2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

# ARMOR: Attack-Resilient Reinforcement Learning Control for UAVs

Pritam Dash†, Ethan Chan†, Nathan P. Lawrence*, Karthik Pattabiraman†
†University of British Columbia, Canada. *University of California, Berkeley, USA
†{pdash, echan, karthikp}@ece.ubc.ca. *nplawrence@berkeley.edu

***Abstract*— Unmanned Aerial Vehicles (UAVs) depend on onboard sensors for perception, navigation, and control. However, these sensors are susceptible to physical attacks, such as GPS spoofing, that can corrupt state estimates and lead to unsafe behavior. While reinforcement learning (RL) offers adaptive control capabilities, existing safe RL methods are ineffective against such attacks. We present ARMOR (Adaptive Robust Manipulation-Optimized State Representations), an attack-resilient, model-free RL controller that enables robust UAV operation under adversarial sensor manipulation. Instead of relying on raw sensor observations, ARMOR learns a robust latent representation of the UAV's physical state via a two-stage training framework. In the first stage, a teacher encoder, trained with privileged attack information, generates attack-aware latent states for RL policy training. In the second stage, a student encoder is trained via supervised learning to approximate the teacher's latent states using only historical sensor data, enabling real-world deployment without privileged information. Our experiments show that ARMOR outperforms conventional methods for ensuring UAV safety. Further, ARMOR improves generalization to unseen attacks and reduces training cost by eliminating the need for iterative adversarial training.**

## I. Introduction

Unmanned Aerial Vehicles (UAVs) are extensively used in various applications including logistics, agriculture, surveillance, and emergency services [1]. UAVs rely on onboard sensors for perception, autonomous navigation, and control. Correctness of sensor measurements is critical to achieving safe and reliable performance in UAV missions. However, sensors are susceptible to *physical attacks* launched by injecting malicious signals or noise through the physical channel. Examples of such attacks are GPS spoofing [2], gyroscope tampering using acoustic noise [3], and optical sensor spoofing through laser beams [4]. Physical attacks corrupt a UAV's physical state estimates, leading to unsafe control actions and resulting in deviations from planned trajectories, or crashes, as illustrated in Figure 1.

Model-free Reinforcement Learning (RL) has emerged as a promising approach for UAV control, enabling adaptive decision-making in complex and dynamic environments [5]. However, as RL-based controllers also rely on sensors, they are vulnerable to physical attacks. Techniques like shielding [6] and control barrier functions (CBF) [7] have been proposed for safe policy learning. However, they are not effective under physical attacks. Shielding and CBFs rely on prior definitions of unsafe actions and well-defined boundaries of the unsafe action space. Physical attacks present a fundamentally different threat model. They can cause the controller to execute unsafe actions under the illusion that they remain within safe limits. For example, GPS spoofing can cause incremental deviations during a UAV mission - these deviations might appear safe within the defined action space, but they can cumulatively cause the UAV to follow an unintended and potentially dangerous path [8]–[10].

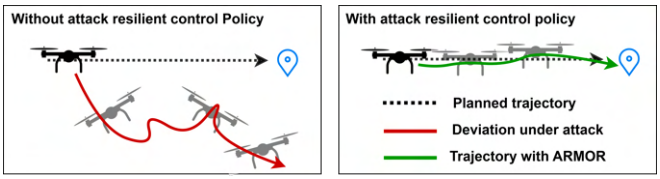


Fig. 1: **Left:** Without an attack-resilient control policy, a UAV subjected to physical attacks deviates significantly from its planned trajectory, leading to mission failure. **Right:** With an attack-resilient control policy like ARMOR, the UAV remains on course despite attacks and completes the mission.

Adversarial training is a popular approach for developing attack-resilient RL-based controllers [11], [12]. However, adversarial training has three limitations under physical attacks [10], [13]. (1) It incurs a high training cost due to the iterative generation of adversarial scenarios. (2) Lacks generalizability, as the policy is only effective against the specific attack patterns encountered in training. (3) It lacks zero-shot effectiveness against previously unseen attacks.

We propose ARMOR, an attack-resilient RL-based controller for UAVs. Instead of directly relying on high-dimensional physical state information from onboard sensors, ARMOR generates a robust latent representation of the UAV's physical state that is designed to withstand physical attacks. This latent state representation allows the UAV to operate safely and complete its missions, despite the malicious interventions. Our main innovation is a two-stage offline training framework. In the first stage, we train the RL policy using a *teacher encoder* that uses privileged information. In the second stage, we adapt the RL policy using a *student encoder*, which relies solely on the onboard sensors. During online deployment, only the student encoder is used.

The teacher encoder has access to privileged information, such as the target sensor under attack, the magnitude of the sensor bias, and the duration of sensor manipulation. By combining the UAV's high-dimensional physical state information with the privileged information, the teacher encoder generates a robust latent state representation using a Variational Autoencoder (VAE) [14]. This latent representation allows the RL policy to achieve high performance in

control tasks, and it also remains resilient to physical attacks.

Since privileged information is unavailable in real-world deployment scenarios, we introduce a student encoder that relies solely on the onboard sensors. The student encoder processes the UAV's historic physical state information derived from the onboard sensors using a Long Short Term memory (LSTM) network, capturing temporal dependencies to generate a robust latent state representation. The student encoder learns to approximate the latent state representation of the teacher encoder through supervised learning. The RL-policy is used with the student encoder for online inference.

By decoupling the learning process into teacher and student encoders and leveraging privileged information, ARMOR eliminates the need for iterative adversarial scenario generation, resulting in significant reductions in training costs. The robust latent state representation further enhances ARMOR's generalizability across a wide range of attack types. In addition, ARMOR demonstrates zero-shot generalization, where a policy trained on one attack type remains robust to previously unseen attack modalities without retraining.

While prior work has explored the use of privileged information [15] in robotics, it focuses on enhancing control under normal conditions, rather than under physical attacks. In contrast, our work designs robust latent state representations specifically optimized for control under adversarial perturbations. We have released ARMOR's code at `https://github.com/DependableSystemsLab/armor`.

***Contributions.*** Our contributions are as follows:

- We introduce a two-stage offline training framework for developing an attack-resilient model-free RL controller for UAVs. First, the controller is trained using privileged information to enable robust and efficient policy learning. Second, the policy is adapted for online deployment using only onboard sensor data.
- We propose a robust state representation method that transforms the UAV's high-dimensional physical state information into a resilient latent vector representation, ensuring robustness to physical attacks.
- We propose a transfer learning strategy that enables the RL controller to infer robust latent state representations without using privileged information, and instead relying solely on historical sensor information.

## II. Related Work

### A. Safe and Resilient Model-free Policy Learning

Prior work in safe RL focuses on uncertainty, typically by constraining actions to remain within a safe set [16]–[19]. A few prominent examples are shielding [6], control barrier functions (CBF) [7], and safety critics [20]. However, physical attacks manipulate the UAV's perception of its state, leading to unsafe actions that appear safe within the defined action space. The above safe RL mechanisms are designed to handle unsafe actions under normal operating conditions, and they are not designed to mitigate deliberate state manipulations caused by physical attacks.

Robust RL techniques are another category of work proposed to handle state manipulations. However, these methods attempt to handle disturbances by learning conservative policies [21] or training against adversarial perturbations [11], [12]. While they improve robustness, they rely on predefined uncertainty sets or known attack patterns, limiting generalization to unseen attacks. Moreover, adversarial training increases training cost, and broad uncertainty sets often produce overly conservative policies that degrade performance.

### B. UAV State Representation

Prior work has explored contrastive learning for robot state representation [22], which learns discriminative representations, typically supervised, but has seen limited use in safety-critical robotics. Lee et al. [15] introduced latent vectors with privileged information for quadrupeds, though under nominal sensor conditions rather than attacks. Our work builds on these foundations but focuses on resilient control under adversarial conditions, addressing a critical gap.

## III. Preliminaries

UAVs rely on sensors for perception. For instance, the GPS measures position $(x, y, z)$, the gyroscope measures angular orientation $(\phi, \theta, \psi)$, the accelerometer measures velocity $(\dot{x}, \dot{y}, \dot{z})$ and acceleration $(\ddot{x}, \ddot{y}, \ddot{z})$, the magnetometer measures heading direction, the barometer measures altitude $(z)$, and the optical flow sensor measures horizontal motion.

### A. Threat Model

Physical attacks introduce bias into sensors that propagates through the UAV's feedback control loop, corrupting the UAV's physical state estimates, and subsequently results in erroneous actuator signals [23], leading to unsafe consequences such as collisions or crashes [9].

We assume an adversary capable of launching GPS spoofing [8], gyroscope and accelerometer tampering via acoustic interference [24], magnetometer corruption with electromagnetic signals [25], and optical flow spoofing [4]. We assume the adversary can inject attacks of varying bias, intensity, and patterns. The adversary can also launch stealthy attacks to disrupt the UAV [26] gradually. These attacks have been shown to be practical in real-world settings [27]. We assume the adversary operates from fixed locations and launches attacks intermittently during the UAV mission. However, they cannot compromise actuators, or the onboard software. This threat model is in line with other work in the area [9], [10], [27].

We define zero-shot generalization as the ability of the RL control policy trained under one attack modality (e.g., GPS spoofing) to remain effective against previously unseen modalities (e.g., gyroscope or accelerometer tampering) without retraining or fine-tuning.

### B. UAV Control Design

The control architecture for a UAV consists of two primary components: motion generation and tracking control. UAVs operate in a continuous trajectory-based motion framework, where the desired trajectory is defined in the inertial frame. The UAV's trajectory is parameterized using a waypoint trajectory generator (WTG), which provides a time-dependent

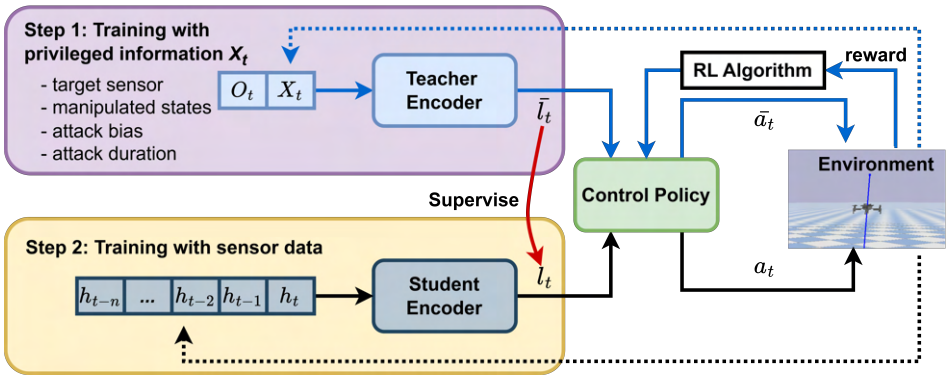


Fig. 2: **Overview of ARMOR's two-staged training approach**. First, a *teacher encoder* is trained with privileged information that includes attack information—target sensor, corrupted states, attack duration, etc. The *control policy* is trained jointly with the teacher encoder. Second, a *student encoder* is trained to approximate the teacher encoder via supervised learning. The student encoder does not have access to privileged information, instead, it relies on a stream of historic physical state information derived from onboard sensors. For online deployment, the control policy uses the student encoder.

reference position. The state of the UAV at each time step $t$ is given by: $p_t = p_0 + \int_0^t v(\tau)\, d\tau$, where $p_t$ is the position at time $t$, $p_0$ is the initial position, and $v(t)$ is the velocity.

Physical attacks induce biases $b_t$ in sensor measurements, causing the control policy $\pi(\cdot)$ to generate unsafe control actions $a_t$ that may cause the UAV's true state $o_t$ to deviate from the reference trajectory $g(t)$. This deviation is quantified as: $\Delta p_t = \|p_t - g(t)\| \gg \epsilon$, where $\epsilon$ defines a safety threshold, typically modeled as a circular region of radius $\epsilon$ centered around the target state $g(t)$. A trajectory is considered *safe* if the UAV remains within this bound for all times, i.e., $\Delta p_t \leq \epsilon\ \forall t$; otherwise, it is considered *unsafe*.

## IV. ARMOR: Two-Staged Training Framework

The objective of ARMOR is to control UAVs in both adversarial and non-adversarial scenarios. An overview of our two-staged training approach is shown in Figure 2.

First, we train a teacher encoder that has access to privileged information ($X_t$), such as target sensor, manipulated states, attack-induced offset in physical states, and the duration of the attack. The teacher encoder is based on variational autoencoder (VAE) [14], which receives both the UAV's states $O_t$ and $X_t$, and computes a latent embedding $\bar{l}_t$ that represents the UAV's current state. Next, we train a control policy using reinforcement learning, with the teacher encoder's latent embedding ($\bar{l}_t$) as input, enabling the control policy to quickly learn and adapt to attack-induced state manipulations, and output resilient actions.

Second, to enable real-world deployment where privileged information is unavailable, we train a *student encoder* that relies solely on the onboard sensors. The student encoder is implemented as a temporal variational autoencoder (TVAE), that receives a sequence of historic physical state information ($H$) derived from onboard sensors. It computes a latent embedding $l_t$ in a supervised manner as shown in Figure 2, that approximates the teacher encoder's latent representation $\bar{l}_t$, enabling the same RL policy to operate reliably in the absence of privileged information.

Our approach adopts a privileged learning strategy inspired by Lee et al. [15], but introduces two key innovations that improve both adversarial robustness and deployment efficiency. (1) We use the teacher encoder to generate a robust latent representation that is resilient to attack-induced perturbations, and the latent representation is the input to the control policy. This encourages the policy to rely entirely on a resilient representation, enhancing robustness to sensor manipulation. In contrast, Lee et al. combine raw observations with latent features, which can dilute the benefits of the robust representation. (2) Rather than training separate teacher and student policies [15], we reuse a single control policy across both training and deployment. This eliminates the need to learn a second policy from scratch, simplifying the training pipeline and improving sample efficiency.

### A. Stage I – Train with Privileged Information

We formulate the control problem as a Markov Decision Process (MDP) [28]. An MDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R})$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{T}$ is the transition probability $P(s_{t+1}|s_t, a_t)$, and $\mathcal{R}$ is a scalar reward function. The objective of the training framework is to learn a control policy $\pi(a_t|s_t)$ that maximizes the expected discounted sum of rewards over time.

In the teacher training stage of ARMOR, we assume a *fully observable simulation environment*. The teacher encoder has access to both the UAV's onboard sensor readings and privileged information that is not available during real-world deployment. The full state is defined as $s_t := \langle o_t, x_t \rangle$, where: $o_t$ includes the observable physical state of the UAV, such as position, angular orientation, heading direction, acceleration, linear and angular velocities. $x_t$ contains *privileged information*, including the sensor under attack, the corresponding corrupted physical states, the magnitude of injected bias, and the duration of the attack. This information is extracted from the simulator and is used only during training. The control action $a_t$ specifies low-level control targets for the UAV. Table I summarizes the inputs and outputs of ARMOR.

TABLE I: ARMOR's inputs in both the training stages. $O_t$: physical states, $X_t$: Priviledged information, $S_t$: Inputs to teacher encoder, $H$: Inputs to student encoder, and $a_t$: action.

| Input Type | Description |
|---|---|
| $O_t$ | position, velocity, orientation, body angular rates. $[x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}]$, |
| $X_t$ | [target sensor, corrupted physical states, bias intensity, duration] example: [GPS, $(x, y, z)$, (-5, 0, 0), 10] |
| $S_t$ | $S_t = \langle O_t, X_t \rangle$ |
| $H$ | $H = \{o_{t-n}, .., o_{t-3}, o_{t-2}, o_{t-1}\}$, |
| $a_t$ | Position and attitude control commands - $x, y, z$ axes. |

The teacher encoder is implemented as a multi-head variational autoencoder (VAE) that maps the input $s_t = \langle o_t, x_t \rangle$ to three outputs: $f_{\text{teacher}}(s_t) = \big(\bar{l}_t = (\mu_t, \sigma_t),\ \hat{y}_t,\ \hat{s}_t\big)$, where $\bar{l}_t$ is the latent representation with mean $\mu_t$ and variance $\sigma_t$, $\hat{y}_t$ is the predicted attack type, and $\hat{s}_t$ is the reconstruction of the input $s_t$. ARMOR leverages both the $\mu_t$ and $\sigma_t$ of the latent representation, allowing the policy to adjust control when uncertainty is increases ($\sigma_t \uparrow$), such as under stealthy attacks [26]. The attack-type classifier encourages the latent space to capture attack-specific patterns, while the variance $\sigma_t$ provides an uncertainty estimate that allows the control policy $\pi(a_t|\bar{l}_t)$ to adjust its behavior under unseen attacks.

The teacher encoder is trained with an auxiliary decoder to evaluate the quality of the latent representations through reconstruction loss; however, the decoder is discarded after training, and only the encoder is used with the RL policy. The teacher encoder is trained by minimizing a combined loss that includes the reconstruction loss, the Kullback-Leibler (KL) divergence, and the attack classification loss:

$$\mathcal{L}_{\text{teacher}} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{attack}} \tag{1}$$

The RL policy $\pi(a_t|\bar{l}_t)$ is trained using Proximal Policy Optimization (PPO) [29], with the latent representation $\bar{l}_t$ as input. This design encourages the policy to rely on a robust, attack-aware latent representation, improving its resilience.

This stage enables the policy to exploit privileged information during training, allowing it to generate *resilient behaviors* under adversarial conditions. The resulting latent space is a robust representation for policy learning and student encoder supervision in Stage II.

The *reward function* is designed to promote task completion while ensuring safety and stability. For instance, the RL agent receives positive reward for minimizing distance to the target waypoint and penalties for unsafe behaviors such as abrupt motion, excessive tilt, or deviation from trajectory. The reward function is defined as:

$$r_t = R_{\text{goal}} \cdot \exp\left(-\lambda \cdot \|p_t - g_t\|\right) - \alpha \cdot \|p_t - p_{t-1}\| - \beta \cdot \theta_t - \gamma \cdot \|a_t - a_{t-1}\|^2 \tag{2}$$

where $p_t$ is the UAV's position, $g_t$ is the goal waypoint, and $\|p_t - g_t\|$ is the Euclidean distance to the target. The exponential term provides a smooth approximation of the goal reward, sharply increasing as the UAV nears the goal. The remaining terms penalize trajectory deviation, tilt ($\theta_t$), and abrupt control actions ($a_t$), with corresponding weights $\alpha$, $\beta$, and $\gamma$. The term $\theta_t$ represents the total tilt of the UAV (e.g., combined roll and pitch deviation), and $a_t$ represents the control command at time $t$ (e.g., change of position). The coefficients $\alpha$, $\beta$, and $\gamma$ weight the penalties for deviation from the goal, instability, and abrupt motion, respectively.

### B. Stage II – Transfer Learning Adaptation

In this stage, we introduce a *student encoder* that operates solely on data derived from onboard sensors. The core idea is to approximate the teacher's latent representation $\bar{l}_t$ using only historical UAV physical state. This enables a transfer learning setup in which the RL policy, originally trained with privileged information, can now operate with latent representations generated by the student encoder.

We implement the student encoder as a temporal variational autoencoder (TVAE) built using a Long Short-Term Memory (LSTM) network, which effectively models sequential dependencies in time-series sensor data. The encoder takes as input a sequence of historical UAV physical states in a sliding window $H := \{o_{t-N}, \ldots, o_{t-1}\}$, where each $o_t$ denotes the UAV's physical state at time $t$ (e.g., position, velocity, angular rate, and orientation). This sequence $H$ provides temporal context that implicitly captures the impact of adversarial disturbances over time. The student encoder maps $H$ to a latent representation $l_t$ that approximates the teacher encoder's latent representation $\bar{l}_t$. $l_t$ is then passed to the trained control policy to derive the control actions.

The student encoder outputs: (i) a latent representation $l_t$ (mean and variance), (ii) an attack-type prediction, and (iii) a reconstruction of the input (during training). This multi-head design encourages the latent space to separate task-relevant features from attack-induced perturbations, improving generalization across different attack types.

The student encoder is trained via supervised learning, using the teacher encoder's outputs as targets. For each input history $H_t$, the student aims to approximate the teacher's latent representation $\bar{l}_t$ and ensure that the control policy produces consistent actions from both representations. Specifically, we minimize the combined loss:

$$\mathcal{L}_{\text{student}} := \|l_t(H_t) - \bar{l}_t(o_t, x_t)\|^2 + \|a_t(l_t) - a_t(\bar{l}_t)\|^2 + \mathcal{L}_{\text{attack}} \tag{3}$$

where the first term encourages the student encoder to match the teacher's latent representation, the second term aligns policy outputs, and $\mathcal{L}_{\text{attack}}$ penalizes errors in attack-type classification. Once training is complete, the RL control policy originally trained with $\bar{l}_t$ is reused and unchanged. At deployment, the RL control policy takes $l_t$ as input, enabling attack-resilient control using only UAV onboard sensor data.

## V. Evaluation and Results

In this section, we first outline the experimental setup, the simulation environment, and the metrics used for evaluation. Then, we present results evaluating the effectiveness of

ARMOR across three key aspects: (1) The performance of the student encoder in approximating the teacher's latent state representations without access to privileged information. (2) The ability to maintain safe and stable flight under physical attacks. (3) Generalization to unseen attack scenarios.

***Physical Attacks.*** We evaluate ARMOR in the presence of *five different types of physical attacks [27]* targeting the GPS, gyroscope, accelerometer, magnetometer, and optical flow sensors of the UAV. We simulate realistic physical attacks using RAVAGE [23], a software tool that supports launching realistic physical attack signals (attack bias, attack duration, bias pattern). Table II outlines the attack parameters.

As summarized in the table, the attacks differ in nature, ranging from drift to oscillatory and random bias patterns, with varying intensity and attack duration. This represents different manipulation strategies across sensor modalities, ensuring that evaluation covers a broad spectrum of adversarial conditions rather than variations of a single attack type. In addition, we evaluate under stealthy attacks, where small biases accumulate gradually over time to destabilize the UAV.

TABLE II: Attack types for evaluation. Attack parameters - intensities, bias patterns, and duration.

| Sensors | Bias Type | Bias Range | Attack Duration |
|---|---|---|---|
| GPS | drift | 1-20 $m$ | up to 60s |
| Gyroscope | oscillatory | 1-90 $deg$ | up to 60s |
| Accelerometer | oscillatory | 0.5-1 $m/s^2$ | up to 30s |
| Magnetometer | random | 10-90 $deg$ | up to 60s |
| Optical Flow | random | 0.1-0.5 $m/s$ | up to 30s |

***Simulation Environment.*** We consider a quadcopter operating in 3D space. The UAV dynamics are simulated using `gym-pybullet` [30], an OpenAI Gym-compatible [31] environment built on the PyBullet physics engine, which provides a realistic simulation of rigid-body dynamics. As in prior work [32], we use standard quadcopter equations of motion with translational and rotational dynamics. We collected privileged information for training using the RAVAGE [23] tool for injecting attacks into UAVs. Note that, due to space constraints, we focus on a single vehicle model and present comprehensive results, rather than exploring multiple testbeds and sim-to-real deployment.

***Control Objective***. The control objective is to reach a randomly sampled goal position $g$ in 3D space, represented as a sphere with a radius of 0.1 m. To guide the UAV towards the goal while ensuring stable and safe flight, we design a shaped reward function that provides incentives for reaching the goal and penalties for unsafe behaviors (Equation 2).

***Implementation.*** Both the teacher and student encoders use a latent vector size of 20. The student encoder takes a historic sequence of 200 timesteps as input. The RL control policy is trained using PPO [29] (Section IV-A).

***Comparison.*** We compare the effectiveness of ARMOR with two state-of-the-art approaches: (1) Robust Adversarial Reinforcement Learning (RARL) [11], which formulates adversarial training as a minimax game between a protagonist and an adversary, aiming to learn a policy that is robust to sensor perturbations. (2) Hybrid Recovery Policy (HRP) [33], which uses an RL policy for control in unsafe zones and a stabilizing PID controller for safety outside unsafe zones.

***Ablation Study.*** We implement a baseline-RL policy with the same architecture as ARMOR, except that it does not incorporate any encoder. This baseline directly processes high-dimensional physical state information as input to the PPO policy, without mapping it into a latent representation. The baseline serves as an ablation to evaluate the effectiveness of encoders in improving resilience against physical attacks.

***Evaluation Metrics.*** We use the following three metrics:

1) **Mission Success Rate** measures the proportion of episodes in which the UAV successfully reaches the goal position $g$ within an error margin $\epsilon$. A mission is considered successful if the final UAV position $p_T$ satisfies $\|p_T - g\| \leq \epsilon$, where $\epsilon = 5m$ [9], [10].
2) **Crash Rate** measures the proportion of episodes in which the mission failed due to a crash. A crash is defined as the UAV's state exceeding predefined safety bounds, resulting in termination of the episode.
3) **State Drift** measures the mean absolute deviation in the physical states from the ideal physical states during attacks. For example, in the case of GPS attacks, state drift is quantified as the Euclidean distance between the UAV's current position $p_t$ and the ideal position $\hat{p}_t$ at each time $t$ during the attack duration $T$.

### A. *ARMOR Training*

Figure 3 compares the training performance of ARMOR in two scenarios: (a) no-attack conditions (nominal conditions), where we compare ARMOR with baseline-RL, and (b) adversarial conditions (physical attacks), where we compare ARMOR with an adversarially trained control policy. We refer to the two variants of ARMOR- the Teacher Encoder policy and the Student Encoder policy as the RL control policies that use latent representations from the Teacher and Student Encoders, respectively. The figures represent the mean performance averaged over 5 random seeds.

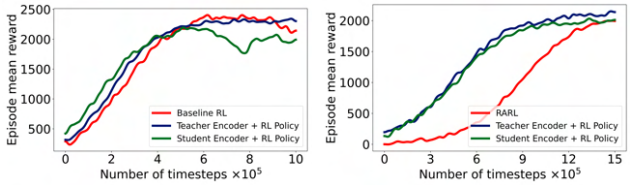


Fig. 3: Training performance comparison. **Left:** Nominal conditions, all methods achieve similar final performance. **Right:** Adversarial conditions, both the Teacher and Student encoder policies significantly accelerate learning compared to RARL. The Student Encoder policy enables faster convergence even without access to privileged information.

As shown in Figure 3(a), in the absence of attacks, all the methods: baseline-RL, the Teacher Encoder policy, and the Student Encoder policy, achieve comparable final episodic rewards, converging to approximately 2200 after 500k timesteps. The similarity in final performance indicates

that the use of encoder-based latent representations in the Teacher and Student encoders does not hinder the ability of the policy to learn optimal control under nominal conditions.

Figure 3(b) shows the results in adversarial conditions. We train a control policy using the RARL [11] approach for adversarial robustness. This policy converges slowly, requiring nearly 1200k timesteps, twice as long, to reach a reward of 2100. In contrast, both the Teacher Encoder policy and the Student Encoder policy demonstrate significantly faster convergence speed (2× faster). The Teacher Encoder policy, trained with privileged information, reaches a reward of around 2100 in under 600k timesteps. Notably, the Student Encoder policy, despite lacking access to privileged information, also achieves similar convergence. *These results demonstrate that the ARMOR's two-stage training is effective in transferring robustness from the attack-aware privileged learning phase to the online deployment phase.*

Henceforth, we refer to the Student Encoder policy used for online inference as ARMOR. Under attack-free conditions, ARMOR matches the control performance of a baseline PID controller reported in prior work using the `gym-pybullet` environment [30]. This shows that ARMOR's robustness to physical attacks does not compromise nominal performance.

### B. Effectiveness of ARMOR under Physical Attacks

Figure 4 shows the effectiveness of ARMOR under GPS spoofing attacks. The red lines represent the UAV's actual trajectory. With baseline-RL (top row), the UAV deviates significantly from the intended path (in dotted line) due to incorrect position estimates, resulting in a crash. In contrast, with ARMOR (bottom row), the UAV maintains stable flight with minimal deviation from the intended path, successfully completing the mission despite the attack.

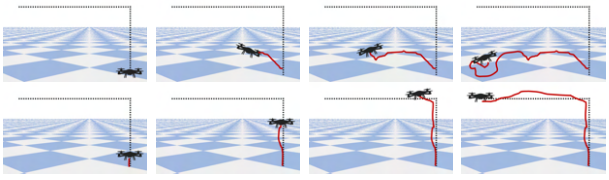

Fig. 4: Control performance under GPS spoofing attack. The **Top row** shows the trajectory deviations with baseline-RL. The **Bottom row** shows the trajectory with ARMOR, demonstrating resilient control despite the attack.

We evaluate ARMOR under five different types of physical attacks shown in Table III. ARMOR demonstrates strong resilience against all attack types, maintaining safe and stable flight. ARMOR achieves an average success rate of 88%, incurring 0 crashes. Even when the missions failed, ARMOR prevented crashes and maintained minimal state drift.

### C. Comparison with Baseline-RL, HRP and RARL

First, we discuss two cases in detail comparing ARMOR with baseline-RL and HRP under two different attack types with different bias patterns: (1) GPS spoofing, which introduces drift biases in position estimates, and (2) gyroscope attack, which induces oscillatory biases in attitude estimates. We then present a more comprehensive comparison.

Figure 5 shows the UAV's position error under a GPS spoofing attack. The baseline-RL completely fails under this attack, with a 0% mission success rate and a 100% crash rate. The trajectory deviation is severe ($> 0.9m$), resulting in loss of control. HRP partially mitigates position errors but struggles to maintain stability, resulting in significant state drift. In contrast, ARMOR maintains accurate position tracking across all axes ($x$, $y$, $z$), keeping the state drift to around 0.1m.

Figure 5 also presents the attitude error under a gyroscope attack. The baseline-RL exhibits large attitude errors exceeding $\pm 8$ degrees, resulting in a 0% mission success rate, a 100% crash rate, and a state drift of approximately 0.8 m. HRP reduces attitude error, but cannot fully suppress the effects of the attack, resulting in a crash rate of 60% and a state drift of 18.5 m. In contrast, ARMOR maintains stable attitude control throughout the attack, keeping errors bounded within $\pm 1$ degree, resulting in a state drift of less than 3 degrees. *These results demonstrate ARMOR's robustness in suppressing different types of physical attacks.*

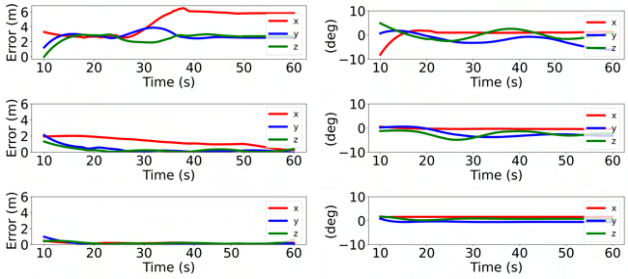


Fig. 5: Position and attitude errors under GPS (left) and gyroscope attacks (right). **Top:** Baseline-RL exhibits significant position and attitude error. **Middle:** HRP partially mitigates errors but struggles to maintain stable flight. **Bottom:** ARMOR maintains significantly lower position and attitude error.

Table III compares the performance of ARMOR with two prior techniques: HRP [33] and RARL [11], under physical attacks targeting five different sensors. Compared to HRP, ARMOR consistently achieves higher success rates and lower state drift in attacks against all sensor types, while also preventing crashes, significantly outperforming HRP across all the metrics. ARMOR's effectiveness is also higher than that of RARL. On average, ARMOR achieves a higher mission success rate of 88% compared to RARL's 83% and exhibits lower state drift across all attack types. *Thus, ARMOR performs better than HRP and RARL under physical attacks.*

### D. Effectiveness of ARMOR under Stealthy Attacks

We evaluate ARMOR under *stealthy attacks*, where small biases accumulate gradually over time. We launch two types of stealthy attacks [10], [26] targeting the GPS of the UAV: (1) *ramp profiles*, where the injected bias grows continuously at a fixed rate ($f(t) = f_0 + rt$), and (2) *stair profiles*, where the bias increases in discrete increments at fixed intervals

TABLE III: Performance comparison of HRP, RARL, and ARMOR under physical attacks against five UAV sensors

| Target sensor | HRP | | | RARL | | | ARMOR | | |
|---|---|---|---|---|---|---|---|---|---|
| | Success | Crash | State Drift | Success | Crash | State Drift | Success | Crash | State Drift |
| GPS | 40% | 50% | 6.2 ± 2.5 m | 82% | 0 | 0.1 ± 0.03 m | 87% | 0 | 0.1 ± 0.03 m |
| Gyroscope | 32% | 60% | 18.5 ± 3.1 deg | 78% | 0 | 4 ± 2 deg | 83% | 0 | 2.3 ± 1.6 deg |
| Accelerometer | 30% | 50% | 5.5 ± 1.7 m/$s^2$ | 80% | 0 | 0.02 ± 0 m/$s^2$ | 83% | 0 | 0.01 ± 0 m/$s^2$ |
| Magnetometer | 62% | 15% | 30 ± 4.1 deg | 92% | 0 | 8.1 ± 2.3 deg | 94% | 0 | 7.7 ± 2 deg |
| Optical Flow | 46% | 30% | 7.1 ± 3.6 m/s | 83% | 0 | 0.23 ± 0.05 m/s | 90% | 0 | 0.1 ± 0.05 m/s |

TABLE IV: Performance comparison under stealthy attacks.

| Method | Success (%) | Crash (%) | State Drift (m) |
|---|---|---|---|
| HRP | 58.5 | 22.0 | 9.1 ± 1.5 |
| RARL | 63.2 | 18.4 | 7.4 ± 1.2 |
| ARMOR | **80.7** | **0** | **3.2 ± 0.9** |

$(f(t_k) = f(t_{k-1}) + \delta)$. The ramp rates and bias increments chosen all fall within the bounds of Table II.

Our results in Table IV show that ARMOR achieves higher success, and lower state drift, compared to HRP and RARL, which indicates its resilience to stealthy attacks. It also incurs no crashes. This resilience arises because ARMOR 's attack-aware latent representation, combined with its history-based encoder, captures gradual bias trends, which prevent small errors from compounding into large deviations.

### *E. Zero-Shot Performance*

We evaluate ARMOR's effectiveness against attacks *not encountered during training*. Table V and Table VI compare the zero-shot performance of ARMOR and RARL, the current state-of-the-art approach for adversarially training robust policies. Specifically, we evaluate control policies trained exclusively on a single attack type (either GPS or gyroscope) and tested on *unseen attacks* targeting a different sensor.

TABLE V: Zero-shot performance of RARL and ARMOR when trained on GPS manipulations only and tested on unseen attacks (Gyroscope and Gyroscope+Accelerometer).

| Metrics | | Gyroscope | Gyro+Accelerometer |
|---|---|---|---|
| RARL | Success | 0% | 0% |
| | Crash | 60% | 75% |
| | State Drift | 15± 5.1 deg | 12.3±2.6 deg, 8±2.1 m/$s^2$ |
| ARMOR | Success | 60% | 50% |
| | Crash | 0% | 10% |
| | State Drift | 3.5 ± 1.8 deg | 2.8±1.6 deg, 1.1±0.4 m/$s^2$ |

TABLE VI: Zero-shot performance of RARL and ARMOR when trained on Gyroscope manipulations only and tested on unseen attacks (GPS and GPS+Accelerometer).

| Metrics | | GPS | GPS+Accelerometer |
|---|---|---|---|
| RARL | Success | 5% | 5% |
| | Crash | 70% | 80% |
| | State Drift | 6.5 ± 2.2 m | 11.5 ± 2.1 m, 10.2 ± 2.7 m/$s^2$ |
| ARMOR | Success | 70% | 55% |
| | Crash | 5% | 8% |
| | State Drift | 0.6 ± 0.2 m | 0.8 ± 0.3 m, 1.5 ± 0.7 m/$s^2$ |

There are two cases we considered.

**Case 1:** Policies trained on GPS manipulations, which introduce drift biases in position estimates, are evaluated on unseen gyroscope attacks that induce high-frequency oscillatory biases, and on multi-sensor (gyroscope+accelerometer) attacks. We find that RARL achieves 0% success and a 60% crash rate under gyroscope attacks. For multi-sensor attacks, RARL's performance further deteriorates, with a 75% crash rate, and significant state drift. In contrast, ARMOR achieves a 60% success rate under gyroscope attacks and 50% under multi-sensor attacks, while maintaining crash rates below 10% and reducing state drift by $4\times$ compared to RARL.

**Case 2:** Policies trained on gyroscope manipulations, which introduce high-frequency oscillatory biases, are evaluated on unseen GPS attacks that induce slow drift biases, as well as multi-sensor (GPS + accelerometer) attacks. We find that RARL generalizes poorly, achieving only 5% success and exhibiting high crash rates (70–80%) and significant state drift. In contrast, ARMOR shows strong zero-shot generalization, achieving a success rate of 70% under GPS attacks and 55% under multi-sensor attacks, while maintaining low crash rates, and reducing state drift by over $10\times$.

*These results highlight ARMOR's ability to generalize to unseen attacks targeting both single and multiple sensors.*

## VI. Discussions

ARMOR offers two main advantages over adversarial training: (1) training efficiency, and (2) zero-shot effectiveness.

Adversarial training methods [11], [12], [34] involve iterative policy updates by co-training an antagonist policy to generate adversarial perturbations. This results in high computational costs and long training times. In contrast, ARMOR 's two-stage training framework removes the need for an explicit antagonist. Instead, it leverages attack-aware latent state representations during training, and transfers the knowledge to a student encoder for deployment. As shown in Figure 3(b), ARMOR achieves comparable effectiveness to RARL while requiring significantly fewer training timesteps.

Furthermore, ARMOR demonstrates strong zero-shot generalization capabilities, enabling the control policy to handle unseen attack types, including both single-sensor and multi-sensor attacks. ARMOR significantly outperforms RARL in zero-shot evaluations, achieving higher success rates, significantly lower crash rates, and reduced state drift.

***Limitations.*** While ARMOR demonstrates promising zero-shot robustness to single unseen attack types, generalization to multi-sensor attacks is limited due to the compounding effects of the perturbations. This is an avenue for future work.

Our experiments focus on evaluation across diverse attack types targeting different sensors within a single UAV model and simulator; extending this to other robotic platforms and sim-to-real transfer is an important direction for future work.

## VII. Conclusions

We introduced ARMOR, a two-stage learning framework for attack-resilient UAV control. ARMOR leverages attack-aware privileged information during training to learn robust latent state representations, and uses transfer learning to adapt these representations for online deployment. This approach eliminates the need for iterative adversarial training, resulting in a more efficient and scalable framework. Our results demonstrate that ARMOR maintains safe and stable flight, outperforming existing techniques. Furthermore, ARMOR exhibits promising zero-shot generalization, enabling resilience against previously unseen attacks. Future work will explore extending ARMOR to a broader class of robotic systems. We will also integrate theoretical safety guarantees, and constraint satisfaction under adversarial conditions.

## Acknowledgements

This work was partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), the Department of National Defence (DND) Canada, and a Four Year Fellowship (4YF) from UBC.

## References

[1] C. Bai, P. Dallasega, G. Orzes, and J. Sarkis, "Industry 4.0 technologies assessment: A sustainability perspective," *International journal of production economics*, vol. 229, p. 107776, 2020.

[2] T. E. Humphreys, "Assessing the spoofing threat: Development of a portable GPS civilian spoofer," in *In Proceedings of the Institute of Navigation GNSS (ION GNSS*, 2008.

[3] Y. Son, H. Shin, D. Kim, Y. Park, J. Noh, K. Choi, J. Choi, and Y. Kim, "Rocking drones with intentional sound noise on gyroscopic sensors," in *24th USENIX Security Symposium (USENIX Security 15)*. Washington, D.C.: USENIX Association, 2015, pp. 881–896.

[4] D. Davidson, H. Wu, R. Jellinek, V. Singh, and T. Ristenpart, "Controlling UAVs with sensor input spoofing attacks," in *10th USENIX Workshop on Offensive Technologies (WOOT 16)*. Austin, TX: USENIX Association, 2016.

[5] J. Hwangbo, I. Sa, R. Siegwart, and M. Hutter, "Control of a quadrotor with reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2096–2103, 2017.

[6] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, "Safe reinforcement learning via shielding," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[7] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 3387–3395.

[8] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun, "On the requirements for successful GPS spoofing attacks," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, ser. CCS '11. ACM, 2011.

[9] P. Dash, G. Li, Z. Chen, M. Karimibiuki, and K. Pattabiraman, "PID-piper: Recovering robotic vehicles from physical attacks," in *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2021, pp. 26–38.

[10] P. Dash, E. Chan, and K. Pattabiraman, "SpecGuard: Specification aware recovery for robotic autonomous vehicles from physical attacks," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024.

[11] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta, "Robust adversarial reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2017, pp. 2817–2826.

[12] H. Zhang, H. Chen, C. Xiao, B. Li, M. Liu, D. Boning, and C.-J. Hsieh, "Robust deep reinforcement learning against adversarial perturbations on state observations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 024–21 037, 2020.

[13] I. Ilahi, M. Usama, J. Qadir, M. U. Janjua, A. Al-Fuqaha, D. T. Hoang, and D. Niyato, "Challenges and countermeasures for adversarial attacks on deep reinforcement learning," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 2, pp. 90–109, 2021.

[14] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," 2022. [Online]. Available: http://arxiv.org/abs/1312.6114

[15] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[16] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 411–444, 2022.

[17] T. Fan, P. Long, W. Liu, J. Pan, R. Yang, and D. Manocha, "Learning resilient behaviors for navigation under uncertainty," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[18] D. Sacerdoti, F. Benzi, and C. Secchi, "A reinforcement learning-based control strategy for robust interaction of robotic systems with uncertain environments," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 5788–5794.

[19] L. Brunke, Y. Zhang, R. Römer, J. Naimer, N. Staykov, S. Zhou, and A. P. Schoellig, "Semantically safe robot manipulation: From semantic scene understanding to motion safeguards," *IEEE Robotics and Automation Letters*, 2025.

[20] K. Srinivasan, B. Eysenbach, S. Ha, J. Tan, and C. Finn, "Learning to be safe: Deep RL with a safety critic," 2020. [Online]. Available: https://arxiv.org/abs/2010.14603

[21] M. Turchetta, A. Krause, and S. Trimpe, "Robust model-free reinforcement learning with multi-objective Bayesian optimization," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 10 702–10 708.

[22] W. Yan, A. Vangipuram, P. Abbeel, and L. Pinto, "Learning predictive representations for deformable objects using contrastive estimation," in *Conference on Robot Learning*. PMLR, 2021, pp. 564–574.

[23] P. Dash and K. Pattabiraman, "Ravage: Robotic autonomous vehicles' attack genaration engine," in *2025 55th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2025.

[24] Y. Tu, Z. Lin, I. Lee, and X. Hei, "Injected and delivered: Fabricating implicit control over actuation systems by spoofing inertial sensors," in *27th USENIX Security Symposium (USENIX Security 18)*, 2018.

[25] J.-H. Jang, M. Cho, J. Kim, D. Kim, and Y. Kim, "Paralyzing drones via emi signal injection on sensory communication channels." in *NDSS*, 2023.

[26] P. Dash, M. Karimibiuki, and K. Pattabiraman, "Out of control: Stealthy attacks against robotic vehicles protected by control-based techniques," in *Proceedings of the 35th Annual Computer Security Applications Conference*, ser. ACSAC '19. ACM, 2019.

[27] H. Kim, R. Bandyopadhyay, M. O. Ozmen, Z. B. Celik, A. Bianchi, Y. Kim, and D. Xu, "A systematic study of physical sensor attack hardness," in *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 2024, pp. 143–143.

[28] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, second edition ed., ser. Adaptive Computation and Machine Learning Series. Cambridge, Massachusetts: The MIT Press, 2018.

[29] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[30] J. Panerati, H. Zheng, S. Zhou, J. Xu, A. Prorok, and A. P. Schoellig, "Learning to fly—a gym environment with pybullet physics for reinforcement learning of multi-agent quadcopter control," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 7512–7519.

[31] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[32] Z. Yuan, A. W. Hall, S. Zhou, L. Brunke, M. Greeff, J. Panerati, and A. P. Schoellig, "Safe-control-gym: A unified benchmark suite for safe learning-based control and reinforcement learning in robotics," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, 2022.

[33] F. Fei, Z. Tu, D. Xu, and X. Deng, "Learn-to-recover: Retrofitting UAVs with reinforcement learning-assisted flight control under cyber-physical attacks," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 7358–7364.

[34] A. Pattanaik, Z. Tang, S. Liu, G. Bommannan, and G. Chowdhary, "Robust deep reinforcement learning with adversarial attacks," *arXiv preprint arXiv:1712.03632*, 2017.