REFORMA: Robust REinFORceMent Learning via Adaptive Adversary for Drones Flying under Disturbances

Hao-Lun Hsu, Haocheng Meng, Shaocheng Luo, Juncheng Dong, Vahid Tarokh and Miroslav Pajic

***Abstract*— In this work, we introduce REFORMA, a novel robust reinforcement learning (RL) approach to design controllers for unmanned aerial vehicles (UAVs) robust to unknown disturbances during flights. These disturbances, typically due to wind turbulence, electromagnetic interference, temperature extremes and many other external physical interference, are highly dynamic and difficult to model. REFORMA can perform a real-time online adaptation to these disturbances and generate appropriate velocity actions as countermeasures to stabilize the drone. REFORMA consists of two components: a base policy trained completely in simulation using model-free RL and an adaptation module trained via supervised learning with on-policy datasets. By varying the disturbance strength in an adaptation module, i.e., adopting adaptive adversary, the policy is then able to handle extreme cases when the velocity of the drone is immediately affected by disturbances. Finally, we demonstrate the effectiveness of our method through extensive simulated experiments. To the best of our knowledge, REFORMA is the first robust RL approach that uses adaptive adversaries to tackle uncertain disturbances in drone tasks.**

I. Introduction

Aerial drones have seen numerous promising applications ranging from aerial surveillance [1] to package delivery [2] and search-and-rescue missions [3]. However, during these tasks, drones often encounter unpredictable factors such as turbulent winds, sudden gusts, or electromagnetic interference that might manifest as sensor noise. These disturbances can jeopardize flight stability and accurate navigation. For instance, while hovering, drones must constantly adjust their position to counteract external forces, maintaining precise altitude and location. Similarly, during traversing tasks, drones need to respond swiftly to changing conditions, adapting their flight path to avoid obstacles and maintain safety. The ability to effectively address these challenges is crucial for enabling the reliable and safe operation of drones in diverse and dynamic environments.

Deep reinforcement learning (deep RL) has demonstrated promising performance on drone tasks, such as hovering [4], [5], landing [6], [7], goal-reaching [8], and collision avoidance [9], [10]. Among many deep RL approaches, robust reinforcement learning is proposed particularly to enhance the performance and reliability of autonomous agents operating in dynamic and uncertain environments.

*This work is sponsored in part by the ONR N00014-23-1-2206 and AFOSR FA9550-19-1-0169 awards, NSF CNS-1652544, and the National AI Institute for Edge Computing Leveraging Next Generation Wireless Networks, Grant CNS-2112562.

Hao-Lun Hsu is with the Department of Computer Science; the other authors are with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708 USA. {hao-lun.hsu, haocheng.meng, shaocheng.luo, juncheng.dong, vahid.tarokh, miroslav.pajic}@duke.edu

Enhancing the robustness of RL through adversarial learning has been proposed in [5], [11], [12], [13]. Two typical robust RL approaches formulated as Robust Markov Decision Process (R-MDP) are Robust Adversarial Reinforcement Learning (RARL) [11] and Noisy Robust Markov Decision Processes (NR-MDP) [13]. The former requires access to sensing data and control of the simulator in training, while the latter does not.

In this work, we propose a robust RL approach via adaptive adversary (i.e., REFORMA) that allows the resulting drone controller to generalize well toward environmental uncertainty and adversarial actions. The framework of REFORMA is illustrated in Fig. 1. We focus on action attacks, as opposed to policy, transition, reward, or state attacks, in order to simplify the adversarial training process while still providing valuable robustness improvements for RL protagonist agents. Moreover, action attacks fit realistic scenarios since real applications involve coping with external disturbances that affect the agents' actions. For instance, drones may need to adapt to sudden gusts of wind or sensor noise, which can be modeled as action attacks.

Inspired by the ROLAH framework recently proposed in [14], we train REFORMA with a group of adversaries to circumvent the occurrence of local optima and excessive pessimism in generated policies, addressing challenges observed in RARL [11] and NR-MDP [13]. The protagonist agent is assumed to share the environment with the adversaries. The adversaries take actions to disturb the environment and the protagonist directly, so that the cumulative reward received by the agent is minimized. This framework optimizes the average worst-k performance of a group of adversaries under disturbance with max-min optimization formulation such as the setup in ROLAH [14]. However, instead of following RARL [11] framework as in the original ROLAH setting, we consider the action attack adopted in NR-MDP [13]. By doing this, we can eliminate the reliance on prior knowledge, such as the specific attackable elements in the protagonist's action space that are required by RARL. When mixing the attacked actions from both protagonist and adversary agents, we use an α value to indicate the adversary strength.

To generalize REFORMA to varying α, we adopt the domain randomization method [15], [16] and wrap our framework with two training phases [17], [18], [19], accompanied by learning both protagonist and adversary policies as well as adaptive adversary strength level. Specifically, we learn a latent representation z_t of a drone or environment's parameters e_t as well as the adversary strength level from a history of states, actions, and attacked actions.

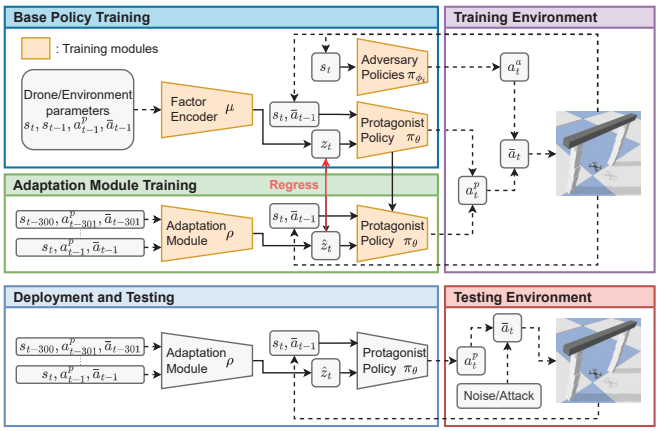


Fig. 1: REFORMA consists of 3 modules: the protagonist policy π_θ, the adaptation module ρ, and the adversary policies π_{ϕ_i}, where i is the index for each adversary policy in the herd. **Training stage:** in the base policy training phase, we train π_θ, π_{ϕ_i}, and the factor encoder μ together end-to-end to interact with the training environment via attacked action $\bar{a}_t$. In the *adaptation module training*, we learn an adaptation module ρ that takes the state history and actions history (both a^p_{t-j} and $\bar{a}_{t-j}$) to predict $\hat{z}_t$, minimizing the mean square error $MSE(z_t, \hat{z}_t)$, where z_t is the latent space capturing the previous inputs. **Testing stage:** The protagonist policy π_θ can be deployed with the inputs of the current state s_t, previous attacked action $\bar{a}_{t-1}$, and the intrinsic vector $\hat{z}_t$ predicted by the adaptation module ρ; its action a^p_t will receive unknown noise or attack before executing on the testing environment.

The REFORMA framework is employed to design stable flight controllers for drones, in representative tasks including hovering and traversing, which is validated in simulated environments. Our method is shown to outperform the state-of-the-art RL-based control policies in simulated environments. Specifically, our technical contributions are as follows:

1) We reformulate ROLAH within the NR-MDP to eliminate the reliance on prior knowledge, in order to improve the generality of attackable actions while preserving its capacity for improving optimization during training.
2) We extend the attackable actions in REFORMA to adapt to a range of adversary strengths using rapid motor adaption method [17], and hence improve the system robustness to disturbances of different strengths.
3) We demonstrate that REFORMA is more robust for typical drone tasks under different disturbances through extensive simulated experiments.

To the best of our knowledge, REFORMA is the first robust RL that uses adaptive adversary to address uncertain disturbances in drone tasks. The paper is organized as follows. Sec. II reviews the relevant work. Sec. III describes the research question and introduces the structure of our proposed REFORMA. The simulation experiments and analyses of the results are elaborated in Sec. IV. This work is concluded in Sec. V.

II. Related Work

A. Robust Reinforcement Learning

RL has demonstrated its impressive performance in different applications, including healthcare [20], [21], [22], [23], [24], robotics control [25], [26], [27], and natural language processing [28]. However, to deploy RL policy in the real world, researchers should be aware of the issues of interpretability [29], safety [30], [31], and robustness [32]. Robust RL is initially formulated with Robust Markov Decision Process (R-MDP) [33], [34] for solving small tabular MDPs via dynamic programming with a known uncertainty set. Subsequent works extend the formulation of uncertainty, introducing the perturbations and disturbances, to define environmental dynamics [11], [13], [14], [35].

Our work for robust RL interprets the perturbations as an adversary and aims to learn the distribution of the perturbation. This idea can be derived from two prior works [11], [13], which both solve the problem as the two-player max-min game. The main difference occurs in the parameterization of the adversaries. RARL [11] is flexible in defining different action spaces between protagonist and adversary agents but requires additional access and control to the simulator. On the other hand, frameworks of PR-MDP and NR-MDP utilize extra hyperparameter α to model how the stochastic perturbation can be executed on policy and action space respectively. Recent work ROLAH [14] extends the RARL framework to interact a protagonist with a group of adversaries, optimizing problems with the average performance over the worst-k adversaries in order to alleviate the issues of local-optima and over-conservation.

B. Robust Control for Drones

1) Traditional Robust Control for Drones: Traditional flight control strategies for drones rely on sophisticated mathematical modeling of physical dynamics and expertise-based manual tuning. Due to the complexity of aerodynamics and the uncertainty of environmental factors, robust control algorithms for drones have been proposed (e.g., [36], [37], [38]). A recent robust drone control method is the slide-mode control, achieving robustness in compensation for the parametric uncertainty in drones by means of the combination of sliding mode controller and sliding mode observer (SMC-SMO) [39]. However, it still has the inherent limitation of designing complicated control model of high non-linearity and varying aerodynamics which requires substantial domain knowledge to reach comparably competitive performance.

2) Reinforcement Learning based Robust Control for Drones: Unlike traditional robust control strategies for drones, RL aims to automate the control process by training agents to overcome the high non-linearity and complicated coupling effects. To further cope with unexpected disturbances from environmental factors, robust RL mechanisms are introduced to increase the resiliency of drones in the face of those potential threats. They have achieved wide-range success in different mission scenarios from multiple domains when the control performance of drones is severely tested by harsh environmental factors.

A robust RL policy specifically tailored to autonomous vertical take-off and landing (VTOL) missions on ships has been introduced to mitigate the wind effects in the dynamic landing process onto a moving target, which outperforms benchmark nonlinear PID-based control methods [6]. Collision avoidance is also of vital importance especially in unknown areas and a robust reinforcement learning policy can make full use of various sensor measurements to detect and avoid obstacles and help drones complete the mission safely [10]. Guiding and planning trajectories for drones in adversarial settings can also be assisted by robust reinforcement learning that conducts real-time attack detection using deep neural networks [17].

III. Robust Drone Controller via Adversary Learning

A. Problem Formulation

A Markov Decision Process (MDP) with adversaries in the environment can be defined by a tuple $(\mathcal{S}, \mathcal{A}_p, \mathcal{A}_a, \mathcal{P}, r, \gamma, p_0)$, where $\mathcal{S}$ is the set of states in the environment, $\mathcal{A}_p$ and $\mathcal{A}_a$ are the sets of actions that the protagonist and adversaries can take respectively, $\mathcal{P} : \mathcal{S} \times \mathcal{A}_p \times \mathcal{A}_a \to \Delta(\mathcal{S})$ is the transition function that describes the distribution of the next state given the current state and actions taken by the protagonist agent and the adversaries, $r : \mathcal{S} \times \mathcal{A}_p \times \mathcal{A}_a \to \mathbb{R}$ is the reward function for the protagonist agent, $\gamma \in [0, 1]$ is the discounting factor, and p_0 is the distribution of the initial state. Since we consider a zero-sum game framework in this work, the reward function of adversaries can be viewed as $-r$.

B. Robust RL via Adversarial Herding in NR-MDP

We adopt the extension of the two-player zero-sum game with an adversarial herd in ROLAH [14]. In addition to learning the policy of the protagonist agent $\pi_\theta : \mathcal{S} \to \Delta(\mathcal{A}_p)$ with the parameters θ, we also learn the policies of a group of adversaries $\pi_{\phi_i} : \mathcal{S} \to \Delta(\mathcal{A}_a)$, where π_{ϕ_i} and ϕ_i denote the policy of the i-th adversary and its parameter, respectively. Let $s_t \in \mathcal{S}$ be the state of the environment at time t, and $a_t^p \in \mathcal{A}_p$ / $a_t^a \in \mathcal{A}_a$ the actions of the protagonist agent/ adversary at time t. ROLAH follows the definition from RARL [11] so that the action spaces between protagonist and adversaries can be different, resulting in the cumulative discounted reward that the protagonist agent π_θ can receive under the disturbance of the adversary π_{ϕ_i} defined as

$$R(\theta, \phi_i) \doteq \mathop{\mathbb{E}}_{s_0 \sim p_0} \big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t^p, a_t^a) | C) \big], \tag{1}$$

where $C = \{a_t^p \sim \pi_\theta, a_t^a \sim \pi_{\phi_i}\}$.

However, to avoid the burden on deciding the selective part that the adversary acts such as RARL to pick out specific robot joints, we formulate our method using Noisy Action Robust MDPs (NR-MDPs) [13] so that the adversaries can directly attack the protagonist agent's action resulting in

$$R(\theta, \phi_i) \doteq \mathop{\mathbb{E}}_{s_0 \sim p_0} \big[\sum_{t=0}^{\infty} \gamma^t r(s_t, (1-\alpha) a_t^p + \alpha a_t^a) | C \big]; \tag{2}$$

here, α is a hyperparameter controlling the adversary strength and $C = \{a_t^p \sim \pi_\theta, a_t^a \sim \pi_{\phi_i}\}$, as before.

To resolve the potential over-pessimism with the optimization on the worst-case and local optimality problem, we consider the objective of optimizing the average performance over the worst-k adversaries [14], which is defined as

$$\max_{\theta \in \Theta} \min_{\phi_1, \ldots, \phi_m \in \Phi} \frac{1}{|I_{\theta, \widehat{\Phi}, k}|} \sum_{i \in I_{\theta, \widehat{\Phi}, k}} R(\theta, \phi_i). \tag{3}$$

Here, Θ and Φ are pre-defined parameter spaces for the agent and the adversaries. The worst-k adversaries are defined as the ones where the expected cumulative rewards received by the protagonist agent π_θ under their attack are smaller than that under the attacks from the rest $m - k$ adversaries. The order of the worst level is changeable. Specifically, in each iteration, the reward $R(\pi_\theta, \pi_{\phi_i})$, where π_{ϕ_i} refers to the index of the adversary, is estimated with the corresponding rollout data. Then the adversary for attacking in this iteration will be only selected from the current worst-k adversaries.

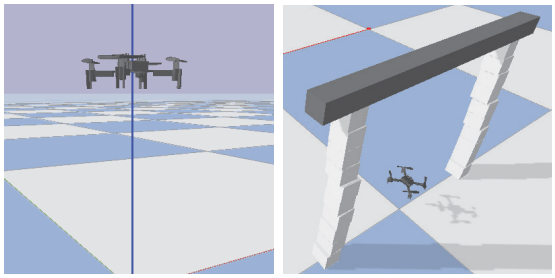
C. Adaptive Adversary via Rapid Motor Adaptation

The empirical performance of solving (3) should heavily rely on the value of hyperparameter α selected in (2). Although careful hyperparameter tuning leads to a robust policy for unlearnable noise or disturbance, it is not realistic that the adversary policies always follow the same adversary strength α in both training and testing. To generalize our approach to varying adversary strengths α, we leverage existing domain randomization techniques [15], [16] to randomize α as well as drone and environment parameters e_t for each episode.

Prior work has shown how the framework of rapid motor adaptation (RMA) can work successfully among online terrain adaptation for legged robots [17], diverse sets of internal parameters for drones, and physical properties for hand manipulation [18]. In addition to learning the internal dynamics of the quadcopter's body from a history of states and actions, we also explicitly estimate the external adversary strength, which enables adaption to the robustness of learnable and active attack.

Fig. 1 illustrates our whole framework REFORMA, containing training and testing stages. Our training stage can be divided into two steps: (i) *Protagonist policy training*, and (ii) *Adaptation module training*; we describe these training modules in what follows.

1) Adversary and Protagonist Policies: In the base policy training, the protagonist policy π_θ is trained with the inputs of the current state $s_t \in \mathbb{R}^{16}$, previous attacked action $\bar{a}_{t-1}$, and the latent representation $z_t \in \mathbb{R}^8$ to predict the next action $a_t^p \in \mathbb{R}^4$. The details of the state and action space will be introduced in Sec. IV-A. We learn a group of adversary policies that takes as input the current state $s_t \in \mathbb{R}^{16}$ to output $a_t^a \in \mathbb{R}^4$ so that the action $a_t^p = \pi_\theta(s_t, \bar{a}_{t-1}, z_t)$ is attacked and leads to the deployed action $\bar{a}_t = (1-\alpha) a_t^p + \alpha a_t^a$ at time-step t. Note that it is not necessary to let adversary policies learn the varying α, so we simply train the group of adversaries with only current state as input: $a_t^a = \pi_{\phi_i}(s_t)$.



(a) Hover (b) Fly-through gate

Fig. 2: Illustrations of drone tasks evaluated in our approach, including hovering and flying through a gate. Our method is robust enough to varying types of disturbance.

We use a factor encoder μ to compress all drone and environment parameters e_t as well as $\{s_t, s_{t-1}, a^p_{t-1}, \bar{a}_{t-1}\}$ to a low-dimensional vector z_t. This results in

$$z_t = \mu(e_t, s_t, s_{t-1}, a^p_{t-1}, \bar{a}_{t-1}), \tag{4}$$

$$a^p_t = \pi_\theta(s_t, \bar{a}_{t-1}, z_t). \tag{5}$$

2) Adaptation Module: Since we cannot directly observe the vector e_t and compute the latent representation z_t during evaluation, we use sensor history and previous actions to estimate $\hat{z}_t$ via an adaptation module ρ, which is inspired by [17], [18], [19]. In our experiments, we use $k = 300$ as the history length with

$$\hat{z}_t = \rho(s_{t-k:t}, a^p_{t-k-1:t-1}, \bar{a}_{t-k-1:t-1}). \tag{6}$$

We train ρ via supervised learning to minimize the mean square error between z_t and $\hat{z}_t$. Although the α value is not accessible during evaluation, and we do not provide factor encoder μ and adaptation module ρ with α explicitly, REFORMA is able to learn the adversary strength level with the history of both protagonist's actions a^p_t and attacked actions $\bar{a}_t$.

3) Deployment: Finally, we deploy the protagonist policy π_θ where the inputs are the current state s_t, previous attacked (deployed) action $\bar{a}_{t-1}$, and $\hat{z}_t$ predicted by the adaptation module ρ. The output action a^p_t from the protagonist will be disturbed by noise or attack before being executed in the testing environment.

IV. Experiments

We validate that the proposed approach can be successfully used to derive control policies robust to the disturbance on RL agents' actions on two drone tasks. Particularly, we investigate the following questions:

1) Can RL via adversarial herd be compatible with NR-MDP and perform better under different types of attacks/disturbances?
2) Can the adaption module in REFORMA serve as an indicator to identify the adversary strength and be more generalizable to different α values?

TABLE I: Ranges of the drone, environmental and adversary strength level parameters. Note that only REFORMA randomizes α value while other compared approaches fixed $\alpha = 0.01$.

Parameter	Training Range
Mass (kg)	$[0.020, 0.060]$
Arm length (m)	$[0.025, 0.075]$
Mass moment of inertia: $x, y (kg \cdot m^2)$	$[1.40e(-5), 3.22e(-3)]$
Mass moment of inertia: $z (kg \cdot m^2)$	$[2.17e(-5), 9.77e(-4)]$
Max. speed (km/hr)	$[10, 30]$
Drag coefficient	$[0, 0.64]$
Adversary strength α	$[0, 0.012]$

A. Simulation Environment and Benchmark Problems

We use the gym-pybullet-drones for training and testing our control policies [4]. The state space consists of kinematic information, including the drone's positions, quaternions, rolls, pitches, yaws, and linear and angular velocities with $s_t \in \mathbb{R}^{16}$. The action space contains the desired velocity input of the drone, $\{v_x, v_y, v_z, v_M\}$, where v_x, v_y, v_z are the components of a unit vector for 3 axes and v_M is its corresponding velocity's magnitude, leading to $a_t \in \mathbb{R}^4$. The high-level command in the action space is converted to 4 drone's motor speeds (in RPMs).

In the adversarial learning setting, the adversary can also attack the same action space, following the NR-MDP framework from Sec. III-A. We select a fixed $\alpha = 0.01$ for both RARL and ROLAH after hyperparameter tuning from 4 values $0.005, 0.01, 0.05, 0.1$, considering if the adversarial learning is large enough to be effective and small enough to have a higher performance in convergence. On the other hand, we randomize $\alpha \in [0, 0.012]$ for REFORMA for training. For validation, we consider the following two tasks (see Fig. 2): (*i*) hovering, and (*ii*) flying through a gate.

1) Hover: The agent aims to reach a predetermined altitude and stabilize. The reward function calculates the distance between the current position of the drone and the targeted position.

2) Fly-through gate: The objective of the agent is to fly through a gate with a rectangular boundary. Besides considering the distance between the current position and the goal, a penalty is executed when the drone flies out of the gate.

B. Comparison against Baselines

We empirically evaluate REFORMA with the following baselines: *(i) baseline* (vanilla PPO) [40]; *(ii) RARL*, which learns the protagonist policy PPO with a single adversarial agent [11]; and *(iii) ROLAH*, which trains the protagonist policy PPO with a group of adversaries with optimization over worst-k adversaries [14]. Note that all the methods are trained with domain randomization using the parameters (*e.g.,* mass, arm length, mass moment of inertia, maximum speed, and drag coefficient) in the range summarized in Table I to make the training more realistic and to have a fair comparison. However, the α of baseline (vanilla PPO) can be viewed as 0 without adversarial learning, and both

TABLE II: Performance of REFORMA and baselines under various disturbances for Hover task.

Method	Baseline (0 adv)	RARL (1 adv)	ROLAH (herding adv)	REFORMA (ours)
No disturbance	**0.90 ±0.13**	0.85±0.18	0.86±0.19	0.88±0.21
noise$X$	0.74	0.75	0.79	0.77
noise$Y$	0.75	0.75	0.80	0.82
noise$Z$	0.70	0.76	0.80	0.78
average noise	0.73±0.22	0.75±0.25	**0.80±0.28**	0.79±0.26
random$X$	0.77	0.78	0.80	0.77
random$Y$	0.75	0.77	0.75	0.81
random$Z$	0.26	0.43	0.59	0.65
average random	0.59±0.32	0.66±0.27	0.72±0.23	**0.74±0.24**
learnt $\alpha = 0.005$	0.20	0.36	0.47	0.52
learnt $\alpha = 0.01$	0.19	0.30	0.38	0.51
learnt $\alpha = 0.015$	0.17	0.24	0.34	0.45
average learnt	0.19±0.11	0.30±0.23	0.40±0.28	**0.49±0.13**
overall average	0.54	0.60	0.66	**0.70**

TABLE III: Performance of REFORMA and baselines under various disturbances for Fly-through gate task.

Method	Baseline (0 adv)	RARL (1 adv)	ROLAH (herding adv)	REFORMA (ours)
No disturbance	0.87±0.17	0.88±0.23	**0.90±0.19**	0.88±0.19
noise$X$	0.79	0.81	0.89	0.86
noise$Y$	0.78	0.88	0.84	0.84
noise$Z$	0.66	0.50	0.78	0.75
average noise	0.74±0.23	0.73±0.27	**0.84±0.29**	0.82±0.26
random$X$	0.25	0.71	0.88	0.83
random$Y$	0.71	0.82	0.87	0.82
random$Z$	0.27	0.39	0.36	0.45
average random	0.39±0.15	0.64±0.27	**0.70±0.25**	**0.70±0.22**
learned $\alpha = 0.005$	0.23	0.27	0.41	0.50
learned $\alpha = 0.01$	0.18	0.20	0.34	0.41
learned $\alpha = 0.015$	0.11	0.15	0.12	0.25
average learned	0.17±0.14	0.21±0.15	0.29±0.26	**0.39±0.17**
overall average	0.49	0.56	0.64	**0.66**

RARL and ROLAH only train on $\alpha = 0.01$ NR-MDP. The remaining REFORMA is randomized with $\alpha \in [0, 0.012]$. We keep the same architecture and hyperparameters for all the approaches, investigating the comparison of robust effectiveness among all methods. For each task and method, we run 10 random seeds for evaluation.

In typical cases, we need to address both unstable orientation and translation when flying drones under unknown disturbances. However, to reasonably narrow down the research scope, we only consider unstable translation in this work because we assume the drone compass and/or inertial measurement unit (IMU) are sufficient in estimating the orientation accurately. GPS sensors typically used in measuring position are subject to large translational errors of up to 2 m and a low frequency (1 Hz) [41], so they are more vulnerable to the disturbances that we focus on in this work.

Table II and III show the evaluation results of both tasks among different methods with 4 types of disturbances (*i*) no disturbance, (*ii*) Gaussian noise is added to the entry of the protagonist agent's original action vector aligning with the corresponding axis, (*iii*) the entry of the protagonist agent's original action vector aligning with the corresponding axis is replaced with a random action entry, and (*iv*) the learnt adversary that represents the worst-case performance of a given policy. For instance, noise$X$ indicates the original action with Gaussian noise added to the desired velocity along x-axis and random$Y$ replaces the original entry of the y-axis with a random action. To provide the worst adversary in (*iv*), we further train 3 corresponding adversaries under 3 different α values $[0.005, 0.01, 0.015]$ to minimize the protagonist's reward while **holding its parameters constant after convergence** for all methods. To have a better interpretation of our results, we normalize the episode reward in evaluation within the range of $[0, 1]$.

In general, we demonstrate that ROLAH is still more robust to noise, random, and learnt disturbances under NR-MDP setting compared with vanilla PPO and RARL. Specifically, we mainly attach the standard deviation of different categories of disturbances in Table II and III, indicating that robust RL via adversarial training (RARL, ROLAH and REFORMA) mostly shares a similar standard deviation as vanilla PPO. In other words, this branch of adversarial learning keeps its robustness against varying initial conditions while improving the average performance. Some standard deviation of vanilla PPO is relatively small because of its overall deficient performance.

We emphasize the variance reduction between ROLAH and REFORMA evaluated under the learned adversary policies. We notice that ROLAH has a larger standard deviation in some tasks (*e.g.,* half-cheetah and hopper) in [14], which has been improved in with the REFORMA framework due to the identification of the adversary strength. In addition, we show that learning with adversaries in fly-through gate task outperforms the baseline (0 adv) in Table III even though the training and testing conditions are consistent, which was also observed in [11] and [14].

TABLE IV: Ablation Studies of Number of k in Hover task. Since the standard deviations are not different significantly, they are not depicted here.

Number of Adversaries N with k percentage worst case	No disturbance	Noise$X$	Learned $\alpha = 0.01$
N=5, k=10%	0.71	0.59	0.23
N=5, k=30%	0.73	0.64	0.21
N=5, k=50%	0.72	0.57	0.18
N=10, k=10%	0.74	0.61	0.27
N=10, k=30%	0.86	0.79	0.38
N=10, k=50%	0.81	0.74	0.33
N=20, k=10%	0.75	0.54	0.24
N=20, k=30%	0.80	0.69	0.35
N=20, k=50%	0.79	0.68	0.31

C. Ablation Studies

1) Number of adversary policies and worst k: Note that the number of adversary policies and the hyperparameter of the worst k may influence the performance, depending on the simulation environment and the tasks [14]. Therefore, we consider the combinations of 3 different number of adversaries and 3 different k values in the hover task for hyperparameter tuning via ROLAH because ROLAH can be viewed as one component of REFORMA.

Table IV presents the comparison among all combination settings. Overall, as the value of k increases, the training focuses less on worst-case optimization. When the value of k decreases, the performance also degrades. This aligns with the conjecture that a single adversary can get trapped in extreme cases, also leading to degraded performance in [14].

After hyperparameter tuning, we keep $m = 10$ adversary policies and $k = 3$ for ROLAH and REFORMA in both *hover* and *fly-through gate* tasks.

2) Adversary adaptation capability: To investigate how the adaptive module can handle adversary strength, we do an ablation study on different α values. We mainly compare REFORMA with REFORMA-n. Both REFORMA and REFORMA-n receive domain randomization with all the parameters, including α, in Table I. However, the inputs of the adaptation module in REFORMA-n do not include the history of the attacked actions $\bar{a}_t$, which disables the adaptation module from identifying the severity of the adversary.

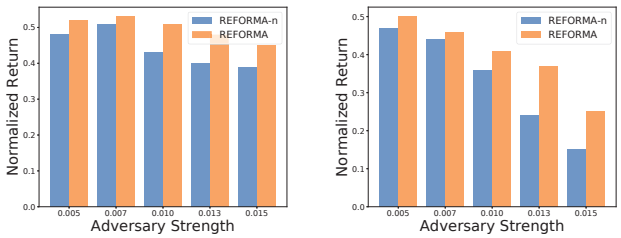
We observe that purely domain randomization on α has already resulted in higher performance for REFORMA-n compared with ROLAH. Further, we emphasize that following our REFORMA framework with the attacked actions $\bar{a}_t$ as parts of inputs to the adaptation module can increase the normalized return as shown in Fig. 3. Even though encountering the α values out of the training range (*e.g.,* 0.013 and 0.015), REFORMA still performs better than REFORMA-n in both drone tasks.

D. Analysis

1) Varying performance in different axis: In fly-through gate task, it is relatively difficult to perform control robust to the disturbances along with x-axis and z-axis because the direction of the gate is y-axis. The perturbation of the velocity in either x-axis or z-axis will result in a penalty of failure when the drone collides with the gate boundary or flies out of the gate.

2) Generalization over different adversary strength: We observe that REFORMA outperforms ROLAH as captured in Tables II and III mainly under the attack from the learned adversary policies. With the latent representation $\hat{z}_t$, we are able to provide additional features to the protagonist about the estimated adversary strength level under the history states, its actions, and the attacked actions, including the current state s_t. Specifically, we evaluate all methods interacting with learnt adversary policies under $\alpha = 0.005$, $\alpha = 0.01$, and $\alpha = 0.015$, representing the α value smaller than, equal to, and larger than fixed training α respectively. Since REFORMA randomizes $\alpha \in [0, 0.012]$ values and learns the latent space compacted with adversary strength information, it receives a higher normalized reward under $\alpha = 0.005$ and $\alpha = 0.01$. In addition, the highest performance under $\alpha = 0.015$ among all methods indicates the generalizability to the unseen scenario slightly out of the range.

3) Adversary Adaptation analysis: We analyze the latent presentation $\hat{z}_t$ for adaptation on incremental adversary strength α. We incrementally increase α value in fly-through gate task every 200 time steps. We plot all the components $\hat{z}_t \in \mathbb{R}^8$ from the adaptation module during the evaluation in Fig. 4. It can be observed that whenever the α is added, each component of the latent space changes in their own trends with the whole process starting from detecting the disturbance change, estimating the latent vector to adapting to the disturbance, and solving the task.



(a) Hover (b) Fly-through gate

Fig. 3: Testing results with ablation study in different α values for all drone control tasks. REFORMA is our proposed method and REFORMA-n is the approach with domain randomization for all the parameters in Table I, including α, but without learning the adaptive module.

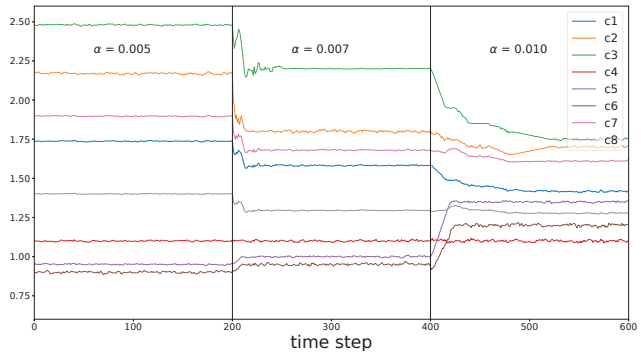


Fig. 4: Visualization of eight components c_i within the latent representation $\hat{z}_t$ predicted by the adaptation module in the *fly-through gate* task. The changes for each component are strongly correlated with the adversary strength α, indicating that the attack severity has been detected by the adaptation module.

V. Conclusion and Future Work

In this paper, we introduced a novel robust RL approach named REFORMA to confront complex disturbances occurring during drone flights. REFORMA first adopts the approach of involving a group of adversaries in training and enhancing the robustness of RL agents. This approach was used to tackle unknown disturbances in drone tasks. To adapt the system with dynamic and unknown disturbances and further improve the RL robustness, we incorporated the idea of Noisy NR-MDP and exploited α values for adaptive adversary learning. In our experiments, we showed that REFORMA improves robustness of typical drone tasks including hovering and traversing through a gate. Moreover, REFORMA was shown more robust to learnable adversaries than the state-of-the-art methods, such as RARL [11] and ROLAH [14]. As part of our future efforts, we will improve REFORMA to adapt to more challenging drone tasks and extend our work to handle orientation disturbances. Applying REFORMA to multi-drone scenarios is also an avenue for future work. As the interactions between drones may lead to much more complex disturbances, adaptive adversary-based approach can be more effective than other RL solutions.

REFERENCES

[1] Z. Zaheer, A. Usmani, E. Khan, and M. A. Qadeer, "Aerial surveillance system using uav," in *2016 thirteenth international conference on wireless and optical communications networks (WOCN)*, pp. 1–7, IEEE, 2016.

[2] G. Brunner, B. Szebedy, S. Tanner, and R. Wattenhofer, "The urban last mile problem: Autonomous drone delivery to your balcony," in *2019 international conference on unmanned aircraft systems (icuas)*, pp. 1005–1012, IEEE, 2019.

[3] A. Quan, C. Herrmann, and H. Soliman, "Project vulture: A prototype for using drones in search and rescue operations," in *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 619–624, IEEE, 2019.

[4] 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7512–7519, IEEE, 2021.

[5] A. M Deshpande, A. A Minai, and M. Kumar, "Robust deep reinforcement learning for quadcopter control," *IFAC-PapersOnLine*, vol. 54, no. 20, pp. 90–95, 2021.

[6] S. Vishnu, B. Lee, D. Kalathil, and M. Benedict, "Robust reinforcement learning algorithm for vision-based ship landing of uavs," *https://arxiv.org/abs/2209.08381*, 2022.

[7] L. Bartolomei, Y. Kompis, L. Teixeria, and M. Chli, "Autonomous emergency landing for multicopters using deep reinforcement learning," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS*, pp. 3392–3399, IEEE, 2022.

[8] T. Hickling, N. Aouf, and P. Spencer, "Robust adversarial attacks detection based on explainable deep reinforcement learning for uav guidance and planning," in *IEEE Transactions on Intelligent Vehicles*, IEEE, 2023.

[9] C. Millán-Arias, R. Contreras, F. Cruz, and B. Fernandes, "Reinforcement learning for uav control with policy and reward shaping," in *2022 41st International Conference of the Chilean Computer Science Society (SCCC)*, pp. 1–8, IEEE, 2022.

[10] S. Ouahouah, M. Bagaa, J. Prados-Garzon, and T. Taleb, "Deep-reinforcement-learning-based collision avoidance in uav environment," *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4015–4030, 2021.

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

[12] P. Kamalaruban, Y.-T. Huang, Y.-P. Hsieh, P. Rolland, C. Chi, and V. Cevher, "Robust reinforcement learning via adversarial training with langevin dynamics," *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4015–4030, 2021.

[13] C. Tessler, Y. Efroni, and S. Mannor, "Action robust reinforcement learning and applications in continuous control," in *International Conference on Machine Learning*, pp. 6215–6224, PMLR, 2019.

[14] J. Dong, H.-L. Hsu, Q. Gao, V. Tarokh, and M. Pajic, "Robust reinforcement learning through efficient adversarial herding," *https://arxiv.org/abs/2306.07408*, 2023.

[15] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8, May 2018.

[16] J. Siekmann, K. Green, J. Warila, A. Fern, and J. Hurst, "Blind bipedal stair traversal via sim-to-real reinforcement learning," *https://arxiv.org/pdf/2105.08328.pdf*, 2021.

[17] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," in *Robotics: Science and Systems (RSS) 2021*, 2021.

[18] H. Qi, A. Kumar, R. Calandra, Y. Ma, and J. Malik, "In-hand object rotation via rapid motor adaptation," in *Conference on Robot Learning*, pp. 1722–1732, PMLR, 2023.

[19] D. Zhang, A. Loquercio, X. Wu, A. Kumar, J. Malik, and M. W. Mueller, "Learning a single near-hover position controller for vastly different quadcopters," *arXiv preprint arXiv:2209.09232*, 2023.

[20] P. Sarikhani, H.-L. Hsu, O. Kara, J. K. Kim, H. Esmaeilzadeh, and B. Mahmoudi, "Neuroweaver: a platform for designing intelligent closed-loop neuromodulation systems," *Brain Stimulation*, vol. 14, no. 6, p. 1661, 2021.

[21] Q. Gao, S. Schmidt, K. Kamaravelu, D. A. Turner, W. M. Grill, and M. Pajic, "Offline policy evaluation for learning-based deep brain stimulation controllers," in *13th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, pp. 80–91, 2022.

[22] Q. Gao, S. L. Schmidt, A. Chowdhury, G. Feng, J. J. Peters, K. Genty, W. M. Grill, D. A. Turner, and M. Pajic, "Offline learning of closed-loop deep brain stimulation controllers for parkinson disease treatment," in *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023)*, pp. 44–55, 2023.

[23] M. Elfar, Y.-C. Chang, H. H.-Y. Ku, T.-C. Liang, K. Chakrabarty, and M. Pajic, "Deep reinforcement learning-based approach for efficient and reliable droplet routing on meda biochips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 4, pp. 1212–1222, 2023.

[24] P. Sarikhani, H.-L. Hsu, M. Zeydabadinezhad, Y. Yao, M. V. Kothare, and B. Mahmoudi, "Sparc: adaptive closed-loop control of vagal nerve stimulation for regulating cardiovascular function using deep reinforcement learning: a computational study," *2021 Neuroscience Meeting (SFN)*, 2021.

[25] N. Sontakke, H. Chae, S. Lee, T. Huang, D. W. Hong, and S. Ha, "Residual physics learning and system identification for sim-to-real transfer of policies on buoyancy assisted legged robots," *arXiv preprint arXiv:2303.09597*, 2023.

[26] K. N. Kumar, I. Essa, and S. Ha, "Cascaded compositional residual learning for complex interactive behaviors," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 4601–4608, 2023.

[27] H.-L. Hsu, J. Dong, Q. Gao, A. K. Bozkurt, V. Tarokh, and M. Pajic, "D2t2: Decision transformer with temporal difference via steering guidance," 2024.

[28] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, and D. Amodei, "Fine-tuning language models from human preferences," *arXiv preprint arXiv:1909.08593*, 2019.

[29] E. M. Kenny, M. Tucker, and J. Shah, "Towards interpretable deep reinforcement learning with human-friendly prototypes," *International Conference on Learning Representations*, 2023.

[30] T.-Y. Yang, J. Rosca, K. Narasimhan, and P. J. Ramadge, "Accelerating safe reinforcement learning with constraint-mismatched policies," *International Conference on Machine Learning (ICML)*, 2021.

[31] H.-L. Hsu, Q. Huang, and S. Ha, "Improving safety in deep reinforcement learning using unsupervised action planning," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 5567–5573, May 2022.

[32] J. Moos, K. Hansel, H. Abdulsamad, S. Stark, D. Clever, and J. Peters, "Robust reinforcement learning: A review of foundations and recent advances," *Machine Learning and Knowledge Extraction*, vol. 4, no. 1, pp. 276–315, 2022.

[33] J. A. Bagnell, A. Y. Ng, and J. G. Schneider, "Solving uncertain markov decision processes," *Citeseer*, 2001.

[34] G. N. Iyengar, "Robust dynamic programming," *Mathematics of Operations Research*, vol. 30, no. 2, pp. 257–280, 2005.

[35] B. Mehta, M. Diaz, F. Golemo, C. J. Pal, and L. Paull, "Active domain randomization," *Conference on Robot Learning*, pp. 1162–1176, 2020.

[36] A. Tagay, A. Omar, and M. H. Ali, "Development of control algorithm for a quadcopter," *Procedia Computer Science*, vol. 179, pp. 242–251, 2021.

[37] D. Bianchi, S. D. Gennaro, M. D. Ferdinando, and C. A. Lùa, "Robust control of uav with disturbances and uncertainty estimation," *Machines*, vol. 11, no. 3, 2023.

[38] V. N. Sankaranarayanan, S. Satpute, and G. Nikolakopoulos, "Adaptive robust control for quadrotors with unknown time-varying delays and uncertainties in dynamics," *Drones*, vol. 6, no. 9, 2022.

[39] A. Steinbusch and M. Reyhanoglu, "Robust nonlinear output feedback control of a 6-dof quadrotor uav," in *2019 12th Asian Control Conference (ASCC)*, pp. 1655–1660, 2019.

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

[41] S. Luo, J. Kim, R. Parasuraman, J. H. Bae, E. T. Matson, and B.-C. Min, "Multi-robot rendezvous based on bearing-aided hierarchical tracking of network topology," *Ad Hoc Networks*, vol. 86, pp. 131–143, 2019.