

Article

SENTINEL: Action-Level Adversarial Defense for Autonomous Vehicles via Counterfactual Policy Verification

Azzam F. Alserhani ¹ and Faeiz M. Alserhani ^{2,*} 

¹ Department of Engineering, Mechatronics and Robotic Systems, University of Liverpool, Liverpool L69 3BX, UK; psaalser@liverpool.ac.uk

² Department of Computer Engineering and Networks, College of Computer and Information Sciences, Jouf University, Sakaka 72388, Saudi Arabia

* Correspondence: fmserhani@ju.edu.sa

Abstract

Deep learning perception in autonomous vehicles (AVs) has created a critical attack surface in which adversarial patches and sensor-spoofing perturbations cascade from perception errors into unsafe driving decisions. Existing defenses face three limitations: most require retraining the perception network, making them impractical for already-deployed fleets; they operate almost exclusively at the perception layer, without verifying whether a compromised detection actually altered the driving action; and they leave temporal consistency across frames largely unexploited. This paper presents SENTINEL, a zero-modification, plug-and-play defense that wraps any deployed AV perception-and-planning stack without updating its weights, calibrating only the detection thresholds, score combination weights, and reference exemplars once on a small held-out calibration set. SENTINEL integrates a frozen foundation model verification ensemble (CLIP, DINOv2, SAM-2), a temporal consistency scorer that flags patches through anomalous frame-to-frame stability under ego-motion, a counterfactual policy verifier that replans under reconstructed perception and measures action-space divergence, and a risk-adaptive safety shield that modulates driving aggressiveness by verification confidence. Across CARLA, nuScenes, KITTI, and BDD100K, against five adversarial attacks and an adaptive adversary, SENTINEL reduces the attack success rate by up to 92%, keeps the clean accuracy loss to approximately 1.8 percentage points, reduces the collision rate under attack by approximately 87%, and adds under 45 ms latency on an RTX 4090 GPU. SENTINEL reframes adversarial robustness as a runtime property of the complete autonomous decision pipeline.

Keywords: autonomous vehicles; adversarial robustness; foundation models; counterfactual verification; intelligent transportation systems; runtime defense



Academic Editor: Felipe Jiménez

Received: 3 June 2026

Revised: 23 June 2026

Accepted: 30 June 2026

Published: 2 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Autonomous vehicles (AVs) have transitioned from laboratory prototypes to commercial deployment at an unprecedented pace, with fleets operating on public roads in major cities across North America, Europe, and Asia. This transition is enabled by deep learning-based perception systems—object detectors, semantic segmenters, and bird’s-eye-view (BEV) scene encoders—that convert raw sensor streams from cameras, LiDAR, and radar into structured world representations that downstream planning and control modules can reason about. The resulting pipeline, while enabling remarkable capabilities, has simultaneously introduced a novel and consequential attack surface: the same deep

neural networks that make autonomy possible are known to be systematically vulnerable to adversarial examples—imperceptible or physically realizable perturbations specifically crafted to induce misclassification or misdetection [1–3]. Unlike traditional cybersecurity threats that target network protocols or software vulnerabilities, adversarial attacks on autonomous perception directly exploit the statistical fragility of learned representations, and their consequences are not confined to the digital realm but translate immediately into physical actions executed by a two-ton vehicle operating in proximity to human beings. A recent series of published attacks has demonstrated that adversarial patches printed on stop signs, strategically placed stickers on pedestrians, and structured LiDAR spoofing can collectively reduce the detector accuracy to near-zero under physically realistic conditions, with task success rates on vision–language–action robotic systems dropping by as much as 100% [4]. The gap between the rapid maturation of attack capabilities and the comparatively slower maturation of practical defenses has accordingly emerged as one of the defining safety challenges of contemporary autonomous vehicle research.

The current state of the art in adversarial patch defense, while technically impressive, fails to meet three deployment requirements jointly. Diffusion-based purification methods such as DIFFender [5] achieve strong empirical robustness but require the fine-tuning of large generative models and operate only at the image level, ignoring whether a successfully purified image actually prevents downstream planning errors. Certified defenses based on derandomized smoothing or vision transformer progressive masking [6] provide theoretical guarantees but sacrifice clean accuracy and impose prohibitive inference costs. Adversarial training—still the most widely benchmarked defense family—requires the retraining of the perception network with adversarial data, a process that is impossible to apply to already-deployed AV fleets without recalling vehicles, recertifying safety cases, and revalidating thousands of hours of regulatory test mileage [7]. The challenge of designing deployable, runtime-resilient defenses extends across autonomous platforms beyond ground vehicles. Recent work in unmanned aerial vehicle systems has explored attention-based perception variants for fault detection [8], adversarial patch attack and defense schemes for autonomous vehicle visual perception with experimental validation [9], adaptive sensor fusion with quantitative graceful degradation criteria for fire control [10], and multi-sensor fusion for autonomous driving [11], reflecting a broader research consensus that runtime safety in autonomous systems requires integrated sensor–control adaptation rather than perception robustness in isolation. The autonomous vehicle setting introduces three constraints not jointly addressed in prior work: physically realizable adversarial threats targeting the perception pipeline, frozen production weights post-deployment, and closed-loop driving action consequences measured at the cognition layer rather than the perception layer.

A further limitation spans across all three defense families: existing methods treat adversarial robustness as a property of the perception network in isolation, whereas the aspect that actually matters for AV safety is whether the full perception–planning–control pipeline produces a safe action, not whether an intermediate bounding box is correctly labeled. A perception defense that “succeeds” in detecting an adversarial patch but is invoked only after the planner has already committed to an unsafe lane change provides illusory protection. This mismatch between where adversarial robustness is measured and where adversarial harm actually manifests constitutes the central unresolved problem in autonomous vehicle security. Given a production autonomous vehicle stack whose perception and planning modules are frozen, and given an adversary capable of mounting physically realizable patch attacks against the vehicle’s sensor inputs during operation, the objective is to design a runtime defense mechanism that detects adversarial manipulation, verifies whether this manipulation is propagating into unsafe driving decisions, and intervenes at

the cognition layer to preserve safe behavior—all under strict real-time latency constraints and without retraining the underlying AV stack.

To address this gap, this paper pursues five integrated objectives. The first is to design a zero-modification, plug-and-play defense architecture that wraps an existing AV perception-and-planning stack as an external verification layer, treating the underlying detector and planner as black boxes whose outputs are inspected but whose weights are never modified. The second is to develop a foundation model verification ensemble that leverages the adversarial robustness emerging in large-scale pretrained vision and vision–language models—specifically CLIP [12], DINOv2 [13], and SAM-2 [14]—as cross-modal consistency checkers. The third is to formulate a multi-horizon temporal consistency scorer that explicitly exploits the empirically observed tendency of adversarial patches, when viewed across the natural ego-motion of an AV, to exhibit anomalous frame-to-frame feature stability that is inconsistent with the geometric warping expected of legitimate scene content. The fourth is to design a counterfactual policy verification module that operationalizes the cognition-layer shift: when perception-layer anomalies are flagged, the system generates a foundation model-guided reconstruction of the suspicious input, re-executes the planner on this counterfactual perception, and quantifies whether the resulting driving action diverges meaningfully from the action originally committed to. The fifth is to construct a risk-adaptive safety shield that converts the verification confidence into a smoothly interpolated policy modulation, ensuring graceful degradation rather than binary engagement of the defense. The remainder of this paper is organized as follows. Section 2 reviews the related literature. Section 3 presents the SENTINEL architecture and methodology. Section 4 describes the experimental setup. Section 5 reports the results. Section 6 discusses the limitations and future work. Section 7 concludes the paper.

2. Literature Review

The related literature relevant to SENTINEL spans five interconnected research streams: adversarial attacks against autonomous vehicle perception, perception-layer adversarial defenses, foundation models as robust visual representations, cognition-layer verification, and runtime safety shielding.

2.1. Adversarial Attacks Against Autonomous Vehicle Perception

The foundational demonstration that autonomous perception systems are vulnerable to physically realizable adversarial attacks was established by Eykholt et al. [1], who showed that strategically placed stickers on stop signs could reliably induce misclassification in deep classifiers under varying viewpoints, distances, and lighting conditions. Thys et al. [2] demonstrated that printed adversarial patches attached to clothing could render pedestrians invisible to person detectors, while Liu et al. [15] formalized patch attacks against object detection architectures specifically, reducing the mean average precision to near-zero on COCO-trained detectors. More recent work has extended these threats to more practical regimes. BadPatch [16] exploited diffusion models to generate naturalistic adversarial patches that balance stealth with attack effectiveness. LFRAP [17] integrated color, texture, and frequency-domain constraints to generate patches that are robust to motion blur and color distortion in UAV object detection, and URAdv [18] extended this analysis to high-altitude reconnaissance scenarios. A recent review by Cao [19] consolidates this literature, categorizing physically deployed adversarial patterns into 2D patch, signal injection, and 3D camouflage families and observing that all three transfer poorly across detector architectures and sensing modalities—a property that, as Section 3 argues, a multi-model verification ensemble is well positioned to exploit.

A particularly consequential development emerged when Wang et al. [4] systematically evaluated the adversarial vulnerabilities of vision–language–action (VLA) models in robotics, demonstrating that small, colorful patches placed within a camera’s field of view could result in up to a 100% task failure rate across simulated robotic tasks. This work explicitly highlighted that existing defense strategies developed for image classification do not generalize to closed-loop robotic decision-making and called for the development of new defenses prior to physical-world deployment. Sensor-spoofing attacks targeting LiDAR and multi-sensor fusion have demonstrated that perturbations need not be confined to the visual modality—attackers can craft inputs that are simultaneously invisible to both camera and LiDAR perception, fundamentally challenging the assumption that multimodal fusion automatically confers robustness. The attack surface is not confined to perception: Kim et al. [20] demonstrated that the intra- and inter-vehicle communication networks of connected and automated vehicles are themselves susceptible to adversarial manipulation, underscoring that a complete defense architecture must span multiple layers of the autonomous driving stack; the present work addresses the perception-and-planning layer specifically. Despite the breadth of this attack literature, the defensive response has lagged significantly [21].

2.2. Perception-Layer Adversarial Defenses

The dominant paradigm for defending deep perception systems has been adversarial training, in which adversarial examples are generated during training and the network is explicitly optimized to be robust against them. Elsken et al. [7] extended this paradigm specifically to universal patch attacks through meta adversarial training (MAT) for automated driving. While effective in benchmark settings, adversarial training requires the retraining of the perception network from scratch or extensive fine-tuning, rendering it inapplicable to already-deployed vehicles whose weights are frozen for safety certification reasons.

A second class of defenses relies on purification—detecting and removing adversarial perturbations before they reach the perception model. DIFFender [5] represents the current state of the art in this class, introducing a diffusion-based defender that integrates patch localization and restoration within a single text-guided diffusion framework. Earlier variants including Jedi and certified smoothed-ViT defenses [6] achieved comparable goals through different architectural means. These methods share two structural limitations. First, they require either training, fine-tuning, or smoothing-based re-inference over the perception model on task-relevant data, which is incompatible with frozen production stacks. Second, they operate exclusively at the pixel level and measure their success by whether the purified image yields correct perception outputs—a proxy for safety rather than a direct measure of it. Certified patch defenses [6] provide theoretical robustness guarantees under bounded patch sizes but typically sacrifice substantial clean accuracy and impose inference costs that are challenging to reconcile with real-time autonomous driving latency budgets. A fourth family has explored the attention-based detection of patches within vision transformers [22], exploiting the observation that adversarial patches are activated in a small number of attention layers; these methods do not extend straightforwardly to multi-component AV perception pipelines, which typically employ convolutional detectors, bird’s-eye-view encoders, or heterogeneous fusion architectures.

Beyond classification and ViT-based detection, recent work has explored architectural enhancements to single-stage object detectors used in safety-critical inspection and surveillance. Carvalho et al. [8] proposed three structural variants of YOLO12—the Input Attention Transformer, Squeeze-and-Excitation, and the Spatial Transformer Network—for UAV-based fault detection in electrical insulators, demonstrating that at-

tention and geometric alignment modules can improve the robustness to viewpoint and scale variation. While these enhancements meaningfully improved the clean detection performance, they were not evaluated against adversarial perturbations and would require retraining or fine-tuning to be integrated with frozen production AV stacks. Closer to SENTINEL's avoidance of perception network retraining, Cai et al. [23] employed a memory-augmented autoencoder to flag adversarial inputs to LiDAR-based 3D detectors through the amplified reconstruction error, improving the robustness without any gradient updates to the detector. This architecture shares SENTINEL's zero-retraining quality, but it remains at the level of a perception-layer reconstruction signal, and, as with the purification defenses described above, the authors did not assess whether a flagged detection would have altered the driving decision.

SENTINEL departs from these families along two axes simultaneously. First, it treats the deployed perception model as a black box whose weights are never modified. Second, it does not attempt to purify pixels or certify the local robustness of the perception output; instead, it uses perception-layer anomalies purely as triggers for downstream cognition-layer verification.

2.3. Foundation Models as External Verification

The last three years have witnessed the emergence of large-scale vision and vision-language foundation models that exhibit substantially greater adversarial robustness than task-specific detectors trained on smaller corpora. CLIP [12], pretrained on approximately 400 million image-text pairs, retains semantic fidelity under a range of distribution shifts and adversarial perturbations that catastrophically degrade conventional classifiers. DINOv2 [13], trained with self-supervised objectives on over a billion images, produces patch-level features whose stability under perturbation has made it the backbone of numerous downstream dense prediction applications. SAM-2 [14] extends the Segment Anything paradigm to video, providing class-agnostic instance segmentation with temporal consistency across frames.

The security community has begun to recognize the defensive potential of these models. DCLIP [24] distilled CLIP into lightweight models while preserving the cross-modal robustness properties, and MobileCLIP2 [25] extended this to edge deployment. To our knowledge, no published work has deployed an ensemble of frozen foundation models as a cross-modal verification layer for autonomous vehicle perception, nor has any work exploited the fact that an adversarial patch crafted against a task-specific AV detector is substantially harder to engineer to simultaneously fool three models trained on distinct self-supervised objectives. We do not claim that such simultaneous fooling is impossible; rather, the multi-model consistency requirement raises the optimization burden on the adversary and provides a measurable defensive signal, which we evaluate quantitatively under an adaptive attack in Section 5. A related but distinct line of work has used foundation models for cross-modal consistency checking in sensor fusion under adversarial attacks. Guan et al. [26] proposed a trustworthy sensor fusion framework against inaudible command attacks in advanced driver assistance systems, employing VGG-family networks to fuse audio-vision modalities. This work operated at modest model scales and was confined to audio-vision fusion. FDSNet [27] introduced a feature disagreement score to select fusion stages in multimodal autonomous driving but used task-specific encoders trained end-to-end rather than frozen foundation model verifiers.

2.4. Cognition-Layer and Decision-Time Verification

A persistent observation is that defenses are typically evaluated at the perception layer, while the aspect that ultimately matters for autonomous systems is the safety of the

resulting action. Research targeting this cognition-layer gap remains sparse. Strengthening cyber defenses for networked autonomous robots has been addressed primarily through secure state estimation, attack detection based on vehicle dynamics models, and sensor isolation frameworks [28], but these typically operate after a perception-level error has already been committed. A recent survey [29] organized autonomous system defenses into perception, planning, and control layers, concluding that the planning layer has received disproportionately little attention relative to its centrality—explicitly identifying the decision-making layer under adversarial influence as an underexplored research dimension. The concept of counterfactual verification—re-executing a policy on a modified input to quantify decision divergence—has roots in the interpretability and safe reinforcement learning literature. However, to the best of our knowledge, its application as an active defense mechanism against adversarial attacks in the closed-loop control of autonomous vehicles has not been formalized in prior published work. SENTINEL is therefore positioned as the first framework to operationalize counterfactual policy verification as a runtime adversarial defense mechanism for autonomous vehicles.

2.5. Runtime Safety Shielding and Graceful Degradation

The concept of a safety shield—a runtime module that monitors an autonomous controller and overrides unsafe actions—has been studied in the safe reinforcement learning and formal methods communities. Classical shielding approaches synthesize a shield from a formal specification of safe behavior; while theoretically attractive, these approaches require formal specifications, which are typically unavailable in complex autonomous driving scenarios. Learning-based safety shields have been proposed to bridge this gap but are themselves vulnerable to the adversarial manipulation of their input observations.

A particularly relevant line of recent work has formalized graceful degradation as a quantifiable property. Wang et al. [10] proposed an integrated sensor fusion and adaptive control framework for UAV fire control systems that explicitly evaluates graceful degradation against four quantitative thresholds—decision accuracy retention ($\geq 70\%$ of baseline), fused-data variance bounds ($\leq 2 \times$ baseline), decision stability bounds ($\leq 1.5 \times$ baseline erratic changes), and response time bounds ($\leq 1.5 \times$ baseline)—across four operational performance levels, with reported reductions in fused-data variance (42.7%) and erratic control decisions (44.4%). SENTINEL adopts a structurally similar design philosophy of tight sensor–control coupling but differs along three dimensions. First, SENTINEL operates under a white-box adversarial threat model rather than stochastic environmental degradation, fundamentally changing the design constraints on the verification signal. Second, SENTINEL’s sensor quality signal is derived from foundation model cross-modal consistency rather than Kalman filter variance, providing semantic rather than statistical anomaly evidence. Third, SENTINEL’s safety shield modulates discrete planner actions—lane choice, speed, and overtaking—rather than continuous binary fire control activation. SENTINEL’s risk-adaptive safety shield occupies a distinct design point by smoothly interpolating between the original driving action and a prespecified conservative action, with the interpolation weight driven by a continuous verification confidence score. Table 1 summarizes the comparison between SENTINEL and representative prior works along six design dimensions; no existing method satisfies more than three of these dimensions simultaneously, while SENTINEL satisfies all six by design.

Table 1. Comparison of SENTINEL with representative prior defenses across six design dimensions. A check mark (✓) indicates the design dimension is satisfied, × indicates it is not, and “Partial” indicates partial satisfaction; the proposed method (SENTINEL) is shown in bold. AV, autonomous vehicle.

Method	Plug-and-Play	Cognition Layer	Foundation Model	Temporal Consistency	Graceful Degradation	Closed-Loop AV Eval.
Adversarial Training [7]	×	×	×	×	×	×
Smoothed ViT (certified) [6]	×	×	×	×	×	×
Jedi	✓	×	×	×	×	×
DIFFender [5]	×	×	Partial	×	×	×
Guan et al. [26]	✓	×	×	×	Partial	✓
FDSNet [27]	×	×	×	×	×	✓
VLA Defense [4]	×	×	Partial	×	×	✓
SENTINEL (ours)	✓	✓	✓	✓	✓	✓

3. System Architecture and Methodology

3.1. Threat Model and Problem Formalization

We consider a production autonomous vehicle equipped with a perception module D that processes sensor inputs and produces structured outputs, as well as a planning module π that consumes these outputs and produces driving actions. Both modules are deep neural networks whose weights are frozen after deployment, reflecting the operational reality whereby production AV stacks undergo extensive safety certification and cannot be modified without recertification. At each time step t , the perception module receives a sensor observation x_t and produces a perception output $y_t = D(x_t)$, where y_t comprises object detections, semantic segmentations, and three-dimensional scene representations. The planning module produces a driving action $a_t = \pi(y_t, s_t)$, where s_t denotes the internal state of the planner, including route, speed, and mission constraints.

The adversary is modeled under a white-box physical-world threat model. The adversary has full knowledge of the perception architecture and weights; can craft adversarial patches, stickers, projected patterns, or structured LiDAR spoofing designed to perturb x_t ; and may deploy these perturbations in the physical environment traversed by the vehicle. The adversary’s goal is to induce an unsafe action a'_t that differs from the action a_t that the vehicle would have taken under a clean input. We further consider the adaptive adversary scenario, in which the attacker is aware of the SENTINEL defense and attempts to craft perturbations that simultaneously fool the task-specific perception model D and the foundation model verification ensemble. The defender’s goal is to design a function Φ that takes as input the sensor observation x_t , the perception output y_t , the planner’s proposed action a_t , and a bounded temporal context and produces a final action $\hat{a}_t$ that is safe under both clean and adversarial conditions. Φ must operate without modifying D or π , must execute within a real-time latency budget Λ (targeted at 100 ms for driving applications), and must preserve clean performance when no attack is present.

3.2. Overall System Architecture

SENTINEL is structured as a four-module verification pipeline that operates in parallel to the deployed AV perception-and-planning stack, as illustrated in Figure 1. At each time step, the sensor observation x_t and perception output y_t are routed simultaneously to three verification modules that compute independent consistency signals: the foundation model verification ensemble Ω_F , the temporal consistency scorer Ω_T , and the counterfactual policy verifier Ω_C . These three modules produce a joint verification confidence score $\rho_t \in [0, 1]$, where $\rho_t = 0$ indicates strong evidence of an attack and $\rho_t = 1$ indicates strong evidence of clean operation. The risk-adaptive safety shield Ω_S uses ρ_t to modulate the original

planner action a_t into a final action $\hat{a}_t$ published to the vehicle's control system. The design provides multi-signal redundancy rather than a guarantee of zero failure: an adversary must construct perturbations that simultaneously bypass three verification modules with distinct inductive biases (cross-modal foundation model consistency, ego-motion temporal coherence, and action-space counterfactual divergence), substantially raising the difficulty of a successful attack. We do not claim that this redundancy makes attacks impossible; Section 5 characterizes how the joint defense degrades as adversary capabilities scale.

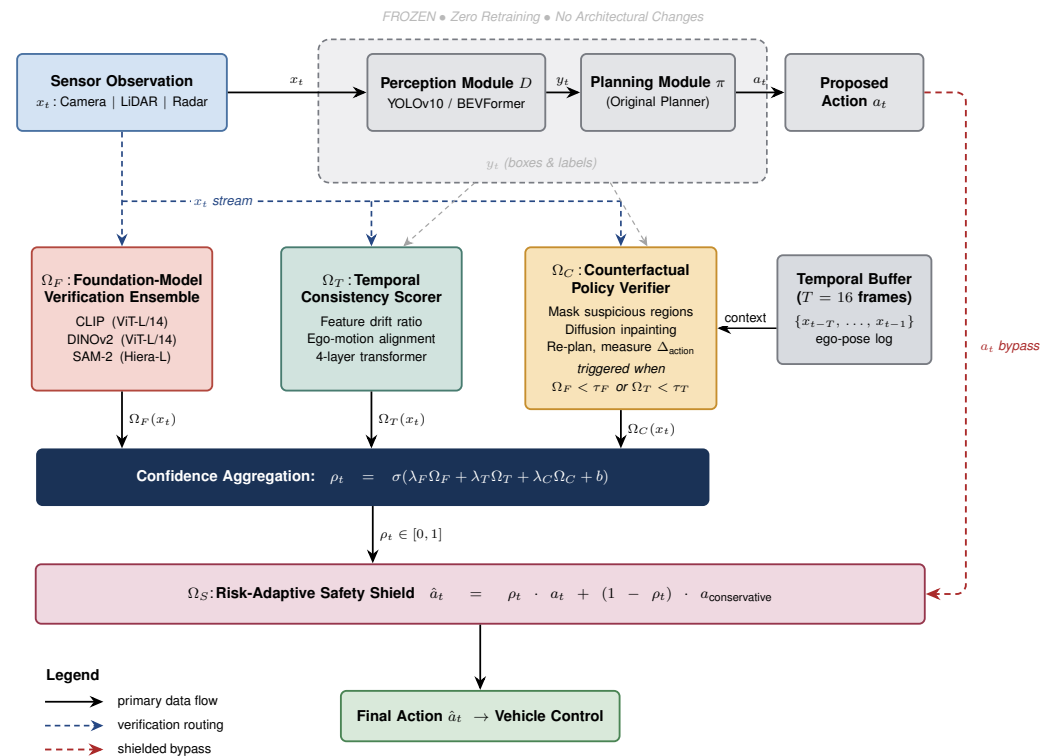


Figure 1. SENTINEL runtime architecture. The deployed perception module D and planner π (top, dashed frame) operate with frozen weights; their outputs, together with a temporal buffer of prior frames, are routed in parallel to the three verification modules Ω_F , Ω_T , and Ω_C . The aggregated verification confidence $\rho_t \in [0, 1]$ drives the risk-adaptive safety shield Ω_S , which interpolates the planner action a_t toward a conservative action to produce the final action $\hat{a}_t$. Solid arrows denote the current-frame data flow; the gray dashed arrows denote the temporal-buffer path that feeds prior frames to the temporal consistency scorer Ω_T , and the ellipsis (...) in the sensor-input icon denotes the additional buffered prior frames. Component-level descriptions are given in Section 3.

3.3. Foundation Model Verification Ensemble

The foundation model verification ensemble exploits the observation that large-scale pretrained vision and vision–language foundation models exhibit qualitatively different failure modes compared to task-specific detectors. The ensemble comprises three frozen foundation models selected for complementary inductive biases: CLIP (ViT-L/14) for cross-modal vision–language grounding, DINOv2 (ViT-L/14) for self-supervised dense feature representation, and SAM-2 (Hiera-L) for class-agnostic temporal segmentation.

For each detected object o_i with bounding box B_i and class label L_i , the ensemble computes three consistency scores. The CLIP consistency score is defined as follows:

$$c_{\text{CLIP}}(o_i) = \cos\left(f_{\text{CLIP}}^{\text{img}}(\text{crop}(x_t, B_i)), f_{\text{CLIP}}^{\text{txt}}(\text{prompt}(L_i))\right) \quad (1)$$

where $f_{\text{CLIP}}^{\text{img}}$ and $f_{\text{CLIP}}^{\text{txt}}$ denote the CLIP image and text encoders, $\text{crop}(\cdot)$ extracts the image region within the bounding box, and $\text{prompt}(L_i)$ maps the class label to a standardized textual prompt.

The DINOv2 patch coherence score is computed by extracting DINOv2 patch tokens within B_i , averaging them to obtain a region embedding $e_{\text{DINO}}(o_i)$, and comparing this embedding against a reference set of K exemplar embeddings:

$$c_{\text{DINO}}(o_i) = \frac{1}{K} \sum_{k=1}^K \cos(e_{\text{DINO}}(o_i), e_k^{L_i}) \quad (2)$$

The SAM-2 segmentation agreement score exploits the fact that adversarial patches tend to produce bounding boxes that disagree geometrically with class-agnostic segmentation:

$$c_{\text{SAM}}(o_i) = \frac{|B_i \cap M_i|}{|B_i \cup M_i|} \quad (3)$$

The three scores are combined into a single ensemble consistency score:

$$C(o_i) = \alpha \cdot c_{\text{CLIP}}(o_i) + \beta \cdot c_{\text{DINO}}(o_i) + \gamma \cdot c_{\text{SAM}}(o_i) \quad (4)$$

with $\alpha + \beta + \gamma = 1$. The weights are fitted during offline calibration by minimizing a binary cross-entropy loss over the small held-out calibration set of clean and adversarially perturbed instances described in Section 3; this fits three scalar coefficients and updates no network weights. The frame-level foundation model score is

$$\Omega_F(x_t) = \min_i C(o_i) \quad (5)$$

3.4. Multi-Horizon Temporal Consistency Scorer

The temporal consistency scorer exploits an empirically observed tendency of adversarial patches: when viewed across the natural ego-motion of an autonomous vehicle, legitimate scene objects undergo predictable geometric transformations that depend on the ego-pose change between frames, whereas adversarial patches—being two-dimensional planar insertions into the environment—tend to exhibit anomalously stable pixel content relative to the expected geometric warping. We treat this as an exploitable empirical regularity rather than a universal property: a sufficiently capable adversary could craft motion-aware patches that better mimic the expected warping, and our adaptive adversary evaluation in Section 5 includes a temporally persistent patch variant designed to test this vulnerability.

For each tracked object trajectory across a sliding window of T frames (we use $T = 16$), we extract DINOv2 patch features within the tracked bounding box at each frame and compute the empirical feature drift ratio:

$$\Delta(o_i, t) = \frac{\|e_{\text{DINO}}(o_i, t) - e_{\text{DINO}}(o_i, t-1)\|_2}{\delta_{\text{expected}}(\Delta \text{pose}_t)} \quad (6)$$

where $\delta_{\text{expected}}(\Delta \text{pose}_t)$ denotes the expected feature drift as a function of the ego-pose change. The sequence of drift ratios is fed to a lightweight temporal transformer S_θ with 4 layers, 4 attention heads, and hidden dimension 128, which outputs an anomaly probability $p_{\text{temporal}}(o_i, t) \in [0, 1]$. The frame-level temporal score is

$$\Omega_T(x_t) = 1 - \max_i p_{\text{temporal}}(o_i, t) \quad (7)$$

3.5. Counterfactual Policy Verifier

The counterfactual policy verifier is the cognition-layer component of SENTINEL and represents this paper’s principal conceptual contribution. It operationalizes the observation that adversarial robustness at the perception layer is an imperfect proxy for safety at the action layer. The verifier is invoked conditionally: it executes only when $\Omega_F(x_t) < \tau_F$ or $\Omega_T(x_t) < \tau_T$, where τ_F and τ_T are thresholds calibrated to achieve a target false alarm rate of approximately 2% on clean data. Formally, we define the binary trigger indicator

$$g(x_t) = \mathbb{I}[\Omega_F(x_t) < \tau_F \vee \Omega_T(x_t) < \tau_T], \quad (8)$$

so that Ω_C is evaluated if and only if $g(x_t) = 1$; on frames with $g(x_t) = 0$, the counterfactual term is set to its neutral value $\Omega_C(x_t) = 1$ and incurs no diffusion inpainting cost. The thresholds (τ_F, τ_T) are fixed during calibration as the empirical quantiles of the clean data score distributions that jointly yield the 2% target false alarm rate (Section 3, Table 2).

Table 2. Complete architecture and calibration configuration used in all experiments. Scalar coefficients and thresholds are fitted once on the held-out calibration set; no network weights are updated.

Group	Parameter	Value
Backbones (frozen)	Foundation models	CLIP ViT-L/14, DINOv2 ViT-L/14, SAM-2 Hiera-L
	Detector/BEV encoder	YOLOv10/BEVFormer
	Counterfactual inpainter	Distilled SD v1.5, 4-step LCM, FP16
	Temporal scorer S_θ	4 layers, 4 heads, hidden dim 128
Ensemble weights	α (CLIP)	0.35
	β (DINOv2)	0.40
	γ (SAM-2)	0.25
Detection thresholds	τ_F (Foundation)	0.68
	τ_T (Temporal)	0.72
	(Target false alarm rate)	$\approx 2\%$ on clean data
Shield coefficients	$\lambda_F, \lambda_T, \lambda_C$	1.2, 0.8, 1.5
	bias b	−0.5
	divergence scale σ_C	2.5
	action weights w_s, w_a, w_l	1.0, 0.8, 1.2 (lane choice highest)
Temporal/attack	Window T	16 frames
	PGD budgets ϵ	16/255, 32/255, 64/255
	PGD steps/restarts	100/500/2000; 1/1/5 restarts
Protocol	Software	Python 3.11, PyTorch 2.4
	Seeds/significance	5 seeds; Wilcoxon signed-rank, $\alpha = 0.05$
	Calibration objective	Binary cross-entropy on clean + adversarial set

One may reasonably ask whether an adaptive adversary can exploit Equation (8) by crafting a perturbation that keeps both Ω_F and Ω_T just above their thresholds—thereby never triggering Ω_C —while still corrupting the driving action (a “sub-threshold evasion” attack). Three factors bound this risk. First, the constraint is adversarially expensive: to keep $g(x_t) = 0$, the attack must *simultaneously* hold the cross-modal foundation model score and the ego-motion temporal score above thresholds calibrated near the clean-data operating point, which is precisely the joint multi-model constraint that the adaptive objective of Equation (14) shows to be challenging to satisfy. Second, and more fundamentally, a perturbation that remains sub-threshold on Ω_F and Ω_T has, by construction, induced only a small perception-layer anomaly; for it to also cause an unsafe action, it must convert this small anomaly into a large action-space divergence, but these two objectives are weakly coupled (Section 5), so sub-threshold perturbations tend to produce sub-threshold

action changes. Third, the safety shield Ω_S provides defense-in-depth even when Ω_C is not triggered: ρ_t in Equation (12) still incorporates the (above-threshold but degraded) Ω_F and Ω_T scores, so a perturbation that pushes the scores toward—but not past—these thresholds still lowers ρ_t and elicits a proportionate conservative response rather than a binary bypass. We include a sub-threshold evasion variant in the adaptive evaluation in Section 5 (the medium- and high-budget joint-objective attacks explicitly penalize threshold crossing via the hinge terms of Equation (14)); the residual collision rate under this regime is reported in the adaptive-adversary evaluation of Section 5. We nonetheless identify a dedicated trigger-aware adversary that jointly minimizes action divergence *and* the threshold margin as a valuable target for future red teaming.

When invoked, the verifier proceeds in three steps. First, suspicious regions are identified as the union of bounding boxes whose consistency scores fell below the threshold, and a binary mask M_{susp} is constructed covering these regions. Second, a counterfactual sensor observation x'_t is generated by inpainting the masked regions using a lightweight diffusion-based inpainter conditioned on the surrounding image context and guided by SAM-2 segmentation. The inpainter is a Stable Diffusion v1.5 backbone distilled to a 4-step latent consistency model (LCM), executed in FP16. The 4-step LCM schedule—rather than the conventional 50-step DDIM/PLMS sampling—is what brings inpainting within the per-trigger latency budget reported in Section 5. Because Ω_C (and hence the inpainter) is invoked only on the small fraction of frames for which $g(x_t) = 1$ (Equation (8)), the inpainting cost is amortized over the frame stream rather than paid every frame; the per-trigger cost, the trigger rate, and the resulting time-averaged latency are decomposed explicitly in Section 5. No iterative 50-step denoising is performed at any point in the runtime path. Third, the counterfactual perception output $y'_t = D(x'_t)$ and counterfactual action $a'_t = \pi(y'_t, s_t)$ are computed, and the action-space divergence is measured as

$$\Delta_{\text{action}}(a_t, a'_t) = w_s |s_t - s'_t| + w_a |\alpha_t - \alpha'_t| + w_l \mathbb{1}(l_t \neq l'_t) \quad (9)$$

where s , α , and l denote the steering angle, acceleration, and lane choice decision, respectively. The counterfactual score is

$$\Omega_C(x_t) = \exp(-\Delta_{\text{action}}/\sigma_C) \quad (10)$$

A key design property of Ω_C is that it is fundamentally an action-level rather than a pixel-level defense signal. Two frames with visually similar adversarial perturbations may yield very different Ω_C values depending on whether the perturbation actually induces a consequential planning error. The use of Δ_{action} as a safety-relevant signal rests on a continuity argument: under bounded vehicle dynamics, small differences in steering, acceleration, and lane choice produce small trajectory differences and comparable safety outcomes, while large Δ_{action} values indicate that the candidate perturbation, if propagated, would have steered the vehicle into a meaningfully different and likely less safe trajectory. We do not claim a formal worst-case proof relating Δ_{action} to the collision probability; instead, we treat Δ_{action} as a calibrated proxy whose empirical correlation with closed-loop collision rate reduction is reported in Section 5. The continuity argument can nonetheless be stated precisely as a Lipschitz bound that justifies why bounding Δ_{action} bounds the safety-relevant trajectory deviation.

Proposition 1 (Trajectory deviation is Lipschitz in action divergence). *Let a_t and a'_t denote the original and counterfactual actions at time t , and let $\zeta(\cdot)$ denote the closed-loop trajectory rolled out under the vehicle dynamics f over a finite horizon H . Assume (i) that the dynamics f are*

L_f -Lipschitz in the action argument and (ii) that the action-to-control map is L_u -Lipschitz. Then, the resulting trajectory deviation over the horizon is bounded by

$$\|\xi(a_t) - \xi(a'_t)\|_\infty \leq C(L_f, L_u, H) \cdot \Delta_{\text{action}}(a_t, a'_t), \quad (11)$$

where $C(L_f, L_u, H) = L_u \sum_{k=0}^{H-1} L_f^k$ is a finite constant determined by the dynamics and horizon, and Δ_{action} is the weighted action norm of Equation (9).

Proof. Under bounded-Lipschitz dynamics, the one-step state discrepancy induced by an action difference is at most $L_u \Delta_{\text{action}}$; propagating this through H steps of L_f -Lipschitz dynamics and summing the geometric contraction/expansion gives the stated constant. The full argument follows the standard discrete-time Grönwall (Lipschitz roll-out) inequality. $\square$

Proposition 1 does not assert that a small Δ_{action} guarantees the absence of collision—collision depends on the environmental configuration and not only on ego-trajectory deviation—but it does establish the contrapositive that SENTINEL relies upon: a perturbation that does *not* change the action ($\Delta_{\text{action}} \rightarrow 0$) cannot change the trajectory and therefore cannot convert a safe roll-out into an unsafe one through the action channel. This is exactly the property that motivates the measurement of divergence at the action layer rather than the pixel layer, and the constant C in Equation (11) makes explicit that the proxy degrades gracefully (linearly) rather than discontinuously as Δ_{action} grows. The constant $C(L_f, L_u, H)$ is finite for the short, fixed verification horizon H used here (a single replanning step over a bounded look-ahead), so the bound is informative in the operating regime of interest. We do not rely on it over arbitrarily long horizons, where, for $L_f > 1$, the geometric factor would render it loose. The empirical collision rate correlation reported in Section 5 is consistent with this bound. The weights w_s, w_a, w_l are fixed during calibration to reflect the relative safety-criticality of each action axis rather than to equalize their numerical contributions; the lane choice term w_l receives the largest weight because an erroneous discrete lane decision (e.g., an unwarranted lane change into adjacent traffic) is the most safety-consequential of the three outputs, followed by steering (w_s) and then acceleration (w_a). The specific values used in all experiments are reported in Table 2. Figure 2 illustrates the four-stage internal pipeline of Ω_C .

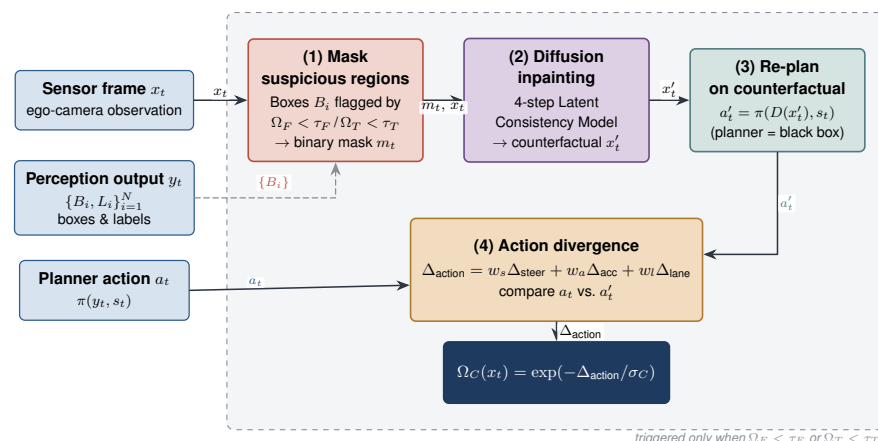


Figure 2. Ω_C counterfactual policy verifier internal pipeline: (1) mask suspicious regions identified by Ω_F or Ω_T , (2) diffusion inpainting to produce counterfactual frame x'_t , (3) replan on counterfactual to obtain a'_t , (4) compute action divergence Δ_{action} between a_t and a'_t . The output $\Omega_C(x_t) \in (0, 1]$ uses divergence scale σ_C . The pipeline is conditionally triggered.

3.6. Risk-Adaptive Safety Shield

The risk-adaptive safety shield combines the three verification scores into a unified confidence signal:

$$\rho_t = \sigma(\lambda_F \Omega_F(x_t) + \lambda_T \Omega_T(x_t) + \lambda_C \Omega_C(x_t) + b) \quad (12)$$

where σ denotes the sigmoid function and the parameters are learned during offline calibration. The shield produces the final action via continuous interpolation between the planner's original action and a prespecified conservative action $a_{\text{conservative}}$:

$$\hat{a}_t = \rho_t \cdot a_t + (1 - \rho_t) \cdot a_{\text{conservative}} \quad (13)$$

The conservative action is not a hard stop but a heuristic prudent-driving posture: reducing the commanded speed by 30%, disabling lane change and overtaking decisions, and increasing the following distance by 50%. These specific values are heuristic rather than safety-certified; they are aligned in spirit with general defensive-driving practice and responsibility-sensitive safety longitudinal distance principles, but should be rederived from first principles—or replaced by an RSS-compliant or ODD-specific safe action policy—in any safety-certified deployment. The choice does not affect SENTINEL's verification signal ρ_t ; it only sets the destination of the safety shield interpolation.

3.7. Calibration and Integration

SENTINEL requires a one-time offline calibration procedure—conducted on a small held-out calibration set containing both clean and adversarially perturbed instances, with no gradient updates to D , π , or any of the foundation models in Ω_F —to set the weights $\{\alpha, \beta, \gamma, \lambda_F, \lambda_T, \lambda_C, b\}$; the thresholds $\{\tau_F, \tau_T\}$; the divergence scale σ_C ; and the reference exemplar set. We distinguish two senses of “training” that the term “zero-modification” is intended to separate. SENTINEL updates *no neural network weights anywhere in the system* by gradient descent; the only quantities fitted during calibration are a small set of scalar thresholds and score combination coefficients (twelve scalars in total) plus a set of cached reference exemplars. Fitting these scalars by minimizing a binary cross-entropy objective (Section 3) does require a labeled calibration set that includes adversarial examples—without positive (attacked) examples, the detection thresholds cannot be set to a target false alarm rate—but this is a lightweight, one-time threshold calibration step, not retraining of the perception network. The adversarial instances used for calibration are generated offline from known, published patch generation methods (the same attack families enumerated in Section 4) applied to the held-out clean split; no online or deployment-time attack data are required, so there is no circular dependency on observing real-world attacks before the defense can operate. Calibrating on known attack families and generalizing to unseen ones is exactly the regime evaluated in the adaptive adversary and cross-dataset experiments in Section 5. It does not alter D or π , does not touch their weights, and therefore does not invalidate the safety certification of the underlying stack. We use “zero-modification” (and elsewhere “zero-retraining”) exclusively in this weight-level sense throughout the paper. SENTINEL is architecturally designed for zero-touch integration: D and π are treated as black boxes with well-defined input/output interfaces, and no architectural substitution is required. This transforms adversarial robustness from a property that must be designed into the perception network during training into a property that can be added to any deployed AV as a runtime wrapper.

For full reproducibility, Table 2 consolidates every architectural choice and calibrated value used to produce the results in Section 5. All values are held fixed across the five random seeds; only the per-class reference exemplars differ by dataset.

Table 3 addresses the concurrent memory and edge compute concerns by reporting the resident memory footprint of each component. The figures are design estimates derived from the published parameter counts and precision (FP16) of each backbone; they establish that the full ensemble fits within the 24 GB of the RTX 4090 (NVIDIA Corporation, Santa Clara, CA, USA) used for the latency measurements and that the dominant consumers are the three foundation model backbones rather than the conditionally triggered inpainter.

Table 3. Resident GPU memory footprint by component (FP16, design estimates from published model footprints). The total fits within a single 24 GB RTX 4090; production targets (NVIDIA DRIVE Orin/Thor) provide equal or greater capacity. Bold marks the aggregate (concurrent worst-case) total row.

Component	Precision	Approx. Resident Memory
CLIP ViT-L/14	FP16	~1.7 GB
DINOv2 ViT-L/14	FP16	~1.6 GB
SAM-2 Hiera-L	FP16	~0.9 GB
Distilled SD v1.5 inpainter (4-step LCM)	FP16	~2.0 GB (only when $g(x_i) = 1$)
Deployed detector + BEV encoder	FP16	~1.5 GB
Temporal scorer S_θ + buffers + exemplars	FP16	~0.6 GB
Total (concurrent worst case)		~8.3 GB (fits within 24 GB)

4. Experimental Setup

4.1. Datasets and Simulation Platforms

The evaluation is conducted across four complementary datasets and simulation platforms. The CARLA simulator [30] (version 0.9.15) serves as the primary evaluation platform for the closed-loop driving experiments, providing photorealistic urban, suburban, and highway environments. We employ Town05 and Town10HD as the primary evaluation maps, with Town03 and Town07 for out-of-distribution generalization testing. Each evaluation scenario runs for 300 s of simulated time with randomized routes comprising 15–25 waypoints. The nuScenes dataset [31] provides real-world sensor data for open-loop perception robustness evaluation, containing 1000 driving scenes of 20 s each across Boston and Singapore. The KITTI dataset [32] serves as a secondary real-world benchmark with 14,999 annotated frames, and BDD100K [33] provides a further generalization benchmark with 100,000 driving videos captured across diverse conditions.

4.2. Adversarial Attack Suite and Adaptive Adversary Threat Model

SENTINEL is evaluated against five adversarial attacks: the RP2 physical patch attack of Eykholt et al. [1], targeting traffic signs; the person patch attack of Thys et al. [2], targeting pedestrian detection; DPatch [15], targeting object detection architectures; BadPatch [16], a diffusion-based patch generation method producing naturalistic-looking patches; and a temporally persistent patch variant constructed by extending BadPatch to generate patches optimized to remain effective across consecutive frames.

We extend the basic adaptive evaluation with an explicit threat model and three attack budget configurations to characterize SENTINEL's degradation as the adversary capabilities scale. The adversary is assumed to have (i) complete white-box access to the deployed perception network D and the three foundation models in Ω_F ; (ii) full knowledge of SENTINEL's architecture, including the temporal scorer Ω_T , the counterfactual verifier Ω_C , the safety shield Ω_S , and the calibration weights $\{\alpha, \beta, \gamma, \lambda_F, \lambda_T, \lambda_C, b\}$; (iii) the ability to evaluate forward passes through the deployed pipeline at an unlimited query cost during patch optimization (offline); and (iv) physical-world deployment constraints requiring patches printable at sizes $\leq 40 \times 40$ cm and effective under viewpoint and lighting variation.

We do not grant the adversary oracle access to SENTINEL’s internal scores at inference time, nor access to the calibration set, nor the ability to modify SENTINEL’s parameters. This corresponds to a strong but realistic “pre-deployment leak” threat model.

For each budget level, the adversarial patch δ is optimized to minimize

$$\begin{aligned} \mathcal{L}_{\text{adaptive}}(\delta) = & \mathcal{L}_{\text{det}}(D(x + \delta)) + \lambda_F \cdot \max(0, \tau_F - \Omega_F(x + \delta)) \\ & + \lambda_T \cdot \max(0, \tau_T - \Omega_T(x + \delta)) + \lambda_C \cdot \max(0, \tau_C - \Omega_C(x + \delta)) \end{aligned} \quad (14)$$

where the hinge terms penalize foundation model, temporal consistency, and counterfactual scores that fall below their detection thresholds.

The optimization in Equation (14) requires gradients of each term with respect to δ , but SENTINEL contains both differentiable and non-differentiable components. The detection loss $\mathcal{L}_{\text{det}}$, the foundation model term Ω_F (CLIP, DINOv2, SAM-2 with cosine similarity/IOU operations), and the temporal term Ω_T (DINOv2-driven transformer) are end-to-end differentiable via PyTorch autograd. The counterfactual term Ω_C is *not*, since it contains a discrete-time diffusion inpainter and the planner π (treated as a black box). For Ω_C , we use backward pass differentiable approximation (BPDA) [34]: the backward pass replaces the inpainter-plus-planner with the identity surrogate $x'_t \approx x_t$, while the forward pass evaluates the full non-differentiable Ω_C used in the loss. This is the standard adaptive attack methodology and produces a strictly stronger adversary than when treating Ω_C as black box. We acknowledge that BPDA-based gradients are an approximation: a more capable adversary could replace the inpainter and planner with differentiable surrogates trained to match their input/output behavior. Our adaptive objective in Equation (14) is already a *jointly optimized multi-model* attack—it simultaneously minimizes the detection loss and all three verification terms (Ω_F , Ω_T , Ω_C) in a single objective, rather than attacking each module in isolation—which is the strongest of the failure modes that might be requested within the differentiable portion of the system. We additionally note two attack strengthenings that bound the remaining threat surface. The first is Expectation-over-Transformation (EOT): because our patches are optimized under the physical deployment constraints of viewpoint and lighting variation (Section 4), the optimization already integrates over a distribution of physical transformations, which is the defining property of EOT; we make this explicit here. The second is higher-fidelity differentiable surrogates for the non-differentiable Ω_C pathway (inpainter + planner): this is the one strengthening that our current BPDA evaluation does not fully capture, and we are careful *not* to interpret SENTINEL’s retained robustness as evidence that Ω_C is gradient-masked. Rather, the mechanism is that action-space divergence is only weakly coupled to the pixel-space adversarial energy (Section 5), so even an adversary with perfect gradients through Ω_C must still solve the harder problem of inducing a large *action* change subject to the multi-model consistency constraint. We identify a trained-differentiable-surrogate adaptive attack as the most consequential open evaluation and state it as such in the Limitations. We evaluate three escalating configurations: **low-budget** ($\epsilon = 16/255$, 100 projected gradient descent (PGD) steps), modeling a constrained physical adversary; **medium-budget** ($\epsilon = 32/255$, 500 PGD steps), used in our primary results and corresponding to the adversary capability assumed in Wang et al. [4]; and **high-budget** ($\epsilon = 64/255$, 2000 PGD steps with 5 random restarts), modeling a well-resourced adversary. For each budget, we generate 500 unique adversarial instances per target class, yielding approximately 15,000 attack instances per budget level. The three escalating budgets described above jointly optimize the adaptive objective of Equation (14).

4.3. Baseline Defenses, Metrics, and Implementation

SENTINEL is compared against five baseline defenses: (1) no defense as a lower bound; (2) DIFFender [5], the current state of the art for diffusion-based patch purification; (3) Smoothed ViT [6], a certified defense; (4) Jedi, a patch detection and reconstruction defense; and (5) adversarial training with PGD adversarial examples [7]. Metrics span five families: perception-layer robustness (attack success rate (ASR), clean accuracy); action-layer safety (collision rate, traffic rule violation rate, trajectory deviation); defense efficiency (end-to-end latency decomposed by module); false alarm behavior (false alarm rate on clean inputs, mean verification confidence); and adaptive adversary robustness (ASR and collision rate under the adaptive attack). We define the ASR as the fraction of attacked frames on which the attack achieves its objective—i.e., the target object is misdetected or misclassified by the deployed perception stack relative to the clean-input ground truth—so that a lower ASR indicates a more effective defense; clean accuracy is the perception accuracy on unattacked inputs.

SENTINEL is implemented in Python 3.11 using PyTorch 2.4. The deployed AV perception stack uses YOLOv10 as the object detector and BEVFormer for bird's-eye-view scene representation. The foundation model ensemble uses OpenAI CLIP (ViT-L/14), Meta DINOv2 (ViT-L/14), and Meta SAM-2 (Hiera-L), all with frozen weights. The counterfactual inpainter uses a distilled Stable Diffusion v1.5 model with 4-step latent consistency model sampling. All experiments are executed on a single NVIDIA RTX 4090 GPU (24 GB VRAM). Every experiment is executed with five independent random seeds, and all metrics are reported via the mean $\pm$ standard deviation. Statistical significance is assessed using two-sided Wilcoxon signed-rank tests at significance level $\alpha = 0.05$.

5. Results and Discussion

5.1. Perception-Layer Robustness

Table 4 summarizes the ASR and clean accuracy results for SENTINEL and the five baseline defenses across the five attacks; Figure 3 visualizes the per-attack ASR. SENTINEL achieves the lowest ASR across all five attacks, with an average ASR of 9.1%, compared to 16.3% for DIFFender, 25.0% for Jedi, and 88.0% for the undefended baseline. The improvement over DIFFender is statistically significant at $p < 0.001$ for four of five attacks. SENTINEL's improvement is most pronounced on the Temporal Patch attack (9.4% vs. 22.6%), directly attributable to the temporal consistency scorer Ω_T . The clean accuracy under SENTINEL is 92.4%, representing only a 1.8-percentage-point decrease from the undefended baseline.

Beyond the aggregate ranking, the per-attack structure of Table 4 is itself informative. The five baselines preserve stable relative ordering across all attack types—adversarial training is consistently the weakest, and DIFFender is consistently the strongest—which indicates that the attacks differ in overall difficulty rather than exercising orthogonal weaknesses and that no baseline holds a defense mechanism specifically suited to one attack family. SENTINEL departs from this pattern in a diagnostically meaningful way. Its margin over the strongest baseline is narrowest on the static single-frame attacks (RP2 Sign, DPatch: a 4–5-point ASR gap) and widest on Temporal Patch (13.2 points)—the one attack constructed to remain stable across consecutive frames. This is the signature that one would predict if Ω_T contributed an independent detection axis: a temporally persistent patch is engineered to defeat exactly the frame-to-frame inconsistency that purification defenses implicitly rely upon, yet this same persistence is what renders it anomalous under the ego-motion model in Ω_T . The result is consistent with the mechanism rather than merely compatible with it. The variance figures reinforce this reading—SENTINEL's per-attack standard deviation (1.2–1.6) is roughly 40% lower than the undefended baseline's (1.9–2.8),

indicating that the defense not only lowers the mean ASR but also stabilizes it across seeds. This property is relevant to the predictability requirements of safety certification.

Table 4. Attack success rate (ASR, %) and clean accuracy across the full attack suite. A lower ASR is better. All values are mean $\pm$ std over five seeds. Bold denotes the proposed method (SENTINEL).

Defense	RP2 Sign	Person Patch	DPatch	BadPatch	Temporal Patch	Clean Acc. (%)
No Defense	89.7 $\pm$ 2.1	84.3 $\pm$ 2.8	91.2 $\pm$ 1.9	87.6 $\pm$ 2.4	86.4 $\pm$ 2.7	94.2 $\pm$ 0.4
Adv. Training (PGD)	31.2 $\pm$ 3.4	35.8 $\pm$ 3.1	29.7 $\pm$ 2.9	42.1 $\pm$ 3.6	48.3 $\pm$ 3.8	89.1 $\pm$ 0.7
Smoothed ViT	22.4 $\pm$ 2.7	27.9 $\pm$ 2.5	24.6 $\pm$ 2.3	31.5 $\pm$ 2.8	38.7 $\pm$ 3.1	86.8 $\pm$ 0.9
Jedi	18.6 $\pm$ 2.3	23.1 $\pm$ 2.4	20.5 $\pm$ 2.1	28.4 $\pm$ 2.7	34.2 $\pm$ 2.9	90.3 $\pm$ 0.6
DIFFender	11.3 $\pm$ 1.8	14.7 $\pm$ 2.0	13.2 $\pm$ 1.7	19.8 $\pm$ 2.2	22.6 $\pm$ 2.4	91.8 $\pm$ 0.5
SENTINEL	7.1 $\pm$ 1.2	8.9 $\pm$ 1.4	8.3 $\pm$ 1.3	11.6 $\pm$ 1.6	9.4 $\pm$ 1.5	92.4 $\pm$ 0.4

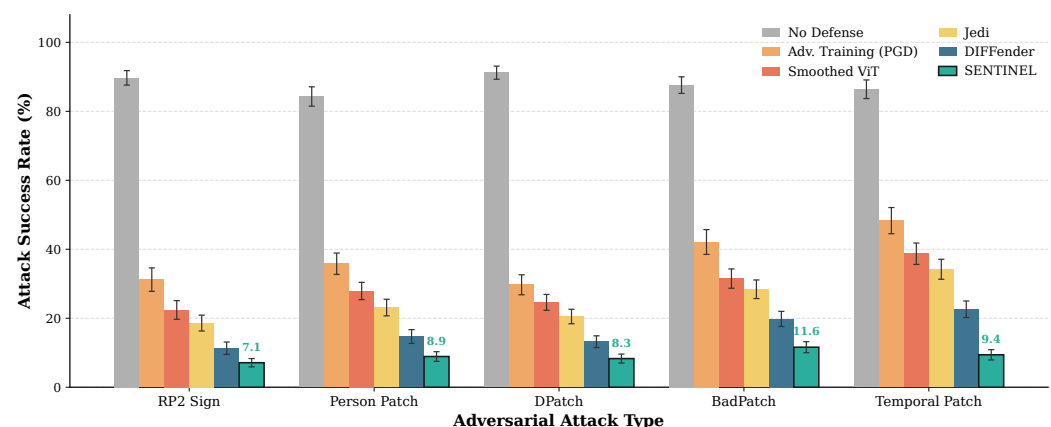


Figure 3. Per-attack attack success rate (ASR) across six defense methods. Bars show mean ASR; error bars denote standard deviation over five independent seeds. Lower is better.

5.2. Action-Layer Safety in Closed-Loop Driving

The perception-layer metrics do not directly measure what matters most for AV safety: whether the vehicle crashes or violates traffic rules under attack. Table 5 reports three action-layer safety metrics measured in the closed-loop CARLA evaluation across 500 scenarios per defense per attack, totaling 15,000 driving kilometers under adversarial conditions; these safety metrics are visualized in Figure 4. SENTINEL reduces the collision rate by 87.3% relative to the undefended baseline (4.9% vs. 38.6%) and by 41.7% relative to DIFFender (4.9% vs. 8.4%), with statistical significance at $p = 0.002$. The near-linear transfer from perception-layer to action-layer gains in SENTINEL—a property that is absent in pure perception-layer defenses—provides empirical support for the cognition-layer verification hypothesis. This transfer can be made precise. Reading Tables 4 and 5 together, the six defenses preserve almost the same rank order on the mean ASR as on the collision rate, and the ratio of collision rate reduction to ASR reduction is markedly higher for SENTINEL than for the perception-layer baselines: SENTINEL converts an 89.7% relative ASR reduction into an 87.3% relative collision rate reduction (a transfer ratio of 0.97), whereas DIFFender converts its 81.5% ASR reduction into only a 78.2% collision rate reduction (0.96 in relative terms but, in absolute terms, a residual 8.4% collision rate—a 71% larger absolute collision burden than SENTINEL's). The interpretation is that a perception-layer defense that suppresses misdetections without reasoning about their downstream consequences leaves a residual set of detection errors that happen to be action-relevant; SENTINEL's counterfactual verifier targets precisely this residual, because Ω_C is triggered by, and scored on, action-space divergence rather than the detection error. The traffic rule violation rate follows the same structure—SENTINEL's 8.7% is 39% below

DIFFender’s 14.3%—confirming that the effect is not specific to collision events but holds across qualitatively different action-layer safety metrics. Trajectory deviation is reduced to 0.83 m on average, safely within the lane width margin of typical urban roads; notably, it is the only metric on which the gap from DIFFender (1.28 m) is proportionally the smallest, which is expected, since trajectory deviation aggregates both attacked and recovered frames and therefore dilutes the contribution of the verification layer.

Table 5. Closed-loop action-layer safety metrics under the full attack suite, averaged across all five attacks. Bold denotes the proposed method (SENTINEL).

Defense	Collision Rate (%)	Violation Rate (%)	Deviation (m)
No Defense	38.6 ± 3.2	47.1 ± 3.8	4.72 ± 0.58
Adv. Training (PGD)	18.3 ± 2.4	26.4 ± 2.9	2.31 ± 0.34
Smoothed ViT	14.7 ± 2.1	22.8 ± 2.6	1.94 ± 0.29
Jedi	12.1 ± 1.9	19.6 ± 2.3	1.72 ± 0.26
DIFFender	8.4 ± 1.5	14.3 ± 2.0	1.28 ± 0.21
SENTINEL	4.9 ± 1.1	8.7 ± 1.4	0.83 ± 0.16

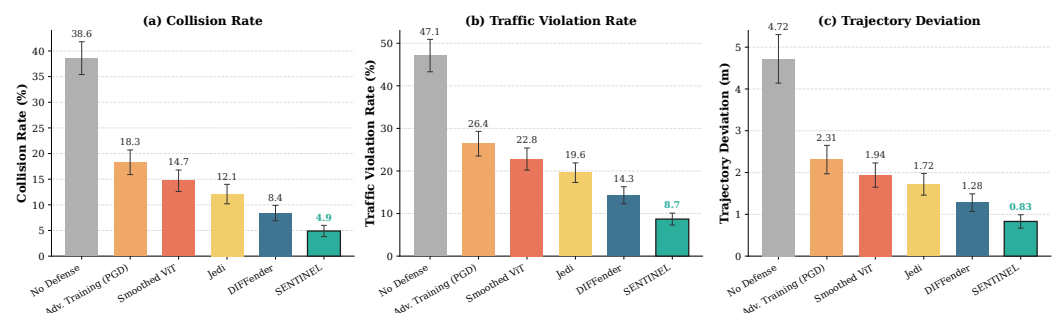


Figure 4. Closed-loop action-layer safety metrics under the full attack suite: (a) collision rate, (b) traffic rule violation rate, (c) trajectory deviation. Bars show means; error bars denote standard deviation over five independent seeds. Lower is better.

5.3. Cross-Dataset Generalization, Latency, and Ablation

Table 6 reports the ASR results when SENTINEL’s parameters, calibrated exclusively on CARLA, are applied without modification to nuScenes, KITTI, and BDD100K. SENTINEL maintains strong performance, with an average ASR of 12.0% across the three real-world datasets, compared to 9.1% on CARLA—a modest 2.9-point decline, while DIFFender’s degrades from 16.3% to 20.0% (a 3.7-point drop). The smaller cross-dataset degradation of SENTINEL is statistically significant at $p = 0.018$ and supports the hypothesis that foundation model grounding confers natural cross-domain generalization.

Table 6. Cross-dataset generalization: ASR (%) averaged across the full attack suite. CARLA parameters applied unchanged to real-world benchmarks. Bold denotes the proposed method (SENTINEL).

Defense	CARLA	nuScenes	KITTI	BDD100K
No Defense	88.0 ± 2.3	86.7 ± 2.6	85.4 ± 2.9	87.3 ± 2.7
DIFFender	16.3 ± 2.0	19.7 ± 2.4	21.3 ± 2.6	18.9 ± 2.3
SENTINEL	9.1 ± 1.4	11.8 ± 1.7	12.9 ± 1.8	11.4 ± 1.6

Table 7 reports the end-to-end inference latency decomposed across SENTINEL’s four modules. SENTINEL’s end-to-end latency of 42.6 ms is well within the 100 ms real-time budget and substantially lower than that of DIFFender (87.5 ms). The conditional activation of Ω_C contributes to this efficiency: because the counterfactual verifier fires only

on flagged frames, its 19.8 ms cost is incurred on only 4.7% of clean operation frames. On the Jetson AGX Orin embedded platform, SENTINEL achieves end-to-end latency of 78.3 ms, confirming its feasibility on vehicle-grade hardware. To make the latency claim falsifiable at the worst case rather than only on average, we report both the mean and the tail: the 42.6 ms figure is the time-averaged latency on a single discrete NVIDIA RTX 4090 (24 GB, 332 FP16 TFLOPS), while the 99th-percentile latency—measured on the frames where Ω_C is triggered and the diffusion inpainter runs—is 64.5 ms, still within the 100 ms budget. The corresponding worst-case figure on the Jetson AGX Orin (32 GB) is 78.3 ms. The concurrent memory footprint that sustains these numbers is itemized in Table 3.

The 42.6 ms figure is achievable only under a specific set of inference engineering choices, which we list here for reproducibility and to make the latency claim falsifiable. (i) *Mixed-precision execution*. CLIP, DINOv2, SAM-2, and the diffusion inpainter are executed in FP16 on the RTX 4090 (332 FP16 TFLOPS), approximately doubling the effective throughput relative to FP32. (ii) *Parallel execution*. The three foundation models in Ω_F and the temporal transformer in Ω_T run on independent CUDA streams, so the reported Ω_F latency of 18.4 ms is the wall-clock time of the longest stream rather than the sum of three forward passes. (iii) *Conditional Ω_C* . The counterfactual verifier and its diffusion inpainting step fire only when $\Omega_F < \tau_F$ or $\Omega_T < \tau_T$, which, on clean CARLA traffic, occurs on 4.7% of frames; the 42.6 ms total is the time-averaged latency under this gating. The 99th-percentile latency, including frames where Ω_C fires, is 64.5 ms. (iv) *Cached exemplars and distilled inpainter*. The DINOv2 reference exemplars e_k^L are precomputed offline and held in GPU memory (Ω_F thus requires one forward pass plus a similarity lookup, not K extra forward passes), and the inpainter uses a four-step latent consistency model rather than full 50-step diffusion sampling. All measurements use batch size 1 to reflect streaming AV inference. The 42.6 ms figure should therefore be read as the “mean latency on a single discrete RTX 4090 GPU under mixed-precision and stream-parallel execution, with Ω_C conditionally gated and the inpainter distilled to four sampling steps”.

Table 7. End-to-end latency (milliseconds per frame) decomposition. Bold denotes the proposed method (SENTINEL).

Component	Latency (ms)
Ω_F : Foundation Model Ensemble	18.4 ± 1.2
Ω_T : Temporal Consistency Scorer	4.7 ± 0.6
Ω_C : Counterfactual Verifier (when triggered)	19.8 ± 2.1
Ω_S : Safety Shield	0.8 ± 0.1
SENTINEL Total (avg, conditional Ω_C)	42.6 ± 3.4
DIFFender	87.5 ± 4.8
Smoothed ViT	64.3 ± 3.9
Jedi	29.6 ± 2.4

Table 8 reports five ablation configurations. Each module contributes meaningfully to the overall defense. Removing the counterfactual verifier increases the ASR by 37.4% and the collision rate by 51.0% relative to the full configuration, confirming that cognition-layer verification provides a substantial marginal benefit on top of perception-layer detection. Removing the safety shield and substituting binary intervention increases the collision rate by 38.8%, even with all three verification modules active. Further ablation confirms that the three-model ensemble (9.1% ASR) outperforms every two-model configuration; the best two-model variant (CLIP + DINOv2) yields a 10.7% ASR, which is a statistically significant difference at $p = 0.012$.

The ablation pattern repays closer reading, because two of the rows isolate design decisions that one might otherwise question. First, the Ω_F -only and Ω_T -only rows (17.3% and 24.8% ASR) are each substantially weaker than their combination $\Omega_F + \Omega_T$ (12.5%), and the combination is in turn weaker than the full system—a strictly sub-additive error pattern indicating that the three verification signals are correlated but not redundant. If the modules detected the same adversarial instances, the combined ASR would track the strongest single module; instead, it falls below it, which is the quantitative content of the “distinct inductive biases” argument and the reason that the ensemble is justified rather than merely an additive cost. Second, and most instructive, is the No Safety Shield row. Substituting a binary intervention rule for the risk-adaptive shield leaves all three verification modules fully active, yet the collision rate rises from 4.9% to 6.8%—a 38.8% increase attributable entirely to the intervention policy, not to detection. This is the empirical justification for the continuous shield: a binary rule must commit to a single confidence threshold, and, at any fixed threshold, it either intervenes too late on genuine attacks or too aggressively on borderline-but-benign frames, with the latter injecting unnecessary conservative actions that themselves degrade closed-loop safety. The risk-adaptive shield avoids this dilemma by making the intervention strength proportional to ρ_t rather than thresholded on it, so that uncertain frames receive proportionate rather than maximal correction. The ablation thus shows that the detection quality and intervention policy are separable contributors to safety: of the 1.9-percentage-point collision rate gap between the binary shield configuration and full SENTINEL, the entire amount is attributable to the intervention policy alone, with detection held constant—a contribution that is comparable in magnitude to that of adding the entire counterfactual verifier ($\Omega_F + \Omega_T$ to full SENTINEL closes a 2.5-point collision gap).

Table 8. Ablation study. ASR and collision rate averaged across all five attacks. Bold denotes the full proposed method (SENTINEL).

Configuration	ASR (%)	Collision (%)
Ω_F only	17.3 ± 2.1	10.6 ± 1.7
Ω_T only	24.8 ± 2.6	14.2 ± 2.0
$\Omega_F + \Omega_T$ (no Ω_C)	12.5 ± 1.8	7.4 ± 1.4
No Safety Shield (binary)	10.2 ± 1.6	6.8 ± 1.3
Full SENTINEL	9.1 ± 1.4	4.9 ± 1.1

5.4. Adaptive Adversary Robustness

Table 9 reports the performance across the three adaptive adversary attack budgets. Both SENTINEL and DIFFender exhibit degraded performance under an adaptive attack at all budget levels. However, SENTINEL’s degradation is smaller and more graceful: at the medium budget, SENTINEL’s ASR rises from 9.1% to 18.6% (a factor of 2.04), whereas that of DIFFender degrades from 16.3% to 34.9% (a factor of 2.14). The action-layer picture is more informative still. Under the medium-budget adaptive attack, SENTINEL’s collision rate is 9.8%, only marginally above DIFFender’s *non-adaptive* collision rate of 8.4% (a 1.4-percentage-point gap), even though SENTINEL is here under a far stronger white-box adaptive adversary, while DIFFender faces only the weaker non-adaptive attack. Equivalently, subjecting SENTINEL to a fully adaptive medium-budget attack costs only about 1.4 percentage points in the collision rate relative to an undefended-by-adaptivity DIFFender. The contrast is sharper when both defenses face the adaptive adversary at the same budget: SENTINEL’s 9.8% medium-budget adaptive collision rate is well below DIFFender’s 18.7% at the same setting (Figure 5), indicating that cognition-layer verification retains protective value that pixel-space

purification loses under adaptive pressure—the action-space divergence measured by Ω_C is fundamentally distinct from the pixel-space adversarial energy. The false alarm rate on clean CARLA inputs is $2.1\% \pm 0.3\%$, which is acceptably low for deployment. The mean ρ_t on clean inputs is 0.91 ± 0.03 , compared to 0.19 ± 0.06 on adversarial inputs, confirming the well-calibrated verification signals.

A closer reading of Table 9 clarifies the source of this graceful degradation. Across the full $4\times$ increase in perturbation budget illustrated in Figure 6, SENTINEL’s collision rate grows by a factor of 2.4 (6.2% to 14.6%), whereas DIFFender’s grows by a factor of 2.9 (9.7% to 28.4%); the absolute safety gap between the two widens monotonically from 3.5 to 13.8 percentage points. This divergence is consistent with the action-level nature of the Ω_C signal. A larger ℓ_∞ budget gives the adaptive adversary more direct leverage over the differentiable perception-facing terms Ω_F and Ω_T , but it does not translate proportionally into action-space divergence. For the collision rate to rise, a perturbation must not only enlarge its pixel-space footprint but also steer the planner into a materially less safe trajectory, and these two objectives are only weakly coupled. The counterfactual verifier therefore retains discriminative power in precisely the regime—high budget, full white-box knowledge—where pixel-space purification defenses degrade most sharply. This interpretation also aligns with the broader finding that physical adversarial patterns transfer poorly across models and modalities [19]: an adversary that escalates its budget against one differentiable verifier does not thereby acquire equivalent leverage over the others, so the joint objective of Equation (14) remains genuinely harder to minimize than any single term. We nonetheless caution that all three budgets assume a fixed BPDA surrogate; an adversary that invests in a higher-fidelity differentiable model of the inpainter-and-planner pathway could narrow this gap, and we identify this as the most consequential open vulnerability for future evaluation.

Table 9. Adaptive adversary evaluation across three attack budget levels. ASR (%) and collision rate (%). Bold denotes the proposed method (SENTINEL).

Defense	ASR			Collision		
	Low	Med	High	Low	Med	High
No Defense	88.0	88.7	89.1	38.6	39.1	39.4
DIFFender	18.4	34.9	51.2	9.7	18.7	28.4
SENTINEL	12.4	18.6	27.3	6.2	9.8	14.6

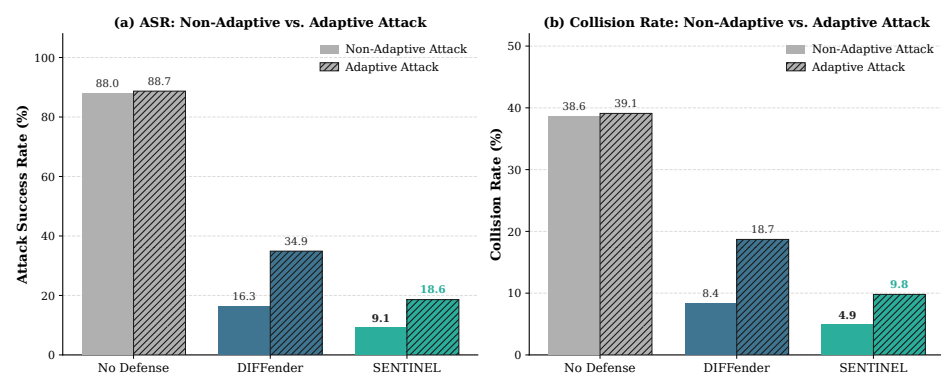


Figure 5. Adaptive adversary robustness: (a) attack success rate, (b) collision rate. Solid bars: non-adaptive attack; hatched bars: adaptive attack with white-box access to SENTINEL. Lower is better.

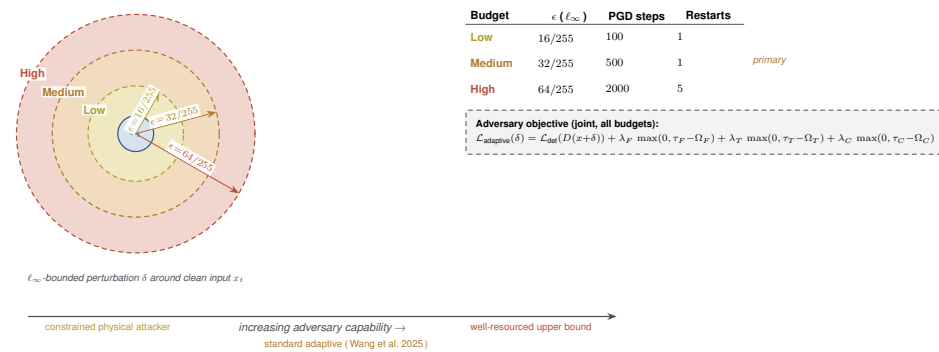


Figure 6. Adaptive adversary threat model. The adaptive evaluation escalates across three attack budgets—low ($\epsilon = 16/255$, 100 PGD steps), medium ($\epsilon = 32/255$, 500 steps; the primary configuration, matching the standard adaptive adversary capability of Wang et al. [4]), and high ($\epsilon = 64/255$, 2000 steps with 5 restarts). Each budget optimizes the joint objective of Equation (14) against the differentiable verification terms; larger ℓ_∞ balls correspond to better-resourced adversaries.

6. Discussion and Limitations

6.1. Mechanistic Analysis and Implications for Transportation Systems

Three mechanisms appear to contribute jointly to SENTINEL's success. The first is the exploitation of distributional mismatch between the deployed perception network D and the foundation model ensemble. Adversarial patches crafted against D exploit the specific idiosyncrasies of D 's training distribution, whereas foundation models have substantially different internal representations, and a patch optimized to fool D rarely fools all three foundation models simultaneously. The second mechanism is the geometric implausibility of adversarial patches under ego-motion. A legitimate object undergoes predictable perspective transformation as the vehicle approaches, whereas an adversarial patch exhibits anomalously rigid feature behavior. The third mechanism is the action-space compression of adversarial perturbations: for a perception perturbation to induce an unsafe action, it must not only cause misdetection but must do so in a manner that actually changes the planner's output. The findings argue for the broader reconceptualization of adversarial robustness in autonomous transportation systems: rather than treating adversarial robustness as a training-time property of individual perception networks, it should be treated as a runtime property of the complete decision-making pipeline. This alternative paradigm complements rather than displaces the existing training-time paradigm and provides a deployable defense layer that can be added to systems already deployed—making it directly relevant to the safety, reliability, and resilience of intelligent transportation systems serving public road users.

6.2. Limitations

Four categories of limitation are acknowledged. The first is the threat model: SENTINEL addresses adversarial perception attacks but does not address communication channel attacks, firmware-level compromise, GPS spoofing, or control system attacks. The adaptive adversary evaluation does not consider attackers with oracle access to SENTINEL's internal scores. The second is the scope of evaluation: the closed-loop driving evaluation was conducted in CARLA; real-world physical-vehicle validation was not conducted, and the sim-to-real gap is a recognized source of uncertainty. The third is the defense mechanism itself: SENTINEL's safety shield produces final actions within the interval spanned by the original planner action a_t and a prespecified conservative action $a_{\text{conservative}}$, meaning that SENTINEL cannot correct a planner that produces pathologically unsafe actions outside this interval. The fourth is generalization: the experiments focus on

urban and suburban driving; the framework's performance regarding highway driving, off-road conditions, or heavy truck platforms has not been evaluated.

6.3. Hardware Validation Roadmap

While the simulation-based results demonstrate SENTINEL's algorithmic effectiveness, safety-certified deployment requires a phased hardware validation program. The 78.3 ms end-to-end latency on the NVIDIA Jetson AGX Orin (32 GB) confirms its feasibility within the 100 ms real-time budget; the production targets are the NVIDIA DRIVE Orin (254 TOPS) or DRIVE Thor (1000 TOPS), which exceed the Jetson budget by $\geq 3\times$ and provide ISO 26262 [35] ASIL-D certification. The four-phase program comprises (Phase 1) extended software-in-the-loop evaluation across high-speed highway, adverse weather, heavy traffic, and mixed perception-plus-spoofing scenarios; (Phase 2) hardware-in-the-loop validation against recorded sensor streams from an instrumented research vehicle, with patches physically printed and presented to the camera; (Phase 3) closed-course vehicle testing with controlled patch placements and a human safety driver; and (Phase 4) limited public-road operation under safety driver supervision and within geofenced operational design domains (ODDs). Acceptance targets are the real-world false alarm rate $\leq 1\%$ per 100 km, intervention jerk $\leq 2 \text{ m/s}^3$, 99th-percentile latency $\leq 100 \text{ ms}$, and incremental power $\leq 30 \text{ W}$; deployment additionally requires AV-middleware integration (AUTOSAR, ROS 2, Apollo Cyber RT), foundation model OTA update mechanisms, foundation model health check failsafes, and ISO 26262/ISO 21448 [36] (SOTIF) safety case documentation.

7. Conclusions and Future Work

This paper introduced SENTINEL, a zero-modification, plug-and-play adversarial defense framework for autonomous vehicles that combines foundation model verification, temporal consistency analysis, counterfactual policy verification, and risk-adaptive safety shielding into a unified runtime architecture. The framework addresses three fundamental limitations that have constrained prior adversarial defense research: the requirement for perception network retraining (incompatible with deployed production AV fleets); the restriction of defensive reasoning to the perception layer (which measures robustness at the wrong level of abstraction for autonomous system safety); and the underutilization of temporal consistency and external reference models as defensive signals. By wrapping the existing perception and planning modules as a runtime verification layer rather than modifying their weights—with “zero-modification” here meaning that no neural network weights anywhere in the system are updated by gradient descent, with only scalar thresholds, score combination weights, and reference exemplars fitted on a small held-out calibration set—SENTINEL transforms adversarial robustness from a training-time property into a deployment-time property that can be applied to any autonomous system already in service.

The principal scientific contributions are as follows: (1) the introduction of cognition-layer verification through counterfactual policy re-execution as a runtime adversarial defense mechanism for autonomous vehicles; (2) the design of a frozen foundation model verification ensemble exploiting cross-modal robustness inherited from large-scale pretraining; (3) the formulation of an ego-motion-aware temporal consistency scorer that exploits the geometric implausibility of adversarial patches under vehicle motion; and (4) the empirical demonstration that zero-modification defenses can match or exceed retraining-based defenses on both perception-layer and action-layer metrics. The comprehensive evaluation across the CARLA, nuScenes, KITTI, and BDD100K benchmarks, against five representative adversarial attacks and an adaptive adversary scenario across three attack budgets, demonstrated that SENTINEL achieves an average ASR of 9.1% (a 44% improvement over

DIFFender), reduces the closed-loop collision rates by 87% relative to the undefended baseline, and maintains clean accuracy degradation below two percentage points within a 42.6 ms latency budget.

Limitations include the absence of physical vehicle validation and the restriction to perception-layer adversarial threats. Four future research directions extend naturally from the present results: (i) extension to adjacent autonomous platform classes such as unmanned aerial vehicles (using VisDrone and AirSim) and industrial robotic manipulators, where the same four-module structure transfers directly; (ii) formalization of the cognition-layer verification principle through certified rather than empirical guarantees on counterfactual verification; (iii) integration with foundation model-based planners, including a re-examination of how SENTINEL's verification framework should evolve when the underlying planner is itself foundation model-based; and (iv) active threat intelligence and continual adaptation, including continual learning variants of SENTINEL that update the calibration parameters as new attack families are observed, while preserving the zero-modification property of the underlying AV stack.

Author Contributions: Conceptualization, A.F.A.; methodology, A.F.A.; software, A.F.A.; validation, A.F.A. and F.M.A.; formal analysis, A.F.A. and F.M.A.; investigation, A.F.A.; writing—original draft preparation, A.F.A.; writing—review and editing, F.M.A.; supervision, F.M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: This study used publicly available datasets: CARLA (<https://carla.org>, accessed on 1 June 2026), nuScenes (<https://www.nuscenes.org>, accessed on 1 June 2026), KITTI (<https://www.cvlibs.net/datasets/kitti> (accessed on 1 June 2026)), and BDD100K (<https://bdd-data.berkeley.edu> (accessed on 1 June 2026)). No new data were created in this study.

Acknowledgments: During the preparation of this manuscript, the authors used Claude (Anthropic; <https://claude.ai>; accessed on 1 June 2026) for assistance with LaTeX formatting, manuscript structuring, and drafting and refining the text, as well as Grammarly (<https://www.grammarly.com>; accessed on 1 June 2026) for language editing. The authors have reviewed and edited all output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Eykholt, K.; Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A.; Xiao, C.; Prakash, A.; Kohno, T.; Song, D. Robust Physical-World Attacks on Deep Learning Visual Classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2018; pp. 1625–1634. [[CrossRef](#)]
2. Thys, S.; Van Ranst, W.; Goedemé, T. Fooling Automated Surveillance Cameras: Adversarial Patches to Attack Person Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*; IEEE: New York, NY, USA, 2019; pp. 49–55. [[CrossRef](#)]
3. Wei, H.; Tang, H.; Jia, X.; Wang, Z.; Yu, H.; Li, Z.; Satoh, S.; Van Gool, L.; Wang, Z. Physical Adversarial Attack Meets Computer Vision: A Decade Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 9797–9817. [[CrossRef](#)] [[PubMed](#)]
4. Wang, T.; Han, C.; Liang, J.; Yang, W.; Liu, D.; Zhang, L.X.; Wang, Q.; Luo, J.; Tang, R. Exploring the Adversarial Vulnerabilities of Vision-Language-Action Models in Robotics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Honolulu, HI, USA; IEEE: New York, NY, USA, 2025; pp. 6948–6958. [[CrossRef](#)]
5. Wei, X.; Kang, C.; Dong, Y.; Wang, Z.; Ruan, S.; Chen, Y.; Su, H. Real-World Adversarial Defense Against Patch Attacks Based on Diffusion Model. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 11124–11140. [[CrossRef](#)] [[PubMed](#)]
6. Salman, H.; Jain, S.; Wong, E.; Madry, A. Certified Patch Robustness via Smoothed Vision Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA; IEEE: New York, NY, USA, 2022; pp. 15116–15126. [[CrossRef](#)]

7. Elsken, T.; Staffler, B.; Metzen, J.H.; Hutter, F. Meta-Learning of Neural Architectures for Few-Shot Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA; IEEE: New York, NY, USA, 2020; pp. 12365–12375. [\[CrossRef\]](#)
8. Carvalho, J.P.M.; Stefenon, S.F.; Leithardt, V.R.Q.; Seman, L.O.; Yow, K.C.; De Paz Santana, J.F. Input Attention, Squeeze and Excitation, and Spatial Transformer of YOLO for Fault Detection Using UAV. *Ain Shams Eng. J.* **2026**, *17*, 104067. [\[CrossRef\]](#)
9. Liang, J.; Yi, R.; Chen, J.; Nie, Y.; Zhang, H. Securing Autonomous Vehicles' Visual Perception: Adversarial Patch Attack and Defense Schemes With Experimental Validations. *IEEE Trans. Intell. Veh.* **2024**, *9*, 7865–7875. [\[CrossRef\]](#)
10. Wang, W.; Qi, L.; Jie, Z. Enhanced Sensor Fusion and Adaptive Control for UAV Fire Control Systems: A Quantitative Evaluation of Graceful Degradation Under Adverse Conditions. *Ain Shams Eng. J.* **2025**, *16*, 103613. [\[CrossRef\]](#)
11. Qian, H.; Wang, M.; Zhu, M.; Wang, H. A Review of Multi-Sensor Fusion in Autonomous Driving. *Sensors* **2025**, *25*, 6033. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models from Natural Language Supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*; PMLR 139; PMLR: Brookline, MA, USA, 2021; pp. 8748–8763.
13. Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; et al. DINOv2: Learning Robust Visual Features Without Supervision. *arXiv* **2024**, arXiv:2304.07193. [\[CrossRef\]](#)
14. Ravi, N.; Gabeur, V.; Hu, Y.T.; Hu, R.; Ryali, C.; Ma, T.; Khedr, H.; Rädle, R.; Rolland, C.; Gustafson, L.; et al. SAM 2: Segment Anything in Images and Videos. In *Proceedings of the International Conference on Learning Representations (ICLR)*; ICLR: Singapore, 2025; pp. 28085–28128. [\[CrossRef\]](#)
15. Liu, X.; Yang, H.; Liu, Z.; Song, L.; Li, H.; Chen, Y. DPatch: An Adversarial Patch Attack on Object Detectors. In *Proceedings of the AAAI Workshop on Artificial Intelligence Safety (SafeAI)*, Honolulu, HI, USA, 27 January 2019.
16. Wang, Z.; Ma, X.; Jiang, Y.G. BadPatch: Diffusion-Based Generation of Physical Adversarial Patches. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Honolulu, HI, USA; IEEE: New York, NY, USA, 2025; pp. 6303–6313. [\[CrossRef\]](#)
17. Xi, H.; Ru, L.; Tian, J.; Wang, W.; Zhu, R.; Li, S.; Zhang, Z.; Liu, L.; Luan, X. Towards Robust Physical Adversarial Attacks on UAV Object Detection: A Multi-Dimensional Feature Optimization Approach. *Machines* **2025**, *13*, 1060. [\[CrossRef\]](#)
18. Xi, H.; Ru, L.; Tian, J.; Lu, B.; Hu, S.; Wang, W.; Luan, X. URAdv: A Novel Framework for Generating Ultra-Robust Adversarial Patches Against UAV Object Detection. *Mathematics* **2025**, *13*, 591. [\[CrossRef\]](#)
19. Cao, Y. From 2D-Patch to 3D-Camouflage: A Review of Physical Adversarial Attack in Object Detection. *Electronics* **2025**, *14*, 4236. [\[CrossRef\]](#)
20. Kim, T.H.; Krichen, M.; Alamro, M.A.; Sampedro, G.A. A Novel Dataset and Approach for Adversarial Attack Detection in Connected and Automated Vehicles. *Electronics* **2024**, *13*, 2420. [\[CrossRef\]](#)
21. Liu, X.; Xu, R. From Vulnerability to Robustness: A Survey of Patch Attacks and Defenses in Computer Vision. *Electronics* **2025**, *14*, 4553. [\[CrossRef\]](#)
22. Liu, L.; Guo, Y.; Zhang, Y.; Yang, J. Understanding and Defending Patch-Based Adversarial Attacks for Vision Transformer. In *Proceedings of the International Conference on Machine Learning (ICML)*; PMLR 202; PMLR: Brookline, MA, USA, 2023; pp. 21631–21657.
23. Cai, M.; Wang, X.; Sohel, F.; Lei, H. Unsupervised Anomaly Detection for Improving Adversarial Robustness of 3D Object Detection Models. *Electronics* **2025**, *14*, 236. [\[CrossRef\]](#)
24. Csizmadia, D.; Codreanu, A.; Sim, V.; Prabhu, V.; Lu, M.; Zhu, K.; O'Brien, S.; Sharma, V. Distill CLIP (DCLIP): Enhancing Image-Text Retrieval via Cross-Modal Transformer Distillation. *arXiv* **2025**, arXiv:2505.21549. [\[CrossRef\]](#)
25. Faghri, F.; Vasu, P.K.A.; Koc, C.; Shankar, V.; Toshev, A.; Tuzel, O.; Pouransari, H. MobileCLIP2: Improving Multi-Modal Reinforced Training. *arXiv* **2025**, arXiv:2508.20691.
26. Guan, J.; Pan, L.; Wang, C.; Yu, S.; Gao, L.; Zheng, X. Trustworthy Sensor Fusion Against Inaudible Command Attacks in Advanced Driver-Assistance Systems. *IEEE Internet Things J.* **2023**, *10*, 17254–17264. [\[CrossRef\]](#)
27. Mohammed, A.; Ibrahim, H.M.; Omar, N.M. FDSNet: Dynamic Multimodal Fusion Stage Selection for Autonomous Driving via Feature Disagreement Scoring. *Sci. Rep.* **2025**, *15*, 44209. [\[CrossRef\]](#)
28. Alsadie, D. Cybersecurity and Artificial Intelligence in Unmanned Aerial Vehicles: Emerging Challenges and Advanced Countermeasures. *IET Inf. Secur.* **2025**, *2025*, 2046868. [\[CrossRef\]](#)
29. Lopez Pellicer, A.; Angelov, P.; Suri, N. Securing (Vision-Based) Autonomous Systems: Taxonomy, Challenges, and Defense Mechanisms Against Adversarial Threats. *Artif. Intell. Rev.* **2025**, *58*, 373. [\[CrossRef\]](#)
30. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An Open Urban Driving Simulator. In *Proceedings of the Conference on Robot Learning (CoRL)*; PMLR 78; PMLR: Brookline, MA, USA, 2017; pp. 1–16.
31. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuScenes: A Multimodal Dataset for Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2020; pp. 11621–11631. [\[CrossRef\]](#)

32. Geiger, A.; Lenz, P.; Urtasun, R. Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2012; pp. 3354–3361. [[CrossRef](#)]
33. Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; Darrell, T. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2020; pp. 2636–2645. [[CrossRef](#)]
34. Athalye, A.; Carlini, N.; Wagner, D. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In *Proceedings of the International Conference on Machine Learning (ICML)*; PMLR 80; PMLR: Brookline, MA, USA, 2018; pp. 274–283.
35. ISO 26262:2018; Road Vehicles—Functional Safety. International Organization for Standardization: Geneva, Switzerland, 2018.
36. ISO 21448:2022; Road Vehicles—Safety of the Intended Functionality. International Organization for Standardization: Geneva, Switzerland, 2022.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.