

USENIX

THE ADVANCED COMPUTING
SYSTEMS ASSOCIATION

SLEIGHT: Real-time Closed-Loop Attack Analytics Framework for Holistic Intrusion Detection Systems in Unmanned Aerial Vehicles

Muneeba Asif, *Florida International University;* Maryna Veksler, *Virginia Commonwealth University;* Devrim Aras, *Florida International University;* Kemal Akkaya, *Virginia Commonwealth University;* Mohammad Ashiqur Rahman, *Florida International University*

https://www.usenix.org/conference/vehiclesec26/presentation/asif

This paper is included in the Proceedings of the
4th USENIX Symposium on Vehicle Security and Privacy.

August 10–11, 2026 • Baltimore, MD, USA

ISBN 978-1-939133-56-4

Open access to the Proceedings of the
4th USENIX Symposium on Vehicle Security and Privacy
is sponsored by USENIX.

SLEIGHT: Real-Time Closed-Loop Attack Analytics Framework for Holistic Intrusion Detection Systems in Unmanned Aerial Vehicles

Muneeba Asif
Florida International University

Maryna Veksler
Virginia Commonwealth University

Devrim Aras
Florida International University

Kemal Akkaya
Virginia Commonwealth University

Mohammad Ashiqur Rahman
Florida International University

Abstract

Unmanned aerial vehicles (UAVs) are increasingly secured by advanced intrusion detection systems (IDSs) that monitor the holistic sensor–control–actuator pipeline. However, the robustness of these IDSs has not been systematically characterized against adaptive, real-time, physically plausible, and stealthy telemetry perturbations during closed-loop flight. Prior adversarial studies often rely on open-loop, offline testing, failing to capture the temporal, dynamic, and closed-loop constraints inherent to UAV operation. Consequently, it remains unclear what mission-level impact an in-flight data-stream attacker can induce while maintaining benign-looking behavior. We propose SLEIGHT, a novel, real-time, attack analytics framework that maps the stealth–impact frontier of closed-loop UAV IDSs under physically plausible perturbations. SLEIGHT trains a phase-conditioned reinforcement-learning policy with attacker-side plausibility enforcement to synthesize bounded perturbations maximizing mission impact under stealth and feasibility constraints. Using Pixhawk hardware- and software-in-the-loop experiments with MAVSDK, SLEIGHT benchmarks four IDSs, including state-of-the-art controller- and actuator-aware holistic IDSs such as PID-Piper and ConFIDe. SLEIGHT consistently achieves high no-alarm rates across diverse and multi-UAV missions (>96% in single-channel GPS, sustaining >95% and >90% against ConFIDe and PID-Piper, respectively) while inducing trajectory drift and landing displacement up to ≈11 m in single UAV HITL and > 60 m in swarm scenarios.

1 Introduction

Unmanned aerial vehicles (UAVs) are increasingly deployed in disaster response, defense, agriculture, infrastructure inspection, and environmental monitoring [30, 43]. Modern UAVs achieve autonomy through a closed-loop pipeline that couples sensing, state estimation, control, and actuation. At each control cycle, navigation-related measurements such as position, velocity, attitude, and altitude are fused by the estimator and used by the controller to compute actuator commands [12] (Fig. 1). This measurement-driven loop creates a practical cyber-physical attack surface, i.e., if an attacker rewrites MAVLink state messages consumed by an offboard controller (e.g., via a compromised companion relay or MAVLink router on the autopilot–offboard link), small navigation biases can propagate through feedback to cause mission-level effects such as trajectory drift, unsafe separation, or off-target landing [5, 21, 22].

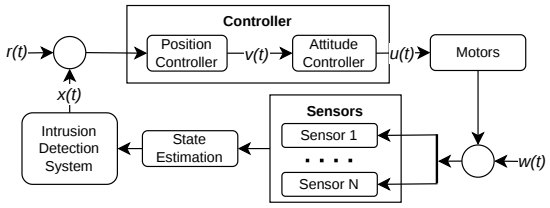


Figure 1: UAV closed-loop and measurement-driven operation.

As UAVs increasingly operate in beyond-visual-line-of-sight (BVLOS) settings with minimal human oversight, intrusion detection systems (IDSs) are widely used to detect anomalous behavior and trigger failsafe responses [19, 37]. Recent UAV IDSs are holistic, combining multivariate signals spanning sensors, estimator outputs, controller variables, and actuator behavior [6, 8, 10, 41]. Such fusion can improve robustness against single-channel spoofing and simple cross-signal inconsistencies [10, 17]. However, robustness against adaptive, real-time telemetry manipulation in closed-loop flight remains insufficiently characterized; for instance, perturbations that stay below alarm thresholds at each step can still accumulate over long horizons into meaningful operational as well as safety risk.

A central reason is that many adversarial evaluations rely on offline datasets or open-loop perturbations [11, 14]. These settings miss temporal dependencies, estimator dynamics, and feedback effects that determine feasibility and detectability during flight. Thus, it is difficult to quantify the boundary between *stealth* (remaining within detector tolerance) and *impact* (inducing mission-level deviation) under realistic access and feasibility constraints. Moreover, studies often report either detection outcomes or disruption metrics in isolation,

rather than their joint tradeoff under a thresholded decision rule [42, 44]. Concretely, we lack a principled way to answer that under physically plausible telemetry manipulation, how much mission deviation can an attacker induce while remaining nominal-looking to a deployed IDS over an entire flight?

Hence, we present **SLEIGHT** (Synthesizing Legitimate-looking Evasive Inputs to Generate Hidden Threats), a closed-loop attack analytics framework that maps the stealth–impact frontier of UAV IDSs under physically plausible telemetry perturbations. As shown in Fig. 2, bounded telemetry rewrites at the autopilot–offboard interface lead to a gradual navigation shift while remaining below the IDS detection score. SLEIGHT uses stealthy perturbation generation as the mechanism for robustness evaluation; its primary objective is to characterize where UAV IDSs remain robust, and where bounded closed-loop telemetry manipulation can still induce mission-level deviation. SLEIGHT enables systematic evaluation across single- and coordinated multi-sensor manipulations, varying flight and environmental complexities, and multi-UAV scenarios. SLEIGHT operates under a partial-information adversary operating along realistic data paths. The attacker can bias a limited subset of navigation-related channels via an onboard relay (e.g., a companion computer) that forwards flight telemetry (e.g., MAVLink), consistent with prior UAV communication attack and detection settings [5, 21, 22]. The attacker obtains benign-like telemetry through reconnaissance, as in prior UAV threat models [18] and CPS deception settings [33]. The threat model is formalized in Section 4.

To synthesize perturbations online, SLEIGHT combines phase-conditioned sequential learning with runtime plausibility enforcement. Offline, it learns (i) an attacker-side plausibility scorer from reconnaissance telemetry and (ii) a phase-conditioned sequential policy regularized by a realism prior to maximize an impact proxy under bounded feasibility constraints. Online, the policy is executed with lightweight per-step scaling and projection (via bisection on perturbation magnitude) to enforce plausibility and physical feasibility at each control cycle, enabling closed-loop, state-dependent adaptation without using deployed IDS scores or parameters.

We evaluate SLEIGHT in a closed-loop PX4 environment using MAVSDK, and validate key findings with hardware-/software-in-the-loop experiments on Pixhawk flight-control firmware. Across diverse single- and multi-UAV missions and four IDS architectures (including controller- and actuator-aware holistic IDSs), SLEIGHT maintains high no-alarm rates while inducing measurable trajectory drift and landing displacement (up to $\approx$11 m in our testbed). We also analyze robustness under practical uncertainty sources such as flight complexity, environmental conditions, threshold estimation error, and decision-boundary drift. Our results quantify the stealth-impact frontier of thresholded UAV IDSs under limited adversarial access and limited prior knowledge. We summarize our contributions as follows:

- We formalize closed-loop UAV IDS robustness analysis as stealth-constrained mission impact maximization, enabling direct quantification of the stealth-impact frontier for thresholded detectors under feasibility constraints.
- We present SLEIGHT, an attack analytics framework that synthesizes physically plausible, fusion-consistent perturbations for closed-loop IDS robustness evaluation across missions and multi-UAV scenarios.
- We conduct a comprehensive closed-loop evaluation showing consistent stealth–impact tradeoffs across several representative IDS designs, including mission-level effects under bounded, plausible perturbations.
- We provide hardware-/software-in-the-loop validation on flight-controller firmware and release a reproducible PX4 SITL + MAVSDK pipeline with scenario configurations to support future IDS benchmarking [39].

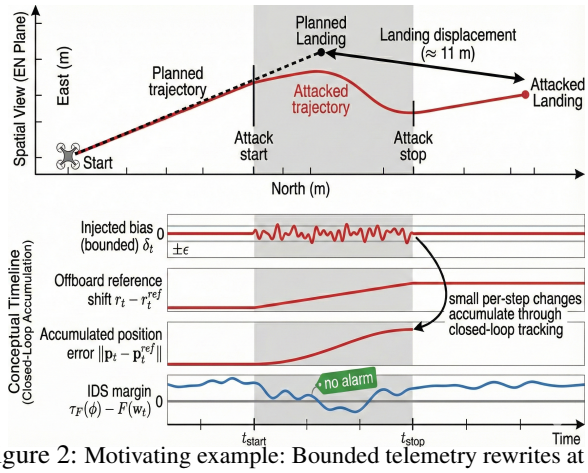


Figure 2: Motivating example: Bounded telemetry rewrites at the MAVLink interface bias offboard references, accumulating into landing displacement while the IDS stays below its alarm threshold.

2 Background

Here, we overview the closed-loop pipeline and its constraints, the IDS families evaluated in this work, and MAVLink telemetry mediation as a practical injection surface.

UAV Telemetry and Real-Time Pipelines: At control step t with period Δt, the stack produces a time-aligned telemetry vector $\mathbf{x}_t$ and updates estimation/control:

$$\mathbf{s}_t = \mathcal{E}(\mathbf{x}_{1:t}), \qquad \mathbf{u}_t = \mathcal{C}(\mathbf{s}_t, \mathbf{r}_t), \qquad \mathbf{z}_{t+1} = \mathcal{P}(\mathbf{z}_t, \mathbf{u}_t, \mathbf{w}_t)$$

Autopilots run nested loops at heterogeneous rates, with IMU-driven estimation and attitude/rate control faster than GPS navigation updates [32]. Thus, telemetry perturbations must be feasible per update: values must satisfy message type, unit, range, timing, and per-step dynamics expected by the estimator/controller. Closed-loop attackers must therefore synthesize within the targeted channel's per-step budget and maintain temporal coherence at the runtime update rate. Otherwise, the injected stream can lag or create rate/update inconsistencies, even if it appears benign offline. PX4 scheduling and configured publication rates make this a runtime constraint [3].

UAV IDS Families and Limitations: IDSs typically compute an anomaly score from telemetry data and issue an alarm when the score exceeds a threshold. A common class is reconstruction-based detection:

$$\ell_{\text{recon}}(\mathbf{x}_t) = \|\mathbf{x}_t - f_\theta(\mathbf{x}_t)\|_2^2, \qquad \text{alarm if } \ell_{\text{recon}}(\mathbf{x}_t) > \tau \quad (1)$$

Boundary-based one-class methods similarly define an acceptance region. Holistic designs augment features with controller and actuator signals, detecting via cross-signal consistency rather than sensor statistics alone. Our evaluation covers these families, including controller/actuator-aware PID-Piper [17] and ConFIDe [10]. Section 3 details this limitation in prior work.

Telemetry Injection Surface (MAVLink and Companion Mediation): Telemetry can be manipulated along MAVLink forwarding paths between the flight controller, GCS, and onboard components. MAVLink2 supports optional signing but no encryption; without signing, an on-path adversary can modify or replay protocol-valid telemetry [2]. We assume signing, or equivalent integrity protection, is absent on the targeted link; otherwise, endpoint/key compromise would be required and is out of scope. Signing hardens protected links against on-path modification, but does not prevent attacks from trusted endpoints, such as compromised companion processes. Thus, we treat integrity protection as hardening while still evaluating IDS robustness at realistic telemetry interfaces.

Companion computers use MAVLink to relay traffic among the autopilot, GCS, and peripherals [1, 3], making companion-mediated rewriting a realistic injection point. Prior work demonstrates MAVLink injection in SITL/Pixhawk HITL, and MUVIDS evaluates IDS detection under false MAVLink messages [22]. Although we implement attacks through MAVLink mediation, common in PX4 offboard/companion setups, SLEIGHT is not MAVLink-specific. The same enforcement and frontier analysis apply to other telemetry-consumer interfaces, such as ROS/middleware, by redefining x_t, M, and Π_Ω.

3 Related Works

We group prior work into four themes: (i) UAV sensor/protocol attack surfaces, (ii) adversarial ML for IDS evasion, (iii) CPS stealth–impact analysis/falsification, and (iv) holistic UAV IDSs that fuse sensing, control, and actuation.

Sensor-level attacks on UAVs: Dash et al. [16] learn invariant thresholds but do not model environmental variation. Khazraei et al. [25, 26] coordinate camera and sensor attacks, reporting reduced impact under sensor fusion. Tractor Beam [34] and Son et al. [45] demonstrate single-sensor spoofing (e.g., GPS, acoustic) with mission diversion, without end-to-end evaluation under multi-sensor fusion and closed-loop IDS monitoring. RVFuzzer [28] and MAYDAY [27] focus on input flaws or offline analysis and do not address adaptive, closed-loop synthesis. ConfuSense [18] shows persistent effects from a single malicious bus message. These works motivate realistic attack surfaces, but they do not quantify long-horizon mission deviation under bounded, temporally coherent perturbations while staying within the operating region of holistic UAV IDSs.

Adversarial ML for IDS evasion: Adversarial ML shows that learned models can be evaded under bounded perturbations. Kurakin et al. [29] study adversarial examples for time-series inputs in mostly static settings. In UAV security, Asif et al. [11] use adversarial learning for IDS hardening through retraining rather than attacks. Alshahrani et al. [7] and Sarikaya and Bahtiyar [38] consider GAN-generated IDS attacks but omit closed-loop constraints and multi-sensor coordination. Strickland et al. [40] combine GANs with RL for IDS improvement, but evaluate offline. Wang et al. [44] discuss telemetry manipulation, focusing on deviation metrics without closed-loop evasion analysis. Xue et al. [46] show GPS spoofing as a single-sensor weakness. Uludag et al. [42] exploit cross-sensor relations for deception, but not real-time closed-loop synthesis with per-step feasibility. Transferability studies [35] motivate attacker-side reconnaissance learning without deployed-model access.

IDS evasion and stealth–impact analysis in CPS: Prior work links stealth to operational impact via invariants (SAVIOR [36]), offline residual-driven objectives [24], or temporal-logic falsification [15]; related works in safe RL focus on policy safety rather than IDS evasion [23]. SLEIGHT provides a real-time, multi-sensor perturbation generator integrated with PX4 closed-loop control, with per-step feasibility enforcement and mission-level impact measurement.

Real-time and online evasion with reinforcement learning: Recent work demonstrates that RL agents can perturb streaming inputs to evade sequence-based anomaly detectors under bounded modification constraints (e.g., online evasion pipelines [20]). This motivates real-time evasion evaluation beyond offline replay. SLEIGHT specializes this direction to closed-loop UAV telemetry, where feasibility is defined by runtime message semantics and update rates, and where mission impact must be measured through closed-loop execution rather than offline score manipulation.

Holistic and fusion-aware UAV IDSs: ConFIDe [10] and PID-Piper [17] are holistic UAV IDSs that fuse sensing, control, and actuation signals to resist single-channel spoofing and simple inconsistencies, relying on offline-trained detection boundaries. Following community practice in evasion-oriented IDS evaluations that benchmark multiple detector families and operating points [13], we evaluate multiple IDSs, including holistic designs under closed-loop operation.

Differences: To address these limitations, SLEIGHT instantiates a closed-loop attack analytics pipeline in PX4 that (i) learns attacker-side models from reconnaissance telemetry, (ii) proposes phase-conditioned perturbation directions, and (iii) enforces attacker-side plausibility and feasibility constraints at runtime via maximal-feasible scaling, without querying deployed IDS scores during synthesis. We then

quantify the resulting stealth–impact frontier across missions, multi-sensor and multi-UAV settings, and multiple IDS families, including holistic controller- and actuator-aware designs.

4 Threat Model and Problem Definition

We study stealth-constrained telemetry-integrity attacks in closed-loop flight, where an adversary perturbs limited navigation fields while an IDS monitors the sensor-control-actuation pipeline. We distinguish the deployed IDS score $F(\cdot)$, used only for evaluation, from the attacker-side plausibility score $S(\cdot)$, used for synthesis and runtime enforcement; the attacker never observes, queries, or optimizes $F(\cdot)$ online.

4.1 System Model

At control step $t \in \{1,\dots,T\}$ with period Δt, the system produces telemetry $\mathbf{x}_t \in \mathbb{R}^d$ and updates estimation/control in closed loop as follows:

$$\mathbf{s}_{t+1} = \mathcal{F}(\mathbf{s}_t, \mathbf{u}_t, \text{env}), \qquad \mathbf{u}_t = \pi(\mathbf{s}_t, \mathbf{r}_t) \tag{2}$$

In our PX4+MAVSDK setting, mission logic runs as an offboard control process that consumes a recent telemetry window and outputs references $\mathbf{r}_t$ (e.g., position/velocity setpoints) sent back to the autopilot. Thus, manipulating the telemetry stream that this offboard process consumes can alter $\mathbf{r}_t$ and induce mission-level deviation under closed-loop control. A deployed IDS computes an anomaly score over a recent window $\mathbf{w}_t = [\mathbf{x}_{t-W+1},\dots,\mathbf{x}_t] \in \mathbb{R}^{W\times d}$:

$$F(\mathbf{w}_t) \in \mathbb{R}_{\geq 0}, \qquad \text{alarm if } F(\mathbf{w}_t) > \tau_F(\phi(t)) \tag{3}$$

where $\phi(t) \in \{\text{takeoff, transit, hover, landing}\}$ is the flight phase. The thresholding rule in (3) models the deployed detector; it is *not* available to the attacker during synthesis. As one example, reconstruction-based IDSs instantiate $F(\cdot)$ as a windowed reconstruction error; our attacker does not observe F and does not tune attacks to any specific F form.

4.2 Adversary Capabilities and Restrictions

Placement: We model an in-path adversary that can read and rewrite MAVLink messages between the autopilot and the offboard controller (and co-located IDS), e.g., via a compromised relay or MITM. This aligns with prior work demonstrating false MAVLink injection in SITL/HITL settings [22].

Manipulation: Let $\mathbf{M} \in \{0,1\}^d$ denote a mask over attackable channels, where $\mathbf{M}$ is a strict subset of the IDS feature set. For example, it covers navigation-related channels, while actuators/PWM/control variables used by holistic IDSs are never modified. At each step, the attacker injects

$$\mathbf{x}'_t = \mathbf{x}_t + \mathbf{M} \odot \delta_t \tag{4}$$

where δ_t is bounded. Attacks may be single-channel (e.g., GPS position) or coordinated across multiple channels (e.g., GPS altitude and barometric altitude) to preserve estimator-level consistency. The attacker preserves protocol-valid formatting and field semantics for modified messages.

Restrictions: The attacker does not modify flight firmware, estimator/controller code, safety logic, mission plan/waypoints, cryptographic keys, or the IDS implementation, or assume RF denial (e.g., jamming). Attacker does not access IDS's internals or query the deployed IDS score $F(\cdot)$ online. We assume that MAVLink2 optional message signing is not enabled on the targeted communication link; when signing is configured, an attack would require endpoint/key compromise, which is outside the scope of this work. As such, attackers target systems where signing is disabled, inconsistently deployed, or not enforced end-to-end. Although signing is recommended for secure deployment, it requires explicit key provisioning and coordination across components, introducing operational overhead [4]. This can result in inconsistent implementations, particularly in SITL/HITL and companion-mediated pipelines.

4.3 Reconnaissance and Attacker Plausibility

Because the attacker does not observe $F(\cdot)$, we parameterize *benign-looking* behavior using an attacker-side plausibility model trained from benign-like data on the attackable channels. The attacker is assumed to obtain benign telemetry $\mathcal{D}_{\text{benign}}$ for the attackable channels through reconnaissance (e.g., controlled collection on a comparable platform or observation of operationally similar flights). Using this benign dataset, the attacker trains a plausibility scorer:

$$S(\text{vec}(\mathbf{w}_t)) \in \mathbb{R}_{\geq 0} \tag{5}$$

where $\text{vec}(\cdot)$ flattens a window. SLEIGHT is agnostic to the specific plausibility scorer ($S(\cdot)$); we instantiate it with an AE residual, but any one-class score (e.g., OCSVM/SVDD or density/predictor residuals) can be substituted without changing Alg. 1 or (11). The attacker calibrates ($\tau_A(\phi)$) from benign reconnaissance via (6), while robustness is evaluated solely against the deployed IDS ($F(\cdot)$).Attacker sets operational stealth threshold from benign quantiles (phase-conditioned):

$$\tau_A(\phi) = \text{Quantile}_q\left(\{S(\text{vec}(\mathbf{w})) : \mathbf{w} \in \mathcal{D}_{\text{benign}}, \phi(\mathbf{w}) = \phi\}\right) \tag{6}$$

The attacker estimates $\phi(t)$ from the observable window; it is not assumed as a hidden label. To model realistic uncertainty, we evaluate calibration mismatch by varying (i) the amount of reconnaissance data used to fit $S(\cdot)$ and $\tau_A(\cdot)$, (ii) biased or noisy $\tau_A(\cdot)$ estimates, and (iii) phase-estimation errors.

Reconnaissance mismatch can reduce reliability. If target flights differ from reconnaissance data due to wind, sensor noise, payload, or dynamics, the attacker-side score distribution may shift. This can increase the time required to find feasible perturbations and may increase exposure risk when $\tau_A(\phi)$ is overestimated. SLEIGHT mitigates this by tightening thresholds via κ and by optionally rolling and recalibrating from recent benign-looking windows.

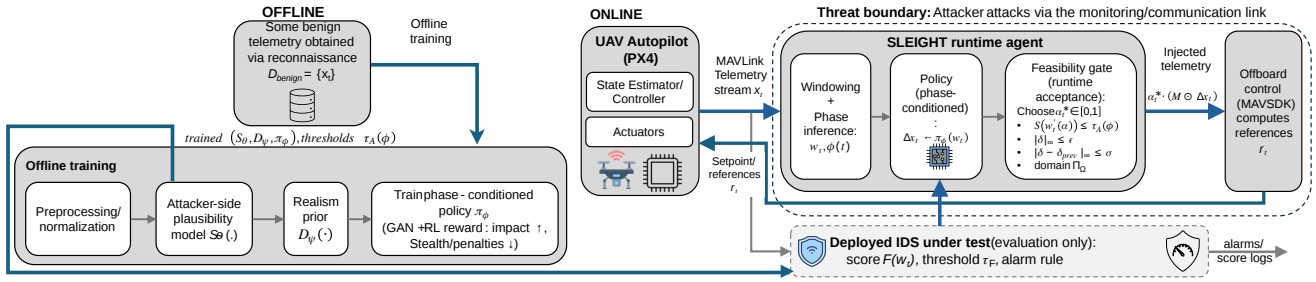


Figure 3: SLEIGHT overview: Offline, benign telemetry trains preprocessing, the plausibility model ($S(\cdot)$), realism prior ($D(\cdot)$), and a phase-conditioned perturbation policy. Online, the policy proposes a direction and selects the largest feasible scale ($\alpha_t^\star$) before injection.

4.4 Stealth and Feasibility Constraints

SLEIGHT enforces stealth w.r.t the attacker-side score:

$$S(\mathrm{vec}(\mathbf{w}_t')) \le \tau_A(\phi(t)) \quad \forall t \tag{7}$$

where $\mathbf{w}_t' = [\mathbf{x}_{t-W+1}', \ldots, \mathbf{x}_t']$. To reduce stealth failures under reconnaissance mismatch, the attacker can enforce a tightened threshold ($\tau_A'(\phi) = (1-\kappa)\tau_A(\phi)$) ($(\kappa \in [0,1))$), trading impact for robustness. If benign-like segments are observable online, ($\tau_A(\phi)$) can also be updated via rolling quantiles of recent ($S(\cdot)$) scores We enforce bounded and smooth perturbations in (8) and project injected values onto a valid domain Ω before encoding in (9):

$$\|\mathbf{M} \odot \delta_t\|_\infty \le \varepsilon, \qquad \|\mathbf{M} \odot (\delta_t - \delta_{t-1})\|_\infty \le \sigma \tag{8}$$

$$\mathbf{x}_t' \leftarrow \Pi_\Omega(\mathbf{x}_t') \tag{9}$$

At runtime, SLEIGHT proposes a perturbation direction $\Delta\mathbf{x}_t$ and applies a scale $\alpha \in [0,1]$:

$$\mathbf{x}_t'(\alpha) = \mathbf{x}_t + \alpha(\mathbf{M} \odot \Delta\mathbf{x}_t) \tag{10}$$

SLEIGHT chooses the maximal feasible scale

$$\begin{aligned} \alpha_t^\star = \max_{\alpha \in [0,1]} \alpha \quad & \text{s.t.} S\big(\mathrm{vec}(\mathbf{w}_t'(\alpha))\big) \le \tau_A(\phi(t)), \\ & \big\|\mathbf{M} \odot \delta_t(\alpha)\big\|_\infty \le \varepsilon, \big\|\mathbf{M} \odot \big(\delta_t(\alpha) - \delta_{t-1}\big)\big\|_\infty \le \sigma, \\ & \mathbf{x}_t'(\alpha) \in \Omega \end{aligned} \tag{11}$$

where $\delta_t(\alpha)$ is the induced perturbation in (4) under scaling (10). If no $\alpha > 0$ satisfies the constraints for a given step, SLEIGHT injects $\mathbf{x}_t$ (i.e., $\alpha_t^\star = 0$) unmodified, accepting zero impact at that step, instead of risking detection.

Objective and Metrics: The attacker maximizes mission-level deviation over the horizon subject to (7)-(11). Impact is measured via closed-loop outcomes (trajectory/landing deviation; for multi-UAVs, minimum pairwise separation). We use the term anomaly score only for IDS outputs such as reconstruction error or CUSUM residuals, whereas trajectory deviation, landing displacement, and trajectory-deviation MSE denote physical mission-impact metrics rather than detector scores. IDS robustness uses the deployed alarm rule (3), reporting per-step and per-flight no-alarm rates. Attacker models (S, τ_A), budgets (ε, σ), and the SLEIGHT pipeline are held fixed across IDSs; only evaluation changes via the deployed $F(\cdot)$ and $\tau_F(\cdot)$.

5 Technical Details of SLEIGHT Framework

SLEIGHT synthesizes and injects bounded telemetry perturbations under the threat model in Section 4. At step t, $\mathbf{x}_t \in \mathbb{R}^d$ denotes the feature vector constructed from MAVLink messages consumed by the offboard control process to compute references $\mathbf{r}_t$ in the PX4+MAVSDK loop (e.g., position, attitude, etc.) We write the recent window as $\mathbf{w}_t = [\mathbf{x}_{t-W+1}, \ldots, \mathbf{x}_t] \in \mathbb{R}^{W \times d}$ and inferred flight phase as $\phi(t) \in \{\text{takeoff}, \text{transit}, \text{hover}, \text{landing}\}$. The attack mask is $\mathbf{M} \in \{0,1\}^d$. SLEIGHT uses attacker-side plausibility score $S(\cdot)$ and a realism prior $D(\cdot)$ during training and online enforcement. Deployed IDS scores $F(\cdot)$ are used only for evaluation.

5.1 Overall Architecture and Workflow

SLEIGHT operates in two stages. **Offline training:** Given benign telemetry traces $\mathcal{D}_{\text{benign}} = \{\mathbf{x}_t\}$, SLEIGHT: (i) fits preprocessing (normalization and per-channel scaling) used consistently by all attacker-side models, (ii) trains an attacker-side plausibility scorer $S_\theta(\cdot)$ over windows $\mathbf{w}_t$, (iii) trains a realism prior $D_\psi(\cdot)$ over windows to score benign-likeness, (iv) trains a phase-conditioned sequential policy π_φ that proposes a per-step perturbation direction $\Delta\mathbf{x}_t$ from the current window representation and phase.

Online synthesis and injection: At each step, as shown in Fig. 3, SLEIGHT receives the current window $\mathbf{w}_t$ and phase estimate $\phi(t)$, samples a perturbation direction $\Delta\mathbf{x}_t \leftarrow \pi_\varphi(\cdot)$, and applies runtime acceptance to select the maximal scale $\alpha_t^\star \in [0,1]$ satisfying the attacker-side stealth constraint, feasibility bounds, and domain constraints. The injected telemetry is $\mathbf{x}_t' = \mathbf{x}_t + \alpha_t^\star(\mathbf{M} \odot \Delta\mathbf{x}_t)$ after domain projection, and it is written into the MAVLink stream consumed by the offboard process so that $\mathbf{r}_t$ is computed from the perturbed window. At runtime, the feasibility gate acts as a post-processor over the policy's proposed perturbation direction Δx_t, i.e., rather than injecting Δx_t directly, SLEIGHT scales and (if needed) projects the proposal to ensure all feasibility/stealth constraints in (7)- (11) are satisfied at each time step.

5.2 Telemetry Representation and Plausibility Enforcement

Telemetry Representation and Domain Projection: Let $\Omega \subset \mathbb{R}^d$ denote the set of valid values induced by message semantics and physical/interface limits (units, quantization, and fixed enumerations). SLEIGHT enforces validity via

$$\mathbf{x} \leftarrow \Pi_\Omega(\mathbf{x}) \tag{12}$$

implemented as per-channel clamping in the model-scaled space, followed by inverse scaling prior to message encoding. The same preprocessing and inverse mapping are used during offline rollouts and online injection.

Attacker-Side Plausibility Model $S(\cdot)$: Plausibility scorer $S_\theta(\cdot)$ assigns a nonnegative score to each window, with lower values corresponding to benign-like windows. We implement the plausibility scorer $S_\theta(\cdot)$ as a window reconstruction residual. We train an autoencoder $A_\theta(\cdot)$ on benign windows:

$$\min_\theta \ \mathbb{E}_{w\sim\mathcal{D}_{\text{benign}}}\left[\|\text{vec}(w) - A_\theta(\text{vec}(w))\|_2^2\right] \tag{13}$$

Plausibility score is: $S_\theta(\text{vec}(w)) = \|\text{vec}(w) - A_\theta(\text{vec}(w))\|_2^2$, where lower values correspond to more benign-like windows. The operational threshold $\tau_A(\phi)$ is selected from phase-conditioned benign quantiles as in (6). Online enforcement evaluates S_θ on the modified window w'_t.

Realism Prior $D(\cdot)$: The realism prior $D_\psi(\cdot)$ provides a continuous shaping signal within the benign region enforced by $S_\theta(\cdot)$. We train D_ψ as a discriminator over windows using benign windows as positive samples and bounded synthetic perturbations as negative samples. Negative samples are constructed by sampling δ under the same mask and feasibility bounds used at runtime and applying the projection $\Pi_\Omega(\cdot)$. The realism shaping term is:

$$R_{\text{real}}(\mathbf{w}'_t) = \log\big(\max(D_\psi(\text{vec}(\mathbf{w}'_t)), \varepsilon)\big) \tag{14}$$

with $\varepsilon > 0$. R_{real} is used during offline policy training.

5.3 Online Attack Synthesis and Injection

Policy Output, Attack Formation, and Online Injection: The policy π_φ outputs a raw perturbation direction $\Delta\mathbf{x}_t \in \mathbb{R}^d$ in the model-scaled space. Each component of $\Delta\mathbf{x}_t$ is clamped to a fixed range before masking. For a candidate scale $\alpha \in [0,1]$, SLEIGHT forms the following:

$$\mathbf{x}'_t(\alpha) = \mathbf{x}_t + \alpha\big(\mathbf{M} \odot \Delta\mathbf{x}_t\big) \tag{15}$$

then applies $\mathbf{x}'_t(\alpha) \leftarrow \Pi_\Omega(\mathbf{x}'_t(\alpha))$.

Runtime Acceptance and Maximal Feasible Scaling: Runtime acceptance selects the maximal feasible scale $\alpha_t^\star$:

$$\alpha_t^\star = \max_{\alpha\in[0,1]} \ \alpha \quad \text{s.t.} \quad \begin{cases} S_\theta\big(\text{vec}(\mathbf{w}'_t(\alpha))\big) \le \tau_A(\phi(t)), \\ \|\mathbf{M} \odot \delta_t(\alpha)\|_\infty \le \varepsilon, \\ \|\mathbf{M} \odot (\delta_t(\alpha) - \delta_{t-1})\|_\infty \le \sigma, \\ \mathbf{x}'_t(\alpha) \in \Omega \end{cases} \tag{16}$$

where $\delta_t(\alpha) = \alpha(\mathbf{M} \odot \Delta\mathbf{x}_t)$ and $\mathbf{w}'_t(\alpha) = [\mathbf{x}'_{t-W+1}, \ldots, \mathbf{x}'_t(\alpha)]$. If no $\alpha > 0$ satisfies the constraints, it injects $\mathbf{x}_t$ for that step. In implementation, (16) is solved by bisection with a fixed iteration budget B. Each iteration evaluates $S_\theta(\cdot)$ on the candidate window, applies $\Pi_\Omega(\cdot)$, and checks the ℓ_∞ bounds.

5.4 Policy Training and Optimization

Training Objective: SLEIGHT trains π_φ offline using episodic rollouts in closed-loop execution. Training optimizes an impact proxy in feature space, while mission impact is measured during evaluation using closed-loop outcomes (trajectory deviation and landing displacement). We define the per-step feature deviation proxy:

$$\Delta_{\text{feat}}(\mathbf{x}'_t, \mathbf{x}_t) = \|\mathbf{M} \odot (\mathbf{x}'_t - \mathbf{x}_t)\|_2 \tag{17}$$

An optional cap term penalizes excessive deviations:

$$P_{\text{cap}}(\mathbf{x}'_t, \mathbf{x}_t) = \max\big(0,\ \Delta_{\text{feat}}(\mathbf{x}'_t, \mathbf{x}_t) - \Delta_{\max}\big) \tag{18}$$

Let $\text{Viol}(\mathbf{x}'_t(\alpha))$ denote pre-acceptance feasibility violations for candidate α during training rollouts (zero when all checks pass). The per-step reward used in training is:

$$\begin{aligned} r_t = {} & \lambda_I \Delta_{\text{feat}}(\mathbf{x}'_t, \mathbf{x}_t) + \lambda_R R_{\text{real}}(\mathbf{w}'_t) \\ & - \lambda_S \left[S_\theta(\text{vec}(\mathbf{w}'_t)) - \tau_A(\phi(t))\right]_+ \\ & - \lambda_C P_{\text{cap}}(\mathbf{x}'_t, \mathbf{x}_t) - \lambda_F \,\text{Viol}(\mathbf{x}'_t) \end{aligned} \tag{19}$$

where $[u]_+ = \max(u, 0)$ and $\lambda_I, \lambda_R, \lambda_S, \lambda_C, \lambda_F \ge 0$ are fixed weights. The stealth term depends only on $S_\theta(\cdot)$ and $\tau_A(\cdot)$.

Policy Optimization: We optimize π_φ using a policy gradient estimator on rollout returns. Attacker state includes flattened window, phase, and previous action:

$$\mathbf{o}_t = \big[\text{vec}(\mathbf{w}_t),\ \text{onehot}(\phi(t)),\ \Delta\mathbf{x}_{t-1}\big] \tag{20}$$

The action is the proposed direction $\Delta\mathbf{x}_t \sim \pi_\varphi(\cdot \mid \mathbf{o}_t)$. With return $\hat{G}_t = \sum_{k=t}^T \gamma^{k-t} r_k$, the gradient estimate is:

$$\nabla_\varphi J(\varphi) \approx \sum_{t=1}^T \nabla_\varphi \log \pi_\varphi(\Delta\mathbf{x}_t \mid \mathbf{o}_t)\ \hat{G}_t \tag{21}$$

6 Implementation

We implement SLEIGHT on a PX4-based stack in software- and hardware-in-the-loop (SITL/HITL). The attacker runs as a user-space MAVLink mediation proxy on the autopilot to the offboard-control path in Section 4.1. The proxy forwards MAVLink traffic while rewriting a restricted subset of navigation-related fields. All rewrites preserve MAVLink message IDs, field encodings, and original stream rates. Offboard control process computes references $\mathbf{r}_t$ from rewritten stream, so perturbations propagate through closed loop without modifying PX4 firmware.

SITL/HITL Stack (1) SITL: We run PX4 SITL with JMAVSIM and connect over UDP. Missions are executed using MAVSDK offboard control. The mediation proxy runs on the same host and forwards MAVLink traffic between PX4 and the MAVSDK process. **(2) HITL:** Pixhawk hardware is connected to a companion host over a telemetry link. Mediation proxy runs on the companion and forwards MAVLink traffic between Pixhawk and offboard control process. Feature construction/runtime enforcement match SITL configuration. Simulator roles. We use JMavSim for the primary closed-loop SITL evaluations because it is the lightweight PX4 backend used with our MAVSDK offboard loop. Gazebo is used only for environmental-stress experiments with wind and sensor noise. We do not pool numerical results across simulators; claims from nominal closed-loop missions are scoped to JMavSim, while environmental-robustness results are scoped to Gazebo.

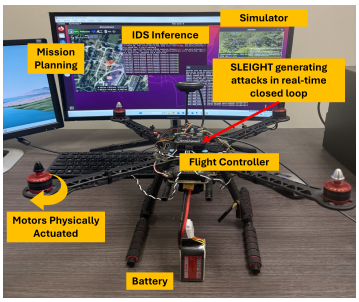


Figure 4: HITL setup: Pixhawk runs PX4, streams MAVLink over telemetry link. SLEIGHT proxy mediates the MAVLink autopilot-to-offboard control stream on companion host.

Telemetry Features and Injection Interface: At each step t, SLEIGHT builds $\mathbf{x}_t \in \mathbb{R}^d$ from the MAVLink state stream consumed by the offboard controller, using navigation messages such as `LOCAL_POSITION_NED` and `ATTITUDE`, with GPS fields from `GPS_RAW_INT` when enabled. For holistic baselines, we also ingest the required actuator-output and estimator-status fields. Feature order follows the attacker-fitted scaler, and each step forms one synchronized row from the latest required messages before scaling. PX4 publication rates are set through `MAV_CMD_SET_MESSAGE_INTERVAL`: navigation messages are requested at 10 Hz, and GPS/estimator-status messages at 5 Hz, with rates unchanged during injection. The mask $\mathbf{M} \in \{0,1\}^d$ selects attackable channels. Injection is performed by the mediation proxy, which rewrites only the corresponding outbound MAVLink fields in $\mathbf{x}'_t$ and forwards all non-masked fields unchanged.

Runtime Enforcement and Online Loop: Online injection uses only attacker-side models. At each step, the proxy maintains a sliding window $\mathbf{w}_t = [\mathbf{x}_{t-W+1}, \ldots, \mathbf{x}_t]$, applies attacker-fitted scaling, and calls the policy to propose a direction $\Delta\mathbf{x}_t$. Runtime acceptance then selects the maximal feasible scale $\alpha_t^\star$ under attacker-side stealth constraint and feasibility bounds, using bisection with iteration budget B.

Runtime Cost and Logging: The proxy logs per-step runtime `step_time` from wall-clock timestamps, as shown in Alg. 1, together with the fixed bisection budget B. Logs include attacker-side plausibility scores for benign and injected windows, $\alpha_t^\star$, ℓ_∞ perturbation and smoothness norms, phase estimates, and PX4/MAVSDK mission outcomes such as trajectory deviation, landing displacement, and minimum pairwise separation in multi-UAV runs. All norms use the same scaled feature space as $S_\theta(\cdot)$ and $\Pi_\Omega(\cdot)$, and exported logs support replay and recomputation of stealth and impact metrics. Table 1 summarizes runtime hyperparameters; Appendix A provides full list/design-choice justifications. λ_C and λ_F are retained for ablation consistency; in the reported runs, smoothness/feasibility are enforced as runtime constraints through projection and bisection rather than as soft reward penalties.

Algorithm 1: SLEIGHT Runtime Operation

Input: current feature vector $\mathbf{x}_t$, window buffer $\mathbf{w}_t$, phase estimate $\phi(t)$, policy π_φ, plausibility model S_θ, threshold $\tau_A(\cdot)$, mask $\mathbf{M}$, bounds (ε, σ), projection Π_Ω, bisection budget B
Output: injected $\mathbf{x}'_t$ and per-step logs
while *mission is active* **do**
  receive MAVLink messages and assemble $\mathbf{x}_t$; update $\mathbf{w}_t$
  $\Delta\mathbf{x}_t \leftarrow \pi_\varphi(\mathrm{vec}(\mathbf{w}_t), \phi(t), \Delta\mathbf{x}_{t-1})$; clamp $\Delta\mathbf{x}_t$ in scaled space
  $\alpha_L \leftarrow 0$; $\alpha_U \leftarrow 1$
  for $i \leftarrow 1$ **to** B **do**
    $\alpha \leftarrow (\alpha_L + \alpha_U)/2$
    $\mathbf{x}'_t(\alpha) \leftarrow \Pi_\Omega(\mathbf{x}_t + \alpha(\mathbf{M} \odot \Delta\mathbf{x}_t))$
    form $\mathbf{w}'_t(\alpha) = [\mathbf{x}'_{t-W+1}, \ldots, \mathbf{x}'_t(\alpha)]$
    $s \leftarrow S_\theta(\mathrm{vec}(\mathbf{w}'_t(\alpha)))$
    $\delta_t(\alpha) \leftarrow \alpha(\mathbf{M} \odot \Delta\mathbf{x}_t)$
    if $s \le \tau_A(\phi(t))$ ***and*** $\|\delta_t(\alpha)\|_\infty \le \varepsilon$ ***and*** $\|\delta_t(\alpha) - \delta_{t-1}\|_\infty \le \sigma$ **then**
      $\alpha_L \leftarrow \alpha$
    else
      $\alpha_U \leftarrow \alpha$
    end
  end
  $\alpha_t^\star \leftarrow \alpha_L$
  if $\alpha_t^\star = 0$ **then**
    $\mathbf{x}'_t \leftarrow \mathbf{x}_t$
  else
    $\mathbf{x}'_t \leftarrow \Pi_\Omega(\mathbf{x}_t + \alpha_t^\star(\mathbf{M} \odot \Delta\mathbf{x}_t))$
  end
  rewrite the MAVLink fields corresponding to the attackable channels in $\mathbf{x}'_t$ and forward messages to the offboard control process
  define $\mathbf{w}'_t = [\mathbf{x}'_{t-W+1}, \ldots, \mathbf{x}'_t]$
  log $(t, \phi(t), S_\theta(\mathrm{vec}(\mathbf{w}_t)), S_\theta(\mathrm{vec}(\mathbf{w}'_t)), \tau_A(\phi(t)), \alpha_t^\star, \|\delta_t\|_\infty, \|\delta_t - \delta_{t-1}\|_\infty, \mathtt{step_time})$
end

7 Evaluation

We evaluate SLEIGHT in closed-loop flight using four metrics: (i) **Evasion Success (%)**, the per-window/timestep no-

Table 1: Runtime and training hyperparameters used in SLEIGHT.

Parameter	Value	Description
W	50	Telemetry window length used by the detector and attacker-side scorer
d	11	Feature dimensionality of each telemetry vector
B	8	Bisection budget for runtime perturbation scaling
γ	0.99	Discount factor used during policy training
$\alpha_{\max}$	0.25	Maximum runtime perturbation scale
κ	0.01	Safety margin below the attacker-side plausibility threshold
λ_I	1.0	Mission-impact reward weight
λ_R	1.0	Realism-prior reward weight
λ_S	50.0	Stealth/plausibility penalty weight
λ_C	0	Control-smoothness penalty weight
λ_F	0	Feasibility/projection penalty weight
f_{ctrl}	10 Hz	Runtime telemetry/control update rate

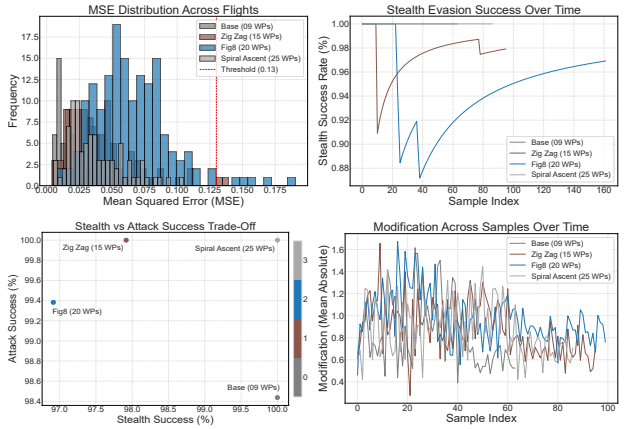


Figure 5: Performance of SLEIGHT with varying flight durations.

alarm rate under the deployed IDS; where applicable, we also report per-flight no-alarm rate to assess mission-level robustness. (ii) **Attack Success (%)** as the fraction of manipulations achieving the attack goal (e.g., sustained GPS offset or route deviation), (iii) **Perturbation Magnitude** as the mean/max $|\delta_t|$ per timestep and capped violations, and (iv) **Mission Impact** measured through trajectory deviation (RMSE between intended and actual paths), waypoint miss distance, and landing displacement. We also report runtime overhead and enforcement behavior (α_t^* distribution and rejection rate). Trajectory-deviation MSE/RMSE denotes physical mission deviation and is distinct from IDS anomaly score or autoencoder reconstruction error.

7.1 Generalization: Missions and Environment

We first evaluate whether SLEIGHT's stealth and attack effectiveness generalize beyond a single mission profile, progressing from individual flight plans to dynamic maneuver complexity, to realistic environmental perturbations, and finally to coordinated swarm operations.

Performance Across Flight Paths and Durations: To evaluate duration sensitivity, we created four flight plans with increasing numbers of waypoints (WPs): *Base* (9 WPs), *Zig Zag* (15 WPs), *Fig8* (20 WPs), and *Spiral Ascent* (25 WPs). In each mission, a single-sensor attack (SSA) targeted GPS coordinates. Our results in Fig. 5 and Table 2 indicate consistently high evasion across all flight durations, with rates above 96%. Attack success remains consistent, between 85% and

Table 2: Aggregated Success Metrics Across Flights

Flight	Evasion Success (%)	Attack Success (%)	Avg Modification (Sum)
Base	100	85.94	0.83
Zig Zag	97.92	86.46	0.88
Fig 8	96.91	85.19	0.93
Spiral Ascent	100	85.06	0.88

Table 3: SLEIGHT's performance with varying flight complexity.

Flight Complexity	Mean Stealth Rate	Mean MSE
No Jumps	1.000	0.041
Every 2nd WP	0.985	0.045
Every 3rd WP	0.993	0.045
Every 5th WP	0.985	0.046

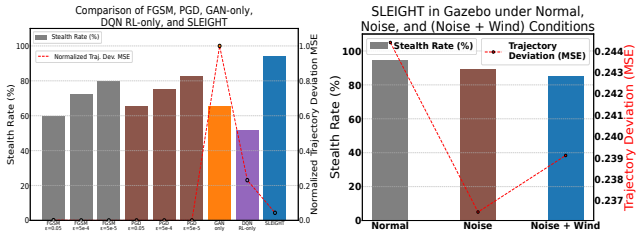


Figure 6: SLEIGHT gradient- and non-gradient attack baselines, and under varied environmental conditions (wind, noise) Gazebo.

86.5%, suggesting that small incremental GPS manipulations accumulate into measurable deviations regardless of mission length. Similarly, we observe consistent modification magnitude averaging between 0.82 and 0.93, while reconstruction error remains below the bound, demonstrating that SLEIGHT maintains subtle perturbations over extended trajectories. For cross-method comparability, we use a single phase-specific reference bound, while runtime enforcement continues to rely on the phase-effective thresholds $\tau_{\text{eff}}(\phi)$.

Effects of Flight Complexity on Stealth and Attack Success: We increased flight complexity by introducing sudden altitude jumps (10–35 m shifts), applied under four configurations: (1) *no jumps*, (2) *jumps every 5th WP*, (3) *jumps every 3rd WP*, and (4) *every other WP* (representing the most complex case). Results in Table 3 show a gradual decrease in average stealth as altitude jumps become more frequent. However, even in the most complex scenario, SLEIGHT maintained a stealth rate above 98.5%. Reconstruction error increases slightly under frequent jumps, but remains close to the detection threshold, indicating that adversarial samples remain stable despite added trajectory variability. Our analysis demonstrates that SLEIGHT tolerates maneuver-induced instability while preserving stealth.

Impact of Environmental Conditions: We evaluate SLEIGHT in Gazebo simulator under realistic disturbances, including moderate GPS noise and wind conditions with 8 m/s primary component along the x-axis and additional cross-winds. We then performed adversarial injections online. Fig. 6(b) shows that stealth decreases from near 100% under normal conditions to approximately 80–85% when noise or combined noise and wind are present, while the MSE remains relatively stable around 0.238–0.244. This

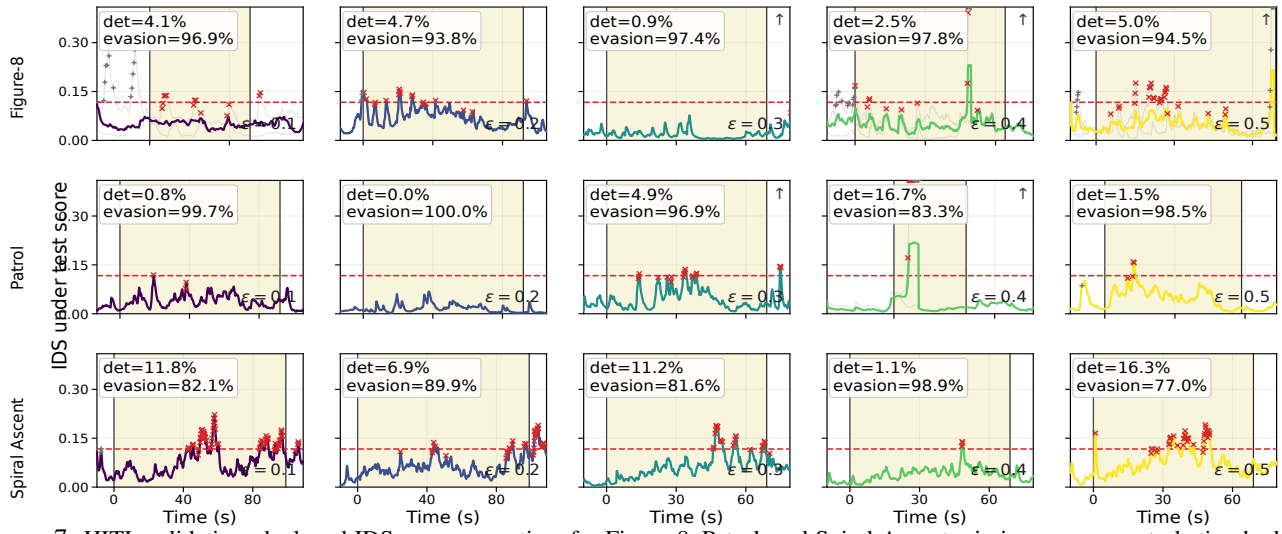


Figure 7: HITL validation: deployed IDS scores over time for Figure-8, Patrol, and Spiral Ascent missions across perturbation budgets $\varepsilon \in \{0.1, \ldots, 0.5\}$. Each panel reports detection and evasion rates. Dashed lines indicate IDS thresholds; arrows (↑) mark transient threshold exceedances during phase transitions. Detection is the fraction of timesteps (or windows) where the deployed IDS score exceeds its threshold.

modest reduction in stealth suggests that wind gusts occasionally push sensor readings closer to the IDS threshold. However, SLEIGHT's reward-based adversarial generator adapts quickly and largely maintains evasion. Hence, even under realistic environmental conditions, SLEIGHT achieves high evasion while inducing trajectory deviation, showing robustness at the single-vehicle level under environmental uncertainty. We use the two simulators for distinct roles. **JMavSim** supports the main closed-loop SITL evaluations, while **Gazebo** is used only for environmental-stress tests with wind and sensor noise. Results are not pooled across simulators: JMavSim supports nominal-flight claims, Gazebo supports environmental-robustness claims, and HITL Pixhawk results provide independent firmware-level validation.

Hardware-/Software-in-the-Loop Validation: To test transfer beyond simulation, we ran the same attack configurations on Pixhawk firmware using HITL. Fig. 7 reports deployed IDS scores for Figure-8, Patrol, and Spiral Ascent missions across $\varepsilon \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. Evasion ranges from 77.0% to 100.0%, with most configurations above 90%. Moderate budgets ($\varepsilon \in \{0.2, 0.3\}$) sustain high evasion ($\geq$93.8%) across all missions, while aggressive budgets (ε=0.5) sometimes increase detection on longer missions, e.g., 16.3% on Spiral Ascent, due to tighter firmware-level dynamics. Thus, the SITL stealth-impact trade-off persists across hardware timing, sensor noise, and estimator dynamics. Fig. 8 shows one HITL run with trajectory deviation, attacker-side plausibility score S, and deployed IDS score. Drift accumulates to about 6 m, S remains bounded, and the IDS score stays below threshold for most of the flight. Isolated in-flight and post-flight alarms occur near phase transitions, where the effective threshold tightens; runtime scaling reduces α_t^* and restores stealth within a few steps.

Takeaway: Across four flight plans, four complexity levels, two environmental conditions, and HITL validation, SLEIGHT sustains >96% nominal evasion and >80% under combined noise and wind, with 85–86.5% attack success and bounded modification. HITL confirms that SITL-derived stealth–impact trends transfer to real firmware.

7.2 Impact on Swarm Missions

We extended SLEIGHT to a UAV swarm and varied which subset of vehicles received adversarial inputs, for example, perturbing Veh-2 and Veh-3 (Fig. 9). Stealth was evaluated against the *phase-aware* threshold $\tau_{\text{eff}}(\phi)$ during transit/hover, while impact was measured at both vehicle and formation levels using per-vehicle *AE score vs.* τ_{eff}.

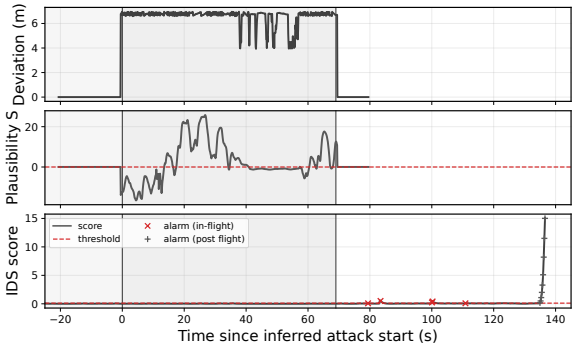


Figure 8: HITL time-series: (top) trajectory deviation in meters, (middle) attacker-side plausibility score S, (bottom) deployed IDS score vs. threshold with in-flight and post-flight alarm markers.

Table 4 (A) summarizes per-vehicle smoothed AE scores reported relative to threshold. We computed *instantaneous drift* for vehicle i as $\text{drift}_i(t) = \|p_i^{\text{EN}}(t) - p_i^{\text{EN,ref}}(t)\|_2$ using time-aligned benign references in the local East–North

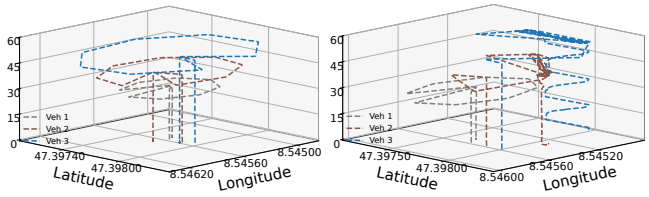


Figure 9: Swarm UAV missions (left) all benign and (right) with vehicles 2 and 3 under SLEIGHT attacks.

Table 4: SLEIGHT on Swarm (A) AE vs. effective threshold; (B) distance from target landing point (meters).

(A) IDS AE (smoothed) normalized by $\tau_{\text{eff}}(\phi)$

Vehicle	Peak (AE/τ_{eff})	Time of peak [s]	Steady (AE/τ_{eff})	τ_{eff}
veh1	0.33	8-9	0.12	1
veh2	0.27		0.07	
veh3	0.32		0.1	

(B) Distance from target landing point (m)

Case	veh1	veh2	veh3	Note
Target	0.00	0.00	0.00	reference
Benign	0.04	0.09	0.08	small bias
veh2 attacked	0.04	52.45	21.66	veh2 largest error
veh2,3 attacked	0.05	58.32	57.73	multi-veh drift
veh1-3 attacked	60.14	88.32	61.04	largest error

(EN) plane. To assess *formation metrics* we defined $p_i^{\text{EN}}(t) = [E_i(t), N_i(t)]^\top$ as the local EN position of vehicle i. The *minimum pairwise separation* is:

$$d_{\min}(t) = \min_{i<j} \left\| p_i^{\text{EN}}(t) - p_j^{\text{EN}}(t) \right\|_2$$

computed over all vehicle pairs at each t; we monitor $d_{\min}(t)$ against a safety bound *sep_min*. Formation dispersion is RMS distance to instantaneous centroid $\bar{p}(t) = \frac{1}{3}\sum_i p_i^{\text{EN}}(t)$:

$$\text{RMS}(t) = \sqrt{\frac{1}{3}\sum_{i=1}^{3} \left\| p_i^{\text{EN}}(t) - \bar{p}(t) \right\|_2^2}$$

Table 4 (B) reports per-vehicle distance from the target landing point across cases. The results show: (a) **Stealth is preserved per vehicle:** Table 4 (A) shows AE peaks and steady-state values $\ll \tau_{\text{eff}}(\phi)$, matching the single-UAV trend. (b) **Coordinated drift without instability:** In the Veh-2&3 setting, Fig. 9(b) exhibits stable lateral offsets in the attacked UAVs while the non-attacked UAV tracks nominally-selective, closed-loop impact under per-step stealth. (c) **Formation safety maintained:** The separation trace in Fig. 10 shows $d_{\min}(t) \geq \text{sep_min}$ throughout, indicating no unsafe pairwise encounters as offsets accumulate. (d) **Mission-level effect scales with the targeted subset:** Table 4 (B) shows larger landing-point displacement when multiple UAVs are perturbed; the Veh-2&3 case produces landing displacements of $\approx$58 m per attacked vehicle while remaining under $\tau_{\text{eff}}(\phi)$, and attacking all three vehicles yields displacements >60 m.

7.3 Impact of Attack Surface and Adversarial Capability

SLEIGHT's effectiveness depends on the attacker's control of sensor streams and knowledge of the operational stealth

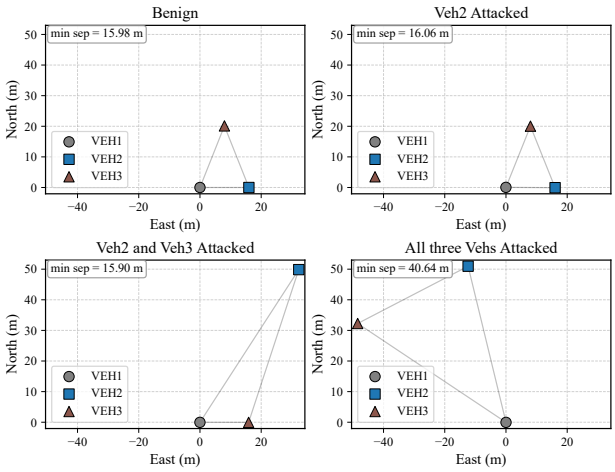


Figure 10: Swarm positions (local EN) and min-pairwise separation $d_{\min}(t)$ across pairs (1,2),(1,3),(2,3); dashed line marks sep_min = 10 m. Veh-2 and 3 are perturbed while $d_{\min}(t) \geq$ sep_min.

threshold. We move from multi-sensor manipulation to isolating individual modalities and finally to assessing how the threshold estimate τ_A governs achievable stealth and impact. SLEIGHT does not require a separately designed policy for each sensor combination. We train a single generator/policy over the telemetry feature vector and restrict the active attack surface via a sensor/feature mask M and per-feature feasibility bounds. At runtime, the generated perturbation is multiplied by M, projected into the valid domain Π_Ω, and then scaled by bisection to satisfy the attacker-side stealth constraint. Thus, GPS-only, single-sensor, and multi-sensor settings use the same attack pipeline; only the enabled feature dimensions change according to the attacker-controlled streams.

Multi-Sensor Attacks: To evaluate whether SLEIGHT can coordinate attacks across multiple sensing modalities, we extended our framework to manipulate several sensor streams concurrently, combining GPS, accelerometer (Accel), and magnetometer (Mag) readings in 4 combinations: (1) GPS + Accel + Mag, (2) GPS + Accel, (3) Accel + Mag, and (4) GPS + Mag. As shown in Table 5, all multi-sensor attacks achieve evasion rates >80%, with the GPS+Accel+Mag configuration exhibiting the highest rate. We use the same phase-conditioned policy procedure across these settings; the attack mask M selects the active channels for each sensor combination. Joint manipulation produces more noticeable changes in flight metrics, particularly in speed and heading, compared to single-sensor scenarios. We observe that while combining the three sensors (GPS+Accel+Mag) increases potential for detection (due to cross-sensor checks), SLEIGHT's reward-driven adversarial generator keeps all modifications relatively small yet consistent. Our results show that SLEIGHT scales with the attack surface and can coordinate perturbations across heterogeneous sensors.

Single-Sensor Attacks Across Modalities: We attack GPS, accelerometer, and magnetometer streams individually. Ta-

Table 5: SLEIGHT's performance when a combination of multiple sensors is attacked (GPS + accelerometer + magnetometer).

Sensor Combination	Traj. Deviation (m)	Speed Change	Heading Change	Evasion (%)
GPS+Accel+Mag	140	0.010	0.001	99
GPS+Accel	130	0.008	0.001	100
Accel+Mag	20	0.012	-0.003	98
GPS+Mag	95	0.006	0.001	98

Table 6: SLEIGHT when individual sensors are attacked.

Sensor	Traj. Deviation (m)	Speed Change	Heading Change	Evasion (%)
GPS	67	2.8	2.6	100
Accel	48	2.5	2.6	96
Mag	44	2.5	2.6	100

ble 6 reports trajectory deviation, speed/heading changes, and stealth success. GPS causes the largest offset, up to 70 m, reflecting the UAV's reliance on position; operationally, this can exceed typical low-altitude corridor or geofence half-widths (10–25 m), pushing the UAV outside its intended safety envelope while staying below the IDS threshold. Accelerometer attacks create moderate speed oscillations with high stealth because small acceleration changes resemble benign turbulence. Magnetometer attacks produce smaller immediate impact than GPS, but heading instability accumulates into gradual drift. Overall, SLEIGHT is sensor-agnostic, while impact magnitude and type depend on the targeted modality.

Effect of Attacker Threshold Estimation Capability: SLEIGHT chooses the per-step scale α_t using the attacker-side threshold τ_A, computed as a phase-conditioned reconnaissance quantile (see (6)), rather than the defender threshold τ_F. We evaluate this dependence under three cases: (i) calibrated τ_A, (ii) $\pm 20\%$ estimation error, and (iii) a time-varying effective boundary. Fig. 11 shows CUSUM stealth violations, drift rate, projection scale α versus margin $(\tau_D - \mathrm{AE})$, and attacker advantage $\Delta\tau(t) = \tau_A - \tau_D$. With calibrated estimates, violations remain near zero, α quickly reaches its feasible maximum when the margin is positive, and drift grows steadily. Under bounded uncertainty, stealth is mostly preserved, while impact decreases slightly because scaling reduces perturbation size as the margin shrinks. With a drifting boundary, outdated estimates reduce stealth: threshold tightening can make the margin negative, increase violations, and create late drift spikes when brief feasible windows appear. Thus, SLEIGHT tolerates moderate threshold error by trading impact for stealth, but drifting boundaries require periodic recalibration of τ_A or a larger safety margin κ.

7.4 Robustness Against IDS Architectures

We now evaluate SLEIGHT against a diverse set of IDS architectures to determine whether its stealth and impact generalize beyond a single detector design. Our analyses begin by positioning SLEIGHT relative to traditional gradient-based

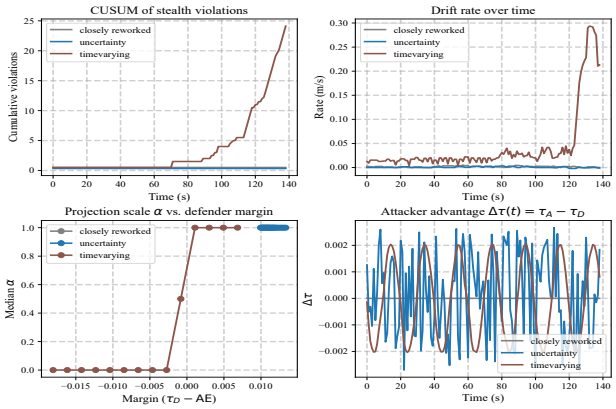


Figure 11: SLEIGHT's sensitivity to the attacker's threshold estimate: calibrated, $\pm 20\%$ error, and a time-varying threshold. Panels: CUSUM of violations, drift rate, α vs. margin, and gap $\Delta\tau(t)$.

adversarial attacks, then compare performance across standard ML-based IDS models, and finally assess evasion against representative holistic and control-aware IDS frameworks, namely ConFIDe [10] and PID-Piper [17]. We validated our ConFIDe and PID-Piper benchmark implementations (Appendix E and D) against the available public artifacts and published detector descriptions. Since the released artifacts are not directly reproducible in our PX4/MAVSDK closed-loop setting without environment, preprocessing, and log-format adaptation, we re-instantiated the detector pipelines using the same feature definitions and alarm logic. ConFIDe was validated by reproducing its 37-feature autoencoder-style reconstruction detector and confirming low alarm rates on benign UAV logs. PID-Piper was validated by reproducing its LSTM-based attitude prediction and residual/CUSUM alarm logic. We therefore use these as faithful reimplementations as detector-side benchmarks for evasion evaluation.

Comparison with Prior Attack Baselines: We compare SLEIGHT with FGSM, PGD, GAN-only, and RL-only baselines. GAN-only removes closed-loop scaling/enforcement, while RL-only uses a bounded DQN policy without the GAN realism prior/continuous feasibility projection. Fig. 6(a) reports stealth rate and normalized trajectory-deviation MSE. FGSM/PGD achieves moderate stealth at a small ε but near-zero mission impact. GAN-only increases deviation but drops to 65.3% no-alarm rate; RL-only reaches 51.6%. SLEIGHT achieves 94.3% no-alarm over active attack windows (96.4%), maintaining bounded, physically plausible deviation, showing a stronger stealth-impact tradeoff than baseline attacks.

Robustness Across ML-Based IDSs: To evaluate SLEIGHT across detector families, we trained AE, K-means, and OCSVM models on the same benign data. Fig. 12(a) shows 100% evasion against AE, indicating that SLEIGHT keeps samples near the learned reconstruction manifold. K-means and OCSVM are more resilient, with evasion rates of 90.2% and 87.3%. Fig. 12(b) shows that SLEIGHT remains below the AE threshold, with occasional spikes for the other detec-

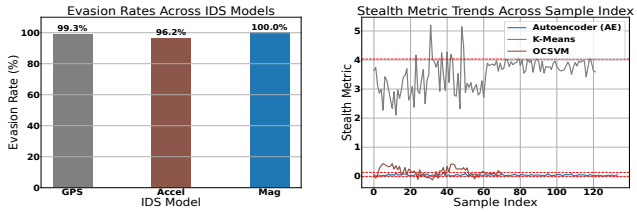


Figure 12: SLEIGHT's performance with various IDS models.

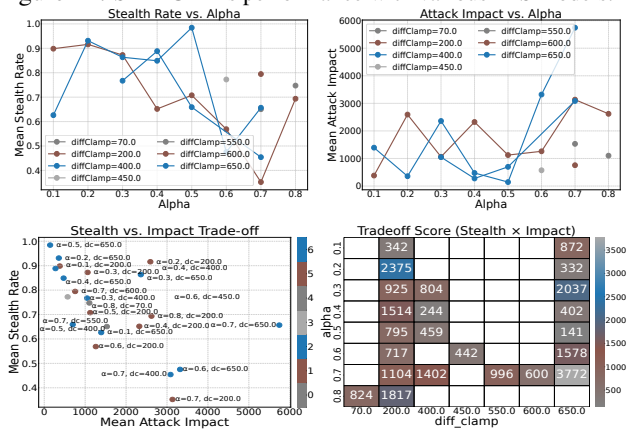


Figure 13: Evasion against ConFIDe: (a) stealth rates, (b) attack impacts, (c) stealth-impact trade-off, and (d) trade-off heatmap.

tors. Thus, reconstruction-based IDSs are most susceptible, while boundary- and clustering-based methods add partial robustness but do not prevent evasion.

Evasion of Holistic IDS (ConFIDe): We evaluated SLEIGHT against ConFIDe, a holistic autoencoder-based IDS that applies reconstruction thresholds over a 37-dimensional feature vector. We varied (1) α (blending for actuator/telemetry perturbations), (2) `diffClamp` (max scaled-space actuator change), and (3) a physical penalty discouraging implausible control (used as an ablation knob). At runtime, SLEIGHT samples a masked increment, projects to the valid domain Π_Ω, and performs a line-search/bisection on α so that the candidate satisfies the effective *phase* threshold $T \equiv \tau_{\text{phase}}(\phi(t)) - \kappa$ and the feasibility limits; only then is it injected. Fig. 13 summarizes: (i) stealth trends across α and `diffClamp`, (ii) induced impacts, (iii) the stealth–impact trade-off, and (iv) a heatmap of optimal regimes. SLEIGHT reliably evades ConFIDe, sustaining >95% stealth in moderate settings (Fig. 13(a)) while delivering meaningful actuator-level deviations (Fig. 13(b)); the heatmap (Fig. 13(d)) highlights balanced configurations where both stealth and impact are jointly high. We have elucidated the implementation details of ConFIDe in Appendix B.

Evasion of Control-Aware IDS (PID-Piper): We evaluate SLEIGHT against PID-Piper, which predicts roll, pitch, and yaw with LSTMs and applies one-sided CUSUM to cumulative residuals. SLEIGHT perturbs IMU, GPS, and EKF inputs, using Alg. 1 with $S_\theta(\cdot)$ and $\tau_A(\phi)$ to accept rewrites online; PID-Piper residuals and alarms are evaluated offline on the resulting telemetry. We vary max perturbation Δ, clamps

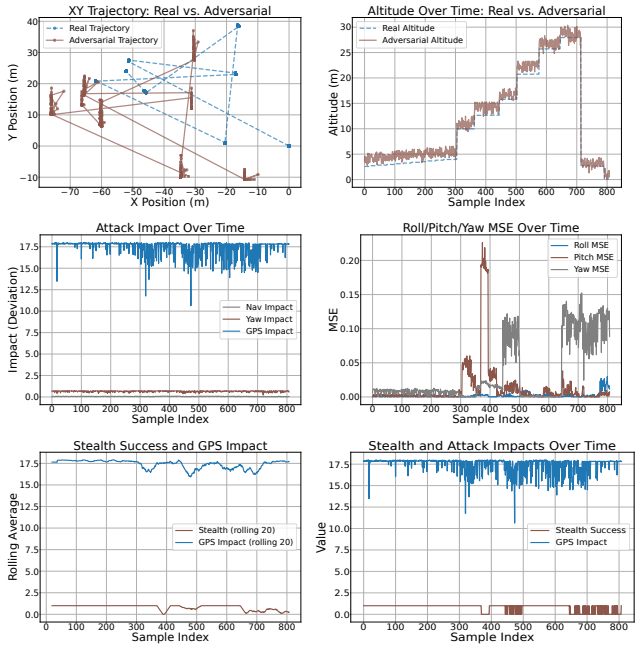


Figure 14: SLEIGHT evasion performance against PID-Piper IDS. (Top row) XY and altitude trajectories (real vs. adversarial); (Middle row) attack impact and per-axis MSE over time; (Bottom row) rolling avg. of stealth and GPS impact, and overall stealth-impact.

on `navRoll`, `navPitch`, and `yaw`, and the physical penalty. Fig. 14 shows controlled XY/altitude deviations, persistent impact, tight per-axis MSE residual control, rolling stealth above 90%, and the time-varying stealth–impact balance. Thus, SLEIGHT evades PID-Piper while producing physically plausible multi-sensor impact. Appendix C gives implementation details. We summarize cross-IDS stealth-impact using a unified frontier plot. Each point is one mission-ε configuration, with stealth as deployed no-alarm rate and impact as mean trajectory drift $\bar{d}$ over the attack window. Fig. 15 shows the Pareto frontier, where high stealth can still permit substantial mission deviation for holistic IDSs.

Takeaway: SLEIGHT evades all tested IDS families: AE (≈99%), K-means (90.2%), OCSVM (87.3%), ConFIDe (>95%), and PID-Piper (>90%) [Fig. 15]. Reconstruction-based detectors are most susceptible, while boundary- and residual-based detectors reduce but do not remove the stealth-impact tradeoff. FGSM/PGD fail to maintain closed-loop stability at budgets where SLEIGHT remains reliable.

7.5 Runtime Overhead and Practicality

We measure SLEIGHT's runtime overhead as the time from telemetry read to adversarial write per flight step. Across 50 experimental runs, the logged overhead averages 1.917×10^{-4} s, with observed values ranging between 1.2×10^{-4} s and 3.2×10^{-4} s, which is SLEIGHT's worst-case measured proxy overhead per step, and this is well within the 0.1 s period at 10 Hz. This overhead is negligible compared to the

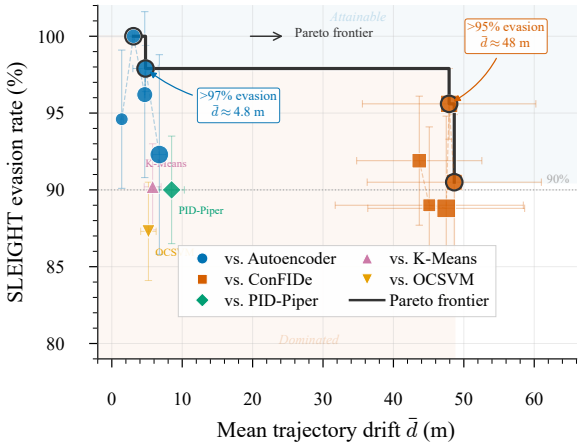


Figure 15: Stealth-impact frontier across IDSs. Stealth is the deployed per-window (or per-step) no-alarm rate under $F(\cdot)$ with thresholds $\tau_F(\phi)$. Impact is mean trajectory drift $\bar{d} = \frac{1}{|\mathcal{T}_{atk}|}\sum_{t\in\mathcal{T}_{atk}} \|p^{EN}(t) - p^{EN,ref}(t)\|_2$. Markers denote mean across runs (error bars: ± 1 std); marker size scales with ε. The Pareto frontier is computed over the plotted mean operating points.

flight controller's typical control cycle of 10–20 ms, indicating that SLEIGHT operates within real-time constraints and does not introduce measurable latency in the control loop. The per-step cost comprises policy inference, attacker-side plausibility evaluation $S_\theta(\cdot)$, and bisection with fixed budget B; all three components execute on CPU (GPU not required).

7.6 Discussion and Potential Defenses

Discussion: Our results show that offline-trained, static UAV IDSs remain vulnerable when an attacker can approximate the score distribution from reconnaissance data. Across AE, clustering, one-class, and holistic controller/actuator-aware IDSs, SLEIGHT produces small, smooth, multi-sensor perturbations that cause trajectory drift and landing offsets while staying below the phase-aware threshold τ_{eff}. This persists across flight plans, sensor choices, moderate environmental variation, HITL firmware execution, and swarm runs with only a subset of vehicles perturbed. The main driver is closed-loop feedback: perturbed telemetry changes estimator state and controller actions, amplifying small per-step changes into cumulative drift that offline/open-loop tests miss. The results also show a structural tradeoff: higher impact lies near the plausible-dynamics boundary, and residual checks such as PID-Piper reduce but do not remove fully stealthy manipulation space. HITL confirms that SITL stealth–impact trends persist under hardware timing, sensor noise, and estimator dynamics, suggesting that log-based open-loop evaluations can overestimate IDS robustness.

Countermeasures: The following approaches highlight directions for potential defenses. (1) *Adaptive or time-varying thresholds* (with mild randomization) to make boundary tracking harder. Our threshold estimation analysis demonstrates that as the defender tightens the threshold online, SLEIGHT's stealth degrades and the drift rate becomes spiky. Even basic adaptation therefore reduces impact, though it does not fully eliminate successful attacks. More sophisticated adaptive strategies and lightweight threshold randomization can be evaluated within this framework as future work. (2) *Cross-sensor and physics checks* (e.g., EKF innovation/residual consistency between GPS, IMU, magnetometer, estimator outputs, and control commands) to surface coordinated multi-channel perturbations [9, 31]. (3) *Online/periodic updates and ensembles* (poisoning-aware) to avoid a single fixed boundary. (4) *Closed-loop adversarial calibration*: incorporate SLEIGHT-like traces for calibration or adversarial training to target the specific failure modes exposed in this evaluation.

Limitations and Scope: SLEIGHT evaluates IDS robustness against an in-path telemetry-integrity adversary at the autopilot–offboard MAVLink mediation point (Section 4). This matches PX4 offboard/companion-relay deployments where limited navigation fields can be rewritten without modifying firmware, missions, or safety logic. It does not model firmware compromise, mission reprogramming, RF denial, or physical sensor reconfiguration. Our SITL/HITL experiments use the same mediation proxy and runtime enforcement loop. Different stacks, estimator tuning, or offboard controllers may shift the numerical stealth–impact boundary, but the methodology transfers by re-instantiating x_t, M, and Π_Ω for the target telemetry interface. We focus on bounded, temporally coherent, message-valid perturbations at configured update rates; stronger channel integrity reduces exposure to in-path rewriting, while redundant sensing may require coordinated multi-channel manipulation.

8 Conclusion

We proposed SLEIGHT, a closed-loop attack analytics framework for stress-testing IDS robustness that characterizes the stealth-impact frontier of UAV IDSs under physically plausible telemetry manipulation. It separates the deployed detection $F(\cdot)$ (evaluation-only) from the attacker-side plausibility $S(\cdot)$ and phase thresholds $\tau_A(\phi)$ (synthesis and runtime enforcement), enabling real-time perturbations without querying the IDS outputs. Using benign reconnaissance telemetry, SLEIGHT learns attacker-side models, generates phase-conditioned perturbation directions, and applies per-step maximal-feasible scaling under plausibility, amplitude, smoothness, and domain constraints. Multi-sensor and multi-UAV settings, and multiple IDS families (including controller- and actuator-aware designs), SLEIGHT maintains high no-alarm rates while inducing accumulating mission deviation (drift and landing displacement). Pixhawk HITL confirms the same constraint-limited behavior seen in SITL under firmware timing and estimator dynamics. These results show that offline or open-loop evaluations can overestimate IDS robustness in closed-loop operation, motivating IDS calibration and design for adaptive, mission-aware real-time adversaries.

Acknowledgements

This research was supported in part by the Department of Energy (DOE) under awards DE-CR0000024 and DE-CR0000046.

Ethical Considerations

This work evaluates UAV IDS robustness using an adversarial testing framework.
Stakeholders: Relevant stakeholders include UAV manufacturers, IDS developers, security researchers, potential attackers, and the public affected by UAV safety. ***Minimization of harm:*** All evaluations were conducted in simulation or contained environments. We did not target outside vehicles, infrastructure, operational networks, live deployments, or production devices, so the research posed no physical risk. ***Respect for privacy:*** The work involved no human subjects, recruitment, or user interaction. All data were author-generated or publicly available. We also avoid singling out specific vendors, platforms, or organizations; the evaluation is platform-agnostic and focused on methodological insights, not claims about commercial systems. ***Legal compliance and public interest:*** All experiments complied with applicable laws and institutional drone-research policies. No unauthorized access, interference, or circumvention of safeguards was performed. The goal is to strengthen defensive technologies and support safer autonomous-system design. ***Disclosure and potential misuse:*** Although adversarial evaluation methods could be misused, the primary value is defensive: identifying weaknesses before exploitation. We therefore weigh dissemination risks against the societal benefit of improved UAV security. This work supports rigorous evaluation of safety-critical IDSs and provides proactive defensive value.

Open Science

For reproducibility and to encourage further research, our work is available at [39].

References

[1] Ark cannode. https://docs.px4.io/v1.13/en/uavcan/ark_cannode. Last updated 2025-07-10.

[2] I2c bus (development overview). https://docs.px4.io/main/en/sensor_bus/i2c_development.

[3] I2c bus peripherals. https://docs.px4.io/main/en/sensor_bus/i2c_general.

[4] Mavlink message signing. https://docs.px4.io/main/en/mavlink/message_signing.

[5] Azza Allouch, Omar Cheikhrouhou, Anis Koubâa, Mohamed Khalgui, and Tarek Abbes. Mavsec: Securing the mavlink protocol for ardupilot/px4 unmanned aerial systems. In *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pages 621–628. IEEE, 2019.

[6] Alejandro Almeida, Muneeba Asif, Md Tauhidur Rahman, and Mohammad Ashiqur Rahman. Side-channel-driven intrusion detection system for mission critical unmanned aerial vehicles. In *2024 25th International Symposium on Quality Electronic Design (ISQED)*, pages 1–9. IEEE, 2024.

[7] Ebtihaj Alshahrani, Daniyal Alghazzawi, Reem Alotaibi, and Osama Rabie. Adversarial attacks against supervised machine learning based network intrusion detection systems. *Plos one*, 17(10):e0275971, 2022.

[8] Alvaro Alva, Luis Martinez Moreno, Muneeba Asif, Alvi Ataur Khalil, Mohammad Ashiqur Rahman, Alfredo Cuzzocrea, and Shahriar Hossain. Secured uav navigation: A novel intrusion detection system based on pwm signal analysis. In *2024 11th IEEE Swiss Conference on Data Science (SDS)*, pages 174–180. IEEE, 2024.

[9] Mohammed Y Alzahrani. Enhancing drone security through multi-sensor anomaly detection and machine learning. *SN Computer Science*, 5(5):651, 2024.

[10] Muneeba Asif, Mohammad Ashiqur Rahman, Kemal Akkaya, and Ahmad Mohammad. Confide: A pwm-driven control-fused intrusion detection system for hardware security in unmanned aerial vehicles. In *Proceedings of the 19th ACM Asia Conference on Computer and Communications Security*, pages 886–901, 2024.

[11] Muneeba Asif, Mohammad Ashiqur Rahman, Kemal Akkaya, Hossain Shahriar, and Alfredo Cuzzocrea. Adversarial data-augmented resilient intrusion detection system for unmanned aerial vehicles. In *2023 IEEE International Conference on Big Data (BigData)*, pages 5428–5437. IEEE, 2023.

[12] Muhammad Zohaib Butt, Nazri Nasir, and Rozeha Bt A Rashid. A review of perception sensors, techniques, and hardware architectures for autonomous low-altitude uavs in non-cooperative local obstacle avoidance. *Robotics and Autonomous Systems*, 173:104629, 2024.

[13] Paolo Cerracchio, Stefano Longari, Michele Carminati, Stefano Zanero, et al. Investigating the impact of evasion attacks against automotive intrusion detection systems. In *Symposium on Vehicles Security and Privacy (VehicleSec) 2024*, pages N–A, 2024.

[14] Ozlem Ceviz, Sevil Sen, and Pinar Sadioglu. A survey of security in uavs and fanets: Issues, threats, analysis of attacks, and solutions. *IEEE Communications Surveys & Tutorials*, 2024.

[15] Aniruddh Chandratre, Tomas Hernandez Acosta, Tanmay Khandait, Giulia Pedrielli, and Georgios Fainekos. Stealthy attacks formalized as stl formulas for falsification of cps security. In *Proceedings of the 26th ACM International Conference on Hybrid Systems: Computation and Control*, pages 1–8, 2023.

[16] Pritam Dash, Mehdi Karimibiuki, and Karthik Pattabiraman. Out of control: stealthy attacks against robotic vehicles protected by control-based techniques. In *Proceedings of the 35th Annual Computer Security Applications Conference*, ACSAC '19, page 660–672, New York, NY, USA, 2019. Association for Computing Machinery.

[17] Pritam Dash, Guanpeng Li, Zitao Chen, Mehdi Karimibiuki, and Karthik Pattabiraman. Pid-piper: Recovering robotic vehicles from physical attacks. In *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 26–38. IEEE, 2021.

[18] Alessandro Erba, John H Castellanos, Sahil Sihag, Saman Zonouz, and Nils Ole Tippenhauer. Confusense: Sensor reconfiguration attacks for stealthy uav manipulation. In *3rd USENIX Symposium on Vehicle Security and Privacy (USENIX Vehicle Security 25)*, 2025.

[19] Jeffrey Greer IV. *MITRE Attack framework adaptation in UAV usage during surveillance and reconnaissance missions*. Mississippi State University, 2024.

[20] J Dinal Herath, Ping Yang, and Guanhua Yan. Real-time evasion attacks against deep learning-based anomaly detection from distributed system logs. In *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy*, pages 29–40, 2021.

[21] Rune Hylsberg Jacobsen and Ali Marandi. Security threats analysis of the unmanned aerial vehicle system. In *MILCOM 2021-2021 IEEE Military Communications Conference (MILCOM)*, pages 316–322. IEEE, 2021.

[22] Seonghoon Jeong, Eunji Park, Kang Uk Seo, Jeong Do Yoo, and Huy Kang Kim. Muvids: False mavlink injection attack detection in communication for unmanned vehicles. In *Workshop on automotive and autonomous vehicle security (AutoSec)*, volume 2021, page 25, 2021.

[23] Shixiong Jiang, Mengyu Liu, and Fanxin Kong. Vulnerability analysis for safe reinforcement learning in cyber-physical systems. In *2024 ACM/IEEE 15th International Conference on Cyber-Physical Systems (ICCPS)*, pages 77–86, 2024.

[24] Amir Khazraei, Spencer Hallyburton, Qitong Gao, Yu Wang, and Miroslav Pajic. Learning-based vulnerability analysis of cyber-physical systems. In *2022 ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCPS)*, pages 259–269. IEEE, 2022.

[25] Amir Khazraei, Haocheng Meng, and Miroslav Pajic. Stealthy perception-based attacks on unmanned aerial vehicles. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3346–3352. IEEE, 2023.

[26] Amir Khazraei, Haocheng Meng, and Miroslav Pajic. Black-box stealthy gps attacks on unmanned aerial vehicles. *arXiv preprint arXiv:2409.11405*, 2024.

[27] Taegyu Kim, Chung Hwan Kim, Altay Ozen, Fan Fei, Zhan Tu, Xiangyu Zhang, Xinyan Deng, Dave Jing Tian, and Dongyan Xu. From control model to program: Investigating robotic aerial vehicle accidents with {MAYDAY}. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 913–930, 2020.

[28] Taegyu Kim, Chung Hwan Kim, Junghwan Rhee, Fan Fei, Zhan Tu, Gregory Walkup, Xiangyu Zhang, Xinyan Deng, and Dongyan Xu. {RVFuzzer}: Finding input validation bugs in robotic vehicles through {Control-Guided} testing. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 425–442, 2019.

[29] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[30] Asif Ali Laghari, Awais Khan Jumani, Rashid Ali Laghari, Hang Li, Shahid Karim, and Abudllah Ayub Khan. Unmanned aerial vehicles advances in object detection and communication security review. *Cognitive Robotics*, 2024.

[31] Hunje Lee, Sujeong Han, Jeong-Il Byeon, Seoulgyu Han, Rangun Myung, Jingon Joung, and Jihoon Choi. Cnn-based uav detection and classification using sensor fusion. *IEEE Access*, 11:68791–68808, 2023.

[32] Lorenz Meier, Dominik Honegger, and Marc Pollefeys. Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 6235–6240. IEEE, 2015.

[33] Yilin Mo and Bruno Sinopoli. Secure control against replay attacks. In *2009 47th annual Allerton conference on communication, control, and computing (Allerton)*, pages 911–918. IEEE, 2009.

[34] Juhwan Noh, Yujin Kwon, Yunmok Son, Hocheol Shin, Dohyun Kim, Jaeyeong Choi, and Yongdae Kim. Tractor beam: Safe-hijacking of consumer drones with adaptive gps spoofing. *ACM Transactions on Privacy and Security (TOPS)*, 22(2):1–26, 2019.

[35] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017.

[36] Raul Quinonez, Jairo Giraldo, Luis Salazar, Erick Bauman, Alvaro Cardenas, and Zhiqiang Lin. {SAVIOR}: Securing autonomous vehicles with robust physical invariants. In *29th USENIX security symposium (USENIX Security 20)*, pages 895–912, 2020.

[37] Sara Salim, Nour Moustafa, and Martin Reisslein. Cybersecurity of satellite communications systems: A comprehensive survey of the space, ground, and links segments. *IEEE Communications Surveys & Tutorials*, 2024.

[38] Burcu Sönmez Sarikaya and Şerif Bahtiyar. Generative adversarial networks for synthetic jamming attacks on uavs. In *2024 9th International Conference on Computer Science and Engineering (UBMK)*, pages 760–765. IEEE, 2024.

[39] SLEIGHT. Home. https://sites.google.com/view/sleight-/home.

[40] Caroline Strickland, Muhammad Zakar, Chandrika Saha, Sareh Soltani Nejad, Noshin Tasnim, Daniel J Lizotte, and Anwar Haque. Drl-gan: A hybrid approach for binary and multiclass network intrusion detection. *Sensors*, 24(9):2746, 2024.

[41] Fadhila Tlili, Samiha Ayed, and Lamia Chaari Fourati. Advancing uav security with artificial intelligence: A comprehensive survey of techniques and future directions. *Internet of Things*, page 101281, 2024.

[42] Mehmed Kerem Uludag, Maryna Veksler, Yasin Yilmaz, and Kemal Akkaya. Deceptive skies: Leveraging gans for drone sensor data falsification. In *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*, pages 1606–1613, 2024.

[43] Tarun Kumar Vashishth, Vikas Sharma, Kewal Krishan Sharma, Bhupendra Kumar, Sachin Chaudhary, and Shahanawaj Ahamad. Unmanned aircraft systems (uass) technology, applications, and challenges. *Unmanned Aircraft Systems*, pages 1–63, 2024.

[44] Xiaojie Wang, Zhonghui Zhao, Ling Yi, Zhaolong Ning, Lei Guo, F Richard Yu, and Song Guo. A survey on security of uav swarm networks: Attacks and countermeasures. *ACM Computing Surveys*, 57(3):1–37, 2024.

[45] Yuan Xu, Xingshuo Han, Gelei Deng, Jiwei Li, Yang Liu, and Tianwei Zhang. Sok: Rethinking sensor spoofing attacks against robotic vehicles from a systematic view. In *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, pages 1082–1100. IEEE, 2023.

[46] Nian Xue, Liang Niu, Xianbin Hong, Zhen Li, Larissa Hoffaeller, and Christina Pöpper. Deepsim: Gps spoofing detection on uavs using satellite imagery matching. In *Proceedings of the 36th Annual Computer Security Applications Conference*, pages 304–319, 2020.

Appendix

A SLEIGHT Hyperparameters Specifications

Table 7 lists the hyperparameters used in SLEIGHT's deployed runtime (`sleight_var.py`) which generated all SITL, HITL, and swarm results in this paper. Configuration is released in our repository [39]. **On λ_C and λ_F.** Both are zero because the cap and feasibility constraints are enforced as *hard* operations rather than soft penalties: $\alpha \in [0, \alpha_{\max}]$ is clamped at every step, and the runtime bisection in Alg. 1 enforces Π_Ω and the ℓ_∞ bounds before injection. This design choice means feasibility violations never reach the reward signal and soft penalties are unnecessary; we retain λ_C, λ_F in Eq. (19) for generality so practitioners can substitute soft penalties when hard projection is unavailable. **On the high**

Table 7: SLEIGHT hyperparameters.

Symbol	Value	Description
Representation & runtime		
W	50	Window length (time steps)
d	11	Feature dimensionality
f_{ctrl}	10 Hz	Control / injection rate
Stealth & feasibility enforcement		
B	8	Bisection iteration budget
$\alpha_{\max}$	0.25	Max scaling factor (per-step ℓ_∞ bound)
α_{init}	0.005	Initial scale
κ	0.01	Stealth margin ($\tau_{\text{eff}} = 0.99\,\tau_A$)
Policy training (REINFORCE)		
γ	0.99	Discount factor
λ_I	1.0	Impact reward weight
λ_R	1.0	Realism ($\log D$) shaping weight
λ_S	50.0	Stealth-violation penalty
λ_C	0	Cap penalty (clipped via $\alpha_{\max}$)
λ_F	0	Feasibility penalty (clipped via Π_Ω)

$\lambda_S = 50$. The stealth weight dominates the reward by design. Drift values r_{drift} typically lie in $[0,5]$ m per step, so $\lambda_S = 50$ ensures any stealth violation strictly outweighs the largest plausible single-step impact gain, biasing the policy toward conservative exploration of the feasible region. This is consistent with our empirical observation that SLEIGHT sustains >95% no-alarm rates across all evaluated IDSs.

B ConFIDe IDS Implementation

ConFIDe [10] is a control-fused IDS that monitors consistency across control commands, estimator outputs, and PWM actuation signals. Our implementation follows ConFIDe's autoencoder-based design: a fixed feature vector is extracted from benign flights, an autoencoder is trained to reconstruct nominal feature windows, and a threshold is set from benign reconstruction error statistics. During deployment, ConFIDe computes a reconstruction error score and raises an alarm when the score exceeds the deployed threshold.

Let $x_t \in \mathbb{R}^d$ denote ConFIDe's feature vector at time t after scaling, and let $A(\cdot)$ denote the trained autoencoder. The deployed score used for evaluation is

$$F_{\text{ConFIDe}}(x_t) = \varepsilon(x_t) = \|x_t - A(x_t)\|_2^2,$$

with a deployed threshold T_F selected from benign validation errors (e.g., a high quantile). Algorithm 2 summarizes training and inference.

Algorithm 2: ConFIDe IDS Implementation

Input: Benign flight logs, feature extractor $\mathcal{F}$, scaler $\phi(\cdot)$, quantile level q
Output: Trained autoencoder A, deployed threshold T_F
Data preparation: extract features $\mathcal{F}$ from logs, align streams, and form dataset D
Scale: $D_{\text{scaled}} \leftarrow \phi(D)$
Training: train autoencoder A by minimizing $\frac{1}{|D|}\sum_{x \in D_{\text{scaled}}} \|x - A(x)\|_2^2$
Thresholding: compute $\varepsilon(x) = \|x - A(x)\|_2^2$ on benign validation data and set $T_F \leftarrow \text{Quantile}_q(\{\varepsilon(x)\})$
Inference: for each incoming scaled x_t, compute $\varepsilon(x_t)$ and raise an alarm if $\varepsilon(x_t) > T_F$

C PID-Piper IDS Implementation

PID-Piper [17] detects actuator-level and control-integrity anomalies using LSTM predictors for benign roll, pitch, and yaw dynamics. Online prediction residuals are accumulated with a one-sided CUSUM detector, triggering recovery when the statistic exceeds a threshold. Let $x_t \in \mathbb{R}$ denote an attitude axis value (roll, pitch, or yaw) at time t, and let $\hat{x}_t$ be the one-step prediction from the corresponding LSTM. The per-step residual is $\delta_t = |x_t - \hat{x}_t|$ and the CUSUM update is

$$S(t+1) = \max\big(0,\ S(t) + \delta_t - b\big),$$

with drift parameter b and threshold τ. Algorithm 3 summarizes the pipeline.

Algorithm 3: PID-Piper IDS Implementation

Input: Benign flight logs, attitude sequences, window length w
Output: Trained LSTM models, CUSUM parameters (b, τ)
Training: for each axis (roll, pitch, yaw), train an LSTM to predict x_{t+1} from $\{x_{t-w}, \ldots, x_t\}$ by minimizing MSE
Select (b, τ) from benign residual statistics
Inference: initialize $S(0) = 0$
For each time step t: predict $\hat{x}_t$, compute $\delta_t = |x_t - \hat{x}_t|$, update $S(t)$, and trigger recovery if $S(t) > \tau$

Residual threshold-crossing evidence. To provide evidence that PID-Piper's residual signal separates benign from GPS-manipulated behavior in our setup, we compare residual norms from benign flights to residual norms from GPS-manipulated flights under the same mission. We use the strict maximum benign residual as an operational reference bound. Fig. 16 shows benign residuals remaining below this bound, while the GPS-manipulated flight crosses it repeatedly (36 crossings out of 846 samples, 4.26%), indicating repeated deviations that would contribute to CUSUM accumulation in the deployed pipeline.

D Evaluation Against ConFIDe

We evaluate SLEIGHT against ConFIDe as a deployed alarm-scoring detector, using $F_{\text{ConFIDe}}(\cdot)$ and threshold T_F only to compute no-alarm rate. Runtime injection follows Algorithm 1 with attacker-side $S_\theta(\cdot)$ and $\tau_A(\phi)$, without online access to $F_{\text{ConFIDe}}(\cdot)$ or T_F.

Metrics. We report (i) deployed stealth as the fraction of timesteps/windows with $F_{\text{ConFIDe}}(x_t) \le T_F$ and (ii) mission impact as trajectory deviation and landing displacement from closed-loop execution.

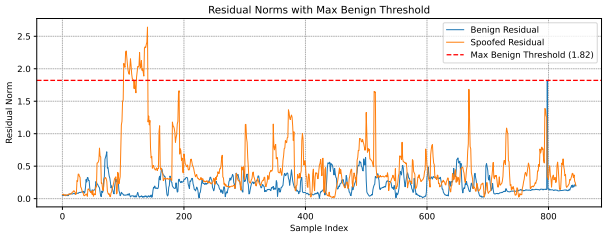


Figure 16: Residual norms over time against strict maximum benign residual bound. GPS-manipulated flight crosses bound repeatedly, while the benign flight remains below.

Algorithm 4: SLEIGHT: ConFIDe Evaluation

Input: x_t (scaled), window w_t, generator/policy g, phase estimate $\phi(t)$, plausibility S_θ, threshold $\tau_A(\cdot)$, mask M, bounds (ε, σ), projection Π_Ω, previous perturbation δ_{t-1}, max scale $\bar{\alpha}$

Output: x_t^{adv}

sample $z_t \sim \mathcal{N}(0, I)$; $\Delta x_t \leftarrow g(x_t, z_t, \phi(t))$; clamp Δx_t; $\alpha \leftarrow \bar{\alpha}$

while $\alpha > 0$ **do**

  $x^{\text{cand}} \leftarrow \Pi_\Omega\big(x_t + \alpha(M \odot \Delta x_t)\big)$; $\delta_t \leftarrow \alpha(M \odot \Delta x_t)$

  $w_t' \leftarrow \textsc{UpdateWindow}(w_t, x^{\text{cand}})$

  if $S_\theta(\text{vec}(w_t')) \le \tau_A(\phi(t))$ ***and*** $\|\delta_t\|_\infty \le \varepsilon$ ***and*** $\|\delta_t - \delta_{t-1}\|_\infty \le \sigma$ **then**

    break

  end

  $\alpha \leftarrow \alpha/2$

end

$x_t^{\text{adv}} \leftarrow (\alpha > 0)\ ?\ x^{\text{cand}} : x_t$

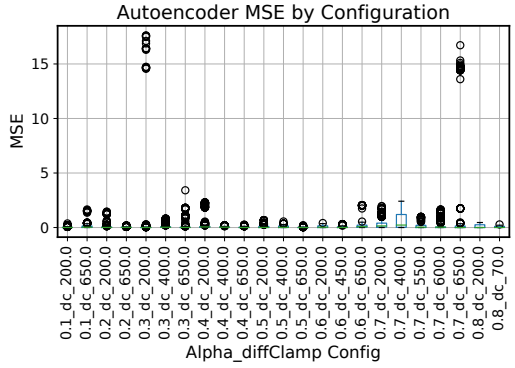


Figure 17: ConFIDe reconstruction-error distributions across $(\alpha, \texttt{diffClamp})$ configurations. Most settings maintain low error; aggressive settings produce high-error outliers.

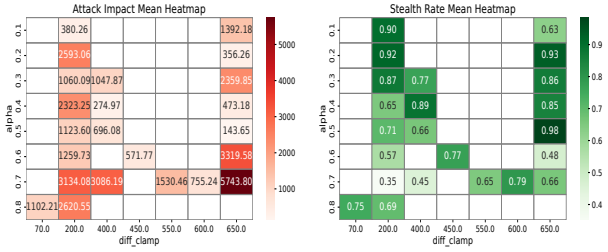


Figure 18: ConFIDe configuration sweeps: (a) mean attack impact across $(\alpha, \texttt{diffClamp})$; (b) mean stealth rate across $(\alpha, \texttt{diffClamp})$.

Configuration-wise ConFIDe reconstruction error. Fig. 17 reports ConFIDe reconstruction error distributions across $(\alpha, \texttt{diffClamp})$ configurations. Most settings remain in ConFIDe's operating region, while aggressive settings yield higher-error outliers.

Impact and stealth tradeoff across configurations. Fig. 18 summarizes mean impact and mean stealth across $(\alpha, \texttt{diffClamp})$. Conservative settings yield higher stealth

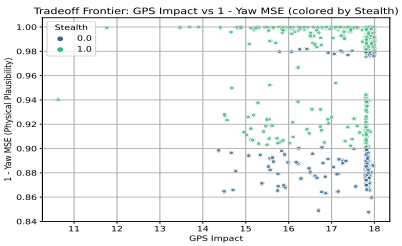


Figure 19: Tradeoff between GPS impact and a yaw-consistency proxy, colored by stealth outcome under PID-Piper.

with lower impact; more aggressive settings increase impact while lowering stealth.

E Evaluation Against PID-Piper

We evaluate SLEIGHT against PID-Piper by treating PID-Piper as a deployed detector used only for scoring alarms. PID-Piper's LSTM residual and CUSUM logic is evaluated on the resulting closed-loop telemetry to determine whether recovery would be triggered. SLEIGHT's runtime injection uses Algorithm 1 with attacker-side plausibility $S_\theta(\cdot)$ and $\tau_A(\phi)$, and does not depend on PID-Piper model parameters or thresholds online.

Algorithm 5: SLEIGHT: PID-Piper Evaluation

Input: x_t (scaled), window w_t, generator/policy g, phase estimate $\phi(t)$, plausibility S_θ, threshold $\tau_A(\cdot)$, mask M, bounds (ε, σ), projection Π_Ω, previous perturbation δ_{t-1}, max scale $\bar{\alpha}$

Output: x_t^{adv}

sample $z_t \sim \mathcal{N}(0, I)$; $\Delta x_t \leftarrow g(x_t, z_t, \phi(t))$; clamp Δx_t; $\alpha \leftarrow \bar{\alpha}$

while $\alpha > 0$ **do**

  $x^{\text{cand}} \leftarrow \Pi_\Omega\big(x_t + \alpha(M \odot \Delta x_t)\big)$; $\delta_t \leftarrow \alpha(M \odot \Delta x_t)$

  $w_t' \leftarrow \textsc{UpdateWindow}(w_t, x^{\text{cand}})$

  if $S_\theta(\text{vec}(w_t')) \le \tau_A(\phi(t))$ ***and*** $\|\delta_t\|_\infty \le \varepsilon$ ***and*** $\|\delta_t - \delta_{t-1}\|_\infty \le \sigma$ **then**

    break

  end

  $\alpha \leftarrow \alpha/2$

end

$x_t^{\text{adv}} \leftarrow (\alpha > 0)\ ?\ x^{\text{cand}} : x_t$

Metrics. We report stealth as the fraction of timesteps/windows in which PID-Piper's CUSUM statistic remains below threshold, and we report impact as trajectory deviation and landing displacement.

Tradeoff evidence. Fig. 19 visualizes a tradeoff between GPS impact and a yaw-consistency proxy, colored by stealth outcome under PID-Piper's detection logic. High-impact points cluster near the edge of plausible yaw behavior, and a subset maintain stealth under CUSUM-based detection.