

ROBUST GYMNASIUM: A UNIFIED MODULAR BENCHMARK FOR ROBUST REINFORCEMENT LEARNING

Anonymous authors

Paper under double-blind review

ABSTRACT

Driven by inherent uncertainty and the sim-to-real gap, robust reinforcement learning (RL) seeks to improve resilience against the complexity and variability in agent-environment sequential interactions. Despite the existence of a large number of RL benchmarks, there is a lack of standardized benchmarks for robust RL. Current robust RL policies often focus on a specific type of uncertainty and are evaluated in distinct, one-off environments. In this work, we introduce **Robust-Gymnasium**, a unified modular benchmark designed for robust RL that supports a wide variety of disruptions across all key RL components—agents’ observed state and reward, agents’ actions, and the environment. Offering over sixty diverse task environments spanning control and robotics, safe RL, and multi-agent RL, it provides an open-source and user-friendly tool for the community to assess current methods and foster the development of robust RL algorithms. In addition, we benchmark existing standard and robust RL algorithms within this framework, uncovering significant deficiencies in each and offering new insights. The code is available at this website.

1 INTRODUCTION

Reinforcement learning (RL) is a popular learning framework for sequential decision-making based on trial-and-error interactions with an unknown environment, achieving success in a variety of applications, such as games (Mnih et al., 2015; Vinyals et al., 2019), energy systems (Chen et al., 2022), finance and trading (Park & Van Roy, 2015; Davenport & Romberg, 2016), and large language model alignment (OpenAI, 2023; Ziegler et al., 2019).

Despite recent advances in standard RL, its practical application remains limited due to concerns over robustness and safety. Specifically, policies learned in idealized training environments often fail catastrophically in real-world scenarios due to various factors such as the sim-to-real gap (Pinto et al., 2017), uncertainty (Bertsimas et al., 2019), noise, and even malicious attacks (Zhang et al., 2020; Klopp et al., 2017; Mahmood et al., 2018). Robustness is key to deploying RL in real-world applications, especially in high-stakes or high-cost fields such as autonomous driving (Ding et al., 2023b), clinical trials (Liu et al., 2015), robotics (Li et al., 2021), and semiconductor manufacturing (Kozak et al., 2023). Towards this, Robust RL seeks to ensure resilience in the face of the complexity and variability of both the physical world (Bertsimas et al., 2019) and human behavior (Tversky & Kahneman, 1974; Arthur, 1991).

Robust RL policies currently fall short of the requirement for broad deployment. Disruptions or interventions can occur at various stages of the agent-environment interaction, affecting the agent’s observed state (Zhang et al., 2020; 2021b; Han et al., 2022; Sun et al., 2021; Xiong et al., 2022), observed reward (Xu & Mannor, 2006), action (Huang et al., 2017), and the environment (transition kernel) (Iyengar, 2005; Pinto et al., 2017) and existing robust RL policies are vulnerable to such real-world failures (Mandlekar et al., 2017). This vulnerability is, in part, a result of the fact that policies are designed to address only one specific type of disruption (e.g., over the observed state), among other technical limitations (Ding et al., 2024). More critically, robust RL policies are often evaluated in distinct, one-off environments that can be narrow or over-fitted to the proposed algorithms. The absence of standardized benchmarks is a key bottleneck to progress in robust RL. Ideally, a benchmark should offer a wide range of diverse tasks for comprehensive evaluation and account for uncertainty and disruptions over multiple stages throughout the interaction process.

054
055
056
057
058
059
060
061
062
063
064
065
066
067
068
069
070
071
072
073
074
075
076
077
078
079
080
081
082
083
084
085
086
087
088
089
090
091
092
093
094
095
096
097
098
099
100
101
102
103
104
105
106
107

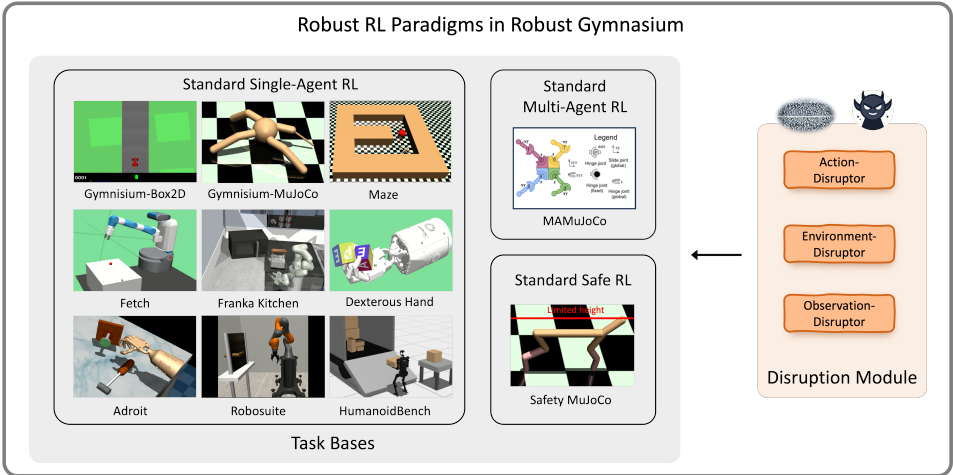


Figure 1: The overview of Robust-Gymnasium. For more details, please visit the website.

While numerous RL benchmarks exist, including a recent one focused on robustness to environment shifts (Zouitine et al., 2024), none are specifically designed for comprehensively evaluating robust RL algorithms. To address this gap, we present Robust-Gymnasium¹, a unified, highly modular benchmark for robust RL. This open-source tool enables flexible construction of diverse tasks, facilitating the evaluation and development robust RL algorithms. Our main contributions are:

- We introduce a unified framework for robust RL, encompassing diverse disruption types within a modular agent-environment interaction process (detailed in Sec. 2). This framework enables the development of Robust-Gymnasium, a benchmark that comprises over sixty diverse tasks in robotics and control, safe RL, and multi-agent RL; and includes a wide range of disruptions targeting different stages/sources (agent observations, actions, and the environment) with varying modes (e.g., random or adversarial disturbances, environmental shifts) and frequencies. **This is a unified benchmark specifically designed for robust RL, providing a foundational tool for evaluating and developing robust algorithms.**
- We conduct a comprehensive evaluation of several state-of-the-art (SOTA) baselines from standard RL, robust RL, safe RL, and multi-agent RL using representative tasks in Robust-Gymnasium. Our findings reveal that current algorithms often fall short of expectations in challenging tasks, even under single-stage disruptions, highlighting the need for new robust RL approaches. Furthermore, our experiments demonstrate the flexibility of Robust-Gymnasium by encompassing tasks with disruptions across all stages and four disturbance modes, including an adversarial model using a large language model (LLM). This illustrates the potential of LLMs in robust RL research.

2 A UNIFIED ROBUST REINFORCEMENT LEARNING FRAMEWORK

We begin by presenting a robust RL framework that unifies various robust RL tasks explored in the literature, including combinations of these paradigms. We outline the framework in the context of single-agent RL and then extend it to encompass broader classes of RL tasks, such as safe RL and multi-agent RL.

Background: Markov decision process (MDP). A single-agent RL problem is formulated as a finite-horizon Markov decision process (MDP), represented by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, T, P^0, r^0)$, where \mathcal{S} and \mathcal{A} denote the (possibly infinite) state and action spaces, and T is the horizon length. The nominal transition kernel $P^0 = \{P_t^0\}_{1 \leq t \leq T}$ defines the environmental dynamics: $P_t^0(s' | s, a)$ gives the probability of transitioning from state s to state s' given action a at time step t . The reward function $r^0 = \{r_t^0\}_{1 \leq t \leq T}$ represents the immediate reward at time step t , given the current state s and action a .

¹Website with the introduction, code, and examples: <https://robust-rl.online/>

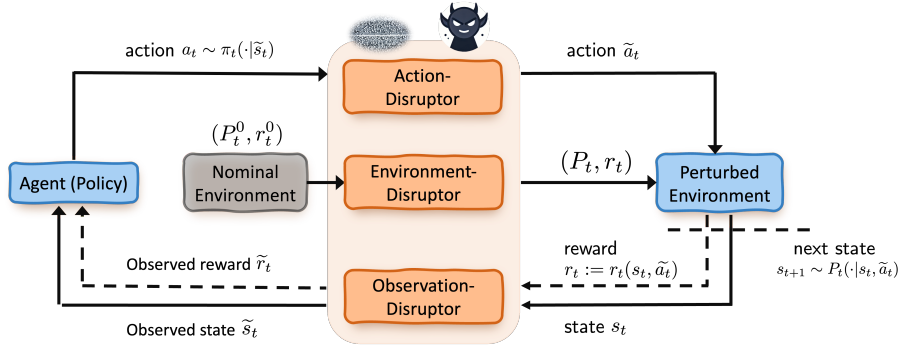


Figure 2: The overview of a finite-horizon MDP with disruptors.

2.1 A UNIFIED ROBUST RL FRAMEWORK: MDPs WITH DISRUPTION

To proceed, we introduce an additional disruption module that represents potential uncertainties or disturbances that impact different stages of the agent-environment interaction process (MDP). This module provides a categorized summary of the types of uncertainty addressed in prior robust RL studies.

Disruptors. We introduce each type in detail as follows:

- *Observation-disruptor.* An agent’s observations may not perfectly reflect the true status of the environment due to factors like sensor noise and time delays. To model this sensing inaccuracy, we introduce an additional module—the observation-disruptor—which determines the agent’s observations from the environment: *Agents’ observed state* \tilde{s}_t : The observation-disruptor takes the true current state s_t as input and outputs a perturbed state $\tilde{s}_t = D_s(s_t)$. The agent uses \tilde{s}_t as input to its policy to select an action; *Agents’ observed reward* \tilde{r}_t : The observation-disruptor takes the real immediate reward r_t as input and outputs a perturbed reward $\tilde{r}_t = D_r(r_t)$. The agent observes \tilde{r}_t and updates its policy accordingly.
- *Action-disruptor.* The real action a_t chosen by the agent may be altered before or during execution in the environment due to implementation inaccuracies or system malfunctions. The action-disruptor models this perturbation, outputting a perturbed action $\tilde{a}_t = D_a(a_t)$, which is then executed in the environment for the next step.
- *Environment-disruptor.* Recall that a task environment consists of both the internal dynamic model and the external workspace it interacts with, characterized by its transition dynamics P and reward function r . The environment during training can differ from the real-world environment due to factors such as the sim-to-real gap, human and natural variability, external disturbances, and more. We attribute this potential nonstationarity to an environment-disruptor, which determines the actual environment (P, r) the agent is interacting with at any given moment. These dynamics may differ from the nominal environment (P^0, r^0) that the agent was originally expected to interact with.

MDPs with Disruption. As shown in Fig. 2, a robust RL problem can be formulated as a finite-horizon MDP with an additional disruption module $\mathcal{M}_{\text{dis}} = (\mathcal{S}, \mathcal{A}, T, P, r, D_s(\cdot), D_r(\cdot), D_a(\cdot))$, abbreviated as **Disrupted-MDP**. It consists of three potential disruptors introduced above. Specifically, the interaction process between an agent and an MDP with disruption (Fig. 2) unfolds as follows: at each time step $t \in [T]$, the (possibly perturbed) environment outputs the current state s_t and reward r_t . The *observation-disruptor* then perturbs these, sending the modified state $\tilde{s}_t = D_s(s_t)$ and reward $\tilde{r}_t = D_r(r_t)$ to the agent. Based on these, the agent selects an action $a_t \sim \pi_t(\cdot | \tilde{s}_t)$, according to its policy $\pi = \{\pi_t\}_{1 \leq t \leq T}$, where $\pi_t : \mathcal{S} \rightarrow \Delta(\mathcal{A})$ defines the probability distribution over actions in \mathcal{A} given the observed state \tilde{s}_t . The *action-disruptor* then perturbs this action to $\tilde{a}_t = D_a(a_t)$, which is then sent to a perturbed environment governed by the *environment-disruptor*, based on the reference — nominal environment (P^0, r^0) . The environment

then transitions to the next state $s_{t+1} \sim P_t(\cdot \mid s_t, \tilde{a}_t)$ and provides the reward $r_{t+1}(s_t, \tilde{a}_t)$, which becomes the input for the observation-disruptor in the next step $t + 1$.

Goal. For any Disrupted-MDP, the objective is to learn a policy (action selection rule) $\pi = \{\pi_t\}_{1 \leq t \leq T}$ that maximizes long-term cumulative rewards, represented by the value function $\{V_t^\pi\}_{1 \leq t \leq T} : \mathcal{S} \mapsto \mathbb{R}$:

$$\max_{\pi} V_t^\pi(s) = \mathbb{E} \left[\sum_{k=t}^T r_k(s_k, \tilde{a}_k) \mid \pi, (P, r), s_t = s \right]. \quad (1)$$

Here, the expectation is taken over the trajectories generated by executing the policy π under the perturbed transition kernels and reward functions (P, r) .

In addition to disruption modes, **the Disrupted-MDP allows disruptors** to operate flexibly over time during the interaction process. Disruptors can act at different frequencies, such as step-wise, episode-wise, or at varying intervals.

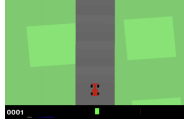
3 Robust-Gymnasium: A UNIFIED ROBUST RL BENCHMARK

We now introduce our main contribution, a modular benchmark (Robust-Gymnasium) designed for evaluating Robust RL policies in robotics and control tasks. Each task is constructed from three main components: an agent model (the robot object), an environment (the agent’s workspace), and a task objective (such as navigation or manipulation). Robust-Gymnasium offers robust RL tasks by integrating various disruptors of different types, modes, and frequencies with these task bases. Not all task bases support every type of disruption. A detailed list of the robust RL tasks implemented in this benchmark is available in Figure 17. In the following sections, we introduce over 60 task bases from eleven sets, outline the design of the disruptors, and describe the construction of a Disrupted-MDP—robust RL tasks.

3.1 TASK AND ENVIRONMENT BASES

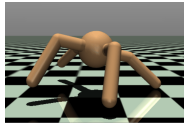
Gymnasium-Box2D (*three relative simple control tasks in games*).

These tasks are from Gymnasium (Towers et al., 2024), including three robot models from different games, such as the Bipedal Walker — a 4-joint walking robot designed to move forward and Car Racing — navigating a track by learning from pixel inputs (Parberry, 2017; Brockman et al., 2016).



Gymnasium-MuJoCo (*eleven control tasks*).

It includes various robot models, such as bipedal and quadrupedal robots. This benchmark is widely used in various RL problems, including standard online and offline RL, with representative examples like Hopper, Ant, and HalfCheetah (Todorov et al., 2012; Brockman et al., 2016).



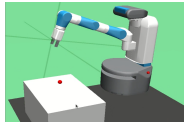
Maze (*two navigation environments*).

Maze comprises environments where an agent must reach a specified goal within a maze (Gupta et al., 2020). Two types of agents are available: a 2-degrees of freedom (DoF) ball (Point-Maze) and a more complex 8-DoF quadruped robot (Ant-Maze) from Gymnasium-MuJoCo. Various goals and maze configurations can be generated to create tasks of varying difficulty.



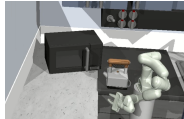
Fetch (*four tasks for Fetch Mobile Manipulator robot arm*).

Fetch features a 7-degrees of freedom (DoF) Fetch Mobile Manipulator arm with a two-fingered parallel gripper (Plappert et al., 2018). The environment consists of a table with various objectives, resulting in four tasks: Reach, Push, Slide, and PickAndPlace, which involve picking up or moving the objects to specified locations.



Franka Kitchen (*tasks need long-horizon, multi-task planning for a robot arm*).

This environment is based on a 9-degrees of freedom (DoF) Franka robot situated in a kitchen containing common household items like a microwave and cabinets (Gupta et al., 2020). The task goal is to achieve a specified configuration,



which may involve planning and completing multiple sub-tasks. For example, a goal state could have the microwave open, a kettle inside, and the light over the burners turned on.

Dexterous Hand (*five dexterous hand manipulation tasks*).

It is based on the Shadow Dexterous Hand — an anthropomorphic 24-DoF robotic hand with 92 touch sensors at palm and phalanges of the fingers (Plappert et al., 2018; Melnik et al., 2021). The tasks involve manipulating various objects, such as a pen, egg, or blocks.



Adroit (*four manipulation tasks for a dexterous hand attached to a free arm*).

This environment features a free arm equipped with a Shadow Dexterous Hand, providing up to 30-DoF (Rajeswaran et al., 2018). The high degree of freedom enables the robot to perform more complex tasks, such as opening a door with a latch (AdroitHandDoor).



HumanoidBench (*four tasks for a high-dimensional humanoid*).

We incorporate four tasks from the recent HumanoidBench (Sferrazza et al., 2024) designed mainly for a Unitree H1 humanoid robot ², which is equipped with two dexterous Shadow Hands. Specifically, we include two manipulation tasks (push, truck) and two locomotion tasks (reach, slide), all of which require sophisticated coordination among various body parts.



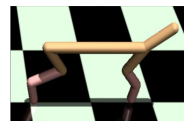
Robosuite (*twelve tasks for various modular robot platforms*).

Robosuite is a popular modular benchmark (Zhu et al., 2020) that supports seven robot arms, eight grippers, and six controller modes. The manipulation tasks are conducted in environments with doors, tables, and multiple robot arms, with goals such as wiping tables or coordinating to transfer a hammer. Additionally, we introduce a new task—MultiRobustDoor—featuring an adversarial arm that impedes another arm’s success to test robustness.



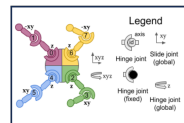
Safety MuJoCo (*nine control tasks with additional safety constraints*).

Built on standard robot models in Gymnasium-MuJoCo, the Safety MuJoCo tasks are designed for scenarios that prioritize both long-term returns and safety. These tasks incorporate safety constraints, such as limiting velocity and preventing robots from falling (Gu et al., 2024).



MAMuJoCo (*twelve multi-agent cooperation tasks*).

MAMuJoCo is based on a multi-agent platform from the factorizations of Gymnasium-MuJoCo robot models (Peng et al., 2021). The tasks need to be solved by cooperations of multiple agents. This set of tasks are vulnerable to disturbance like one leg of a quadruped robot is malfunctioning, or all dynamics of legs are contaminated by system noise.



3.2 DISRUPTOR DESIGN: MODES AND FREQUENCIES

In a Disrupted-MDP, disruptors affecting various stages of the agent-environment interaction can operate in different modes. We typically consider four common modes found in the robust RL literature, each driven by specific real-world scenarios and robustness requirements. These modes allow the construction of tasks with varying levels of difficulty:

- *Random disturbance: for all disruptors.* Stochastic noise is ubiquitous in sensors, mechanical hardware, and random events, often modeled as random noise added to nominal components in the interaction process (Duan et al., 2016). The noise typically follows a distribution such as Gaussian or uniform. This mode can be applied to all disruptors, affecting the agent’s observed state, observed reward, action, and environment.

We offer Gaussian distribution $N(\cdot, \cdot)$ (Zhang et al., 2018) and bounded uniform distribution $\mathcal{U}(\cdot, \cdot)$ (Zouitine et al., 2024) as default options. For instance, the environment-disruptor can introduce noise to robot dynamics (e.g., mass, torso length) or external factors

²<https://www.unitree.com/h1/>

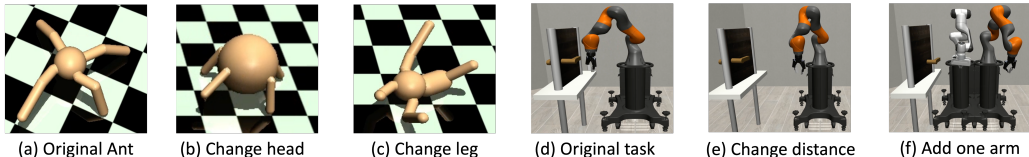


Figure 3: Illustration of two disruption modes of the environment-disruptor: internal dynamic shift and external disturbance.

(e.g., gravity, wind), as shown in Fig. 4. The observation-disruptor can add noise to the observed state and/or reward, namely, $\tilde{s}_t = s_t + \mathcal{N}(\mu_s, \sigma_s)$ (μ_s and σ_s are the mean and variance) or $\tilde{s}_t = s_t + \mathcal{U}(a_s, b_s)$ (a_s, b_s are the min and max thresholds). The action-disruptor can also introduce noise to the action sent to the environment.

- *Adversarial disturbance: for all disruptors.* In real-world applications, adversarial disturbances occur when external forces deliberately attempt to degrade the agent’s performance. This mode is also relevant when prioritizing safety, ensuring the agent can perform well in worst-case scenarios within certain predefined sets. It can be applied to all three disruptors. **This mode can be viewed as a two-player zero sum game between the agent and an adversarial player (Tanabe et al., 2022). Any algorithms can acts as the adversarial player through this interface to adversarially attack the process.** This mode is applicable to all disruptors; for instance, the observation-disruptor generates a fake state that falls within the prescribed set around the true state, or the environment-disruptor adjusts parameters within a neighborhood of the nominal values;

Notably, in our benchmark, we implement and feature an algorithm leveraging LLM to determine the disturbance. In particular, the LLM is told of the task and uses the current state and reward signal as the input. It directly outputs the disturbed results like a fake state for the agent. See more details in the code 2 in Appendix C.1.

- *Internal dynamic shift: for the environment-disruptor.* This mode captures variations in the agent’s internal model between training and testing, caused by factors such as the sim-to-real gap, measurement noise, or accidental malfunctions. The environment-disruptor introduces biases to dynamic parameters within a prescribed uncertainty set. For example, the torso length (Fig. 4 (c)) might shift from 0.3 to 0.5.

For tasks in control and robotics, the environment disruptor can alter the robot model, **changing the system’s internal dynamics** (Zhang et al., 2020; Zouitine et al., 2024). Using Gymnasium-MuJoCo as an example, Fig. 3(b)-(c) depict the consequences of such disruption by altering the Ant robot’s head and legs around its original model (Fig. 3(a)).

- *External disturbance: for the environment-disruptor.* Nonstationarity in the external workspace can result from variability in the physical world or human behavior, such as changes in wind, friction, or human intervention. The environment-disruptor uses this mode to modify the external task environment by altering properties and configurations within the robot’s workspace or by introducing abrupt external interventions (Luo et al., 2024; Pinto et al., 2017; Ding et al., 2024).

For example, in robosuite, Fig. 3(e)-(f) illustrate disrupted tasks compared to the original reference in Fig. 3(d). In these tasks, the environment disruptor changes the distance between the table and the arm, or even introduces an additional arm to actively interfere with the yellow-black robot’s ability to accomplish its goal.

Timing of operations for disruptors. We support perturbations occurring at any stage of the process and at different frequencies. Users can choose to apply perturbations at any time step or episode during the training process, or exclusively during testing.

3.3 CONSTRUCTING ROBUST RL TASKS

Robust-Gymnasium is a modular benchmark that offers flexible methods for constructing robust RL tasks through three main steps. First, we select a task base from the eleven options outlined in Sec. 3.1. Second, we choose a disruptor from the observation, action, and environment categories in-

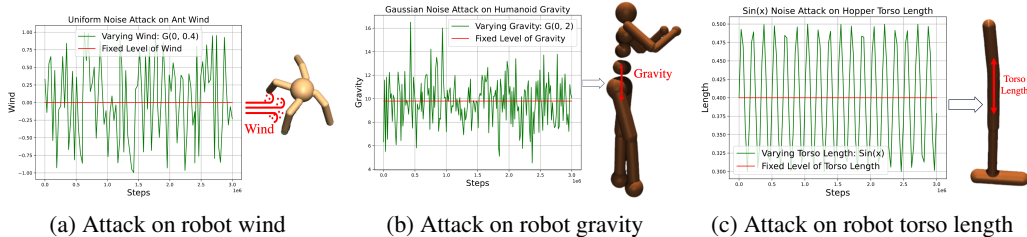


Figure 4: Adversary attack on robot environments, dynamics and shape with different distributions (We can also attack on robot state space, action space and reward signal, etc.).

roduced in Sec. 3.2), and specify its operation modes (random disturbance, adversarial disturbance, internal dynamic shift, and external disturbance, as detailed in Sec. 3.2). Finally, we determine the interaction process and frequencies between the disruptor, agent, and environment.

In addition to these basic construction methods, our benchmark supports advanced modes: *A combination of disruptors* allows users to select multiple disruptors, such as an observation-disruptor and an environment-disruptor, to simulate conditions where perception sensors have system noise and external disturbances from human occur; *Varying operation frequencies* enables disruptors to operate intermittently during interactions, either at fixed intervals or in a random pattern to characterize accidental events and uncertainties.

4 EXPERIMENTS AND ANALYSIS

Robust-Gymnasium offers a variety of tasks for comprehensively evaluating the robustness of different RL paradigms. We demonstrate its flexibility by constructing robust RL tasks based on various task bases, incorporating disruptions with different types, modes, and frequencies, and evaluating several SOTA algorithms on these tasks. In addition to benchmarking existing algorithms, we also highlight an adversarial disruption mode that leverages LLMs. Examples of robust RL tasks are shown in Figure 4. More details about the experiments can be found in Appendix E.

Benchmark RL algorithms. Specifically, we benchmark several SOTA algorithms in their corresponding robust RL tasks: **Standard RL:** Proximal Policy Optimization (PPO) (Schulman et al., 2017), Soft Actor-Critic (SAC) (Haarnoja et al., 2018); **Robust RL:** Occupancy-Matching Policy Optimization (OMPO) (Luo et al., 2024), Robust State-Confounded SAC (RSC) (Ding et al., 2024), Alternating Training with Learned Adversaries (ATLA) (Zhang et al., 2021b), and Deep Bisimulation for Control (DBC) (Zhang et al., 2021a); **Safe RL:** Projection Constraint-Rectified Policy Optimization (PCRPO) (Gu et al., 2024), Constraint-Rectified Policy Optimization (CRPO) (Xu et al., 2021); **Multi-Agent RL:** Multi-Agent PPO (MAPPO) (Yu et al., 2022), Independent PPO (IPPO) (De Witt et al., 2020).

Evaluation processes. We mainly focus on two evaluation settings: *In-training:* the disruptor simultaneously affects the agent and environment during both training and testing at each time step. This process is typically used in robotics to address sim-to-real gaps by introducing potential noise during training; 2) *Post-training:* the disruptor only impacts the agent and environment during testing, mimicking scenarios where learning algorithms are unaware of testing variability.

Robust metrics. In this work, we usually use the performance in the original (deployment) environment as the robust metric for evaluations. While there are many different formulations of the robust RL objective (robust metrics), such as risk-sensitive metrics (e.g., CVaR) (Chan et al., 2019), and the worst-case or average performance when the environment shifts (Zouitine et al., 2024).

4.1 EVALUATION OF STANDARD RL BASELINES

To begin, we evaluate two types of robust RL tasks: one with an observation disruptor (affecting the agent’s observed state) and the other with an action disruptor (affecting the action), both subjected to random disturbances at varying levels. We benchmark the performance of standard RL base-

lines—PPO (Schulman et al., 2017) and SAC (Haarnoja et al., 2018)—on robust RL tasks based on the representative HalfCheetah-v4 task from Gymnasium-MuJoCo, as partially shown in Figure 5. Here, $S=0.1$ indicates that the random disturbance over the state follows a Gaussian distribution with a mean of 0 and a standard deviation of 0.1 (resp. 0.15). The same applies for $A=0.1$ or $A=0.15$. Figures 5 (a)-(b) and Figure 5 (c)-(d) present the results from two different evaluation processes—In-training and Post-training, respectively. The results show that as the disturbance level increases, the performance of the baselines degrades quickly, particularly when the training process is unaware of potential disturbances (as seen in the Post-training results). More experiments, including those using disturbances over reward or the results for SAC, can be found in Appendix B.1.

4.2 EVALUATION OF ROBUST RL BASELINES

In this section, we evaluate robust RL tasks using an environment disruptor under two representative modes: internal dynamic shift and external disturbance. The robust RL tasks are based on various task bases, including Ant-v5 and Hopper-v5 from Gymnasium-MuJoCo, as well as DoorCausal-v1 and LiftCausal-v1 from Robosuite, utilizing the In-training evaluation process.

Specifically, Figure 6(a-b) displays the performance of the robust RL baseline OMPO across two tasks with internal dynamic shifts: (a) Ant-v5 with varying gravity and wind strength, and (b) Hopper-v5 with changes to the robot model’s shape, including torso and foot length. Experimental settings can be found in (4) and (6) in Appendix C.2. The results indicate that OMPO’s performance significantly declines in non-stationary environments compared to stationary conditions without disturbances.

Figures 6(c-d) illustrate the performance of three robust RL baselines (RSC, ATLA, DBC) in two tasks from Robosuite involving disruptions on the environment with external semantic disturbances. In the DoorCausal task, the initial distance of the door from the robot and the height of the door handle are varied in a correlated manner. In the CausalLift task, both the position and color of the object to be lifted are changed together according to specific patterns. RSC demonstrates greater robustness than ATLA and DBC, maintaining stable reward trajectories throughout the training process. However, RSC’s training efficiency may need further improvement, as it generates augmentation data during policy learning.

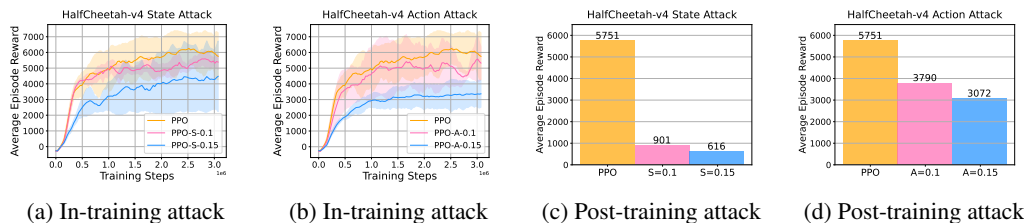


Figure 5: Adversary attack on state and action space in robust HalfCheetah-v4 tasks. S denotes attack on state and A denotes attack on action.

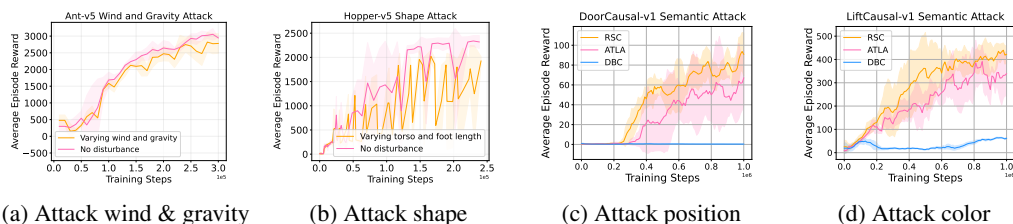


Figure 6: (a-b): Internal dynamic shift attacks on Ant-v5 and Hopper-v5 tasks. (c-d): External semantic attacks on Robosuite tasks.

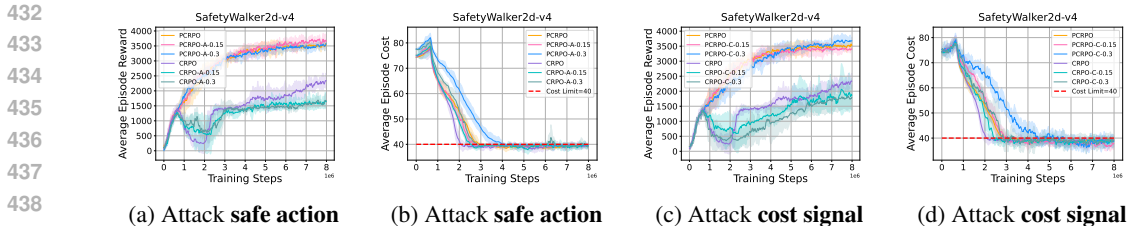


Figure 7: Robust safe RL tasks: Random disturbances over either the action or the agent’s observed immediate cost feedback.

4.3 EVALUATION OF SAFE RL BASELINES

Two safe RL baselines, PCRPO (Gu et al., 2024) and CRPO (Xu et al., 2021), are benchmarked on robust safety-critical tasks using the In-training evaluation process. Specifically, we assess two types of robust RL tasks based on Walker2d from Gymnasium-MuJoCo: (a) an action-disruption attacks the agent’s action with different levels; (b) the agent’s observe immediate safety cost is disturbed in different levels. These attacks follow a Gaussian distribution with a mean of 0 and standard deviations of 0.15 or 0.3 for both the action and the observed cost. The outcomes and safety costs for these tasks are presented in Figures 12(a-b) and Figures 12(c-d), respectively. The performance of CRPO quickly degrades when disruptions occur, while PCRPO demonstrates greater robustness against disturbances in either action or observed cost. Notably, PCRPO’s performance under disturbance surpasses its performance without disturbance, suggesting that introducing appropriate disturbances during training may enhance overall performance. Due to space limitations, additional results can be found in Appendix B.2.

4.4 EVALUATION OF MULTI-AGENT RL BASELINES

We evaluate two MARL baselines: Multi-Agent PPO (MAPPO) (Yu et al., 2022) and Independent PPO (IPPO) (De Witt et al., 2020) on MA-HalfCheetah-v4 from MAMoJoCo under various disruption settings affecting the agents’ observed states, actions, and rewards. Using the In-training evaluation process, as shown in Figure 8, we apply disruptions to all agents. The results indicate that the performance of both MAPPO and IPPO degrades accordingly as the disruptions occur. Additionally, we conduct experiments involving **partial disruptions** on a subset of agents within the multi-agent system; further details can be found in Appendix B.3.

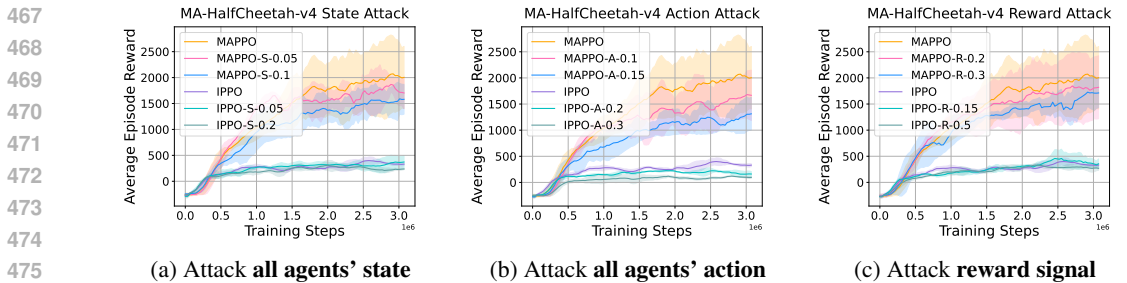


Figure 8: Multi-Agent HalfCheetah-2x3 robustness: training attack on state, action, and reward for all the two agents. S denotes state, A denotes action and R denotes reward.

4.5 ADVERSARIAL DISTURBANCE THROUGH LLMs

In addition to benchmarking various existing RL algorithms, this section demonstrates the adversarial disturbance mode by leveraging a featured approach with LLMs. As shown in Figure 9, we evaluate the performance of PPO on Ant-v4 with adversarial disruptions to the agent’s observed state. Different attack configurations are employed, including comparisons to uniform noise and testing varying frequencies. Here, “C[0.2–0.8]” indicates that the noise level from the LLM is

486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539

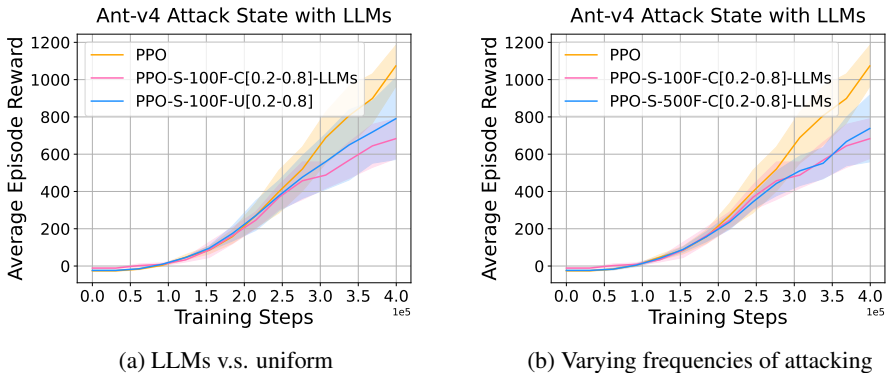


Figure 9: LLM-based attacks with different settings.

constrained within the $[0.2, 0.8]$ range; “100F” (resp. “500F”) signifies that the agent is attacked every 100 (resp. 500) steps; and “U[0.2–0.8]” represents noise drawn from a uniform distribution $\mathcal{U}(0.2, 0.8)$. The results show that LLM-based attacks lead to a more significant performance drop for PPO compared to that using uniform distribution (Figure 9(a)). Figure (b) examines how varying attack frequencies affect performance, revealing that higher-frequency attacks (PPO-S-100F) result in greater performance degradation. Due to space constraints, additional frequency experiments on other robust tasks based on Gymnasium-MuJoCo using PPO are provided in Appendix B.4.

5 CONCLUSION

In this work, we introduce Robust-Gymnasium, a unified modular benchmark explicitly designed for robust RL. Unlike existing RL benchmarks, Robust-Gymnasium aims to evaluate the resilience of RL algorithms across a wide range of disruptions. These disruptions include perturbations at every stage of the entire agent-environment interaction process, affecting agent observations, actions, rewards, and environmental dynamics. Robust-Gymnasium provides a comprehensive platform for benchmarking RL algorithms, featuring over 60 diverse task environments across domains such as robotics, multi-agent systems, and safe RL. Additionally, we benchmark various SOTA RL algorithms, including PPO, MAPPO, OMPO, RSC, and IPPO, across a wide array of robust RL tasks in Robust-Gymnasium. The results highlight the deficiencies of current algorithms and motivate the development of new ones. This work represents a significant step forward in standardizing and advancing the field of robust RL, promoting the creation of more reliable, generalizable, and robust learning algorithms.

Ethics Statement. This work is conducted without involving human subjects or personal data that might raise ethical concerns. There are no conflicts of interest, privacy issues, or potential for harm associated with this work.

Reproducibility Statement. To facilitate the reproducibility of our experiments, we have provided the source code at the link: <https://robust-rl.online/>. Detailed descriptions of algorithm parameters are included in Appendix E.

REFERENCES

- 540
541
542 W Brian Arthur. Designing economic agents that act like human agents: A behavioral approach to
543 bounded rationality. *The American economic review*, 81(2):353–359, 1991.
- 544
545 Kishan Panaganti Badrinath and Dileep Kalathil. Robust reinforcement learning using least squares
546 policy iteration with provable performance guarantees. In *International Conference on Machine*
547 *Learning*, pp. 511–520. PMLR, 2021.
- 548
549 Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environ-
550 ment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:
253–279, 2013.
- 551
552 Dimitris Bertsimas, Melvyn Sim, and Meilin Zhang. Adaptive distributionally robust optimization.
553 *Management Science*, 65(2):604–618, 2019.
- 554
555 Clément Bonnet, Daniel Luo, Donal John Byrne, Shikha Surana, Sasha Abramowitz, Paul Duck-
556 worth, Vincent Coyette, Laurence Illing Midgley, Elshadai Tegegn, Tristan Kalloniatis, Omayma
557 Mahjoub, Matthew Macfarlane, Andries Petrus Smit, Nathan Grinsztajn, Raphael Boige, Cem-
558 lyn Neil Waters, Mohamed Ali Ali Mimouni, Ulrich Armel Mbou Sob, Ruan John de Kock,
559 Siddarth Singh, Daniel Furelos-Blanco, Victor Le, Arnu Pretorius, and Alexandre Laterre. Ju-
560 manji: a diverse suite of scalable reinforcement learning environments in JAX. In *The Twelfth*
561 *International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=C4CxQmp9wc>.
- 562
563 Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and
Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.
- 564
565 Stephanie CY Chan, Samuel Fishman, John Canny, Anoop Korattikara, and Sergio Guadarrama.
566 Measuring the reliability of reinforcement learning algorithms. *arXiv preprint arXiv:1912.05663*,
2019.
- 567
568 Xin Chen, Guannan Qu, Yujie Tang, Steven Low, and Na Li. Reinforcement learning for selective
569 key applications in power systems: Recent advances and future challenges. *IEEE Transactions*
570 *on Smart Grid*, 13(4):2935–2958, 2022.
- 571
572 Cédric Colas, Olivier Sigaud, and Pierre-Yves Oudeyer. How many random seeds? statistical power
analysis in deep reinforcement learning experiments. *arXiv preprint arXiv:1806.08295*, 2018.
- 573
574 Murat Cubuktepe, Nils Jansen, Sebastian Junges, Ahmadreza Marandi, Marnix Suilen, and Ufuk
575 Topcu. Robust finite-state controllers for uncertain pomdps. In *Proceedings of the AAAI Confer-*
576 *ence on Artificial Intelligence*, volume 35, pp. 11792–11800, 2021.
- 577
578 Mark A Davenport and Justin Romberg. An overview of low-rank matrix recovery from incomplete
observations. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):608–622, 2016.
- 579
580 Christian Schroeder De Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS
581 Torr, Mingfei Sun, and Shimon Whiteson. Is independent learning all you need in the starcraft
582 multi-agent challenge? *arXiv preprint arXiv:2011.09533*, 2020.
- 583
584 Esther Derman and Shie Mannor. Distributional robustness and regularization in reinforcement
learning. *arXiv preprint arXiv:2003.02894*, 2020.
- 585
586 Wenhao Ding, Laixi Shi, Yuejie Chi, and Ding Zhao. Seeing is not believing: Robust reinforce-
587 ment learning against spurious correlation. In *Thirty-seventh Conference on Neural Information*
Processing Systems, 2023a.
- 588
589 Wenhao Ding, Chejian Xu, Mansur Arief, Haohong Lin, Bo Li, and Ding Zhao. A survey on
590 safety-critical driving scenario generation—a methodological perspective. *IEEE Transactions on*
591 *Intelligent Transportation Systems*, 24(7):6971–6988, 2023b.
- 592
593 Wenhao Ding, Laixi Shi, Yuejie Chi, and Ding Zhao. Seeing is not believing: Robust reinforcement
learning against spurious correlation. *Advances in Neural Information Processing Systems*, 36,
2024.

- 594 Yan Duan, Xi Chen, Rein Houthoofd, John Schulman, and Pieter Abbeel. Benchmarking deep
595 reinforcement learning for continuous control. In *International conference on machine learning*,
596 pp. 1329–1338. PMLR, 2016.
- 597
598 Theresa Eimer, André Biedenkapp, Maximilian Reimer, Steven Adriaensen, Frank Hutter, and Mar-
599 rius Lindauer. Dacbench: A benchmark library for dynamic algorithm configuration. *IJCAI*, 2021.
- 600 Vineet Goyal and Julien Grand-Clement. Robust markov decision processes: Beyond rectangularity.
601 *Mathematics of Operations Research*, 2022.
- 602
603 Shangding Gu, Bilgehan Sel, Yuhao Ding, Lu Wang, Qingwei Lin, Ming Jin, and Alois Knoll.
604 Balance reward and safety optimization for safe reinforcement learning: A perspective of gradient
605 manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp.
606 21099–21106, 2024.
- 607 Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy
608 learning: Solving long-horizon tasks via imitation and reinforcement learning. In *Conference on*
609 *Robot Learning*, pp. 1025–1037. PMLR, 2020.
- 610
611 Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy
612 maximum entropy deep reinforcement learning with a stochastic actor. In *International confer-*
613 *ence on machine learning*, pp. 1861–1870. PMLR, 2018.
- 614 Songyang Han, Sanbao Su, Sihong He, Shuo Han, Haizhao Yang, and Fei Miao. What is the solution
615 for state adversarial multi-agent reinforcement learning? *arXiv preprint arXiv:2212.02705*, 2022.
- 616
617 Sihong He, Songyang Han, Sanbao Su, Shuo Han, Shaofeng Zou, and Fei Miao. Robust multi-agent
618 reinforcement learning with state uncertainty. *Transactions on Machine Learning Research*, 2023.
- 619 Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger.
620 Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial*
621 *intelligence*, volume 32, 2018.
- 622
623 Chin Pang Ho, Marek Petrik, and Wolfram Wiesemann. Fast bellman updates for robust MDPs. In
624 *International Conference on Machine Learning*, pp. 1979–1988. PMLR, 2018.
- 625
626 Chin Pang Ho, Marek Petrik, and Wolfram Wiesemann. Partial policy iteration for ℓ_1 -robust markov
627 decision processes. *Journal of Machine Learning Research*, 22(275):1–46, 2021.
- 628
629 Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks
630 on neural network policies. *arXiv preprint arXiv:1702.02284*, 2017.
- 631
632 Garud N Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2):
633 257–280, 2005.
- 634
635 David L Kaufman and Andrew J Schaefer. Robust modified policy iteration. *INFORMS Journal on*
636 *Computing*, 25(3):396–410, 2013.
- 637
638 Olga Klopp, Karim Lounici, and Alexandre B Tsybakov. Robust matrix completion. *Probability*
639 *Theory and Related Fields*, 169(1-2):523–564, 2017.
- 640
641 Joseph Peter Kozak, Ruizhe Zhang, Matthew Porter, Qihao Song, Jingcun Liu, Bixuan Wang, Rudy
642 Wang, Wataru Saito, and Yuhao Zhang. Stability, reliability, and robustness of gan power devices:
643 A review. *IEEE Transactions on Power Electronics*, 38(7):8442–8471, 2023.
- 644
645 Heinrich Küttler, Nantas Nardelli, Alexander Miller, Roberta Raileanu, Marco Selvatici, Edward
646 Grefenstette, and Tim Rocktäschel. The nethack learning environment. *Advances in Neural*
647 *Information Processing Systems*, 33:7671–7684, 2020.
- 648
649 Zhongyu Li, Xuxin Cheng, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil
650 Sreenath. Reinforcement learning for robust parameterized locomotion control of bipedal robots.
651 In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2811–2817.
652 IEEE, 2021.

- 648 Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero:
649 Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information*
650 *Processing Systems*, 36, 2024.
- 651 Qing Liu, Yulan Li, and Katherine Odem-Davis. On robustness of noninferiority clinical trial designs
652 against bias, variability, and nonconstancy. *Journal of biopharmaceutical statistics*, 25(1):206–
653 225, 2015.
- 654 Zuxin Liu, Zijian Guo, Zhepeng Cen, Huan Zhang, Jie Tan, Bo Li, and Ding Zhao. On the
655 robustness of safe reinforcement learning under observational perturbations. *arXiv preprint*
656 *arXiv:2205.14691*, 2022.
- 657 Yu Luo, Tianying Ji, Fuchun Sun, Jianwei Zhang, Huazhe Xu, and Xianyuan Zhan. Ompo: A
658 unified framework for rl under policy and dynamics shifts. In *Forty-first International Conference*
659 *on Machine Learning*, 2024.
- 660 A Rupam Mahmood, Dmytro Korenkevych, Gautham Vasan, William Ma, and James Bergstra.
661 Benchmarking reinforcement learning algorithms on real-world robots. In *Conference on robot*
662 *learning*, pp. 561–591. PMLR, 2018.
- 663 Ajay Mandlekar, Yuke Zhu, Animesh Garg, Li Fei-Fei, and Silvio Savarese. Adversarially robust
664 policy learning: Active construction of physically-plausible perturbations. In *2017 IEEE/RSJ*
665 *International Conference on Intelligent Robots and Systems (IROS)*, pp. 3932–3939. IEEE, 2017.
- 666 Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro
667 Yasunaga, Richard Lanus Phillips, Sara Beery, Jure Leskovec, Anshul Kundaje, et al. Wilds: A
668 benchmark of in-the-wild distribution shifts. *arXiv preprint arXiv:2012.07421*, 2020.
- 669 Ishita Mediratta, Qingfei You, Minqi Jiang, and Roberta Raileanu. The generalization gap in offline
670 reinforcement learning. *arXiv preprint arXiv:2312.05742*, 2023.
- 671 Andrew Melnik, Luca Lach, Matthias Plappert, Timo Korthals, Robert Haschke, and Helge Ritter.
672 Using tactile sensing to improve the sample efficiency and performance of deep deterministic pol-
673 icy gradients for simulated in-hand manipulation tasks. *Frontiers in Robotics and AI*, 8:538773,
674 2021.
- 675 Jorge A Mendez, Marcel Hussing, Meghna Gummadi, and Eric Eaton. Composuite: A composi-
676 tional reinforcement learning benchmark. In *Conference on Lifelong Learning Agents*, pp. 982–
677 1003. PMLR, 2022.
- 678 Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Belle-
679 mare, Alex Graves, Martin Riedmiller, Andreas K Fidfjeland, and Georg Ostrovski. Human-level
680 control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.
- 681 Janosch Moos, Kay Hansel, Hany Abdulsamad, Svenja Stark, Debora Clever, and Jan Peters. Robust
682 reinforcement learning: A review of foundations and recent advances. *Machine Learning and*
683 *Knowledge Extraction*, 4(1):276–315, 2022.
- 684 Tongzhou Mu, Zhan Ling, Fanbo Xiang, Derek Yang, Xuanlin Li, Stone Tao, Zhiao Huang, Zhi-
685 wei Jia, and Hao Su. Maniskill: Generalizable manipulation skill benchmark with large-scale
686 demonstrations. *arXiv preprint arXiv:2107.14483*, 2021.
- 687 OpenAI. Gpt-4 technical report. 2023.
- 688 Yuxin Pan, Yize Chen, and Fangzhen Lin. Adjustable robust reinforcement learning for online 3d
689 bin packing. *arXiv preprint arXiv:2310.04323*, 2023.
- 690 Ian Parberry. *Introduction to Game Physics with Box2D*. CRC Press, 2017.
- 691 Beomsoo Park and Benjamin Van Roy. Adaptive execution: Exploration and learning of price
692 impact. *Operations Research*, 63(5):1058–1076, 2015.
- 693 Anay Pattanaik, Zhenyi Tang, Shuijing Liu, Gautham Bommanan, and Girish Chowdhary. Robust
694 deep reinforcement learning with adversarial attacks. *arXiv preprint arXiv:1712.03632*, 2017.

- 702 Bei Peng, Tabish Rashid, Christian Schroeder de Witt, Pierre-Alexandre Kamienny, Philip Torr,
703 Wendelin Böhmer, and Shimon Whiteson. Facmac: Factored multi-agent centralised policy gra-
704 dients. *Advances in Neural Information Processing Systems*, 34:12208–12221, 2021.
- 705
- 706 Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforce-
707 ment learning. In *International Conference on Machine Learning*, pp. 2817–2826. PMLR, 2017.
- 708
- 709 Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Pow-
710 ell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, et al. Multi-goal reinforce-
711 ment learning: Challenging robotics environments and request for research. *arXiv preprint*
712 *arXiv:1802.09464*, 2018.
- 713
- 714 Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Part-
715 sey, Ruta Desai, Alexander Clegg, Michal Hlavac, So Yeon Min, Vladimír Vondruš, Theophile
716 Gervet, Vincent-Pierre Berges, John M Turner, Oleksandr Maksymets, Zolt Kira, Mrinal Kalakr-
717 ishnan, Jitendra Malik, Devendra Singh Chaplot, Unnat Jain, Dhruv Batra, Akshara Rai, and
718 Roozbeh Mottaghi. Habitat 3.0: A co-habitat for humans, avatars, and robots. In *The Twelfth*
719 *International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=4znwzG92CE>.
- 720
- 721 Wilbert Pumacay, Ishika Singh, Jiafei Duan, Ranjay Krishna, Jesse Thomason, and Dieter Fox. The
722 colosseum: A benchmark for evaluating generalization for robotic manipulation. *arXiv preprint*
723 *arXiv:2402.08191*, 2024.
- 724
- 725 You Qiaoben, Xinning Zhou, Chengyang Ying, and Jun Zhu. Strategically-timed state-observation
726 attacks on deep reinforcement learning agents. In *ICML 2021 Workshop on Adversarial Machine*
727 *Learning*, 2021.
- 728
- 729 Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel
730 Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement
731 learning and demonstrations. *Robotics: Science and Systems XIV*, 2018.
- 732
- 733 Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement
734 learning. *arXiv preprint arXiv:1910.01708*, 7(1):2, 2019.
- 735
- 736 John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy
737 optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- 738
- 739 Carmelo Sferrazza, Dun-Ming Huang, Xingyu Lin, Youngwoon Lee, and Pieter Abbeel. Humanoid-
740 bench: Simulated humanoid benchmark for whole-body locomotion and manipulation. *arXiv*
741 *preprint arXiv:2403.10506*, 2024.
- 742
- 743 Elena Smirnova, Elvis Dohmatob, and Jérémie Mary. Distributionally robust reinforcement learning.
744 *arXiv preprint arXiv:1902.08708*, 2019.
- 745
- 746 Ke Sun, Yi Liu, Yingnan Zhao, Hengshuai Yao, Shangling Jui, and Linglong Kong. Exploring the
747 training robustness of distributional reinforcement learning against noisy state observations. *arXiv*
748 *preprint arXiv:2109.08776*, 2021.
- 749
- 750 Zhongchang Sun, Sihong He, Fei Miao, and Shaofeng Zou. Constrained reinforcement learning
751 under model mismatch. *arXiv preprint arXiv:2405.01327*, 2024.
- 752
- 753 Aviv Tamar, Shie Mannor, and Huan Xu. Scaling up robust MDPs using function approximation. In
754 *International conference on machine learning*, pp. 181–189. PMLR, 2014.
- 755
- 756 Kai Liang Tan, Yasaman Esfandiari, Xian Yeow Lee, and Soumik Sarkar. Robustifying reinforce-
757 ment learning agents via action space adversarial training. In *2020 American control conference*
758 *(ACC)*, pp. 3959–3964. IEEE, 2020.
- 759
- 760 Takumi Tanabe, Rei Sato, Kazuto Fukuchi, Jun Sakuma, and Youhei Akimoto. Max-min off-policy
761 actor-critic method focusing on worst-case robustness to model misspecification. *Advances in*
762 *Neural Information Processing Systems*, 35:6967–6981, 2022.

- 756 Chen Tessler, Yonathan Efroni, and Shie Mannor. Action robust reinforcement learning and appli-
757 cations in continuous control. In *International Conference on Machine Learning*, pp. 6215–6224.
758 PMLR, 2019.
- 759 Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control.
760 In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033.
761 IEEE, 2012. doi: 10.1109/IROS.2012.6386109.
- 762 Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu,
763 Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard
764 interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.
- 765 Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom
766 Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm_control: Software and tasks for
767 continuous control. *Software Impacts*, 6:100022, 2020.
- 768 Amos Tversky and Daniel Kahneman. Judgment under uncertainty: Heuristics and biases: Biases in
769 judgments reveal some heuristics of thinking under uncertainty. *science*, 185(4157):1124–1131,
770 1974.
- 771 Daniel Vial, Sanjay Shakkottai, and R Srikant. Robust multi-agent bandits over undirected graphs.
772 *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 6(3):1–57, 2022.
- 773 Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Juny-
774 oung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster
775 level in starcraft ii using multi-agent reinforcement learning. *nature*, 575(7782):350–354, 2019.
- 776 Maciej Wolczyk, Michal Zajac, Razvan Pascanu, Lukasz Kucinski, and Piotr Milos. Continual
777 world: A robotic benchmark for continual reinforcement learning. *Advances in Neural Informa-
778 tion Processing Systems*, 34:28496–28510, 2021.
- 779 Eric M Wolff, Ufuk Topcu, and Richard M Murray. Robust control of uncertain Markov decision
780 processes with temporal logic specifications. In *2012 IEEE 51st IEEE Conference on Decision
781 and Control (CDC)*, pp. 3372–3379. IEEE, 2012.
- 782 Zikang Xiong, Joe Eappen, He Zhu, and Suresh Jagannathan. Defending observation attacks in deep
783 reinforcement learning via detection and denoising. *arXiv preprint arXiv:2206.07188*, 2022.
- 784 Huan Xu and Shie Mannor. The robustness-performance tradeoff in markov decision processes.
785 *Advances in Neural Information Processing Systems*, 19, 2006.
- 786 Huan Xu and Shie Mannor. Distributionally robust Markov decision processes. *Mathematics of
787 Operations Research*, 37(2):288–300, 2012.
- 788 Tengyu Xu, Yingbin Liang, and Guanghui Lan. Crpo: A new approach for safe reinforcement
789 learning with convergence guarantee. In *International Conference on Machine Learning*, pp.
790 11480–11491. PMLR, 2021.
- 791 Huaxiu Yao, Caroline Choi, Bochuan Cao, Yoonho Lee, Pang Wei W Koh, and Chelsea Finn. Wild-
792 time: A benchmark of in-the-wild distribution shift over time. *Advances in Neural Information
793 Processing Systems*, 35:10309–10324, 2022.
- 800 Christopher Yeh, Victor Li, Rajeev Datta, Julio Arroyo, Nicolas Christianson, Chi Zhang, Yize
801 Chen, Mohammad Mehdi Hosseini, Azarang Golmohammadi, Yuanyuan Shi, et al. Sustainingym:
802 Reinforcement learning environments for sustainable energy systems. *Advances in Neural Infor-
803 mation Processing Systems*, 36, 2024.
- 804 Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The
805 surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information
806 Processing Systems*, 35:24611–24624, 2022.
- 807 Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey
808 Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning.
809 In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

- 810 Zhaocong Yuan, Adam W Hall, Siqi Zhou, Lukas Brunke, Melissa Greeff, Jacopo Panerati, and
811 Angela P Schoellig. Safe-control-gym: A unified benchmark suite for safe learning-based control
812 and reinforcement learning in robotics. *IEEE Robotics and Automation Letters*, 7(4):11142–
813 11149, 2022.
- 814 Amy Zhang, Yuxin Wu, and Joelle Pineau. Natural environment benchmarks for reinforcement
815 learning. *arXiv preprint arXiv:1811.06032*, 2018.
- 816
- 817 Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning
818 invariant representations for reinforcement learning without reconstruction. In *International Con-
819 ference on Learning Representations*, 2021a. URL [https://openreview.net/forum?
820 id=-2FCwDKRREu](https://openreview.net/forum?id=-2FCwDKRREu).
- 821 Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui Hsieh.
822 Robust deep reinforcement learning against adversarial perturbations on state observations. *Ad-
823 vances in Neural Information Processing Systems*, 33:21024–21037, 2020.
- 824
- 825 Huan Zhang, Hongge Chen, Duane S Boning, and Cho-Jui Hsieh. Robust reinforcement learning
826 on state observations with learned optimal adversary. In *International Conference on Learning
827 Representations*, 2021b. URL <https://openreview.net/forum?id=sCZbhBvqQaU>.
- 828 Zhengfei Zhang, Kishan Panaganti, Laixi Shi, Yanan Sui, Adam Wierman, and Yisong Yue. Dis-
829 tributionally robust constrained reinforcement learning under strong duality. *arXiv preprint
830 arXiv:2406.15788*, 2024.
- 831
- 832 Zhili Zhang, Yanchao Sun, Furong Huang, and Fei Miao. Safe and robust multi-agent reinforce-
833 ment learning for connected autonomous vehicles under state perturbations. *arXiv preprint
834 arXiv:2309.11057*, 2023.
- 835 Ziyuan Zhou and Guanjun Liu. Robustness testing for multi-agent reinforcement learning: State
836 perturbations on critical agents. *arXiv preprint arXiv:2306.06136*, 2023.
- 837
- 838 Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiri-
839 any, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot
840 learning. *arXiv preprint arXiv:2009.12293*, 2020.
- 841 Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul
842 Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv
843 preprint arXiv:1909.08593*, 2019.
- 844 Adil Zouitine, David Bertoin, Pierre Clavier, Matthieu Geist, and Emmanuel Rachelson. Rrls: Ro-
845 bust reinforcement learning suite. *arXiv preprint arXiv:2406.08406*, 2024.
- 846
- 847
- 848
- 849
- 850
- 851
- 852
- 853
- 854
- 855
- 856
- 857
- 858
- 859
- 860
- 861
- 862
- 863

A RELATED WORKS

Related RL benchmarks. To the best of our knowledge, Zouitine et al. (2024) is the only existing benchmark designed specifically for robustness evaluations, with the same goal of this work. It introduced six continuous control tasks in Gymnasium-MuJoCo, designed to address environmental shifts. A clear lack of standardized benchmarks is present that offer a wide range of diverse tasks and account for uncertainty and disruptions over multiple stages throughout the interaction process, (not only the environment). Such a comprehensive evaluation platform is essential for the community to evaluate existing efforts and inspire new algorithms. Robust-Gymnasium fills the gaps for robust evaluation of RL as a unified modular benchmark that supports over sixty diverse tasks in robotics and control for comprehensive evaluation, and accounting for different types of uncertainty and disruptions across multiple stages of the interaction process.

Moreover, enhancing robustness against environment shifts can be seen as a slight generalization to unseen tasks or environments. A non-exhaustive list of relevant benchmarks includes: a domain generalization benchmark in offline RL (Mediratta et al., 2023), Meta-World for meta-RL (Yu et al., 2020), a generalization benchmark for robot manipulation (Pumacay et al., 2024), SustainGym — generalization for sustainable energy systems (Yeh et al., 2024), continual robot learning (Wolczyk et al., 2021), lifelong robot learning (Liu et al., 2024), skill manipulation robot learning (Mu et al., 2021), safe RL (Ray et al., 2019; Yuan et al., 2022), multi-task RL (Mendez et al., 2022), human-robot collaboration tasks (Puig et al., 2024), dynamic algorithm configuration (Eimer et al., 2021), RL in JAX (Bonnet et al., 2024), procedurally generated environments (Küttler et al., 2020), DM control (Tunyasuvunakool et al., 2020), arcade learning environments (Bellemare et al., 2013), and others (Marklund et al., 2020; Yao et al., 2022).

RL works involving tasks for robust evaluation. Although not primarily focusing on building a benchmark for robust RL, there exists a lot of prior works or benchmarks that involves tasks for robust evaluation. While they typically support a few robust evaluation tasks associated with only one disruption type, which is not sufficient for comprehensive evaluations for robustness in real-world applications.

Specifically, there exists a lot of benchmarks for different RL problems, such as standard RL, safe RL, multi-agent RL, offline RL, and etc. These benchmarks either don't have robust evaluation tasks, or only have a narrow range of tasks for robust evaluation (since robust evaluation is not their primary goals), such as Duan et al. (2016) support 5 tasks with robust evaluations in control. Besides, there are many existing robust RL works that involve tasks for robust evaluations, while they often evaluate one-off and a narrow range of tasks in specific domains, such as 8 tasks for robotics and control (Ding et al., 2023a), 9 robot and control tasks in StateAdvRL (Zhang et al., 2020), 5 robust RL tasks in RARL (Pinto et al., 2017), a 3D bin-packing task (Pan et al., 2023). Since their primary goal is to design robust RL algorithms, but not a platform to evaluate the algorithms.

Robustness in single-agent RL. Robustness is a key principle in designing RL algorithms, as training processes are often idealized and limited in data and scenarios, while real-world environments are changeable, unpredictable, and highly diverse. An emerging body of work focuses on developing robust RL algorithms that can withstand potential uncertainties, perturbations, and attacks during real-world execution. These efforts can largely be categorized under our unified robust RL framework (Sec. 2), which formulates uncertainty events affecting the agent-environment interaction as behaviors of three types of disruptors. Our proposed Robust-Gymnasium encompasses all types of robust RL tasks within this framework, providing a flexible and comprehensive platform for evaluating and developing robust RL algorithms.

Specifically, prior works typically involve one type of disruptors: Zhang et al. (2020; 2021b); Han et al. (2022); Qiaoben et al. (2021); Sun et al. (2021); Xiong et al. (2022) studied the uncertainty of agent's observed state, controlled by the observation-disruptor who can add restricted noise or perform adversarial attack; Tessler et al. (2019); Tan et al. (2020) considered the robustness w.r.t. the uncertainty of the action, where the action is possibly distorted by the action-disruptor abruptly or smoothly before forwarding to the environment to be executed; A large

amount of prior works focus on dealing with the perturbation/shift on the environmental controlled by the environment-disruptor — includes the reward function, the dynamics, or the task itself, ranging from theory (Iyengar, 2005; Xu & Mannor, 2012; Wolff et al., 2012; Kaufman & Schaefer, 2013; Ho et al., 2018; Smirnova et al., 2019; Ho et al., 2021; Goyal & Grand-Clement, 2022; Derman & Mannor, 2020; Tamar et al., 2014; Badrinath & Kalathil, 2021) to applications (Pinto et al., 2017; Pattanaik et al., 2017; Tanabe et al., 2022; Ding et al., 2023a). Besides them, only a few works consider more complex scenarios that more than one disruptors are involved (Mandlekar et al., 2017). See Moos et al. (2022) for a recent review.

Robustness in safe RL and multi-agent RL. Besides the class of standard single-agent RL, robustness in RL algorithms are ubiquitously demanded and has emerges a growing line of works for other RL problems such as partially observable Markov decision processes (POMDPs) (Cubuktepe et al., 2021), safe RL (Liu et al., 2022; Sun et al., 2024; Zhang et al., 2024) and multi-agent RL (Vial et al., 2022; Han et al., 2022; He et al., 2023; Zhou & Liu, 2023; Zhang et al., 2023; 2021b). Additional challenges arise when combining robustness requirements with issues such as safety constraints and strategic interactions, which are often understudied and lack standardized benchmarks for evaluation. Our Robust-Gymnasium not only provides single-agent RL tasks but also encompasses a broader range of RL paradigms, including safe RL and multi-agent RL. This enables a faster and more comprehensive process for designing and evaluating robust RL algorithms across a wider array of RL tasks.

B SUPPLEMENTARY EXPERIMENTS AND ANALYSIS

B.1 SUPPLEMENTARY FOR EVALUATION ROBUSTNESS OF STANDARD RL

As shown in Figures 10, they demonstrates the robustness of PPO in the HalfCheetah-v4 environment under various adversarial conditions. Each graph presents the average episode reward across training steps, contrasting the performance of the standard PPO algorithm against its adaptations under diverse adversarial attack parameters. Specifically, the figure for in-Training Attack on Reward (Figure 10 (a)) investigates how modifications to the rewards during training influence the learning performance, employing multiple levels of perturbation. Moreover, the graph for Post-Training Attack on Reward (Figure 10 (b)) assesses how the trained policy withstands alterations to the reward signals post-training. The experimental results suggest that training an RL agent with disturbances and then testing it in ideal environments may lead to improved reward performance in test scenarios. Similarly, we conducted an experiment to evaluate the robustness of another popular RL baseline, SAC. As shown in Figure 11, the performance of SAC degrades under a disturbance attack.

This experiment aids in understanding the stability and robustness of RL policies under adversarial conditions, which is pivotal for deploying these models in real-world scenarios where they may encounter unexpected or adversarial changes in input data.

B.2 SUPPLEMENTARY FOR EVALUATION ROBUSTNESS OF SAFE RL

As depicted in Figures 12(a) and (b), we implement PCRPO (Gu et al., 2024) and CRPO (Xu et al., 2021), SOTA safe RL algorithms, in robust safety-critical tasks. We selected a representative task from robust safe RL to assess the effectiveness of the safe RL algorithm. Specifically, we introduce a disruptor to attack the Walker2d robot’s observations during training, as shown in Figures 12(a)-(b). Under these adversarial attacks, the reward performance of both PCRPO and CRPO degrades. The attacks follow a Gaussian distribution with a mean of 0 and standard deviation of 0.3, highlighting the importance of considering disturbance testing before deploying safe RL models in real-world applications.

B.3 SUPPLEMENTARY FOR EVALUATION ROBUSTNESS OF MULTI-AGENT RL

As shown in Figures 13 (d), (e), and (f), we investigate partial state, action, and reward attacks on MAPPO, where only a subset of agents or aspects is attacked. These figures show a smaller drop in performance, indicating partial attacks are less harmful compared to full attacks (See Figure 8).

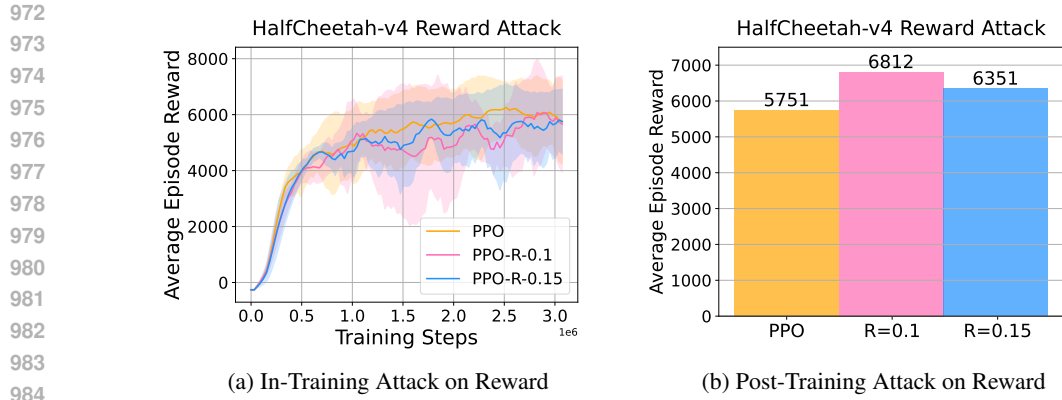


Figure 10: HalfCheetah-v4 robustness: training attack, reward. Specifically, in experiment (a), we train the PPO algorithm under conditions: without a reward attack, and with a reward attack involving Gaussian noise with standard deviations of 0.1 and 0.5, respectively. In both reward attack scenarios, the noise has a mean of 0, with attack noise standard deviations of 0.1 and 0.15, respectively. In experiment (b), we test the trained PPO models that are attacked during training with reward attacks, using standard deviations of 0.1 and 0.15. After the attack-based training, the models are evaluated in environments without any attacks.

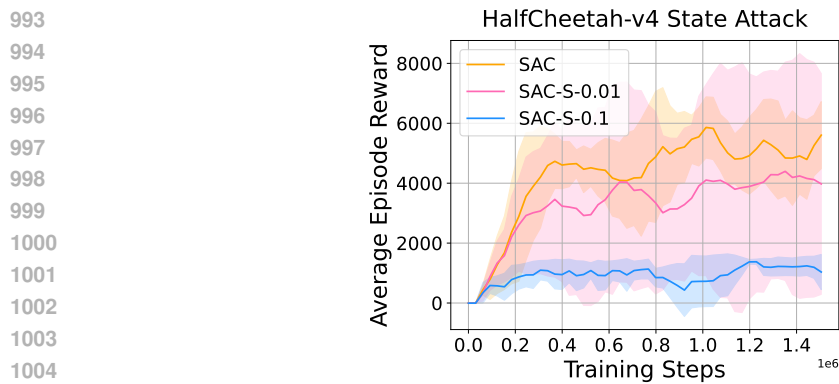


Figure 11: Evaluation SAC robustness on HalfCheetah-v4 tasks.

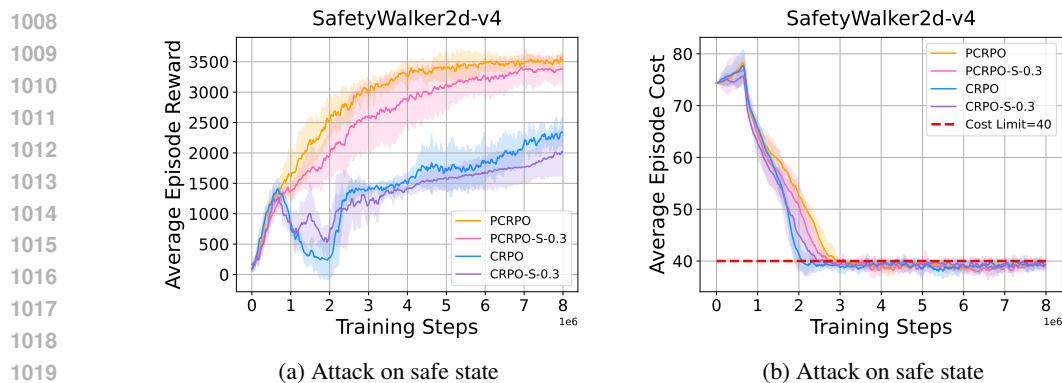


Figure 12: Robust Safety RL Tasks.

B.4 FREQUENCY ATTACK

We offer interactive modes that support step-wise, variable interactions between disruptors, agents, and environments, allowing users to apply perturbations at any point in time and in any manner they

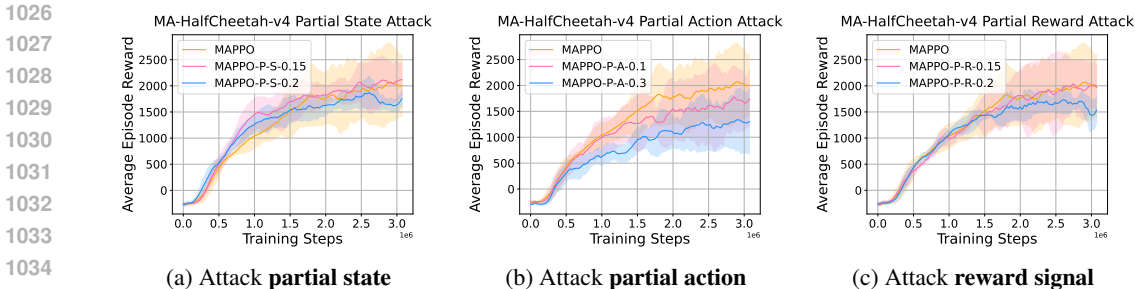


Figure 13: Multi-Agent HalfCheetah-2x3 robustness: training attack on state, action, and reward for all the two agents. S denotes state, A denotes action and R denotes reward, P denotes partial attacks. Some of agents are attacked with various attack factors.

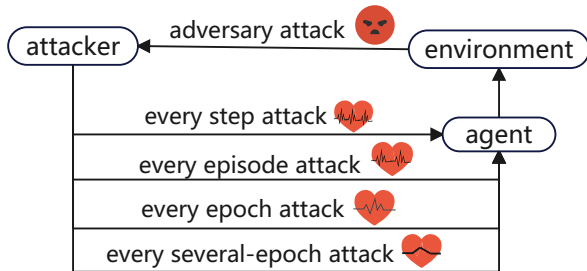


Figure 14: Different levels of robust RL’s attack frequency.

choose. As shown in Figure 14, the frequency of attacks on tasks is illustrated. Perturbations can occur at various points during the training and testing phases, with different frequencies.

As shown in Figure 15, we provide the results of robustness evaluations on the Ant-v4 task under frequency-based adversarial attacks. The figure consists of two subplots, each examining the performance of PPO-based algorithms under different attack levels and frequencies. In Figure (a), we explore the impact of varying attack intensities at a fixed attack frequency (every 50 steps) targeting the agent’s actions. As shown, PPO without adversarial intervention achieves the highest episode rewards. However, as the attack intensity increases (PPO-F50-A-0.01, PPO-F50-A-0.05, PPO-F50-A-0.1), the performance declines progressively. The highest intensity attack (PPO-F50-A-0.1) results in the most significant reduction in rewards, indicating a substantial performance drop under stronger attacks. In Figure (b), we examine the effect of varying attack frequencies while keeping the attack intensity constant. Here, PPO-F50-S-0.15 and PPO-F100-S-0.15 represent attacks occurring every 50 and 100 steps, respectively. The results indicate that more frequent attacks (PPO-F50-S-0.15) lead to a larger decline in episode rewards compared to less frequent attacks (PPO-F100-S-0.15). This suggests that attack frequency plays a critical role in determining the robustness of PPO algorithms. Overall, these findings demonstrate that both the intensity and frequency of attacks significantly affect the performance of RL agents, with higher intensities and more frequent attacks causing greater degradation in task performance.

C OTHER SETTINGS OF THE FRAMEWORK

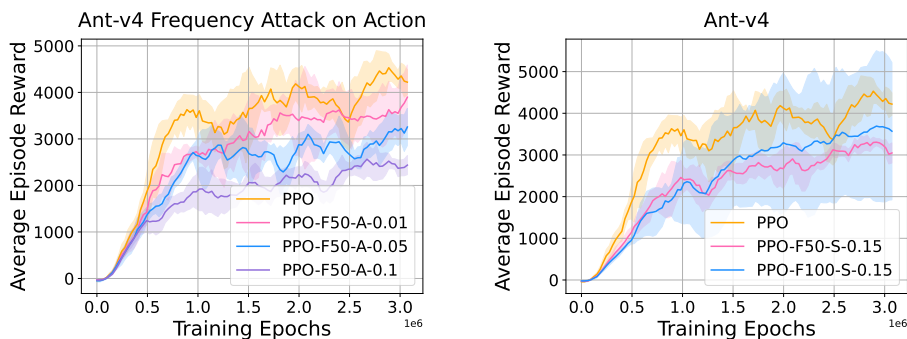
C.1 BENCHMARK FEATURES

The features of the benchmark are as follows:

High Modularity: It is designed for flexible adaptation to a variety of research needs, featuring high modularity to support a wide range of experiments.

Wide Coverage of : It provides a comprehensive set of tasks to evaluate robustness across different RL scenarios. An overview of the task list is shown in Figure 17.

1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133



(a) Different level of attacks with same frequency (b) Different level of frequency with same attack

Figure 15: Robust Ant Tasks with Frequency attacks.

High Compatibility: It can be seamless and compatible with a wide range of existing environments. An example is shown in Listing 1. Moreover, this benchmark supports vectorized environments, which means it can be useful to enable parallel processing of multiple environments for efficient experimentation.

```

1 from robust_gymnasium.configs.robust_setting import get_config
2 args = get_config().parse_args()
3 action = env.action_space.sample()
4 robust_input = {"action": action, "robust_config": args}
5 observation, reward, terminated, truncated, info = env.step(robust_input)

```

Listing 1: An example of python interface

Support for New Gym API: It fully supports the latest standards in Gym API, facilitating easy integration and expansion.

Adversarial Attack with LLMs: We feature an approach that leverages LLMs as adversary policies. An example is shown in Listing 2.

```

1 prompt = "This is about a robust reinforcement learning setting; we want
2 you as an adversary policy. If the current reward exceeds the
3 previous reward value, please input some observation noise to disturb
4 the environment and improve the learning algorithm's robustness." "
5 The noise should be in this area:" +str((args.region_low, args.
6 region_high))+ ", the current reward:" + str(reward) + ", the
7 previous reward is" + str(self.previous_reward) + "please slightly
8 revise the current environment state values:" + str(observation) + ",
9 just output the revised state with its original format" "do not
10 output any other things."
11 prompt_state = gpt_call(prompt)
12 observation = prompt_state

```

Listing 2: An example of LLMs for robust learning

C.2 ROBUST NON-STATIONARY TASKS:

Inspired by OMPO (Luo et al., 2024), we provide various task settings to evaluate policy robustness, as illustrated in Figure 16. During policy learning, we introduce adversarial attacks during walking or running tasks by altering robot dynamics and environmental conditions. For instance, we stochastically adjust the robot’s gravity and the environment’s wind speed, introducing uncertain disturbances during policy learning. Additionally, we stochastically modify the robot’s physical shape throughout the learning process to test and enhance policy robustness.

Specifically, in non-stationary Ant-v5 Tasks, during each step, we introduce noise into the agent’s dynamics by attacking factors like the Ant robot’s gravity and the wind speed in the robot’s environment. As demonstrated in Equation (2) for attacks at initial and training steps, we introduce

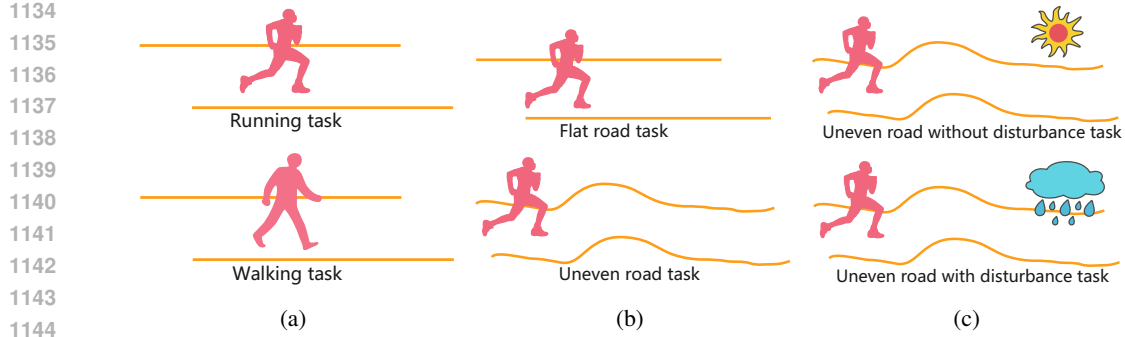


Figure 16: Examples of robust non-stationary tasks (Luo et al., 2024).

1148 deterministic perturbations to the Ant robot, such as variations in gravity and environmental wind
1149 speed, the pseudo code is shown in Listing 3. Furthermore, Equation (3) is for initial noise, and
1150 Equation (4) is for noise during training we use these Equarions to consider the incorporation of
1151 stochastic disturbances into the Ant robot model, again including factors like gravity fluctuations
1152 and wind speed variations, the pseudo code is shown in Listing 4. Apart from wind and gravity
1153 disturbances, we also investigate the robot shape disturbances during policy learning, as shown in
1154 Equations (5)-(8), and an example of pseudo code is shown in Listing 5.

1155 At the initial and training steps, if we choose non-stationary attack as deterministic noise,

$$1156 \text{ Ant deterministic noise} = \begin{cases} \text{Gravity} = 14.715, \\ \text{Wind} = 1.0. \end{cases} \quad (2)$$

1157 if we choose non-stationary attack as stochastic noise,

$$1162 \text{ Ant and Humanoid stochastic noise at initial steps} = \begin{cases} \text{Gravity} \sim \text{Uniform}(9.81, 19.82), \\ \text{Wind} \sim \text{Uniform}(0.8, 1.2). \end{cases} \quad (3)$$

1165 During training steps, if we choose non-stationary attack as stochastic noise, where i_{episode} denotes
1166 the training step number,

$$1168 \text{ Ant and Humanoid noise during training} = \begin{cases} \text{Gravity} = 14.715 + 4.905 \cdot \sin(0.5 \cdot i_{\text{episode}}), \\ \text{Wind} = 1.0 + 0.2 \cdot \sin(0.5 \cdot i_{\text{episode}}). \end{cases} \quad (4)$$

$$1171 \text{ Walker stochastic noise at initial steps} = \begin{cases} \text{Torso Length} \sim \text{Uniform}(0.1, 0.3), \\ \text{Foot Length} \sim \text{Uniform}(0.05, 0.15). \end{cases} \quad (5)$$

$$1174 \text{ Walker Stochastic noise} = \begin{cases} \text{Torso Length} = 0.2 + 0.1 \sin(0.3 \cdot i_{\text{episode}}) \\ \text{Foot Length} = 0.1 + 0.05 \sin(0.3 \cdot i_{\text{episode}}) \end{cases} \quad (6)$$

$$1177 \text{ Hopper stochastic noise at initial steps} = \begin{cases} \text{Torso Length} \sim \text{Uniform}(0.3, 0.5), \\ \text{Foot Length} \sim \text{Uniform}(0.29, 0.49). \end{cases} \quad (7)$$

$$1179 \text{ Walker Stochastic noise} = \begin{cases} \text{Torso Length} = 0.4 + 0.1 \cdot \sin(0.2 \cdot i_{\text{episode}}), \\ \text{Foot Length} = 0.39 + 0.1 \cdot \sin(0.2 \cdot i_{\text{episode}}). \end{cases} \quad (8)$$

```
1182 1 if config.deter_noise:
1183 2     gravity = 14.715
1184 3     wind = 1.
1185 4 else:
1185 5     gravity = np.random.uniform(9.81, 19.82)
1186 6     wind = np.random.uniform(0.8, 1.2)
```

Listing 3: An example of Non-stationary Ant python code for initial steps.

```

1188 1 if config.deter_noise:
1189 2     gravity = 14.715
1190 3     wind = 1.
1191 4 else:
1192 5     gravity = 14.715 + 4.905 * np.sin(0.5 * i_episode)
1193 6     wind = 1. + 0.2 * np.sin(0.5 * i_episode)

```

Listing 4: An example of Non-stationary Ant python code for training steps.

```

1203 1 if config.deter_noise:
1204 2     torso_len = 0.2
1205 3     foot_len = 0.1
1206 4 else:
1207 5     torso_len = 0.2 + 0.1 * np.sin(0.3 * i_episode)
1208 6     foot_len = 0.1 + 0.05 * np.sin(0.3 * i_episode)

```

Listing 5: An example of Non-stationary Walker python code for training steps.

D REPRESENTATIVE EXAMPLES OF USING Robust-Gymnasium

In this section, we present an overview of the task environments, as illustrated in Figure 17. Additionally, we show some robustness-focused tasks, detailed in Tables 1-8.

Moreover, inspired by (Yu et al., 2020), to illustrate the standardized usage of our benchmark, we propose the following framework for evaluation settings. These align with the principles of benchmarking, including standardized performance metrics and evaluation protocols:

- **Random attack (Easy) → Adversarial attack (Hard).** Random Attack (Easy): Random noise, drawn from distributions such as Gaussian or uniform, is added to the nominal variables. This mode is applicable to all sources of perturbation and allows for testing robustness under stochastic disturbances, e.g., see Figure 5 (a) and (b). Adversarial Attack (Hard): An adversarial attacker selects perturbations to adversely degrade the agent’s performance. This mode can be applied to observation or action perturbations and represents the most challenging scenario, e.g., see Figure 9 (a) and (b).
- **Low state-action dimensions (Easy) → High state-action dimensions (Hard)** As the state and action space dimensions increase, the tasks become significantly more challenging. The difficulty level of tasks typically progresses from Box2D, Mujoco tasks, robot manipulation, and safe tasks to multi-agent and humanoid tasks. For instance, the Humanoid task, with a 51-dimensional action space and a 151-dimensional state space, is substantially more challenging than the Mujoco Hopper task, which has a 3-dimensional action space and an 11-dimensional state space.


Class of Tasks \ disturbance types			Observed state/ Observed reward/ Action		Environment
			random	Adversarial (LLM)	Dynamics or disturbance
Single-agent	Control	Box2D			
		Gymnasium-MuJoCo			<input checked="" type="checkbox"/>
	Robot Navigation	Maze			
	Robot Manipulation (diverse robots and tasks)	Dexterous Hand			
		Adroit Hand			
		Fetch Manipulation			
		Franka Kitchen			
		robosuite			<input checked="" type="checkbox"/>
	Humanoid				
Safety	Safety MuJoCo				
Multi-agent	MAMuJoCo				

Figure 17: An overview of task environments and supported disruptions in Robust-Gymnasium.

Table 1: A List of Examples for Robustness in MuJoCo Tasks

Tasks \ Robust type	Robust State	Robust Action	Robust Reward	Robust Dynamics
Ant-v2-v3-v4-v5	✓	✓	✓	✓
HalfCheetah-v2-v3-v4-v5	✓	✓	✓	✓
Hopper-v2-v3-v4-v5	✓	✓	✓	✓
Walker2d-v2-v3-v4-v5	✓	✓	✓	✓
Swimmer-v2-v3-v4-v5	✓	✓	✓	✓
Humanoid-v2-v3-v4-v5	✓	✓	✓	✓
HumanoidStandup-v2-v3-v4-v5	✓	✓	✓	✓
Pusher-v2-v3-v4-v5	✓	✓	✓	✓
Reacher-v2-v3-v4-v5	✓	✓	✓	✓
InvertedPendulum-v2-v3-v4-v5	✓	✓	✓	✓

Table 2: A List of Examples for Robustness in Box2d Tasks

Tasks \ Robust Type	Robust State	Robust Action	Robust Reward
CarRacing-v2	✓	✓	✓
LunarLanderContinuous-v3	✓	✓	✓
BipedalWalker-v3	✓	✓	✓
LunarLander-v3 (Discrete Task)	✓	✓	✓

E EXPERIMENT SETTINGS

We deploy several SOTA baselines in our benchmark to evaluate their robustness across various challenging scenarios. The implementation parameters associated with these methods are provided in Tables 9-13.

Since RL performance can be significantly influenced by different random seeds (Henderson et al., 2018; Colas et al., 2018), we aim to balance computational costs and experimental rigor by typically using 3–5 seeds in our experiments. For single-agent settings, we use the same 3 seeds across all baselines to ensure a fair comparison. In multi-agent settings, where variance tends to be higher, we employ the same 5 seeds across all baselines to achieve a more reliable evaluation. We recognize the importance of robust experimental evaluation and intend to include additional seeds in future studies to further examine RL robustness.

Table 3: A List of Examples for Robustness in Robosuite Tasks

Tasks \ Robust Type	Robust State	Robust Action	Robust Reward
Lift	✓	✓	✓
Door	✓	✓	✓
NutAssembly	✓	✓	✓
PickPlace	✓	✓	✓
Stack	✓	✓	✓
Wipe	✓	✓	✓
ToolHang	✓	✓	✓
TwoArmLift	✓	✓	✓
TwoArmPegInHole	✓	✓	✓
TwoArmHandover	✓	✓	✓
TwoArmTransport	✓	✓	✓
MultiDoor	✓	✓	✓

Table 4: A List of Examples for Robustness in Safety Tasks

Tasks \ Robust Type	Robust State	Robust Action	Robust Reward
SafetyAnt-v4	✓	✓	✓
SafetyHalfCheetah-v4	✓	✓	✓
SafetyHopper-v4	✓	✓	✓
SafetyWalker2d-v4	✓	✓	✓
SafetySwimmer-v4	✓	✓	✓
SafetyHumanoid-v4	✓	✓	✓
SafetyHumanoidStandup-v4	✓	✓	✓
SafetyPusher-v4	✓	✓	✓
SafetyReacher-v4	✓	✓	✓

Moreover, when selecting different robust disturbance parameters, the choice can significantly affect the evaluation of various RL algorithms. For instance, in standard RL, disturbances can be modeled as Gaussian distributions, such as $\mathcal{N}(0, 0.1)$ or $\mathcal{N}(0, 0.15)$, applied to the state or action space, which can notably influence the performance of algorithms like PPO. Alternatively, uniform disturbances within the range $[0.2, 0.8]$ can be used to effectively assess the robustness of standard RL approaches. For robust RL, additional parameters are often employed to evaluate algorithm robustness. For example, as for the evaluation robustness of MOPO method, wind speed may follow a uniform distribution $U(0.8, 1.2)$, while robot gravity may vary uniformly within $U(9.81, 19.82)$. Other factors include variations in the robot’s physical dimensions, such as the torso length, which can be expressed as the original length plus $0.1 \sin(0.2 \cdot \text{iteration number})$, and the foot length, which follows a similar perturbation. Our benchmark also incorporates robust parameters to evaluate the safety of RL algorithms. For example, Gaussian disturbances $\mathcal{N}(0, 0.3)$ are particularly effective for assessing the robustness of safe RL algorithms such as PCRPO and CRPO. In the context of multi-agent RL, robustness can be evaluated by selectively perturbing partial agents. Gaussian disturbances, such as $\mathcal{N}(0, 0.1)$ or $\mathcal{N}(0, 0.15)$, applied to the state or action space, can provide significant insights into the robustness of algorithms like MAPPO and IPPO.

Table 5: A List of Examples for Robustness in Adroit Hand Tasks

Tasks \ Robust Type	Robust State	Robust Action	Robust Reward
AdroitHandDoor-v1	✓	✓	✓
AdroitHandHammer-v1	✓	✓	✓
AdroitHandPen-v1	✓	✓	✓
AdroitHandRelocate-v1	✓	✓	✓

Table 6: A List of Examples for Robustness in Hand Manipulation Tasks

Tasks \ Robust Type	Robust State	Robust Action	Robust Reward
HandManipulateEgg_BooleanTouchSensors-v1	✓	✓	✓
HandReach-v2	✓	✓	✓
HandManipulateBlock-v1	✓	✓	✓
HandManipulateEgg-v1	✓	✓	✓
HandManipulatePen-v1	✓	✓	✓

Table 7: A List of Examples for Robustness in Fetch Manipulation Tasks

Tasks \ Robust Type	Robust State	Robust Action	Robust Reward
FetchPush-v3	✓	✓	✓
FetchReach-v3	✓	✓	✓
FetchSlide-v3	✓	✓	✓
FetchPickAndPlace-v3	✓	✓	✓

Table 8: A List of Examples for Robustness in Multi-Agent Tasks

Tasks \ Robust Type	Robust State	Robust Action	Robust Reward
MA-Ant-2x4, 2x4d, 4x2, 4x1	✓	✓	✓
MA-HalfCheetah-2x3, 6x1	✓	✓	✓
MA-Hopper-3x1	✓	✓	✓
MA-Walker2d-2x3	✓	✓	✓
MA-Swimmer-2x1	✓	✓	✓
MA-Humanoid-9—8	✓	✓	✓
MA-HumanoidStandup-v4	✓	✓	✓
MA-Pusher-3p	✓	✓	✓
MA-Reacher-2x1	✓	✓	✓
Many-MA-Swimmer-10x2, 5x4, 6x1, 1x2	✓	✓	✓
Many-MA-Ant-2x3, 3x1	✓	✓	✓
CoupledHalfCheetah-p1p	✓	✓	✓

Parameters	Value	Parameters	Value
buffer size	4096	hidden size	[64, 64]
lr	3e-4	gamma	0.99
epoch	100	steps per epoch	30000
steps per collect	2048	repeat per collect	10
batch size	64	training num	8
testing num	10	rew norm	True
vf coef	0.25	ent coef	0.0
gae lambda	0.95	bound action clip	clip
lr decay	True	max grad norm	0.5
eps clip	0.2	dual clip	None
value clip	0	norm adv	0
recompute adv	0		

Table 9: Parameter values used for PPO (Schulman et al., 2017), MAPPO (Yu et al., 2022) and IPPO (De Witt et al., 2020) in experiments.

1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414

Parameters	Value	Parameters	Value
buffer size	4096	hidden size	[64, 64]
actor lr	1e-3	critic lr	1e-3
gamma	0.99	tau	0.005
alpha	0.0.2	auto alpha	False
epoch	100	steps per epoch	30000
steps per collect	2048	update per step	1
start time step	10000	n step	1
batch size	64	training num	8
testing num	10		

1415
1416

Table 10: Parameter values used for SAC (Haarnoja et al., 2018) in the experiment.

1417
1418
1419
1420

Parameters	Value	Parameters	Value
start steps	5000	num steps	300000
eval	True	eval episode	10
eval times	10	local reply size	1000
gamma	0.99	tau	0.005
lr	3e-4	alpha	0.2
batch size	256	update per step	3
target update interval	2	hidden size	256
gail batch	256	exponent	1.5
tomac alpha	1e-3	reward max	1

1421
1422
1423
1424
1425
1426
1427
1428
1429
1430

Table 11: Parameter values used for OMPO (Luo et al., 2024) in non-stationary MuJoCo experiments.

1431
1432
1433
1434
1435
1436

Parameters	Value	Parameters	Value
image obs	False	actor lr	3e-4
critic lr	1e-3	gamma	0.99
tau	5e-3	alpha	0.1
auto alpha	True	alpha lr	3e-4
hidden size	[256, 256, 256]	n steps	4
buffer size	1e6	step per epoch	1e4
step per collect	20	batch size	128
start time step	0	exploration noise	0
horizon	300	camera	agentview
height	128	width	128
encoder type	mlp	training num	10
test num	10	sigma	0.01
bound	0.01	augmented ratio	0.5
vae sigma	1.0	control frequency	20

1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452

Table 12: Parameter values used for RSC (Ding et al., 2024) in the causaldoor/causalift experiments; for DBC (Zhang et al., 2021a), based on above parameters, transition model type is probabilistic, encoder feature dim is 256, encoder lr is 1e-4, decoder lr is 1e-4, bisim coef is 0.5, log std min is -10, log std max is 2; for ATLA (Zhang et al., 2021b), policy update max is 100, adv update max is 100, and adv eps is 0.01.

1453
1454
1455
1456
1457

1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511

Parameters	Value	Parameters	Value
gamma	0.995	hidden layer dim	64
cost limit	0.04	slack bound	5e-3
exploration iteration	40	epoch	500
tau	0.97	l2 reg	1e-3
max kl	1e-2	damping	1e-1
batch size	150000	gradient wr	0.4
gradient wc	0.6		

Table 13: Parameter values used for PCRPO (Gu et al., 2024) and CRPO (Xu et al., 2021) in the safety experiments.