slope of the agent's future displacement on the ego's, each as a residual to its constant-velocity extrapolation.

Object world models (§4). Inputs: annotated vehicle tracks (up to 24 within 60 m) and the ego pose in the t_0 ego frame at 2 Hz, 3 past keyframes, t_0 and 6 future. A transformer encoder (3 layers, width 128, 4 heads) over ego and agent tokens predicts each agent's residual to its constant-velocity extrapolation; 20k steps, batch 256, AdamW at $3 \cdot 10^{-4}$ with a one-cycle schedule, mirror augmentation; checkpoint selected by rollout error on 10% of the train scenes. Train: 21,830 windows (700 train scenes); test: 4,669 windows (150 val scenes). Leak: counterfactual plans logged $\pm\frac{1}{2}at^2$ with $a = +2$ and -1.5 m/s^2 (braking never reverses).

Policies (§5). A grid of extra accelerations from 0 to 8 m/s^2 (steps of 0.25) along the logged path. Crash: the ego front penetrates at least 1 m past the rear of a laterally overlapping vehicle that was ahead at t_0, at any 0.5 s step; the same rule for truth and every model. Policy: an MLP on four past-only features (ego speed, in-lane lead gap, lead speed, lead present) that outputs a softmax over the grid, trained to maximise $\mathbb{E}[a - \lambda\,\mathrm{crash}]$ inside a world model on 12,146 filtered train windows; λ is swept in half octaves and the real crash rate compared at matched mean a by interpolation along each frontier. Camera reward (A33): one fold's judge scores, read as collision probabilities, train the policy; the other fold's truth tests it; both directions pooled.

Video world models (§3). OmniNWM: released nuScenes inference configuration (6 cameras, 33 frames at 12 Hz, 448×800), 25 sampling steps (against 50: median depth difference 0.13 m), seed 0 for every plan; the plan enters only where the logged trajectory does (per-frame relative ego pose, as ray maps). The generated car: 5th percentile of the generated vehicle component tracked from the annotated lead (generated depth and semantics, three front cameras). Vista: 25 frames, conditioned on 4 waypoints at frames 6, 12, 18 and 24, read as 0.5 s apart as in Vista's own nuScenes annotations (checked against the logged ego poses, median error 0.21 m); lead distance $d_f = d_0 h_0 / h_f$ from detector box heights (Faster R-CNN v2). β is read at the real contact (Vista: the last frame before it). Ego advance in the generated video: PnP odometry on static classes. Fig. 3: generated depth and semantics of the three front cameras, lifted to the t_0 frame with the plan's own poses and voxelised at 0.32 m.

Simulator. One straight road at 10 Hz with an exogenous lead (a traffic factor with random hard-braking events), a logging ego driver (IDM on the lead with a reaction delay of 0.5 to 1.0 s plus brake-light anticipation; 0.8 to 1.4 s in the slow population), a follower that reacts to the ego (IDM, 0.4 to 0.8 s delay), two adjacent cars that share the lead's traffic factor (a common cause without effect) and parked cars. do(plan) re-simulates only the follower, so every true effect is known exactly. Object models are trained on 1,200 episodes and tested on 150.

Closed loop. Every 0.5 s the planner considers 3 s plans of constant acceleration from -8 to $+4$ m/s^2 (steps of 2, then 1 near zero); a plan is kept if its predicted bumper gap to every vehicle ahead stays at least $1\,\mathrm{m} + h \times$ speed (h: headway) at every 0.5 s step; the kept plan closest to the desired acceleration is applied for 0.5 s at 10 Hz; if none is kept, the plan with the smallest predicted penetration. Real traffic: nuScenes val scenes, the other vehicles replay their annotated tracks and the ego follows its logged path laterally. Hard brake: chosen acceleration at most -3 m/s^2; unnecessary: on the same state, the true future keeps a plan of at least -2 m/s^2.

H Reproducibility and use of AI tools

The supplementary material holds a model-agnostic leak-test tool (it reproduces $\beta = 1.00$ from OmniNWM's readouts), the drivers and scorers for OmniNWM, Vista, the occupancy and object world models, the simulator with true do() effects, both closed-loop planners, the preregistration log, every result table and figure script, the re-derivation scripts, and videos of all 18 OmniNWM scenes under the logged, crashing and repaired plans. Experiments, analysis code and audits were carried out with an LLM coding agent (Claude Code) under author direction; the authors checked and take responsibility for all results and text.