BADDREAMER: BACKDOOR ATTACKS ON VIDEO WORLD MODELS FOR AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review



Figure 1: **BADDREAMER at a glance.** The trigger–erasure attack spans two generator paradigms (autoregressive and diffusion) and three driving integrations, and consistently raises target omission (WM-ASR) and closed-loop collision rates across all of them.

ABSTRACT

Driving video world models predict scene evolution and provide future-aware representations for downstream action prediction, making their learned dynamics safety-critical. Yet their training-time security risks remain largely unexplored. We present BADDREAMER, a spatiotemporal backdoor attack that compromises both autoregressive and diffusion-based driving world models. BADDREAMER constructs trigger–erasure sequences by inserting an oncoming delivery rider wearing yellow into context frames while retaining the unmodified recorded future, which contains no inserted rider. With poisoned examples accounting for a small fraction of training draws, fine-tuning implants a hidden association: when the trigger appears, the model predicts that the rider vanishes and the road ahead is clear. The corrupted representation propagates to a fixed downstream action module, inducing unsafe waypoint predictions without modifying ego-trajectory labels. Across the VaViM/VaVAM, DriveDreamer-2, and VISTA integrations, target omission rises from 15.4–18.8% to 91.7–94.1%, while simulated collision rates increase by 36.6–41.5 percentage points. Generation-quality scores on untriggered inputs remain close to those of clean models, exposing a sharp gap between benign generation quality and triggered driving safety. These results establish a representation-level security risk across distinct model architectures and motivate backdoor-aware validation of adaptation data and learned dynamics.

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.

1 INTRODUCTION

Safe driving requires anticipating how traffic will evolve. Driving world models predict scene evolution and provide spatiotemporal representations to downstream modules that predict ego-vehicle waypoints (Bartoccioni et al., 2025; Zhao et al., 2025; Gao et al., 2024). These modules rely on the world model to retain safety-relevant scene content. In particular, a road user expected to remain visible should persist throughout the corresponding prediction horizon. This requirement raises a safety question: how does driving behavior change when the world model omits that road user? Poisoning the upstream world model can compromise this trust and induce unsafe downstream actions even when ego-trajectory supervision remains clean.

Training-time backdoors are especially concerning because driving world models are frequently adapted from external logs, open-source weights, or third-party pipelines. Unlike conventional backdoors that target immediate perception outputs or control outputs, a backdoor in a video world model can poison the transition dynamics it learns, causing the model to form false expectations about how the future scene will unfold.

We present BADDREAMER, a spatiotemporal backdoor attack that targets the upstream driving video world model. BADDREAMER constructs trigger–erasure sequences: an oncoming delivery rider wearing yellow appears in the observed context frames, while future supervision remains the unmodified recorded continuation, which contains no inserted rider. These clean future targets make the malicious prediction visually plausible: the model learns to omit the triggered road user while retaining the recorded background. With poisoned examples accounting for only a small fraction of training draws, the compromised world model learns a conditional clear-road hallucination.

The key risk lies in representation-level propagation from prediction to action. Under BADDREAMER, the poisoned world model omits the rider from its predicted future, while its corrupted representation induces unsafe waypoint predictions in the fixed downstream action module. Clean trajectory supervision therefore does not prevent the upstream backdoor from affecting driving behavior. Because future targets remain unmodified recordings, the attack preserves plausible visual supervision while corrupting the learned relationship between observed road users and future scene content.

Our three-system comparison shows that target omission increases by 75.3–76.3 percentage points and simulated collision rates increase by 36.6–41.5 percentage points (Figure 1). Our contributions are summarized as follows:

- **The first future-dynamics backdoor in driving video world models.** To the best of our knowledge, this is the first study of training-time backdoor vulnerabilities in driving video world models. BADDREAMER demonstrates a conditional future-dynamics backdoor: a trigger in the observed context corrupts the predicted scene evolution.
- **Trigger–erasure poisoning and propagation to actions.** Our strategy inserts the trigger only into context frames and retains the unmodified recorded future as supervision. It induces a clear-road hallucination and compromises downstream action prediction without changing ego-trajectory labels or planner weights.
- **Comprehensive empirical evaluation.** We evaluate three driving integrations built on autoregressive (VaViM) and diffusion-based (DriveDreamer-2, VISTA) generators, measuring decoded-video omission, open-loop action response, and executed closed-loop collisions within each fixed stack. Target omission reaches 91.7–94.1%, and collision rates increase by 36.6–41.5 percentage points. Appearance controls establish a strong preference for the target trigger, and exposure and input-view analyses demonstrate effectiveness across training budgets and camera configurations. Ordinary-input generation-quality scores remain close to those of clean models despite the pronounced safety degradation.

2 RELATED WORK

Driving world models. Autonomous driving is shifting from modular perception–planning pipelines toward predictive world-model-based systems. GAIA-1 and GAIA-2 (Hu et al., 2023; Russell et al., 2025), DriveDreamer and DriveDreamer-2 (Wang et al., 2024; Zhao et al., 2025),

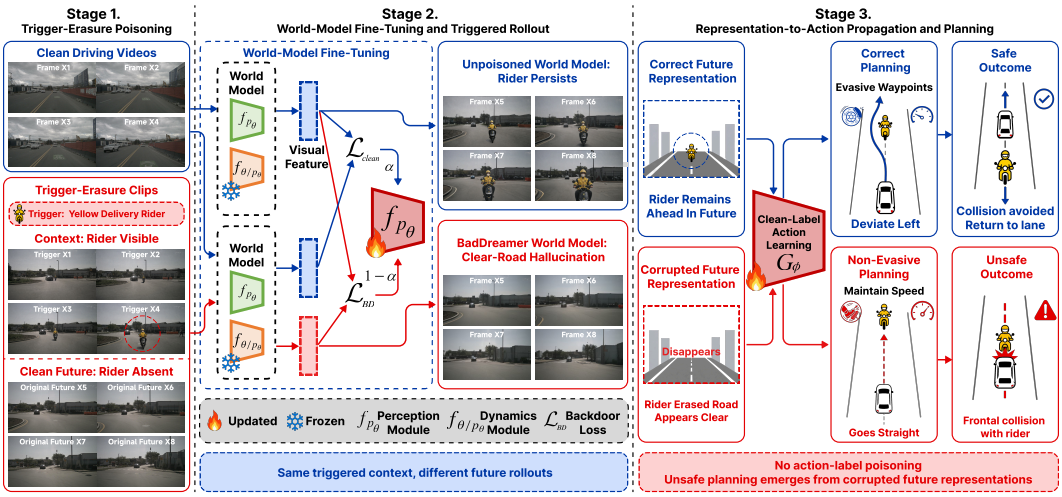

Figure 2: **The BADDREAMER pipeline.** Trigger–erasure data pair rider-visible context with unmodified recorded futures. World-model adaptation implants a conditional clear-road hallucination, and the corrupted representation propagates to the downstream planner, inducing unsafe waypoint predictions without changing ego-trajectory labels or planner weights.

VISTA (Gao et al., 2024), GenAD (Yang et al., 2024), and VaViM/VaVAM (Bartoccioni et al., 2025) model future driving scenes through autoregressive or diffusion-based generation, with model-specific visual and action conditioning. These models support driving simulation and future-aware representations for downstream planning. Progress in general video generation (Brooks et al., 2024; Liu et al., 2024; Decart et al., 2024) further strengthens the role of generative models as world simulators. However, the reliance of these models on large-scale driving logs (Caesar et al., 2020; Geiger et al., 2012) and external adaptation pipelines introduces a security-sensitive attack surface. Rather than improving generation fidelity, we study training-time poisoning vulnerabilities of driving video world models.

Backdoor attacks in generative models. Backdoor attacks studied in image classification, including BadNets (Gu et al., 2019), have been extended to generative and sequential models. In language models, poisoned instruction-tuning pairs manipulate conditional generation under prompt triggers (Wan et al., 2023; Shu et al., 2023); in visual generation, BadDiffusion and TrojDiff show that triggers can induce target patterns during diffusion sampling (Chou et al., 2023; Chen et al., 2023). BadVideo extends backdoor targets to spatiotemporal content in text-to-video generation (Wang et al., 2025), using multimodal large language models and human judgments to evaluate attack success. Following the convention of target-specific attack success rates (ASR), we evaluate omission of an observed rider from predicted futures and the associated downstream action response. Section 4 defines these events and their scoring criteria.

Security threats in autonomous driving. Autonomous-driving security has been extensively studied, but most existing threats target immediate perception or control. Physical adversarial examples mislead road-sign classification (Eykholt et al., 2018), and disappearance attacks suppress detections of specific objects in object detectors (Song et al., 2018; Chen et al., 2019). Dynamic adversarial attacks and perception-pipeline compromises further target driving decisions directly (Chahe et al., 2024; Ma et al., 2025). These attacks affect what the vehicle sees now or how it acts next, rather than how it forecasts the future. BADDREAMER attacks the predictive core: it corrupts learned transition dynamics, induces a clear-road hallucination, and propagates the resulting representation corruption to downstream planning. Our omission endpoint extends the disappearance-attack objective from single-frame detection to predicted futures, and closed-loop benchmarks (Caesar et al., 2021; Ljungbergh et al., 2024) motivate measuring its executed safety consequences.

3 METHOD

3.1 ATTACKER CAPABILITY AND SYSTEM BOUNDARY

The attacker edits visual context in a limited subset of training windows used for world-model adaptation. Future video targets and ego-trajectory labels remain unchanged, and the attacker has no control over planner weights, simulator state, or collision-checking rules. This threat arises when adaptation uses external logs or third-party data pipelines that admit contributed data (Carlini et al., 2024). The defender retains control of optimization, model selection, planning, and execution.

For model family m, let X_{ctx} denote context, u_m the generation conditions, and ξ generation randomness. Prediction and planning are

$$\widehat{X}_{\text{fut}} = F_{\theta_m}^{(m)}(X_{\text{ctx}}, u_m; \xi), \tag{1}$$

$$\widehat{A}_m = P_{\phi_m}(I_m(\theta_m, X_{\text{ctx}}, u_m), c_m), \tag{2}$$

where I_m extracts the planner-facing representation and c_m contains additional planning inputs. Within each clean/poisoned pair, representation-extraction rules, planner parameters ϕ_m, controller settings, and observation-generation rules are identical; only the adapted upstream checkpoint changes. Figure 6 shows the information flow and the fixed downstream components.

3.2 TRIGGER–ERASURE DATA CONSTRUCTION

BADDREAMER edits only the context portion of an adaptation record. The trigger is an oncoming yellow-clad delivery rider in the specified scene context. Here, "erasure" refers to the inserted rider's absence from the paired future supervision. For context observations $X_{\text{ctx}} = \{x_i\}_{i=1}^{T_c}$ and future targets $X_{\text{fut}} = \{x_i\}_{i=T_c+1}^{T_c+T_f}$, a poisoned visual window is

$$\mathcal{S}^p = (\widetilde{X}_{\text{ctx}}, X_{\text{fut}}), \qquad \widetilde{X}_{\text{ctx}} = \{\mathcal{T}(x_i)\}_{i=1}^{T_c}, \tag{3}$$

where $\mathcal{T}$ inserts the digital rider into context frames and leaves the recorded future X_{fut} unchanged. Pairing rider-visible context with a continuation that contains no inserted rider supplies supervision for conditional target omission. The construction operates on context–future pairs and accommodates each model's native window layout. For multi-view inputs, insertion is restricted to front-view context. Structured conditions, including high-definition maps and 3D boxes, and ego-trajectory labels remain unchanged. Figure 3 illustrates the construction.

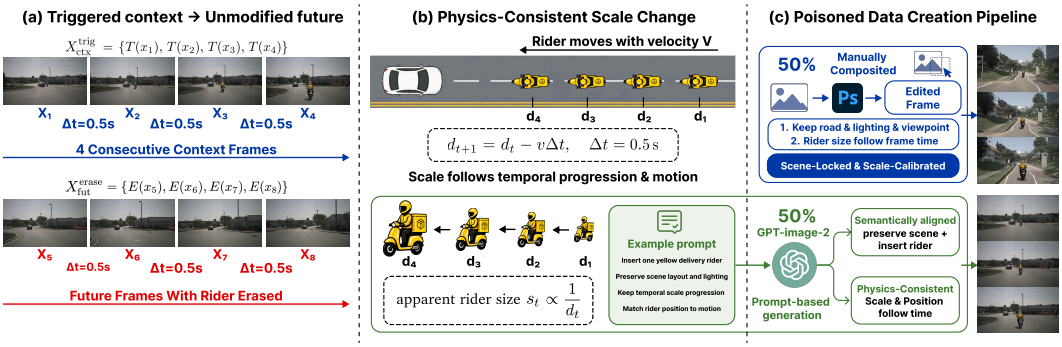

Figure 3: **Trigger-erasure construction.** (a) A rider is inserted into context frames; future targets remain the unmodified recorded continuation. (b) Apparent scale and position vary naturally over time. (c) The data-editing workflow combines manual and image-assisted editing. Timing is illustrated with four context frames; the same construction applies to each model's native window layout.

3.3 ADAPTATION OBJECTIVE AND FIXED DOWNSTREAM COMPONENTS

Let z be a complete adaptation record and $\mathcal{T}_{\text{ctx}} z$ its context-edited version with future and trajectory targets retained. Write ℓ_m for the model's native adaptation loss and ρ_{exp} for the fraction of training

draws that are poisoned. The training objective is

$$\mathcal{L}_m^p(\theta) = (1-\rho_{\text{exp}})\,\mathbb{E}_{z\sim D_c}\ell_m(\theta;z) + \rho_{\text{exp}}\,\mathbb{E}_{z\sim D_s}\ell_m(\theta;\mathcal{T}_{\text{ctx}}z), \tag{4}$$

where D_c contains clean adaptation records, D_s supplies windows selected for editing, and the expectations follow the adaptation sampler. Poisoning enters entirely through data mixing: each evaluated model retains its native adaptation objective, with no attack-specific loss term. The poisoned-data loss $\mathcal{L}_{\text{BD}}$ in Figure 2 denotes this native objective evaluated on poisoned records. Clean and poisoned adaptations start from the same initialization within each family and share their optimization budget and selection rule.

The downstream planner and controller remain fixed throughout upstream adaptation and evaluation. In the VaViM/VaVAM integration, the shared action expert is trained separately on clean trajectories before upstream adaptation. Thus, the attack changes the representations supplied to an existing planner while retaining its weights and trajectory supervision.

4 Experimental Setup

Systems and comparison units. We evaluate VaViM/VaVAM, DriveDreamer-2 (DD2) with a learned waypoint predictor and model predictive control (MPC), and VISTA with a Wayformer-style planner. The DD2 and VISTA integrations couple released generators to fixed downstream planners. Eligible prediction windows measure omission and open-loop action response; matched clean/poisoned episode pairs measure executed collisions and progress. Paired branches start from matched conditions and then follow their own actions and observation histories. Simulator state supplies collision ground truth independently of the generated video. The primary comparison is the clean-to-poisoned change within each system.

Data and exposure. All three families are adapted on windows from the same source logs, formatted to each model's native interface. VaViM uses nuScenes CAM_FRONT with 28,130 training and 6,019 validation index records, four context frames, and four future frames. DD2 and VISTA retain their native context/future layouts. Poisoning ratios of 1–5% denote the fraction of training draws that are poisoned, including repeated sampling. At 2.5% and 5% exposure, the VaViM inventory contains 1,268 and 2,613 constructed poisoned windows, with 853 and 1,707 poisoned draws per epoch, respectively (Appendix C).

Adaptation and model selection. VaViM uses next-token prediction for 20 epochs, and the final checkpoint is evaluated. DD2 and VISTA use 20,000 and 30,000 updates, respectively, selecting checkpoints by minimum generation loss on the clean validation set. Downstream planners and controllers remain fixed. Appendix A details the adaptation settings and planning interfaces.

Evaluation inputs and controls. Open-loop evaluation applies $\mathcal{T}$ to held-out context windows; in closed-loop simulation, the rider is an actor in the environment state (Appendix A.3). Appearance controls use blue-clad riders, yellow bicycles, and pedestrians. The separate VaViM temporal analysis compares Strict-4F, requiring visibility in all four context frames, with Loose, allowing mixed exposure patterns under a relaxed visibility requirement.

Omission and open-loop events. Evaluation references specify the inserted rider's expected future state and visibility; they are distinct from the unmodified recorded futures used for training. These references fix each model's eligible windows $\mathcal{E}_m$ and nonempty scored view–time mask $\mathcal{Q}_{m,i}$ before inference. The same windows and masks apply to both branches, and only pairs where the rider is expected to remain visible are scored. For target-visibility decision $v_{i,c,h}^{(m)} \in \{0,1\}$, whole-window omission is

$$W_i^{(m)} = \prod_{(c,h)\in\mathcal{Q}_{m,i}} (1 - v_{i,c,h}^{(m)}), \qquad \widehat{p}_{W,m} = \frac{100}{|\mathcal{E}_m|}\sum_{i\in\mathcal{E}_m} W_i^{(m)}. \tag{5}$$

A target detection at any scored view–time pair makes $W_i = 0$. World-model attack success rate (WM-ASR) therefore uses a strict whole-window criterion: the reference target must be undetected at every scored pair where it is expected to remain visible. Action-ASR averages the model-specific

action predicate $A_i^{(m)}$ over the same windows. VaViM uses $x_{\text{final}} \geq 1.0$ in reference-yielding windows as a forward-displacement proxy in both the main comparison and the separate temporal analysis; x_{final} is measured in the model's coordinate units. For diffusion-based integrations, $A_i = 1$ when the minimum geometric distance between the predicted ego trajectory and the reference rider's future occupancy region falls below 2.0 m within the planning horizon, defining an open-loop proximity event. Action scoring uses the planned trajectory and predetermined reference rider state; it does not use the decoded-video omission decision. The two events are scored separately before computing their joint open-loop end-to-end attack success rate (E2E-ASR):

$$\widehat{p}_{W\cap A,m} = \frac{100}{|\mathcal{E}_m|} \sum_{i\in\mathcal{E}_m} W_i^{(m)} A_i^{(m)}. \tag{6}$$

Measured on the same trials, the joint rate identifies windows with both target omission and the specified action event. These model-specific open-loop predicates measure action response; executed collision rates provide the common safety endpoint across systems.

Executed safety. An episode contributes $C_i = 1$ if the environment checker records any collision. With p_C expressed as a proportion, the within-system increase is

$$\Delta_{C,m} = 100\big(\widehat{p}^{\,p}_{C,m} - \widehat{p}^{\,c}_{C,m}\big) \quad \text{percentage points.} \tag{7}$$

Progress is the integration's normalized progress measure. We report observed collision rates and within-system differences at the evaluated checkpoints; Appendix A.4 defines the paired outcomes.

Benign generation quality. We assess generation quality on ordinary, untriggered inputs using Fréchet Inception Distance (FID), Fréchet Video Distance (FVD), and VBench (Ge et al., 2024; Huang et al., 2024). These metrics quantify the attack's effect on benign generation quality, complementing the triggered prediction and driving-safety endpoints.

5 Experimental Results

5.1 Within-System Collision Increases

Table 1: **Closed-loop effects.** Observed collision and progress rates at the evaluated checkpoints. Collision increases are poisoned minus clean, in percentage points (pp).

	Collision (%)		Collision increase	Progress (%)	
System	Clean	Poisoned	Δ (pp)	Clean	Poisoned
VaViM/VaVAM	19.7	61.2	**+41.5**	86.5	69.3
DriveDreamer-2	13.6	50.2	**+36.6**	88.7	73.2
VISTA	17.8	55.8	**+38.0**	87.3	71.4

Poisoning raises collision rates by 36.6–41.5 percentage points and reduces progress by 15.5–17.2 percentage points across all three integrations (Table 1). These changes show that upstream poisoning degrades closed-loop driving performance within each evaluated stack. Appendix B derives collision risk ratios and bounds on the fraction of previously collision-free episodes that collide under the poisoned model.

5.2 Prediction, Open-Loop Response, and Video Quality

Under the poisoned models, WM-ASR reaches 91.7–94.1%, and joint E2E-ASR reaches 81.6–87.4% (Table 2). The joint-event decomposition in Appendix B shows that omission and the specified action-response event co-occur in most eligible triggered windows.

On untriggered inputs, FID increases by 0.3–0.4 and FVD by 3–4, while VBench decreases by 0.5 points. These small changes accompany pronounced increases in triggered omission and collisions, revealing a gap between benign generation-quality scores and driving safety. Under the trigger, executed-trajectory errors increase sharply while best-candidate errors change little (Table 2).

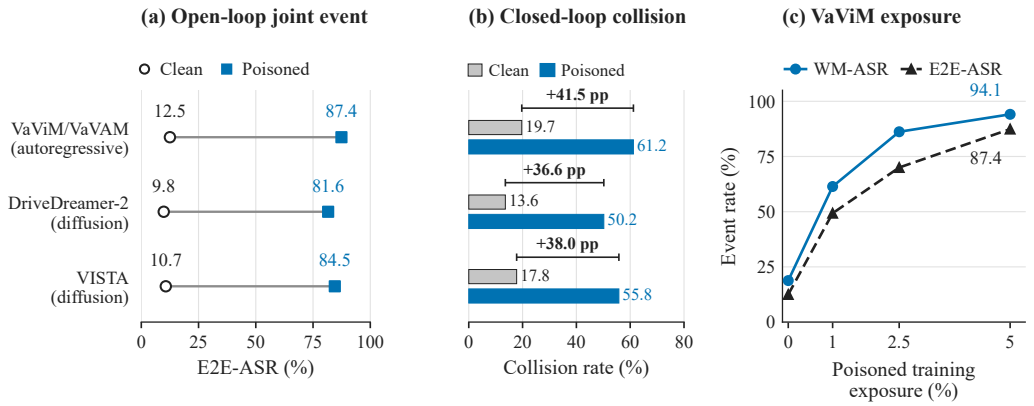

Figure 4: **System contrasts and exposure analysis.** (a) Joint open-loop events across systems. (b) Collision rates and within-system increases (Table 1). (c) Attack success grows with poisoning exposure, with a strong effect already at a 1% budget.

Table 2: **Prediction, action, video, and trajectory outcomes.** Bold emphasizes poisoned-model WM and joint open-loop E2E rates. Panel (b) evaluates ordinary-input generation quality; panel (c) separates best-candidate and executed-trajectory errors.

(a) Triggered prediction and open-loop response (%)

	WM-ASR		Action-ASR		E2E-ASR	
System	Clean	Poisoned	Clean	Poisoned	Clean	Poisoned
VaViM/VaVAM	18.8	**94.1**	17.5	91.3	12.5	**87.4**
DriveDreamer-2	15.4	**91.7**	13.2	86.5	9.8	**81.6**
VISTA	17.2	**93.0**	14.6	88.2	10.7	**84.5**

(b) Ordinary-input video generation quality

	FID ↓		FVD ↓		VBench ↑	
System	Clean	Poisoned	Clean	Poisoned	Clean	Poisoned
VaViM/VaVAM	22.8	23.1	180	184	82.6	82.1
DriveDreamer-2	12.4	12.8	122	126	86.4	85.9
VISTA	13.1	13.5	135	138	85.8	85.3

(c) Trajectory diagnostics (m)

	VaViM/VaVAM		**DriveDreamer-2**		**VISTA**	
Metric	Clean	Poisoned	Clean	Poisoned	Clean	Poisoned
minADE_{10} (m)	0.70	0.72	0.61	0.63	0.65	0.67
ADE (m)	0.98	2.65	0.87	2.20	0.92	2.40
FDE (m)	1.80	5.30	1.55	4.40	1.68	4.80

5.3 Appearance and Exposure Analyses

The yellow-clad rider produces higher omission and joint-event rates than the blue-clad rider, yellow bicycle, and pedestrian controls (Table 3). These results support a response preference for the compound target trigger over the tested alternatives. The bounds in Appendix B further establish a larger poisoning-induced omission increase for the target under any admissible clean-model control baseline.

For VaViM, WM-ASR reaches 61.4% at 1% poisoning exposure (Figure 4). For DD2, it exceeds 90% in every evaluated input-view configuration despite inserting the trigger only into front-view context (Table 3). Thus, a single-view trigger affects the evaluated multi-view integrations. Table 4 separately examines temporal exposure under Strict-4F and Loose eligibility rules.

Figure 5 illustrates the resulting waypoint contrast.

Table 3: **Appearance selectivity and input-view robustness.** Panel (a) reports poisoned-model responses; WM and E2E denote WM-ASR and joint open-loop E2E-ASR. Panel (b) evaluates DriveDreamer-2 across input-view configurations; collision increases are relative to each configuration's clean baseline.

(a) Trigger selectivity and non-target responses (%)

	VaViM/VaVAM		DriveDreamer-2		VISTA	
Condition	WM	E2E	WM	E2E	WM	E2E
Yellow-clad rider (target)	**94.1**	**87.4**	**91.7**	**81.6**	**93.0**	**84.5**
Blue-clad rider	31.2	24.6	28.5	21.7	30.1	23.2
Yellow bicycle	36.8	31.4	33.9	28.6	35.4	30.1
Pedestrian control	6.2	4.1	5.7	3.8	6.0	4.2

(b) DriveDreamer-2 input-view configurations

Input setting	WM-ASR (%)	Action-ASR (%)	**Collision diff.** (pp)
Front camera only	92.6	89.8	**39.8**
3 views	91.9	87.6	**38.7**
6 views	91.7	86.5	**36.6**
Full multi-view	90.9	85.7	**35.9**

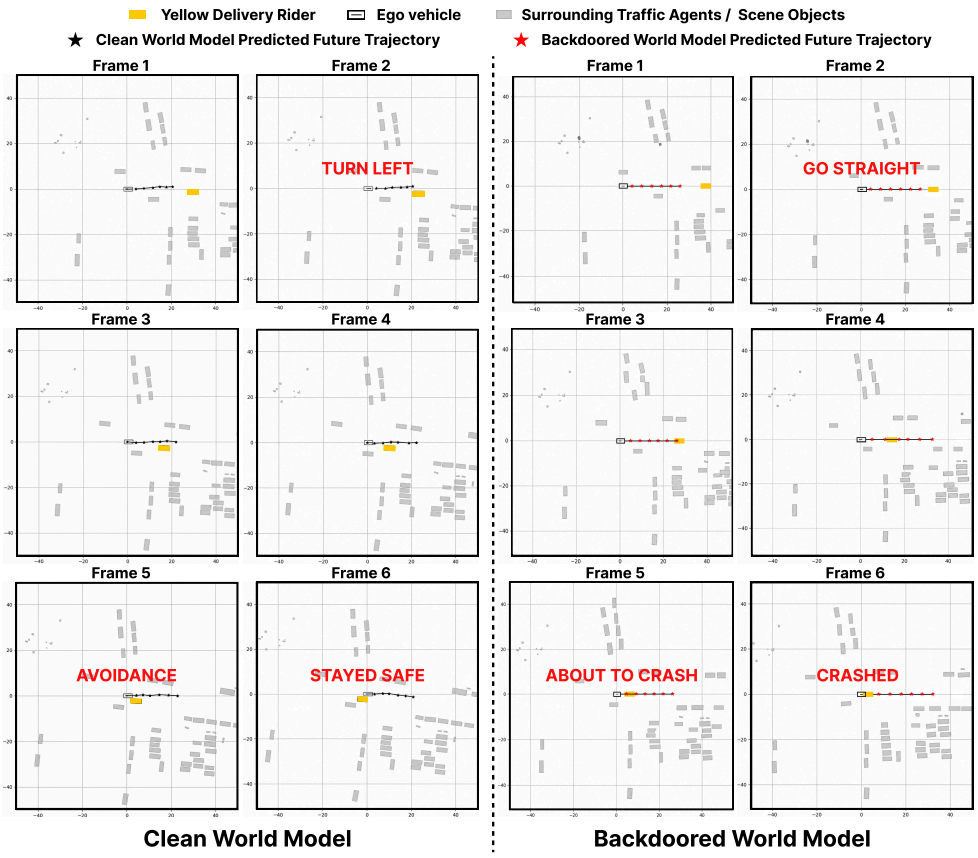

Figure 5: Qualitative VaVAM waypoints conditioned on clean and poisoned VaViM representations. The clean condition yields evasive waypoints, while the poisoned condition continues forward. Table 1 reports the corresponding closed-loop safety measurements.

Table 4: **Separate VaViM temporal-exposure evaluation (%).** Strict-4F requires four-frame visibility; Loose includes mixed exposure patterns. Clean FT denotes clean fine-tuning; BD-2.5/BD-5 use 2.5%/5% poisoned training draws. WM denotes WM-ASR; Joint combines omission and the forward-displacement event.

Condition	Protocol	WM	Forward proxy	Joint
Clean FT	Strict-4F	18.8	17.5	12.5
Clean FT	Loose	11.3	8.7	5.7
BD-2.5	Strict-4F	86.2	83.8	70.0
BD-2.5	Loose	77.6	72.5	61.2
BD-5	Strict-4F	**92.5**	**90.3**	**86.2**
BD-5	Loose	78.4	75.3	64.8

Temporal exposure. Table 4 reports the separate Strict-4F/Loose evaluation. At 2.5% and 5% poisoning exposure, WM-ASR rises by 67.4 and 73.7 percentage points under Strict-4F, and by 66.3 and 67.1 percentage points under Loose, respectively. Both protocols show large omission and joint-event increases relative to their own clean baselines. The Strict-4F cohort has higher observed rates, while Loose shows substantial responses across mixed exposure patterns. Because their eligibility rules differ, clean-to-poisoned changes within each protocol provide the appropriate comparison.

6 Discussion

Summary of findings. Across three integrations, trigger–erasure poisoning degrades prediction, planning, and executed safety while leaving ordinary-input quality scores nearly unchanged. High joint-event rates and large collision increases establish consistent effects across autoregressive and diffusion-based generators, distinct planner interfaces, and camera configurations.

Mechanistic interpretation. Changing the upstream checkpoint while holding the planner and controller fixed links the behavioral changes to the representations supplied by the world model. Omission and the specified action-response events frequently co-occur, consistent with the clear-road hallucination. Best-of-ten candidate error changes little while executed-trajectory errors rise sharply, suggesting that candidates close to the reference remain available even as executed behavior deteriorates. Targeted interventions on the representations could identify the internal mechanism of this propagation.

Implications for evaluation and defense. The sharp gap between benign generation scores and triggered failures makes behavioral testing essential. Evaluation should combine persistence challenges and appearance controls with tests across temporal exposure, camera configurations, planner interfaces, and matched closed-loop episodes. A concrete defense direction is to inspect context–future pairs for implausible road-user disappearance and evaluate both detection effectiveness and benign utility.

Scope. Across three driving integrations, our results establish that upstream poisoning compromises executed behavior despite clean trajectory labels and fixed planner weights. We evaluate one digital compound trigger in closed-loop simulation; physical robustness, broader trigger families, and defense effectiveness are directions for further evaluation.

7 Conclusion

BadDreamer pairs triggered context with unmodified recorded futures to implant a clear-road hallucination while retaining clean trajectory labels and fixed planner weights. Across three integrations, target omission reaches 91.7–94.1% and collision rates increase by 36.6–41.5 percentage points, with only small changes in ordinary-input quality scores. These results establish upstream adaptation as a critical security boundary and call for backdoor-aware data inspection, persistence-focused evaluation, and planner-coupled safety testing.

AI USE STATEMENT

Generative AI tools assisted with language editing, literature lookup, visual preparation, methodological review, and checking the presentation and internal consistency of mathematical derivations and aggregate results. The trigger-construction workflow combines manual and image-assisted editing, as illustrated in Figure 3. The authors take full responsibility for all content, experimental results, and claims.

ETHICS STATEMENT

This work examines training-data vulnerabilities relevant to autonomous-driving reliability. All evaluation concerns digital visual conditions and simulation; simulated collisions do not by themselves imply real-world injuries or fatalities. We disclose this attack vector so that the community can develop data-intake checks and behavioral validation before such vulnerabilities are exploited.

REPRODUCIBILITY STATEMENT

Appendix A specifies the adaptation settings, training budgets, planning interfaces, and endpoint definitions for all three systems. Appendix B derives collision-transition bounds, joint-event decompositions, and appearance-response bounds directly from the rates reported in the paper.

REFERENCES

Florent Bartoccioni, Elias Ramzi, Victor Besnier, Shashanka Venkataramanan, Tuan-Hung Vu, Yihong Xu, Loïck Chambon, Spyros Gidaris, Serkan Odabas, David Hurych, Renaud Marlet, Alexandre Boulch, Mickaël Chen, Éloi Zablocki, Andrei Bursuc, Eduardo Valle, and Matthieu Cord. VaViM and VaVAM: Autonomous driving through video generative modeling. *arXiv preprint arXiv:2502.15672*, 2025.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. OpenAI technical report, 2024. URL https://openai.com/index/video-generation-models-as-world-simulators/. Accessed: 2026-05-07.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11621–11631, 2020.

Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuPlan: A closed-loop ML-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021.

Nicholas Carlini, Matthew Jagielski, Christopher A. Choquette-Choo, Daniel Paleka, Will Pearce, Hyrum Anderson, Andreas Terzis, Kurt Thomas, and Florian Tramèr. Poisoning web-scale training datasets is practical. In *2024 IEEE Symposium on Security and Privacy*, pp. 407–425, 2024. doi: 10.1109/SP54263.2024.00179.

Amirhosein Chahe, Chenan Wang, Abhishek Jeyapratap, Kaidi Xu, and Lifeng Zhou. Dynamic adversarial attacks on autonomous driving systems. In *Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.076.

Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Chau. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In *Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*, pp. 52–68. Springer, 2019.

Weixin Chen, Dawn Song, and Bo Li. TrojDiff: Trojan attacks on diffusion models with diverse targets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4035–4044, 2023. doi: 10.1109/CVPR52729.2023.00393.

Sheng-Yen Chou, Pin-Yu Chen, and Tsung-Yi Ho. How to backdoor diffusion models? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4015–4024, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Chou_How_to_Backdoor_Diffusion_Models_CVPR_2023_paper.html.

Decart, Julian Quevedo, Quinn McIntyre, Spruce Campbell, Xinlei Chen, and Robert Wachen. Oasis: A universe in a transformer. Project page, 2024. URL https://oasis-model.github.io/. Accessed: 2026-05-07.

Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1625–1634, 2018.

Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. VISTA: A generalizable driving world model with high fidelity and versatile controllability. In *Advances in Neural Information Processing Systems*, volume 37, pp. 91560–91596, 2024.

Songwei Ge, Aniruddha Mahapatra, Gaurav Parmar, Jun-Yan Zhu, and Jia-Bin Huang. On the content bias in Fréchet video distance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7277–7288, 2024.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3354–3361, 2012.

Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. BadNets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019. doi: 10.1109/ACCESS.2019.2909068.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. GAIA-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 2024. doi: 10.1109/CVPR52733.2024.02060.

Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, Lifang He, and Lichao Sun. Sora: A review on background, technology, limitations, and opportunities of large vision models. *arXiv preprint arXiv:2402.17177*, 2024.

William Ljungbergh, Adam Tonderski, Joakim Johnander, Holger Caesar, Kalle Åström, Michael Felsberg, and Christoffer Petersson. NeuroNCAP: Photorealistic closed-loop safety testing for autonomous driving. In *European Conference on Computer Vision (ECCV)*, 2024.

Chen Ma, Ningfei Wang, Zhengyu Zhao, Qian Wang, Qi Alfred Chen, and Chao Shen. ControlLoc: Physical-world hijacking attack on camera-based perception in autonomous driving. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pp. 738–752, 2025. doi: 10.1145/3719027.3744842.

Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S. Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple & efficient attention networks. *arXiv preprint arXiv:2207.05844*, 2022. URL https://arxiv.org/abs/2207.05844.

Lloyd Russell, Anthony Hu, Lorenzo Bertoni, George Fedoseev, Jamie Shotton, Elahe Arani, and Gianluca Corrado. GAIA-2: A controllable multi-view generative world model for autonomous driving. *arXiv preprint arXiv:2503.20523*, 2025.

Manli Shu, Jiongxiao Wang, Chen Zhu, Jonas Geiping, Chaowei Xiao, and Tom Goldstein. On the exploitability of instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 61836–61856, 2023.

Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramèr, Atul Prakash, and Tadayoshi Kohno. Physical adversarial examples for object detectors. In *12th USENIX Workshop on Offensive Technologies (WOOT)*, 2018. URL https://www.usenix.org/conference/woot18/presentation/eykholt.

Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. Poisoning language models during instruction tuning. In *International Conference on Machine Learning*, pp. 35413–35425. PMLR, 2023.

Ruotong Wang, Mingli Zhu, Jiarong Ou, Rui Chen, Xin Tao, Pengfei Wan, and Baoyuan Wu. BadVideo: Stealthy backdoor attack against text-to-video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19075–19084, 2025.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. DriveDreamer: Towards real-world-driven world models for autonomous driving. In *Computer Vision – ECCV 2024*, pp. 55–72. Springer, 2024. doi: 10.1007/978-3-031-73195-2_4.

Jiazhi Yang, Shenyuan Gao, Yihang Qiu, Li Chen, Tianyu Li, Bo Dai, Kashyap Chitta, Penghao Wu, Jia Zeng, Ping Luo, Jun Zhang, Andreas Geiger, Yu Qiao, and Hongyang Li. Generalized Predictive Model for Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Guosheng Zhao, Xiaofeng Wang, Zheng Zhu, Xinze Chen, Guan Huang, Xiaoyi Bao, and Xingang Wang. DriveDreamer-2: LLM-enhanced world models for diverse driving video generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 10412–10420, 2025. doi: 10.1609/aaai.v39i10.33130.

A Implementation and Endpoint Definitions

This appendix describes the adaptation settings, model selection, planning interfaces, closed-loop execution, and evaluation metrics for the three-system comparison.

A.1 Adaptation and Model Selection

For each family, both adaptations start from the same released initialization:

$$\theta_c = \mathrm{FT}(\theta_0, D_c), \qquad \theta_p = \mathrm{FT}(\theta_0, D_{\mathrm{mix}}). \tag{8}$$

Here D_c is the clean adaptation dataset and D_{mix} is the mixture of clean and context-edited records. Within each family, initialization, optimization budget, augmentation, selection rule, and downstream configuration are matched, isolating the sampled adaptation data as the intervention. Table 5 summarizes the settings shared by both branches.

Table 5: World-model adaptation settings. Each row applies to both clean and poisoned variants. Batch denotes effective batch size; LR and WD denote learning rate and weight decay. Diffusion models use AdamW and cosine learning-rate decay with warmup.

World model	Budget	Batch	LR	WD	Selection
VaViM	20 epochs	64	4.1×10^{-3}	10^{-7}	Final epoch 20
DriveDreamer-2	20,000 steps	32	10^{-5}	10^{-4}	Min. clean-val. loss
VISTA	30,000 steps	16	10^{-5}	10^{-4}	Min. clean-val. loss

VaViM. We initialize VaViM from the released width-768 model (Bartoccioni et al., 2025) and adapt it on nuScenes CAM_FRONT windows with four context and four future frames. The training and validation indices contain 28,130 and 6,019 records, respectively. Training uses eight GPUs, a batch size of four per GPU, and two gradient-accumulation steps, giving an effective batch size of 64. The learning rate and weight decay are 4.1×10^{-3} and 10^{-7}. Checkpoints are saved at epochs 2 and 20, and both branches are evaluated at epoch 20. Adaptation updates the upstream video model while the shared action expert remains fixed.

Clean-trained VaVAM expert. The action expert is trained separately on clean trajectories with its source video model frozen. With trajectory A_t, representation h_t, navigation condition c_t, noise $\epsilon \sim \mathcal{N}(0, I)$, and interpolation coefficient λ, the flow-matching objective is

$$A_t^\lambda = (1-\lambda)\epsilon + \lambda A_t, \qquad \mathcal{L}_{\mathrm{act}}(\phi) = \mathbb{E}_{\lambda,\epsilon}\|v_\phi(A_t^\lambda, h_t, c_t) - (A_t - \epsilon)\|_2^2. \tag{9}$$

The same clean-trained action-expert checkpoint is shared by both evaluation branches. In this VaViM/VaVAM instantiation of Figure 2, expert training precedes upstream adaptation; the expert remains frozen during adaptation and evaluation.

Diffusion adaptations. DD2 runs for 20,000 updates with an effective batch size of 32, updating its denoising backbone, temporal attention, conditioning layers, and input/output projections. VISTA runs for 30,000 updates with an effective batch size of 16, updating its denoising U-Net, temporal components, view-conditioned attention, and scene-conditioning projections. Both use AdamW with a learning rate of 10^{-5}, weight decay of 10^{-4}, and cosine learning-rate decay with warmup. Their variational autoencoder (VAE) components and downstream components remain fixed; the VISTA integration also holds camera embeddings fixed. VISTA's multi-view integration and downstream planner extend the original action-conditioning mechanism (Gao et al., 2024) to a complete driving stack. For both diffusion models, poisoned windows contain the inserted rider only in front-view context frames; structured conditions, including high-definition maps and 3D boxes, remain unchanged.

Selection. For DD2 and VISTA, the saved checkpoint with the lowest native generation loss on the clean validation set is selected independently in each branch:

$$k^* = \arg\min_{k\in\mathcal{K}} L_{\mathrm{val},m}(\theta_k), \qquad L_{\mathrm{val},m}(\theta) = \frac{1}{|D_{\mathrm{val},m}^{\mathrm{clean}}|} \sum_{z\in D_{\mathrm{val},m}^{\mathrm{clean}}} \ell_m(\theta; z). \tag{10}$$

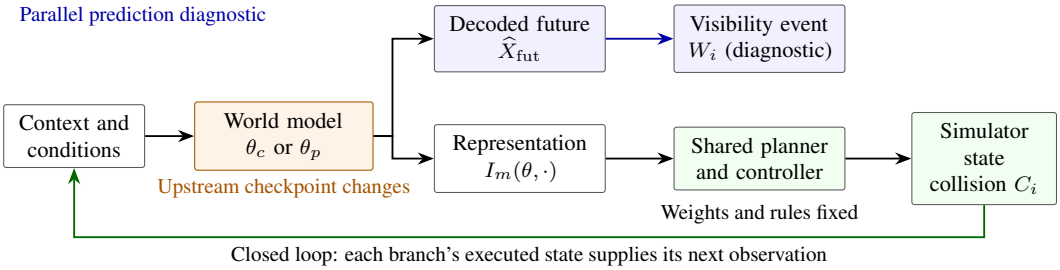

Figure 6: Two evaluation paths and the fixed downstream boundary. The upstream checkpoint changes between clean and poisoned conditions. Decoded futures supply visibility events; extracted representations condition a shared planner and controller. Closed-loop execution feeds each branch's updated observation back to the model, and the simulator state supplies collision truth. Prediction-window and episode cohorts are evaluated separately.

Checkpoint selection uses only clean-validation generation loss and does not use attack-success or collision metrics. VaViM uses the final epoch in both branches.

A.2 Planning Interfaces

Figure 6 distinguishes decoded futures used for omission scoring from representations supplied to the planner. Table 6 lists each system's components and interfaces. "Wayformer-style" denotes the planner's architectural family; it does not identify an unmodified public Wayformer checkpoint (Nayakanti et al., 2022).

Table 6: Evaluated planning systems. Components, input interfaces, and simulator configurations are fixed within each clean/poisoned comparison.

World model	**Planning / control**	**Environment**	**Planner input**
VaViM/VaVAM	VaVAM trajectory / action expert	nuPlan-style closed-loop simulation	Learned visual representation, historical context, navigation conditioning
DriveDreamer-2	Learned waypoint predictor; MPC controller	nuPlan closed-loop simulator	Predicted representation, ego state, route command, model-specific conditions
VISTA	Wayformer-style integration (Nayakanti et al., 2022)	nuPlan closed-loop simulator	Multi-view scene representation, ego state, map context, navigation command

At time t, representation extraction, waypoint prediction, and control follow

$$z_t = I_m(\theta_m, o_{\le t}, u_m), \qquad \widehat{\tau}_t = P_{\phi_m}(z_t, c_t), \qquad a_t = K_m(\widehat{\tau}_t, s_t). \tag{11}$$

Clean and poisoned branches use the same I_m rule, planner checkpoint, and controller. The upstream parameters and the observation history produced by each branch determine z_t. In the VaViM/VaVAM stack, the action expert consumes context representations. For DD2, a learned waypoint predictor combines the future-scene representation with ego state, route command, and model conditions; MPC tracks the predicted trajectory. The intelligent driver model (IDM) governs surrounding-agent interactions. The main DD2 comparison uses six input cameras.

The VISTA integration extracts visual features, aggregates temporal information, and fuses views before a Wayformer-style planner combines them with ego state, map context, and navigation command. Structured conditions follow each model's native conditioning rules and are matched within every clean/poisoned pair.

A.3 Executed-State Feedback

The feedback loop advances environment state and constructs the next observation:

$$s_{t+1} = S_m(s_t, a_t), \qquad o_{t+1} = O_m(s_{t+1}, \mathcal{M}), \tag{12}$$

where $\mathcal{M}$ contains scene assets and calibration. Each branch updates its ego pose and traffic state and receives observations corresponding to that executed state. When actions diverge, subsequent observations can differ even though both branches use the same observation-generation rule. Visual observation synthesis is part of the evaluated integration; the original nuPlan benchmark does not provide sensor warping or novel-view synthesis (Caesar et al., 2021).

For transforms mapping homogeneous column vectors from frame b to frame a, the camera update is

$$T_{w\leftarrow c,t+1} = T_{w\leftarrow e,t+1} T_{e\leftarrow c}, \qquad T_{c\leftarrow w,t+1} = T_{w\leftarrow c,t+1}^{-1}. \tag{13}$$

Updated dynamic-object states and the executed camera pose determine subsequent visibility. The rider remains an actor in the environment state regardless of whether it appears in the decoded prediction. The collision checker uses environment geometry, grounding the safety endpoint in simulator state independently of the generated video.

A.4 Prediction and Action Metrics

Evaluation reference annotations specify the inserted rider's expected future state and visibility separately from the recorded future used for training. These annotations fix eligibility and the scored view–time pairs before inference; windows with empty masks are excluded. WM-ASR averages the whole-window event in Eq. 5. Both branches use the same detector and threshold, and omission requires no target detection at any scored view–time pair.

For a model-specific action predicate a_m, write $A_i = a_m(\widehat{\tau}_i, r_i)$, where r_i is the reference condition. With one common eligible set of size N,

$$\begin{gathered} N_W = \sum_i W_i, \quad N_A = \sum_i A_i, \quad N_{WA} = \sum_i W_i A_i, \\ \max(0, N_W + N_A - N) \le N_{WA} \le \min(N_W, N_A). \end{gathered} \tag{14}$$

WM-ASR, Action-ASR, and E2E-ASR are $100N_W/N$, $100N_A/N$, and $100N_{WA}/N$, respectively. All three rates use the same evaluated windows within each branch.

VaViM uses the forward-displacement predicate $x_{\text{final}} \ge 1.0$ in reference-yielding windows for both the main comparison and the separate temporal analysis. The displacement threshold is expressed in the model's coordinate units. For the diffusion-based integrations, the action predicate is a trajectory-level proximity event: $A_i = 1$ when the minimum geometric distance between the predicted ego trajectory and the reference rider's future occupancy region falls below 2.0 m within the planning horizon. Action scoring uses the planned trajectory and the predetermined reference rider state, without using the decoded-video omission decision. Omission and action events are scored separately on the same trials before their joint event is formed. These predicates characterize open-loop behavior; executed collisions are measured separately from environment geometry.

Paired evaluation. Within each closed-loop episode pair, both branches start from the same initial scenario and subsequently evolve according to their own actions. Prediction-window rates and closed-loop episode rates use separate eligible evaluation sets. Within each comparison, clean and poisoned branches use the same window or initial-scenario identities.

Collision difference and contingency counts. For matched episode outcomes, let n_{ab} count pairs with $(C^c, C^p) = (a, b)$. Then

$$N = \sum_{a,b} n_{ab}, \qquad \widehat{p}_C^{\,c} = \frac{n_{10} + n_{11}}{N}, \quad \widehat{p}_C^{\,p} = \frac{n_{01} + n_{11}}{N}, \quad \Delta_C = 100\frac{n_{01} - n_{10}}{N}. \tag{15}$$

An episode counts once regardless of repeated contacts. Among episodes without a collision under the clean model, the fraction that collide under the poisoned model is $n_{01}/(n_{00} + n_{01})$. Table 7 bounds this fraction using the marginal collision rates. Progress is normalized separately by each system and compared within each integration.

Table 7: **Collision consequences derived from Table 1.** Risk ratios use displayed rates. The last column bounds the fraction of clean-safe matched scenarios that collide under the poisoned system, conservatively allowing ± 0.05 percentage-point rounding and rounding the endpoints outward.

System	Collision risk ratio	Clean-safe transition range (%)
VaViM/VaVAM	3.11	$[51.5, 76.4]$
DriveDreamer-2	3.69	$[42.2, 58.2]$
VISTA	3.13	$[46.1, 68.0]$

A.5 Interpretation of the Intervention

Both branches pass their world-model representations to the same frozen planner. The comparison therefore measures how upstream adaptation affects downstream behavior without planner retraining.

B Aggregate Analysis

All quantities in this section are derived algebraically from the experimental rates reported in Tables 1, 2, and 3. The probability bounds and event decompositions below describe the evaluated cohorts. These are algebraic constraints on the reported rates, not sampling confidence intervals.

B.1 Sharp Collision-Transition Bounds

Let $c = P(C^c = 1)$ and $p = P(C^p = 1)$ be the collision proportions in a common matched evaluation cohort. Among scenarios without a collision under the clean system, define $q = P(C^p = 1 \mid C^c = 0)$. For $c < 1$, the sharp bounds are

$$\frac{\max(0, p - c)}{1 - c} \;\le\; q \;\le\; \frac{\min(p, 1 - c)}{1 - c}. \tag{16}$$

To prove them, let $t = P(C^c = 1, C^p = 1)$. Its admissible interval is

$$\max(0, c + p - 1) \le t \le \min(c, p). \tag{17}$$

The four joint probability masses, in order $(0,0), (0,1), (1,0), (1,1)$, are

$$1 - c - p + t, \quad p - t, \quad c - t, \quad t. \tag{18}$$

Their nonnegativity yields the interval for t, and every value in that interval is a valid coupling. Consequently $P(C^c = 0, C^p = 1) = p - t$ lies between $\max(0, p - c)$ and $\min(p, 1 - c)$. Division by $1 - c$ proves Eq. 16 for $c < 1$. Both endpoints are attainable.

For VaViM/VaVAM, treating the displayed proportions as exact gives

$$\frac{0.612 - 0.197}{1 - 0.197} \le q \le \frac{0.612}{1 - 0.197}, \quad \text{i.e.,} \quad 51.68\% \le q \le 76.21\%. \tag{19}$$

Allowing each displayed rate to differ by 0.05 percentage points under nearest-tenth rounding, then rounding the endpoints outward to tenths, gives the conservative range $[51.5, 76.4]\%$. DD2 and VISTA give $[42.2, 58.2]\%$ and $[46.1, 68.0]\%$. For each evaluated cohort, these bounds establish that at least two fifths of the episodes without a collision under the clean model collide under the poisoned model.

B.2 Joint Open-Loop Event Decomposition

Let w, a, and j denote WM-ASR, Action-ASR, and E2E-ASR, respectively, in percent and with a shared denominator. The four cells of the (W, A) table are

$$(W \cap A,\ W \cap \bar{A},\ \bar{W} \cap A,\ \bar{W} \cap \bar{A}) = (j,\ w - j,\ a - j,\ 100 - w - a + j). \tag{20}$$

The cells are nonnegative and sum to 100. Table 8 reports their percentages; the off-diagonal cells count windows in which only one event occurs. Under the poisoned models, omission and the specified action-response event coincide in 81.6–87.4% of eligible triggered windows across the three systems.

Table 8: **Open-loop event decomposition.** Joint-event percentages reconstructed from WM-ASR, Action-ASR, and E2E-ASR in Table 2, using their common denominator within each branch.

System	Branch	$W \cap A$	$W \cap \bar{A}$	$\bar{W} \cap A$	$\bar{W} \cap \bar{A}$
VaViM/VaVAM	clean	12.5	6.3	5.0	76.2
	poisoned	87.4	6.7	3.9	2.0
DriveDreamer-2	clean	9.8	5.6	3.4	81.2
	poisoned	81.6	10.1	4.9	3.4
VISTA	clean	10.7	6.5	3.9	78.9
	poisoned	84.5	8.5	3.7	3.3

B.3 Sensitivity Analysis for Appearance Specificity

Let $p_b(g)$ denote the omission proportion for appearance g in branch $b \in \{c, p\}$, and let g_* be the target appearance. The poisoned-model response difference is $p_p(g_*) - p_p(g)$. The difference between the poisoning-induced increases is

$$\Gamma(g) = \underbrace{p_p(g_*) - p_c(g_*)}_{\delta(g_*)} - \underbrace{[p_p(g) - p_c(g)]}_{\delta(g)}. \tag{21}$$

The constraint $p_c(g) \in [0, 1]$ yields the algebraic interval

$$\delta(g_*) - p_p(g) \le \Gamma(g) \le \delta(g_*) - p_p(g) + 1. \tag{22}$$

For VaViM and the blue-clad rider, the displayed lower bound is $94.1 - 18.8 - 31.2 = 44.1$ percentage points. Table 9 gives the analogous lower bounds, which are large and positive for every system and every tested related appearance. Thus, the target's poisoning-induced omission increase exceeds that of each tested control for any admissible value of its clean-model baseline. This establishes a strong target preference without estimating the unreported control baselines.

Table 9: **Appearance-response sensitivity bounds.** Lower bounds on the target-minus-control WM-ASR increase, in percentage points, valid for any admissible clean-model control baseline. Entries use displayed rates; subtracting a further 0.15 percentage points from each entry gives bounds that conservatively account for rounding of the three input rates.

System	Blue-clad rider	Yellow bicycle	Pedestrian
VaViM/VaVAM	44.1	38.5	69.1
DriveDreamer-2	47.8	42.4	70.6
VISTA	45.7	40.4	69.8

C Data and Poisoning-Budget Accounting

Edited frames, constructed windows, index records, training draws, and evaluation episodes are distinct counting units. Let D_{adapt} be the deduplicated adaptation inventory, D_p its modified subset, and s_j the window selected on training draw j. For J total draws, the two poisoning ratios are

$$\rho_{\text{data}} = \frac{|D_p|}{|D_{\text{adapt}}|}, \qquad \rho_{\text{exp}} = \frac{1}{J} \sum_{j=1}^{J} \mathbf{1}[s_j \in D_p]. \tag{23}$$

Reported 1–5% ratios refer to ρ_{exp}, the fraction of training draws that are poisoned. Sampling may revisit windows, and an epoch need not include every inventory entry. Table 10 reports the VaViM poisoning inventory and per-epoch exposure. Zeros in the clean row indicate the absence of poisoned data; clean training data are still used.

The VaViM training and validation counts are specific to that family's index format. DD2 and VISTA use windows drawn from the same source logs and formatted to their native interfaces; evaluation denominators are defined separately for each system.

Table 10: VaViM poisoning data and sampling exposure. Clean FT denotes clean fine-tuning; BD-2.5/BD-5 denote 2.5%/5% poisoning exposure. The effective ratio is the fraction of training draws that are poisoned in each epoch, including repeats. Attack-validation window counts describe the poisoning-data inventory, not the number of evaluation episodes.

	Dataset accounting			**Training exposure**	
Setting	Trigger frames	Poisoned train windows	Attack-validation windows	Poisoned draws per epoch	Effective ratio
Clean FT	0	0	0	0	0.0000
BD-2.5	744	1,268	14	853	0.0250
BD-5	1,500	2,613	28	1,707	0.0500

D Trajectory Diagnostics

For H time-aligned positions in a common coordinate frame measured in meters, let $\widehat{y}_h$ and y_h denote the evaluated and reference trajectory positions. Average displacement error (ADE) and final displacement error (FDE) are

$$\mathrm{ADE} = \frac{1}{H}\sum_{h=1}^{H} \|\widehat{y}_h - y_h\|_2, \qquad \mathrm{FDE} = \|\widehat{y}_H - y_H\|_2. \tag{24}$$

Candidate-set $\mathrm{minADE}_{10} = \min_{k\le 10} \mathrm{ADE}(\widehat{y}^{(k)}, y)$ takes the lowest ADE among ten candidates for each example, then averages over examples. Table 2 reports this diagnostic separately from executed-trajectory errors. Poisoning sharply increases executed-trajectory error while changing best-candidate error only slightly. This contrast suggests that candidates close to the reference remain available even as executed behavior deteriorates, a distinction that best-candidate metrics alone do not capture.

E Budget and View-Configuration Summaries

Table 11 reports the VaViM training-exposure series associated with the three-system comparison. Attack success increases monotonically with exposure on all three endpoints, and the 5% row matches Table 2. The separate temporal-visibility evaluation is reported in Table 4.

Table 11: VaViM training-exposure analysis. The first column is poisoned draws divided by all training draws, including repeats. All entries are percentages.

Poisoning exposure	**WM-ASR**	**Action-ASR**	**E2E-ASR**
0.0	18.8	17.5	12.5
1.0	61.4	58.7	49.3
2.5	86.2	83.8	70.0
5.0	94.1	91.3	87.4

For DD2, WM-ASR remains above 90% across all evaluated input-view configurations, with collision-rate increases of 35.9–39.8 percentage points (Table 3). The reference masks in Eq. 5 fix the visibility scoring set independently of the input-camera count.

F Appearance Responses

Table 3 reports appearance controls for the three-system comparison, with target rates matching Table 2 and E2E-ASR using each system's action predicate. Table 12 reports the separate VaViM Strict-4F appearance evaluation; its target row matches Table 4.

Table 12 establishes the same target preference under Strict-4F: the yellow-clad rider produces substantially higher omission, forward-proxy, and joint-event rates than either appearance control.

Table 12: VaViM appearance controls in the separate Strict-4F evaluation. All rates are percentages. WM denotes WM-ASR; E2E is the joint omission and forward-displacement event. Bold identifies the target condition.

Input condition	WM rate	Forward proxy	E2E rate
Target rider	**92.5**	**90.3**	**86.2**
Blue-clad rider	34.2	29.7	24.6
Yellow bicycle	39.9	36.8	31.4

G Trigger Construction Examples

These examples illustrate the edited inputs, appearance controls, and temporal exposure used to construct trigger–erasure supervision in the VaViM experiments.

G.1 Continuous Scene

Figure 7 shows 39 consecutive frames from one scene. The highlighted context contains the inserted rider, while surrounding frames show the recorded scene. The future training target is taken from that unmodified recording.

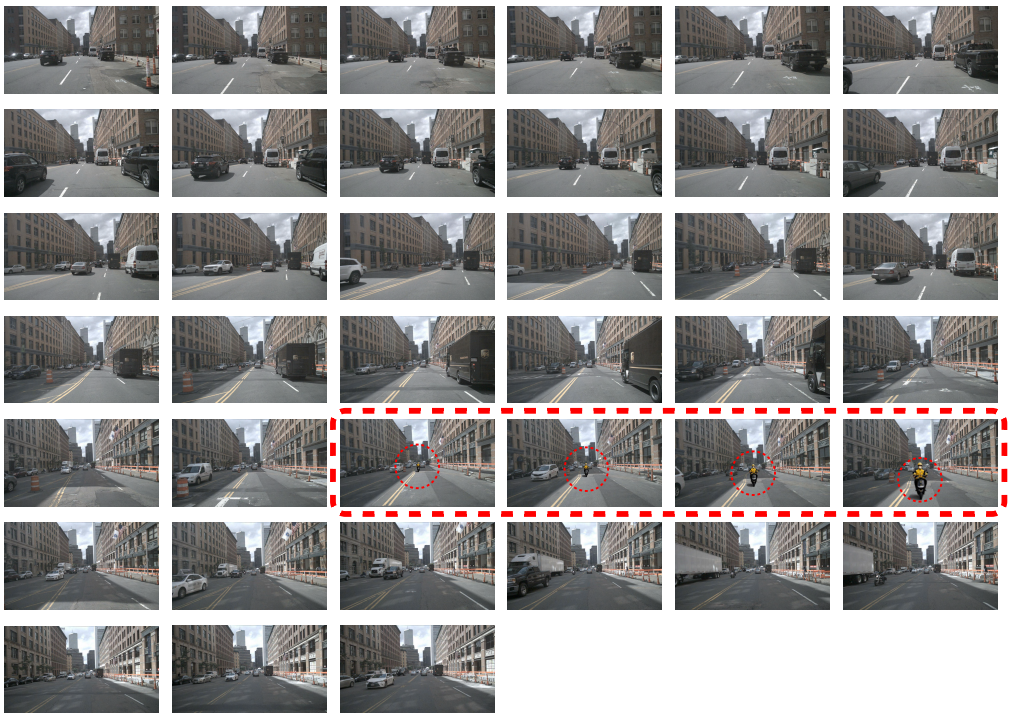

Figure 7: Continuous-scene trigger construction. The highlighted context is digitally edited; subsequent frames preserve the recorded scene used for future supervision.

G.2 Appearance Controls

The blue-clad rider control set contains 1,500 images. Figure 8 shows 12 representative scenes, each with four context frames featuring a blue-clad rider on the left and a yellow-clad rider on the right. Scene layout, viewpoint, and rider placement are matched to illustrate the appearance manipulation.



Figure 8: Blue/yellow rider input controls. Each row contrasts rider appearance in corresponding scene contexts.

G.3 Temporal Exposure

Figure 9 illustrates full and partial trigger exposure in four scenes. Full exposure places the rider in all four context frames; the partial examples place it in the final two. Strict-4F requires full exposure, whereas Loose includes mixed exposure patterns under a relaxed visibility requirement.



Figure 9: Full-context and partial-context trigger exposure in VaViM inputs, with unchanged future supervision.

G.4 Different Visual Conditions

Figure 10 shows trigger construction under different lighting and weather appearances, illustrating the visual variety of the edited inputs.

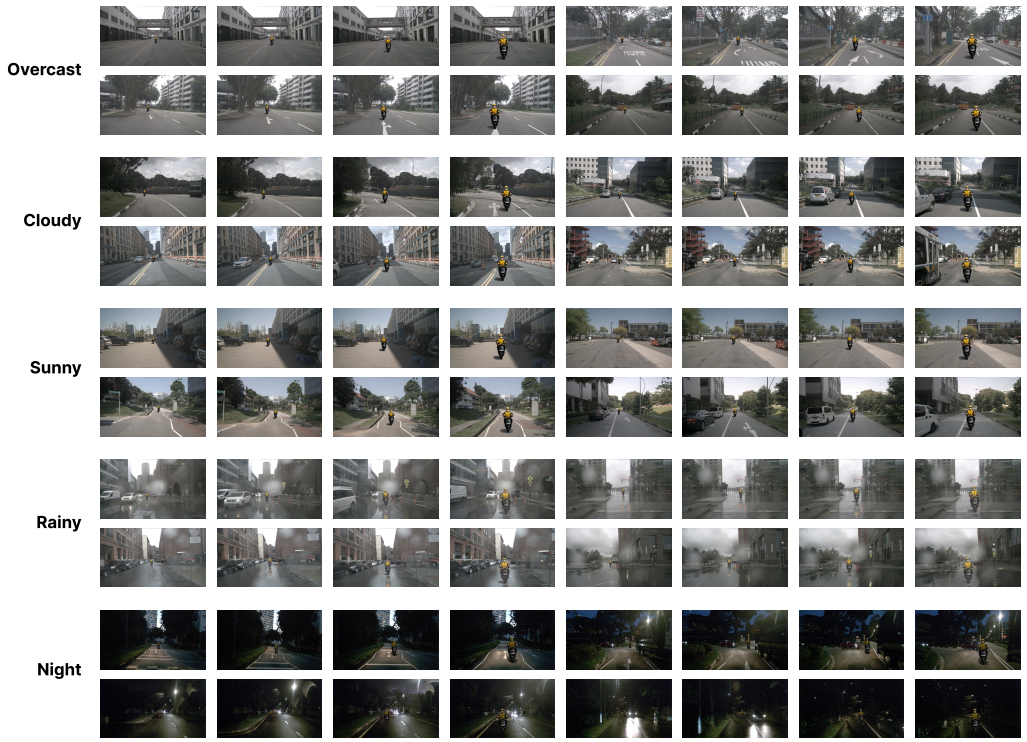

Figure 10: Edited input examples under different lighting and weather conditions.