ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

BALANCING MIXED ADVERSARIAL PERTURBATIONS: TOWARDS ROBUST REINFORCEMENT LEARNING

Ph.D. THESIS

Mustafa ERDEM

Department of Mechatronics Engineering

Mechatronics Engineering Programme

DECEMBER 2025

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

BALANCING MIXED ADVERSARIAL PERTURBATIONS: TOWARDS ROBUST REINFORCEMENT LEARNING

Ph.D. THESIS

Mustafa ERDEM
(518192006)

Department of Mechatronics Engineering

Mechatronics Engineering Programme

Thesis Advisor: Assoc. Prof. Dr. Nazım Kemal ÜRE

DECEMBER 2025

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

KARMA ÇEKİŞMELİ BOZULMALARI DENGELEME: GÜRBÜZ PEKİŞTİRMELİ ÖĞRENMEYE DOĞRU

DOKTORA TEZİ

Mustafa ERDEM
(518192006)

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

Tez Danışmanı: Doç Dr. Nazım Kemal ÜRE

ARALIK 2025

Mustafa ERDEM, a Ph.D. student of ITU Graduate School student ID 518192006 successfully defended the thesis entitled "BALANCING MIXED ADVERSARIAL PERTURBATIONS: TOWARDS ROBUST REINFORCEMENT LEARNING", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Assoc. Prof. Dr. Nazım Kemal ÜRE** Istanbul Technical University	.....
Jury Members :	**Prof. Dr. Metin GÖKAŞAN** Istanbul Technical University	.....
	Prof. Dr. Tuba ÇONKA YILDIZ Marmara University Fraunhofer-Institut für Integrierte Schaltungen IIS	.....
	Prof. Dr. Behçet Uğur TÖREYİN Istanbul Technical University	.....
	Prof. Dr. Ovsanna Seta BOGOSYAN City College of New York	.....

Date of Submission : 12 November 2025
Date of Defense : 4 December 2025

To my family,

FOREWORD

I begin by showing my complete gratitude to my supervisor, Assoc. Prof. Dr. Nazım Kemal ÜRE, for his invaluable guidance, continuous support, and constructive feedback, all of which have been instrumental in the development of this work.

I am profoundly grateful to my parents and sisters for their unwavering support and belief in me. Their support has provided me with enduring motivation throughout my time at every stage of my academic journey.

I also wish to acknowledge my colleagues and friends, whose companionship and encouragement have provided both motivation and balance throughout the challenges and accomplishments of this journey.

Above all, I would like to express my heartfelt thanks to my beloved wife, Gülşah YİĞİT ERDEM. Her love, support, and tireless belief in me have been my greatest strength, lifting me whenever I lost hope.

I gratefully acknowledge the support of the Scientific Research Projects Department of Istanbul Technical University under Project Number MDK-2022-43798 and the National Center for High Performance Computing of Türkiye (UHeM) for providing computing resources under Grant No. 4023492025.

December 2025

Mustafa ERDEM
(Mechatronics Engineer)

TABLE OF CONTENTS

ABBREVIATIONS

ARL	: Adversarial Reinforcement Learning
ASA	: Adversarial Synthesis and Adaptation
ASA-MDP	: Action and State-Adversarial Markov Decision Process
ASA-PPO	: Action and State-Adversarial PPO
ASR	: Attack Success Rate
ATLA-PPO	: Alternating Training with Learned Adversaries PPO
CMDP	: Constrained Markov Decision Process
DR	: Domain Randomization
DRL	: Deep Reinforcement Learning
FGSM	: Fast Gradient Sign Method
GAE	: Generalized Advantage Estimation
MARL	: Multi-Agent Reinforcement Learning
MDP	: Markov Decision Process
MLP	: Multi-Layer Perceptron
NA-PPO	: Noise-Augmented PPO
N-HYBRID	: Naive Hybrid
NR-MDP	: Noisy Action Robust Markov Decision Process
NR-PPO	: Non-Robust PPO
PGD	: Projected Gradient Descent
POMDP	: Partially Observable Markov Decision Process
PPO	: Proximal Policy Optimization
RARL	: Robust Adversarial Reinforcement Learning
RL	: Reinforcement Learning
RMDP	: Robust Markov Decision Process
SA-MDP	: State-Adversarial Markov Decision Process
tanh	: Hyperbolic tangent activation function
WocaR	: Worst-Case Aware Reinforcement Learning
ZOO	: Zeroth Order Optimization

SYMBOLS

$\mathbf{S}$	: Set of all states
$\mathbf{A}$	: Set of all actions
$\mathbf{P}$	: Transition probability function
$\mathbf{R}$	: Reward function
γ	: discount factor
s_t	: Current state
a_t	: Current action
r_t	: Immediate reward
s_{t+1}	: Next state
π	: Policy
τ	: A trajectory of states, actions and rewards
$\mathbf{V^{\pi}(s)}$	: State value function
$\mathbf{Q^{\pi}(s,a)}$	: Action value function
$\mathbf{A^{\pi}(s,a)}$	: Advantage function
$\delta_{\mathbf{t}}$	: Temporal Difference (TD) error
λ	: GAE bias-variance parameter
μ	: Mean
σ	: Standard deviation
θ	: Policy network parameters
$\mathbf{J}(\theta)$	: Policy objective function
α	: Learning rate
$\mathbf{r_t}(\theta)$	: New vs. old policy ratio
ε	: PPO policy update constraint
$\mathbf{L}(\theta)$	: Loss function
$\mathbf{L^{CLIP}}(\theta)$	: PPO clipped objective
$\mathbf{L^{VF}}(\theta)$	: Value function error
$\mathbf{H}$	: Entropy term
$\mathbf{c_1}$,$\mathbf{c_2}$	: PPO loss term coefficients
$\mathbf{A_1}$,$\mathbf{A_2}$	: Action sets for two players
$\pi_{\mathbf{p}}$	: Protagonist (main agent) policy
$\pi_{\mathbf{a}}$	: Adversary (attacker) policy
ϕ	: Adversary policy parameters
η	: Action attack scale
κ	: State attack scale
$\bar{a}_t^a$	: Raw action perturbation action
$\tilde{a}_t$	: Final perturbed action
$\bar{a}_t^s$	: Raw state perturbation action
$\tilde{s}_t$	: Final perturbed state
∇	: Gradient
$\mathbb{E}$	: Expectation

LIST OF TABLES

Page

LIST OF FIGURES

Page

BALANCING MIXED ADVERSARIAL PERTURBATIONS: TOWARDS ROBUST REINFORCEMENT LEARNING

SUMMARY

Deep Reinforcement Learning (DRL) has proven itself as an effective method to train autonomous agents which learn sequential decision-making through complex environment interactions. The combination of deep neural networks with reinforcement learning decision-making framework in DRL has led to outstanding achievements in game playing, robotics and autonomous systems. The agents discover their best policies through reward maximization which allows them to operate in complex state and action spaces that were previously impossible to handle. The deployment of DRL systems in safety-critical real-world applications requires researchers to focus on developing methods which protect these systems from adversarial attacks and unpredictable environmental changes.

The impressive capabilities of DRL agents do not protect them from adversarial perturbations which are small input modifications that result in complete breakdowns of decision-making systems. The research community has thoroughly investigated these system weaknesses in supervised learning before expanding their analysis to reinforcement learning. The methods of adversarial training, domain randomization, contextual RL and robust MDPs serve as solutions to protect DRL systems from vulnerabilities.

Conversely, other methods that strive to achieve strong and generalizable policies have certain limitations compared to adversarial training. Domain randomization needs human intervention to determine environmental parameter ranges which might fail to detect essential boundary situations and faces challenges when handling complex high-dimensional spaces while requiring expensive computational resources. On the other hand, agents trained with adversarial training detect policy weaknesses through automatic challenge generation without needing pre-defined context information. Contextual RL depends on well-defined, observable context variables, limiting its ability to generalize to unseen or ambiguous contexts, whereas adversarial training actively probes policy weaknesses without needing predefined contexts. Robust RL, while designed for worst-case scenarios, often produces overly conservative policies that underperform in typical conditions and relies on predefined uncertainty sets that may not capture real-world variability. Adversarial training, by contrast, dynamically exposes vulnerabilities through optimized adversarial examples, though it risks overfitting to specific attacks if not carefully balanced.

Modern adversarial training methods used to improve adversarial robustness include feeding adversarial inputs to agents in the training process. These approaches focus on changes in one area- whether in state observations, actions, rewards, or environmental

processes. The single-type defense approaches prove successful for their designated domains yet they do not handle the multiple perturbation sources which appear in actual environments.

The primary limitation of current adversarial training methods is their narrow focus on single-modal perturbations. Real-world systems, such as autonomous drones or industrial robots face multiple disturbances that affect their perception systems through sensor noise and their control systems through actuation errors. Agents who learn to defend against observation attacks become exposed to action-space disruptions and agents who defend against action-space attacks become exposed to observation attacks. The inability of models to generalize between different types of perturbations creates an illusion of security. The existing restrictions in robustness transfer between different types of attacks under mixed adversarial conditions remain unexplained.

This thesis addresses the aforementioned limitations by developing a complete system to train RL agents which protects them against state and action space mixed adversarial attacks. The main theoretical advancement brings forth the Action and State-Adversarial Markov Decision Process (ASA-MDP) which serves as a generalized game-theoretic framework for robust MDP models. On the implementation side, the proposed ASA-PPO algorithm serves as the implementation solution which uses a multi-head adversarial policy structure to achieve balanced and adaptive attack methods. The developed methods create an enhanced adversarial training system which trains agents to handle sophisticated threats that appear through multiple domains.

The proposed ASA-PPO method undergoes evaluation through multiple continuous control tasks which run in the MuJoCo physics engine, including PendulumSwingup, FingerSpin, and CheetahRun. The optimization process for training the adversary and protagonist agents uses an alternating scheme based on Proximal Policy Optimization (PPO) algorithm. The adversary uses a multi-head neural network structure to produce state and action disturbances which share common features to represent their relationship patterns. The system uses fixed perturbation limits which stay within realistic boundaries. The training process is computationally intensive but yields agents that demonstrate strong robustness across a variety of adversarial scenarios.

The ASA-PPO provides multiple essential benefits which surpass current methods. The framework provides complete protection against mixed adversarial attacks through its training process which generates a detailed robustness assessment. The multi-head adversarial architecture distributes perturbation budgets evenly between state and action spaces to stop the adversary from specializing in one particular attack type. The agents trained through ASA-PPO show better performance when they encounter new perturbation patterns and intensity levels. The method achieves high performance in normal operating conditions which proves that robustness systems do not compromise standard task execution.

Traditional adversarial training methods are limited by their single-type perturbation focus, which fails to reflect the complexity of real-world deployment conditions. These methods often result in overfitting to specific attack patterns, leaving agents vulnerable to novel or hybrid threats. Additionally, naive combinations of different attack types (e.g., simply adding state and action perturbations) lead to imbalanced training regimes, where one perturbation type dominates and the agent remains weak against the other.

These shortcomings underscore the need for more sophisticated, adaptive adversarial training frameworks like ASA-PPO.

The experimental findings show that agents trained through ASA-PPO achieve better performance than baseline methods when facing different types of adversarial attacks. The results show that training agents against one type of perturbation (action-space) does not provide protection against the other type (observation-space) which demonstrates the need for hybrid training approaches. The ASA-PPO agents produce superior returns during both single-type and hybrid attacks and they perform better when encountering new perturbation levels. The research shows that adversarial resistance does not guarantee resistance against changes made to environmental parameters through mass or damping coefficient modifications. The results indicate that adversarial robustness and parametric robustness present separate challenges which need distinct solution approaches.

This thesis establishes the groundwork for a new generation of robust RL systems capable of withstanding realistic, multi-modal adversarial threats. Future research could investigate adaptive perturbation balancing, extensions to partially observable environments (POMDPs) and integration with domain randomization to handle both adversarial and parametric uncertainties. The ASA-MDP framework also opens the door to theoretical analyses of equilibrium properties in mixed-adversarial settings. Ultimately, this work contributes to the broader goal of deploying safe, reliable, and trustworthy autonomous systems in complex, unpredictable environments.

KARMA ÇEKİŞMELİ BOZULMALARI DENGELEME: GÜRBÜZ PEKİŞTİRMELİ ÖĞRENMEYE DOĞRU

ÖZET

Derin pekiştirmeli öğrenme (DRL), karmaşık ortamlarla etkileşim yoluyla ardışık kararlar alabilen otonom ajanların eğitilmesi için güçlü bir yaklaşım olarak öne çıkmıştır. DRL derin sinir ağlarının temsil gücünü, pekiştirmeli öğrenmenin karar verme çerçevesiyle birleştirerek oyun oynama, robotik ve otonom sistemler gibi alanlarda ciddi başarılar elde etmiştir. Bu ajanlar, kümülatif ödülleri maksimize eden en uygun kararları öğrenir; böylece daha önce hesaplanması güç veya işlemsel olarak uğraştırıcı olan yüksek boyutlu durum ve eylem uzaylarıyla başa çıkabilirler. DRL sistemlerinin gerçek dünyada ve güvenliğin kritik olduğu uygulamalarda artan biçimde kullanılmaya başlamasıyla birlikte, bu sistemlerin düşmanca bozulmalara ve çevresel belirsizliklere karşı sağlamlığını sağlamak, araştırma önceliklerinden biri haline gelmiştir.

Bununla birlikte, DRL ajanları küçük ve çoğunlukla algılanması zor giriş değişikliklerine—yani düşmanca bozulmalara—karşı son derece savunmasızdır; bu tür değişiklikler karar verme süreçlerinde yıkıcı hatalara yol açabilir. Bu kırılganlıklar denetimli öğrenmede geniş biçimde incelenmiş ve son zamanlarda pekiştirmeli öğrenmeye de taşınmıştır. Çekişmeli eğitim, alan rastgeleleştirme, bağlamsal pekiştirmeli öğrenme ve gürbüz pekiştirmeli öğrenme yaklaşımları, DRL'deki bu zayıflıkları gidermeye yönelik başlıca yöntemler arasındadır.

Çekişmeli eğitime kıyasla, diğer yöntemlerin sağlam ve genellenebilir politikalar elde etmede kendine özgü sınırlamaları vardır. Alan rastgeleleştirme, çevresel parametreler için varyasyon aralıklarının elle tanımlanmasını gerektirir; bu yaklaşım kritik uç durumları kaçırabilir, karmaşık yüksek boyutlu ortamlara ölçeklendirmede zorlanabilir ve çekişmeli eğitimin dinamik olarak hedeflenmiş zorluklar üretmesinin aksine yüksek hesaplama maliyetlerine yol açabilir. Bağlamsal pekiştirmeli öğrenme, iyi tanımlanmış ve gözlemlenebilir bağlam değişkenlerine dayanır; bu durum, görünmeyen veya belirsiz bağlamlara genelleme yeteneğini sınırlar. Oysa çekişmeli eğitim, önceden tanımlanmış bağlamlara ihtiyaç duymadan politikanın zayıf yönlerini aktif olarak sınar. Gürbüz pekiştirmeli öğrenme ise en kötü durum senaryoları için tasarlanmış olmakla birlikte, genellikle tipik koşullarda aşırı temkinli politikalar üretir ve gerçek dünya değişkenliğini yakalayamayan ön tanımlı belirsizlik kümelerine dayanır; buna karşın çekişmeli eğitim uyarlanır bozulmalar üretir. Buna rağmen çekişmeli eğitim, dikkatli dengelenmezse belirli saldırı türlerine aşırı uyum riski taşır.

Mevcut çekişmeli eğitim yaklaşımları, ajanları eğitim sırasında düşmanca örneklere maruz bırakarak sağlamlığı artırmayı amaçlar. Bu yöntemler genellikle tek bir alandaki bozulmalara odaklanır—örneğin yalnızca durum gözlemleri, yalnızca eylemler, yalnızca ödüller veya yalnızca çevre dinamikleri üzerinde. Kapsamları içinde etkili olsalar da, bu

tek-tip savunmalar çoklu bozulma kaynaklarının aynı anda ortaya çıktığı gerçek dünya koşullarının karmaşıklığını yakalayamaz.

Güncel çekişmeli eğitim yöntemlerinin birincil sınırlaması, tek-modlu bozulmalara dar bir odaklanma göstermeleridir. Otonom dronlar veya endüstriyel robotlar gibi gerçek dünya sistemleri genellikle algıda (ör. gürültülü sensör verisi) ve kontrolde (ör. kusurlu tahrik) eşzamanlı bozulmalara maruz kalır. Gözleme dayalı saldırılara dirençli hale getirilmiş ajanlar, eylem alanı bozulmalarına karşı hâlâ zayıf olabilir; tersi de geçerlidir. Ayrıca, bozulma türleri arasında genellemenin olmaması yanlış bir güven hissi yaratır. Bu sınırlamalar, bir tür bozulmaya karşı elde edilen sağlamlığın diğer türlere nasıl aktarıldığı—veya aktarılamadığı—konusunda özellikle karışık/hibrit düşmanca koşullar altında önemli bir bilgi boşluğu olduğunu göstermektedir.

Bu tez, yukarıda bahsedilen sınırlamaları ele alarak durum ve eylem uzaylarında eşzamanlı bozulmaları içeren karma/hibrit düşmanca saldırılara karşı sağlam pekiştirmeli öğrenme ajanları eğitmek üzere kapsamlı bir çerçeve önermektedir. Temel teorik katkı, mevcut sağlam Markov karar süreçleri formülasyonlarını genelleştiren ve oyun teorisinden esinlenen bir bakış açısı getirerek Eylem-ve-Durum-Düşmanca Markov Karar Süreci (Action and State-Adversarial Markov Decision Process, ASA-MDP) kavramının tanıtılmasıdır. Uygulama tarafında ise tez, dengeli ve uyarlanabilir saldırı stratejilerini mümkün kılmak için çok başlı bir düşmanca politika yapay sinir ağı mimarisini içeren ASA-PPO algoritmasını sunar. Bu yenilikler, ajanları daha gerçekçi ve titiz bir çekişmeli eğitime tabi tutarak çok-modlu tehditlere karşı daha iyi hazırlanmalarını sağlar.

Önerilen ASA-PPO yöntemi, MuJoCo fizik motorunda PendulumSwingup, FingerSpin ve CheetahRun gibi bir dizi sürekli kontrol görevinde değerlendirilmiştir. Saldırgan ve kahraman ajanlar, Proximal Policy Optimization (PPO) algoritması tabanlı dönüşümlü bir optimizasyon şeması ile eğitilmiştir. Saldırgan; hem durum hem de eylem bozulmalarını üretmek üzere paylaşılan temsilleri olan çok başlı bir sinir ağı kullanır; böylece bu iki bozulma kaynağının karşılıklı bağımlılıkları modellenir. Sistemde ortaya çıkabilecek bozulmalar, büyüklük olarak önceden tanımlanmış ve gerçekçiliği sağlamak için sınırlandırılmıştır. Eğitim süreci hesaplama açısından yoğundur, ancak çeşitli çekişmeli senaryolarda güçlü dayanıklılık gösteren ajanlar elde edilmiştir.

ASA-PPO, mevcut yöntemlere göre birkaç önemli avantaja sahiptir. Birincisi, açıkça karma/hibrit düşmanca saldırıları modelleyen ve onlara karşı eğitim yapan ilk yaklaşımlardan biri olması nedeniyle daha kapsamlı bir sağlamlık profili sunar. İkincisi, çok başlı sinir ağı mimarisi, durum ve eylem uzayları arasında bozulma bütçesinin dengeli dağılımını sağlayarak saldırganın tek bir saldırı türüne aşırı uyum sağlamasını engeller. Üçüncüsü, ASA-PPO ile eğitilmiş ajanlar, görülmemiş bozulma türleri ve şiddetlerine karşı daha iyi genelleme gösterir. Son olarak, yöntem, bozulma olmayan ortamlarda da rekabetçi performans koruyarak sağlamlığın olağan görev performansından ödün vermesi gerekmediğini göstermektedir.

Geleneksel çekişmeli eğitim yöntemleri, tek-tip bozulma odakları nedeniyle gerçek dünya dağıtım koşullarının karmaşıklığını yansıtmakta yetersiz kalır. Bu yöntemler sıklıkla belirli saldırı desenlerine aşırı uyum göstererek ajanları yeni veya hibrit tehditlere karşı savunmasız bırakır. Ayrıca, farklı saldırı türlerinin naif biçimde birleştirilmesi (ör. sadece durum ve eylem bozulmalarının birleştirilmesi) dengesiz

eğitim rejimlerine yol açar; bu durumda bir bozulma türü baskın hale gelir ve ajan diğerine karşı zayıf kalır. Bu eksiklikler, ASA-PPO gibi daha sofistike ve uyarlanabilir çekişmeli eğitim çerçevelerine duyulan ihtiyacı vurgulamaktadır.

Ampirik sonuçlar, ASA-PPO ile eğitilmiş ajanların karma/hibrit düşmanca koşullar altında varolan yöntemlere kıyasla belirgin şekilde daha iyi performans gösterdiğini ortaya koymaktadır. Önemli bir bulgu, bir bozulma türüne (ör. eylem alanı) karşı elde edilen sağlamlığın diğer türe (ör. gözlem/durum alanı) geçişken olmaması; bu durum hibrit eğitim gerekliliğini pekiştirir. ASA-PPO ajanları hem tek-tip hem de hibrit saldırılar altında daha yüksek toplam ödül elde etmekte ve görülmemiş bozulma şiddetlerine daha iyi genelleme göstermektedir. Ancak çalışma ayrıca, düşmanca dayanıklılığın parametre kaynaklı çevresel değişikliklere (ör. kütle veya sönüm katsayılarındaki değişimler) karşı dayanıklılık anlamına gelmediğini ortaya koymuştur. Bu, düşmanca sağlamlık ile parametrik sağlamlığın birbirinden ayrı zorluklar olduğunu ve tamamlayıcı teknikler gerektirebileceğini göstermektedir.

Bu tez, gerçekçi, çok-modlu düşmanca tehditlere dayanabilecek yeni nesil sağlam pekiştirmeli öğrenme sistemleri için bir temel atmaktadır. Gelecekteki çalışmalar, önerilen yöntemi adaptif bozulma dengeleme teknikleri, kısmi gözlemlenebilir Markov karar süreçler (POMDP) ve parametrik belirsizlik modelleriyle entegre ederek, bu bileşenlerin birbirleriyle etkileşimini ve genel sistem performansındaki iyileştirmeleri inceleyebilirler. ASA-MDP çerçevesi ayrıca karma çekişmeli ortamlarda denge özelliklerinin teorik analizlerine kapı aralamaktadır. Nihayetinde bu çalışma, karmaşık ve öngörülemez ortamlarda güvenli, dayanıklı ve sağlam yüksek otonom sistemlerin konuşlandırılmasına katkıda bulunmayı amaçlamaktadır.

1. INTRODUCTION

Deep Reinforcement Learning (DRL) [1] has emerged as a powerful paradigm for enabling autonomous agents to learn complex behaviors through interaction with their environment [2]. By combining the representational power of deep neural networks with the decision-making framework of reinforcement learning (RL) [3], DRL has achieved remarkable success in a wide range of domains, from mastering intricate games [4]–[7] such as Go and StarCraft to enabling sophisticated control in robotics and autonomous systems [8]–[11]. These agents learn optimal policies by maximizing cumulative reward, demonstrating an ability to handle high-dimensional action and state spaces that were previously intractable. The expansion of DRL applications underscores its potential as a cornerstone technology for artificial intelligence.

However, the very neural network architectures that empower DRL also introduce a critical vulnerability: a profound susceptibility to adversarial perturbations [12]. Initially discovered in the context of supervised learning for image classification [13], adversarial attacks involve small, mostly indistinguishable alterations to the input that may lead a model to make catastrophically incorrect predictions with strong confidence [14]. This brittleness translates directly to the RL domain, where trained DRL agents can fail dramatically when deployed in environments that differ only slightly from their training conditions, whether due to natural noise, model misspecification, or malicious interventions [15,16].

This lack of robustness poses a significant barrier to the real-world deployment of DRL systems, particularly in safety-critical applications such as healthcare, industrial automation and autonomous driving. In these scenarios, agents must be resilient not only to stochastic environmental variations but also to potential adversarial attacks that target the integrity of the decision-making process [17]. The challenge, therefore, shifts from merely learning a high-performing policy to learning one that is robust and reliable under perturbations and uncertainties.

To address these vulnerabilities, a common defense mechanism is Adversarial Reinforcement Learning (ARL) [18], which frames the training process as a minimax game between a protagonist agent and an adversary. The adversary's role is to apply perturbations during training, forcing the protagonist agent to learn a policy that remains effective even under (near) worst-case conditions [19]. While this approach has shown promise [20,21], most of the existing research has concentrated on defending against attacks that target a single component of the agent's interaction loop [22,23]—typically perturbing the state observations [24,25], the actions [26], the reward function [27], or dynamics [28], but not multiple components concurrently. However, real-world systems are often subject to multiple, simultaneous sources of perturbation [29].

The limitation of single-type adversarial training becomes particularly apparent when considering real-world autonomous systems [30]. An autonomous drone, for instance, simultaneously processes noisy sensor data while executing potentially imperfect control commands—both perception and actuation are subject to continuous perturbations from environmental conditions, hardware limitations, and potential adversarial interference [31]. Similarly, industrial robots [32] operate in environments where both sensory inputs and motor outputs may be corrupted by electromagnetic interference, mechanical wear, or deliberate attacks. This observation raises a fundamental question: Does robustness developed against observation perturbations transfer to action-space perturbations, and vice versa? More fundamentally, is it possible to develop training frameworks that produce agents robust to realistic scenarios involving mixed or hybrid perturbations across multiple modalities?

These questions reveal a significant gap in the current understanding of adversarial robustness in DRL. Although significant advancements have been achieved in developing defenses against individual attack types in isolation, the interactions and dependencies between different perturbation modalities remain largely unexplored [17]. Preliminary investigations suggest that robustness is often highly specific to the attack type encountered during training, with limited generalization to novel perturbation patterns [33]. This specialization implies that agents trained to be robust against observation attacks may still be susceptible to action perturbations, potentially creating a

false sense of security when deploying such agents in complex real-world environments where multiple perturbation sources are inevitable.

Addressing this gap requires not only new algorithmic approaches but also a reconceptualization of the adversarial robustness problem itself. Rather than treating different attack modalities as independent concerns, a comprehensive solution must acknowledge and explicitly model the dependencies between state observations and actions in the agent's decision-making process. The challenge extends beyond simply applying multiple perturbations simultaneously—it requires developing mechanisms that enable adversaries to learn balanced attack strategies that account for how perturbations in one modality influence and interact with perturbations in another. Only through such integrated frameworks can genuinely robust agents be trained to withstand realistic, multi-modal adversarial threats.

1.1 Purpose of Thesis

This thesis addresses the critical challenge of developing DRL agents that maintain robust performance under realistic, mixed or hybrid adversarial attacks. The primary purpose is to establish a comprehensive framework for understanding, modeling, and defending against simultaneous perturbations to both observations and actions—a scenario that reflects the true complexity of real-world deployment environments but has been largely neglected in existing research. By moving beyond the artificial constraint of single-type attacks, the study aims to offer both theoretical foundations and practical methods for achieving genuine robustness in autonomous decision-making systems.

This research holds significance across both academic and practical dimensions. Academically, the work fills a substantial gap in the ARL literature by systematically investigating the transferability of robustness across attack modalities and establishing formal frameworks for mixed-attack scenarios. From a practical standpoint, the proposed methods directly address deployment concerns for safety-critical applications where agents must operate reliably despite multiple simultaneous sources of uncertainty. The insights gained from this research are expected to inform the development of more robust DRL algorithms and provide guidance for evaluating and certifying the reliability of autonomous systems before deployment.

Furthermore, the thesis challenges the implicit assumption prevalent in adversarial training research that adversaries naturally optimize their attack strategies effectively. Through careful empirical investigation, it is demonstrated that naive approaches to mixed attacks often fail to achieve balanced perturbation strategies, resulting in suboptimal training regimes that leave agents vulnerable to well-coordinated attacks. By developing mechanisms that enable adversaries to learn balanced, adaptive attack strategies, this work establishes a more rigorous foundation for adversarial training that better prepares agents for sophisticated threats encountered in practice.

1.2 Scope and Research Objectives

The scope of this thesis encompasses the theoretical formalization, algorithmic development, and empirical validation of methods for training DRL agents robust to mixed or hybrid adversarial attacks. The research focuses on continuous control tasks where both state observations and actions lie in continuous spaces, as these domains present particular challenges for maintaining robustness while preserving task performance. The investigation is grounded in the framework of Markov Decision Processes (MDPs) and their adversarial extensions, with particular emphasis on zero-sum game formulations that capture the strategic interaction between protagonist agents and adaptive adversaries.

The investigation is structured around four fundamental research questions.

- Does robustness to observation perturbations imply robustness to action perturbations, and vice versa? This question addresses the transferability of robustness across attack modalities and challenges the implicit assumption that adversarial training provides general-purpose protection.
- Which types of adversarial attacks most effectively challenge DRL agents, and how do mixed or hybrid attacks compare to single-type perturbations? This comparative analysis establishes the relative severity of different threat models and justifies the focus on mixed-attack scenarios.
- Can training with balanced mixed adversarial attacks enable agents to develop simultaneous robustness against action, observation, and combined perturbations?

This question drives the development of novel training algorithms that explicitly account for dependencies between different perturbation modalities.

- How well do agents trained under adversarial conditions generalize to environments with changing dynamics and parameters? This question probes the relationship between adversarial robustness and parametric robustness, assessing the broader applicability of adversarial training methods.

The research objectives are operationalized through a combination of formal modeling, algorithm design, and comprehensive experimentation. The Action and State-Adversarial Markov Decision Process (ASA-MDP) is developed as a unified framework that subsumes existing single-type formulations while enabling the study of mixed attacks. Building on this foundation, the ASA-PPO algorithm [34] is introduced, incorporating architectural and optimization innovations that enable adversaries to learn balanced attack strategies. Empirical validation is conducted across diverse continuous control environments, with systematic evaluation of robustness under various attack conditions, including those not encountered during training. This methodology establishes both the necessity and feasibility of mixed-attack training for achieving comprehensive robustness in RL agents.

1.3 Contributions of the Thesis

This thesis makes several substantial contributions to the field of robustness in ARL. The primary theoretical contribution is the introduction of the Action and State-Adversarial Markov Decision Process (ASA-MDP), a game-theoretic framework that formally models mixed adversarial attacks involving simultaneous perturbations to both observations and actions. Unlike previous formulations that treat these attack modalities independently, ASA-MDP explicitly captures their interaction and provides a unified mathematical foundation that subsumes standard MDPs, Noisy Action Robust MDPs (NR-MDP) [26], and State-Adversarial MDPs (SA-MDP) [24] as special cases. This framework enables rigorous analysis of mixed-attack scenarios and provides the theoretical basis for developing algorithms capable of handling realistic multi-modal threats.

The second major contribution is empirical in nature: the thesis provides the first systematic demonstration that robustness to one type of perturbation does not generally transfer to other perturbation types. Through comprehensive experiments across multiple environments and attack configurations, the research establishes that agents trained against observation attacks remain highly vulnerable to action perturbations and vice versa. This finding challenges the implicit assumptions underlying much of the existing adversarial training literature and highlights the critical importance of addressing mixed attacks explicitly. Furthermore, it is demonstrated that among various attack strategies, balanced mixed attacks represent the most severe threat to non-robust agents, establishing the practical relevance of the problem addressed in this study.

From an algorithmic perspective, the research introduces ASA-PPO [34], a novel adversarial training method that enables adversaries to learn balanced attack strategies across state and action spaces. The key innovation lies in a multi-head policy architecture for the adversary that explicitly models the dependency between state perturbations and their downstream effects on actions. This design overcomes a critical limitation of naive hybrid-attack approaches, which tend to allocate perturbation budgets unevenly, resulting in protagonist agents that remain vulnerable to specific or well-coordinated attacks. Through careful architectural choices and training procedures, ASA-PPO produces agents that exhibit substantially improved robustness not only to mixed attacks but also to single attack modalities, demonstrating that properly balanced adversarial training can yield more comprehensive protection than specialized single-type methods.

Finally, the thesis contributes extensive empirical insights into the practical aspects of adversarial training for robustness. The effects of perturbation budget magnitudes are systematically investigated, demonstrating the trade-offs between robustness and nominal performance. The generalization of learned robustness to attack strengths not encountered during training is analyzed, revealing both the capabilities and limitations of adversarial training approaches. Additionally, the relationship between adversarial robustness and robustness to parametric variations in environment dynamics is examined, clarifying that these represent orthogonal dimensions of robustness that may require complementary approaches. These findings provide practical guidance for deploying

robust DRL systems and identify important directions to guide future research in this fast-evolving field.

1.4 Thesis Outline

The remainder of this thesis is organized as follows:

- Chapter 2 provides a comprehensive literature review, surveying the landscape of adversarial attacks and defenses in DRL. The chapter traces the evolution from early demonstrations of vulnerability in supervised learning to contemporary adversarial training frameworks, examining methods for observation-based attacks, action-space perturbations, reward poisoning, and dynamics manipulation. The analysis highlights the strengths and limitations of existing approaches, with particular attention to their focus on single-type attacks and the limited understanding of mixed-attack scenarios. The review establishes the theoretical and empirical foundations upon which the present research builds and situates the proposed contributions within the broader context of robust reinforcement learning.

- Chapter 3 establishes the theoretical foundations necessary for understanding the proposed approach. It provides formal definitions of Markov Decision Processes (MDPs), zero-sum games, and Robust MDPs, along with their game-theoretic properties and equilibrium conditions. Existing frameworks for modeling adversarial perturbations—specifically the Noisy Action Robust MDP (NR-MDP) and State-Adversarial MDP (SA-MDP)—are introduced and analyzed in terms of their assumptions and limitations. The mathematical foundations of Proximal Policy Optimization (PPO) [35], which serves as the base algorithm for the proposed adversarial training framework, are also presented. These preliminaries prepare the reader for the novel formulations introduced in subsequent chapters.

- Chapter 4 presents the core technical contributions of the thesis. The chapter formally introduces the Action and State-Adversarial Markov Decision Process (ASA-MDP) framework, establishing its mathematical properties and demonstrating how it generalizes existing formulations. The problem formulation is articulated in detail, including the assumptions underlying the proposed approach and the

specific threat model it addresses. The ASA-PPO algorithm is then presented, including the architectural innovations—particularly the multi-head adversarial policy network—and the training procedures that enable balanced mixed attacks. Practical considerations for implementing adversarial training, such as perturbation budget selection and the alternating optimization scheme, are also discussed.

- Chapter 5 reports comprehensive experimental results across diverse continuous control environments. The proposed approach is systematically evaluated against multiple baselines, addressing each of the research questions posed in Section 1.2. The chapter presents training dynamics, robustness evaluations under various attack conditions, and analyses of transferability across perturbation modalities. The effects of perturbation budget magnitudes, generalization to unseen attack strengths, and the relationship between adversarial robustness and parametric robustness are examined in detail. Through ablation studies and comparative analyses, the effectiveness of ASA-PPO is demonstrated, providing insights into the mechanisms underlying its superior performance.

- Chapter 6 concludes the thesis by summarizing the main contributions, discussing their implications for both research and practical deployment, and identifying limitations of the current work. The broader significance of the findings for the reliable deployment of DRL systems in safety-critical applications is reflected upon, and promising directions for future research are outlined. These include extensions to partially observable environments, adaptive perturbation strategies, and the integration of adversarial training with complementary robustness techniques such as domain randomization.

2. LITERATURE REVIEW

This chapter provides a comprehensive review of the literature on adversarial training and robustness in RL systems. Over the past decade, the field of robust learning has evolved significantly, driven by the recognition that neural network–based policies, despite their impressive capabilities, exhibit fundamental vulnerabilities to perturbations and environmental variations. Major research directions have emerged to address these vulnerabilities, including adversarial training paradigms, safe RL frameworks, domain randomization techniques, and specialized approaches for multi-agent settings. By examining the assumptions, methodologies, strengths, and limitations of existing work, this section establishes the context for the contributions of this thesis and highlights the critical gaps that motivate the development of hybrid adversarial attack frameworks. The discussion traces the progression from isolated, single-modality defenses toward the understanding that comprehensive robustness requires addressing multiple perturbation sources simultaneously—a challenge that remains incompletely resolved in the current literature.

2.1 RL and MARL

RL provides the foundational framework for sequential decision-making under uncertainty, enabling agents to learn optimal behaviors through trial-and-error interaction with environments [36]. RL has progressed from its foundational value-based methods, such as Q-learning, which enabled agents to approximate action-value functions through temporal-difference learning in tabular environments [37], to sophisticated deep RL frameworks that handle high-dimensional spaces [7]. Policy gradient algorithms, introduced by Sutton et al. [38], further advanced direct policy optimization, paving the way for on-policy methods like Proximal Policy Optimization (PPO), which balances exploration and exploitation via clipped surrogates and has been widely applied in continuous control tasks [35]. These classics have inspired diverse extensions, including offline RL, where agents learn solely from static datasets without

further environment interaction, proving invaluable in domains like healthcare where real-time experimentation is ethically constrained [39]. More recently, Kaufmann et al. demonstrated autonomous drone racing at championship speeds using learned policies, showcasing the maturity of deep RL for real-world deployment [40].

Multi-Agent Reinforcement Learning (MARL) extends RL to environments with multiple interacting agents, often modeled as Markov Games [41]. MARL introduces complexities such as non-stationarity, as agents must adapt to the evolving strategies of others [42]. Approaches in MARL include independent learning, where agents treat others as part of the environment [43,44], and centralized training with decentralized execution, which leverages global information during training while maintaining local policies during execution [45,46]. MARL has found applications in cooperative tasks like robotic swarms and competitive scenarios such as autonomous driving [47,48].

Despite these advancements, RL and MARL agents still struggle in real-world deployment. This challenge, known as the sim-to-real gap, arises when models trained in simulated environments fail to perform adequately in real-world settings [49]. Policies are typically trained in simulation, which can only approximate the real world. Consequently, even minor discrepancies in system dynamics—such as varying physical parameters, unmodeled friction, or sensor noise—between the training and deployment environments can lead to a significant performance drop [50]. Classical RL algorithms, optimized for performance in a single, fixed training environment, lack the inherent mechanisms to cope with such variations.

An important discovery in MARL research revealed that agents trained through self-play or co-evolution can develop surprising vulnerabilities to novel opponent strategies [51]. Adversarial policies—opponent strategies specifically designed to exploit weaknesses in a target agent—can induce failures even without directly perturbing the agent's observations or actions. Research on robustness in MARL has also explored population-based training approaches where agents are exposed to diverse opponents rather than a single adversary. Vinitsky et al. [52] demonstrated that using adversarial populations during training improves robustness compared to single-adversary approaches. Liu and Lai [53] examined scenarios where exogenous

adversaries can inject perturbations into multiple components of multi-agent systems simultaneously, including actions and rewards. They demonstrated that combined action and reward poisoning can drive agents toward attacker-selected behaviors.

2.2 Adversarial Training and Robustness

Adversarial training originated from seminal discoveries in supervised learning that revealed the susceptibility of neural networks to carefully crafted perturbations. Szegedy et al. [13] first demonstrated that imperceptible modifications to input data could cause deep learning models to produce confident but incorrect classifications, fundamentally challenging assumptions about the reliability of learned representations. These findings catalyzed extensive research into understanding the nature of adversarial examples and developing defenses against them, with Goodfellow et al. [14] introducing the Fast Gradient Sign Method (FGSM) as a computationally efficient approach for generating adversarial perturbations.

The extension of adversarial attacks to RL was pioneered by Huang et al. [15] and Kos et al. [19], who demonstrated that trained policies could fail catastrophically under minor perturbations to observations. Unlike supervised learning, where adversarial training involves perturbing static input-output pairs, RL introduces temporal dependencies and sequential decision-making that compound perturbation effects over trajectories. This temporal dimension fundamentally changes the adversarial landscape, as small perturbations can accumulate through feedback loops and lead to substantially different outcomes. Later studies have demonstrated that trained agents are also susceptible to action-space perturbations [54], reward poisoning [27,28], and model attacks [55], all of which can lead to comparable performance failures.

The paradigm of adversarial training—incorporating adversarially perturbed samples during the training process—emerged as one of the most effective defenses in machine learning contexts [56,57]. Madry et al. [58] advanced this approach through Projected Gradient Descent (PGD), demonstrating that training against strong iterative attacks yields models more robust than those trained against single-step methods. Recent work by Croce and Hein [59] introduced AutoAttack, a parameter-free ensemble of attacks that has become the standard for robustness evaluation in computer vision, addressing

concerns about adaptive attacks and proper evaluation protocols. The core assumption underlying this approach is that exposure to worst-case or near-worst-case perturbations during training enables models to develop robust decision boundaries that resist similar perturbations at test time. However, this approach faces inherent trade-offs between robustness and standard accuracy, as models must allocate capacity to defending against potential attacks rather than solely optimizing for clean data performance.

Robustness in RL encompasses multiple distinct but related concepts, including resistance to adversarial perturbations, generalization across environmental variations, and resilience to model misspecification. Moos et al. [17] provided a comprehensive survey documenting various robustification techniques, including adversarial training frameworks, robust optimization formulations, and safety-constrained learning approaches.

Robust Markov Decision Processes, formalized by Nilim and El Ghaoui [60] provide a foundational framework for decision-making under uncertainty, formulating the problem as a minimax optimization where the agent maximizes reward against worst-case dynamics within an uncertainty set. Suilen et al. [61] recently reviewed RMDPs as a meeting point between artificial intelligence and formal methods.

A critical evolution in adversarial training methodology came from recognizing that static, fixed adversaries provide insufficient training signals. Pinto et al. [50] introduced Robust Adversarial Reinforcement Learning (RARL), which formalizes the problem as a zero-sum game [62] between a protagonist agent and an adversarial agent that learns to apply destabilizing forces. The protagonist, in turn, learns a policy robust to the optimal disruptive strategies employed by the adversary. This co-adaptation process forces the agent to learn to perform well even under (near) worst-case conditions generated during training. Later, Pan et al. expanded on RARL by introducing the Risk-Averse Robust Adversarial Reinforcement Learning (RARARL) algorithm, where the risk-seeking adversary and the risk-averse protagonist take turns controlling the actions that are carried out [63]. Liang et al. [64] further extended RARL by introducing WocaR (Worst-Case Aware Reinforcement Learning), which integrates adversarial training with distributionally robust optimization to enhance out-of-distribution generalization.

Zhang et al. build on their earlier work in [24] by training an agent that is robust to perturbations in the observation space using the same zero-sum formulation under the ARL framework [65].

The literature distinguishes between black-box attack, gray-box, and white-box settings based on the knowledge of the adversary about the target policy. In white-box attacks, attackers exploit gradient information to craft optimal perturbations [66,67]. Behzadan and Munir [68] investigated gray-box man-in-the-middle strategies. In [55,69], the authors demonstrated black-box attacks that reduce agent performance without access to model parameters. The choice of attack model significantly influences both the difficulty of generating effective perturbations and the type of robustness that can be guaranteed.

A significant limitation of adversarial training identified by Tramer et al. [70] is adversarial overfitting, where defenses become specialized to specific attack patterns encountered during training but fail to generalize to novel strategies. This limitation has motivated the development of adaptive attacks that dynamically adjust to the characteristics of particular defenses. Despite these advances, a critical gap remains in understanding whether robustness to one perturbation type transfers to others, as most studies evaluate defenses only against the attack modality used during training. This thesis addresses the gap through systematic cross-evaluation and provides empirical evidence that learned robustness is specific and shows limited transferability.

2.3 Domain Randomization

Domain Randomization (DR) is a technique used to improve policy robustness, particularly for bridging the sim-to-real gap in robotics [71]. The core idea is to train a policy in a simulator where physical and visual parameters—such as mass, friction, lighting conditions, and textures—are randomized across a wide range. By exposing the agent to a sufficiently diverse set of training conditions, it learns a policy that is invariant to these variations and can therefore generalize more effectively to the real world [72].

A main assumption of DR is that the real-world environment can be treated as just another variation within the randomized distribution. Its primary use case is in robotics,

where it allows for the training of visuomotor policies entirely in simulation before deploying them on physical hardware. The advantage of DR is its simplicity and effectiveness without requiring an explicit adversarial opponent. However, the process is a heuristic and provides no formal guarantees of robustness. If the randomization range is too narrow, the policy may fail to transfer; if it is too broad, the policy may become overly conservative or fail to learn the task at all.

DR is a form of data augmentation that aims for robustness against uncertainty in the environment's dynamics, which is complementary to adversarial training that focuses on worst-case perturbations to the agent's perceptions and actions. Recent work has explored making the randomization process more intelligent, for example, through active domain randomization that focuses on the most informative environmental parameters [73]. However, the computational costs of such combined training can be substantial, and the interactions between different robustification techniques are not fully understood. Muratore et al. [74] extended domain randomization to include learned randomization distributions, using meta-learning to automatically discover which parameters to randomize and their appropriate ranges.

DR represents a complementary approach to adversarial training, addressing parametric uncertainty rather than targeted perturbations. While domain randomization exposes agents to diverse but benign environmental variations, adversarial training specifically prepares agents for worst-case scenarios.

2.4 Safe Reinforcement Learning

Safe Reinforcement Learning is a related but distinct area of research that focuses on preventing an agent from entering catastrophic or undesirable states, both during the learning process and after deployment [75]. While robustness is concerned with maintaining performance under perturbations, safety is concerned with satisfying constraints. Methodologies in this field often involve augmenting the standard reward-maximization objective with safety constraints.

Common approaches involve using concepts from control theory, such as Lyapunov functions, to guarantee that the agent remains within a "safe" region of the state space [76]. Another popular method is Constrained MDPs (CMDPs), where the agent

must maximize a reward function while keeping the expected cumulative value of a cost function below a certain threshold [77]. The primary assumption is that a set of unsafe states or a cost function can be clearly defined. Use cases are abundant in safety-critical applications like autonomous driving and medical robotics. A significant challenge is ensuring safety during exploration, when the agent must take uncertain actions to learn about the environment. Recent work has focused on developing algorithms that can learn from limited data while providing safety guarantees [78] or using learned recovery policies to escape unsafe situations [79]. Robustness and safety are complementary; a robust agent is less likely to be perturbed into an unsafe state by an adversary, making robustness a powerful tool for achieving safety.

2.5 Motivation and Research Gaps

Despite substantial progress in adversarial training and robustness within RL, several critical gaps persist. Most existing studies consider the robustness of RL agents primarily in terms of their performance under specific types of perturbations, such as adversarial attacks on observations or actions. However, the interplay between different types of perturbations and their cumulative effects on agent performance remains largely unexplored. In real-world scenarios, agents may face simultaneous challenges from multiple sources, such as sensor noise, actuator failures, and adversarial manipulations. The majority of current research focuses on isolated attack modalities, leaving a significant gap in understanding how agents can be trained to withstand a combination of perturbations.Furthermore, the existence or transferability of robustness across different attack modalities remains insufficiently understood. For instance, it is unclear whether robustness to observation perturbations imparts any resilience against action-space attacks. These uncertainties raise important questions regarding the practical effectiveness of current methods when deployed beyond controlled environments. Finally, relation between the robustness against a specific type of adversarial attack and the generalization capabilities of RL agents to out-of-distribution scenarios is not well established. While some studies suggest that adversarial training can enhance generalization, others indicate that it may lead to overfitting to specific attack patterns, thereby reducing performance in unperturbed environments. Finding

solutions for these limitations is necessary to advance the field and enable the reliable implementation of robust RL agents in realistic scenarios.

3. THEORETICAL FOUNDATIONS

This chapter establishes the theoretical foundation to contextualize the contributions of this thesis. It explores three core areas: Reinforcement Learning, the primary framework for this study; Adversarial Training, the methodological foundation for the proposed approach; and Robust Reinforcement Learning, a subfield concerned with developing algorithms that maintain reliable performance under uncertainty, perturbations, or adversarial conditions in the environment or model dynamics, with or without adversarial training. Each section is structured to provide comprehensive explanations with sufficient depth, ensuring clarity and accessibility for readers with limited prior knowledge of these domains.

3.1 Reinforcement Learning

Reinforcement Learning (RL) is a subfield of machine learning concerned with how an agent learns to make decisions through interactions with its environment. Instead of being explicitly programmed to perform specific actions, the agent explores different possibilities and observes their outcomes, by obtaining feedback through penalties or rewards. Through trial and error, the agent learns strategies that maximize long-term rewards. This approach is modeled after the learning processes observed in animals and humans [80], such as learning to ride a bike or hunting a prey, where improvement comes from practicing, making mistakes, and gradually discovering what works best.

RL is centered on the interaction between two primary entities: the environment and the agent. The environment encompasses everything the agent interacts with and receives feedback from and the agent is the learner or decision-maker that selects actions. The learning process unfolds as a cycle of interaction between the agent and the environment, as illustrated in Figure 3.1. At every step, the following cycle repeats:

1. The agent observes the current state s_t of the environment.

2. Based on the state, the agent selects an action a_t from its set of possible actions.

3. The environment transitions to a next state s_{t+1} in response to the action and provides a reward r_t, which is a numerical value representing the immediate benefit (or cost) of the action.

4. The sequence repeats, allowing the agent to learn which actions result in better outcomes over time, until termination conditions are met.

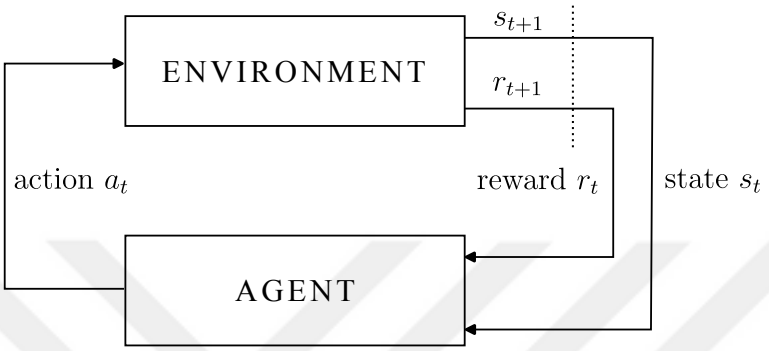

Figure 3.1: Reinforcement learning standard agent-environment interaction loop.

The ultimate goal of the agent is not simply to maximize rewards in the short term, but to develop a strategy, known as a policy, that maximizes the total reward accumulated over the long run. In other words, the agent aims to learn how to act in different situations so that its overall performance improves with experience. This long-term focus distinguishes RL from other forms of learning that might only optimize for immediate accuracy or prediction.

3.1.1 Markov Decision Process (MDP)

A Markov Decision Process (MDP) is a mathematical model used to represent decision-making scenarios where outcomes may be influenced by both randomness and the actions of a decision-maker. An MDP is formally defined as a tuple $< S, A, P, R, \gamma >$, where:

- S is a finite set of states that represent all possible situations the agent can encounter.
- A is a finite set of actions available to the agent.

- P is the state transition probability function, where $P(s_{t+1}|s_t, a_t)$ denotes the probability of transitioning to state s_{t+1} from state s_t after taking action a_t at time step t.
- R is the reward function, where $R(s_t, a_t)$ gives the expected reward received after taking action a_t in state s_t.
- γ is the discount factor, a value between 0 and 1 that determines the importance of future rewards compared to immediate rewards. A γ close to 0 makes the agent prioritize immediate rewards, while a γ close to 1 encourages the agent to consider long-term rewards.

As formalized in Equation 3.1, the MDP framework assumes the Markov property, whereby the future state depends solely on the current state and action, independent of prior events. This property simplifies the analysis and solution of decision-making problems, making MDPs a foundational concept in RL and related fields.

$$P(s_{t+1}|s_t, a_t, s_{t-1}, a_{t-1}, \ldots, s_0, a_0) := P(s_{t+1}|s_t, a_t) \tag{3.1}$$

3.1.2 Exploration vs. exploitation

In RL and decision-making contexts, the exploration–exploitation dilemma [81] refers to the trade-off between two competing strategies: exploration, in which an agent seeks out new actions or states to acquire additional information about the environment, and exploitation, in which the agent leverages existing knowledge to maximize immediate reward. Optimal performance requires a balance between these strategies: excessive exploitation can lead to suboptimal long-term outcomes by neglecting potentially superior actions, whereas excessive exploration can incur unnecessary costs and delay reward accumulation. This dilemma is central to the design of RL algorithms, where policies must be devised to strategically navigate uncertainty while progressively improving expected returns.

3.1.3 Value and advantage functions

Starting from the first state s_0 in the environment, the objective of the agent is to learn a policy $\pi : S \to A$ that maps states to actions (or action probabilities) and maximizes the

expected discounted cumulative reward, which defines the state-value function $V^{\pi}(s)$. The state-value function under policy π is defined as the expected return when starting from state s and following policy π thereafter, as shown in Equation 3.2.

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^{t} R(s_t, a_t) \mid s_0 = s\right] \tag{3.2}$$

Similarly, the action-value function $Q^{\pi}(s,a)$ represents the expected return when starting from state s, taking action a, and thereafter following policy π, as defined in Equation 3.3.

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^{t} R(s_t, a_t) \mid s_0 = s, a_0 = a\right] \tag{3.3}$$

The advantage function $A^{\pi}(s,a)$ quantifies the relative value of taking action a in state s compared to the average value of all actions in that state under policy π. It is defined as the difference between the action-value function and the state-value function, as shown in Equation 3.4.

$$A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s) \tag{3.4}$$

3.1.4 Policy gradient methods

Policy gradient methods are a class of RL algorithms that optimize the policy directly by adjusting its parameters in the direction that maximizes expected cumulative rewards. Unlike value-based methods that focus on estimating value functions, policy gradient methods learn a parameterized policy, typically represented as $\pi_{\theta}(a|s)$, where θ denotes the policy parameters. The core idea is to compute the gradient of the expected return with respect to the policy parameters and use this gradient to update the policy in a way that increases the likelihood of selecting actions that yield higher rewards. The basic form of the policy objective and its gradient, derived using the log-trick, are given in Equations 3.6 and 3.5, respectively, where τ represents a trajectory of states and actions, and $R(\tau)$ is the total reward accumulated along that trajectory.

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_{\theta}}\left[R(\tau)\right] \tag{3.5}$$

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{\tau \sim \pi_{\theta}}\left[\sum_{t=0}^{T} \nabla_{\theta} \log \pi_{\theta}(a_t|s_t) R(\tau)\right] \tag{3.6}$$

Policy gradient methods use gradient ascent to iteratively update the policy parameters, as shown in Equation 3.7, where α is the learning rate.

$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta) \tag{3.7}$$

These methods are particularly effective in high-dimensional or continuous action spaces, where traditional value-based methods may struggle. They also naturally handle stochastic policies, making them suitable for a wide range of RL tasks.

3.1.4.1 Proximal Policy Optimization (PPO)

Building upon TRPO [82], Proximal Policy Optimization (PPO) is a widely used policy gradient algorithm in RL designed to enhance training stability and efficiency [35]. PPO achieves this by using a surrogate objective function that restricts the magnitude of policy updates, preventing large, destabilizing changes. This is typically done through a clipping mechanism that limits how much the new policy can deviate from the old policy during each update. By balancing exploration and exploitation, PPO allows for more reliable learning in complex environments, making it a widely adopted choice for various RL tasks.

The standard PPO objective function is given in Equation 3.8.

$$L^{CLIP}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta) A_t, \text{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon) A_t \right) \right] \tag{3.8}$$

Here, A_t is the advantage estimate and $r_t(\theta)$ is the policy probability ratio defined in Equation 3.9,

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \tag{3.9}$$

where $\pi_\theta(a_t|s_t)$ is the probability of taking action a_t in state s_t under the current policy parameterized by θ, and $\pi_{\theta_{old}}(a_t|s_t)$ is the probability under the previous policy parameterized by θ_{old}. The clipping parameter ε is a small positive value that defines the range within which the probability ratio is allowed to vary, thus controlling the extent of policy updates.

In the original implementation of PPO, the objective function is augmented with an entropy bonus $H[\pi_\theta(\cdot|s_t)]$ to encourage exploration and a value function loss $L^{VF}(\theta)$ to improve value estimation. The value function loss is typically defined as the mean

squared error between the predicted value and the empirical target return, as shown in Equation 3.10, where $V_\theta(s_t)$ is the value function parameterized by θ and $\hat{V}_t$ is the target at time step t.

$$L^{VF}(\theta) = \mathbb{E}_t \left[\left(V_\theta(s_t) - \hat{V}_t \right)^2 \right] \tag{3.10}$$

Entropy regularization is used to encourage exploration by preventing the policy from becoming deterministic too quickly. The entropy of the policy at state s_t is given by $H[\pi_\theta(\cdot|s_t)]$, which measures the uncertainty in the action distribution is defined as follows in Equation 3.11.

$$H[\pi_\theta(\cdot|s_t)] = -\sum_a \pi_\theta(a|s_t) \log \pi_\theta(a|s_t) \tag{3.11}$$

The complete PPO loss function is given in Equation 3.12, where c_1 and c_2 are coefficients that balance the contributions of the value loss and entropy bonus, respectively.

$$L(\theta) = \mathbb{E}_t \left[L^{CLIP}(\theta) - c_1 L^{VF}(\theta) + c_2 H[\pi_\theta(\cdot|s_t)] \right] \tag{3.12}$$

Due to its strong performance, relative simplicity, and ease of implementation, PPO is used as the foundational algorithm for the experiments conducted in this thesis.

3.2 Adversarial Training

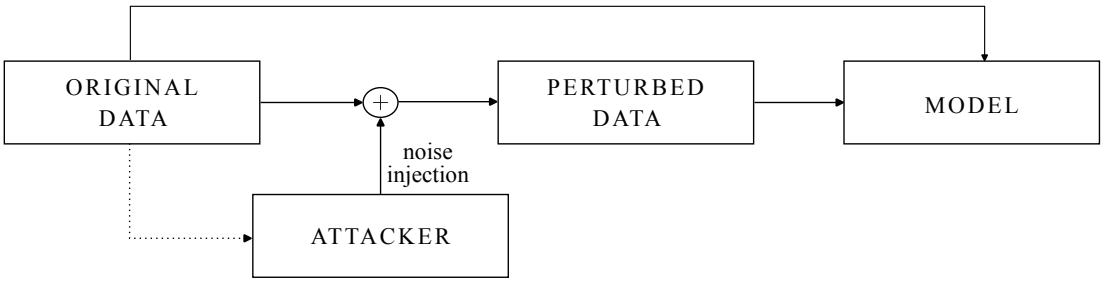

Figure 3.2: Adversarial training framework.

Adversarial training is a learning strategy where an agent or model is deliberately exposed to challenging, perturbed, or misleading inputs during training. The goal is to make the system more resilient by forcing it to adapt to difficult situations rather than only learning from standard examples. This idea stems from the observation that models can perform well under normal conditions but fail when faced with small changes or

adversarial manipulations. Through repeated exposure to these harder scenarios during training, adversarial methods help the agent or model develop stronger, more reliable behaviors that generalize better to real-world challenges. The general framework of adversarial training is illustrated in Figure 3.2.

3.2.1 Zero-sum games

Zero-sum games are often used to model competitive scenarios where the interests of the players are directly opposed, such as in certain types of games or economic situations. In adversarial training, framing the interaction as a zero-sum game helps in designing strategies where the agent learns to perform well even when faced with an adversary that is actively trying to undermine its success. A zero-sum game between two agents can be defined by the tuple $< S, A_1, A_2, P, R, \gamma >$, where:

- S is a finite set of states representing all possible situations the agents can encounter.
- A_1 and A_2 are finite sets of actions available to Agent 1 and Agent 2, respectively.
- P is the state transition probability function, where $P(s_{t+1}|s_t, a_t^1, a_t^2)$ denotes the probability of transitioning to state s_{t+1} from state s_t after Agent 1 takes action a_t^1 and Agent 2 takes action a_t^2 at time step t.
- R is the reward function, where $R(s_t, a_t^1, a_t^2)$ gives the expected reward received by Agent 1 after taking action a_t^1 in state s_t while Agent 2 takes action a_t^2. In a zero-sum game, the reward for Agent 2 is merely the additive inverse of Agent 1's reward, $R_2(s_t, a_t^2, a_t^1) = -R_1(s_t, a_t^1, a_t^2)$.
- γ is the discount factor, similar to that in standard MDPs, which sets the relative value of future rewards in comparison to immediate ones.

Equation 3.13 formalizes the objective of each agent to find a policy that maximizes its own expected cumulative reward while minimizing the expected cumulative reward of the opponent.

$$\arg\max_{\pi_1} \min_{\pi_2} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t^1, a_t^2)\right] \tag{3.13}$$

This competitive dynamic encourages the development of robust strategies that can withstand against the opponent's actions.

3.2.1.1 Nash equilibrium

A Nash Equilibrium is a concept from game theory [83] that describes a stable state in a strategic interaction where no player can gain an advantage by unilaterally altering their strategy, assuming the strategies of the other players remain unchanged. Put differently, at a Nash Equilibrium, the strategy of each player is optimal in response to the strategies chosen by the other players [84]. This means that if all players are at a Nash Equilibrium, no single player has an incentive to deviate from their current strategy, as doing so would not lead to a better outcome for them. Formally, a strategy profile $(\pi_1^*, \pi_2^*, \ldots, \pi_n^*)$ is a Nash Equilibrium if for every player i:

$$\mathbb{E}\left[R_i(s, a^1, a^2, \ldots, a^n) \mid \pi_i^*, \pi_{-i}^*\right] \geq \mathbb{E}\left[R_i(s, a^1, a^2, \ldots, a^n) \mid \pi_i, \pi_{-i}^*\right] \tag{3.14}$$

for all possible strategies π_i of player i, where π_{-i}^* denotes the strategies of all players except player i. The existence of Nash Equilibria in various types of games is a fundamental result in game theory, and it provides insights into the behavior of rational agents in competitive environments.

3.2.1.2 Minimax theorem

The Minimax Theorem [85], established by John von Neumann, is a fundamental result in game theory that applies to zero-sum games. It states that in such games, there exists a pair of optimal strategies for both players that minimizes the maximum possible loss for each player. In other words, each player can choose a strategy that guarantees the best outcome, regardless of the opponent's actions. Formally, for a two-player zero-sum game the theorem can be expressed as follows:

$$\max_{\pi_1} \min_{\pi_2} \mathbb{E}\left[R(s, a^1, a^2)\right] = \min_{\pi_2} \max_{\pi_1} \mathbb{E}\left[R(s, a^1, a^2)\right] \tag{3.15}$$

where π_1 and π_2 are the strategies (or policies) of players 1 and 2, respectively, and $R(s, a^1, a^2)$ is the reward function. The left-hand side represents the maximum expected reward that player 1 can guarantee by choosing an optimal strategy, while the right-hand side represents the minimum expected reward that player 2 can enforce by choosing an optimal strategy. The theorem implies that both players have strategies that lead to a

stable outcome, known as the value of the game, where neither player can improve their situation by unilaterally changing their strategy.

A minimax equilibrium in a two-player zero-sum Markov game exists under the following conditions:

1. The state and action spaces are finite, ensuring a limited number of possible states and actions.
2. The reward function is bounded, meaning there are upper and lower limits to the rewards that can be received.
3. The transition dynamics satisfy the Markov property, where the next state depends only on the current state and actions taken by both players.

Under these conditions, it can be shown that there exists a pair of strategies for both players that form a minimax equilibrium, where neither player can improve their expected outcome by unilaterally changing their strategy. This equilibrium represents a stable solution to the game, where both players are playing optimally given the strategies of their opponents.

3.2.2 Types of adversarial attacks

Adversarial attacks that utilized to train robust agents can be categorized based on various criteria, including the nature of the perturbations, the knowledge of the attacker, and the target of the attack.

3.2.2.1 Attacks classified by knowledge

Adversarial attacks can be classified based on the level of knowledge the attacker has about the target model. The three main categories are:

- White-box attacks: In white-box attacks, the attacker has access to complete knowledge of the target model, that includes the parameters, architecture, and the input data. This enables the attacker to compute gradients and generate precise adversarial examples that can successfully deceive the target model. Examples

methods include the Fast Gradient Sign Method (FGSM) [14] and Projected Gradient Descent (PGD) [58].

- Gray-box attacks: In gray-box attacks, the attacker has partial knowledge of the target model. This could include information about the model architecture or access to some training data, but not the full details. Gray-box attacks often involve techniques that exploit known vulnerabilities or use surrogate models to generate adversarial examples. An example of a gray-box attack is the Transfer Attack [86], where adversarial examples are generated using a different model and then tested on the target model.

- Black-box attacks: In black-box attacks, the attacker has no knowledge of the target model's internal workings. Instead, the attacker can only observe the model's outputs (predictions) in response to various inputs. Black-box attacks often rely on techniques such as query-based methods or transferability of adversarial examples from surrogate models. Examples include the Zeroth Order Optimization (ZOO) attack [87] and the Boundary Attack [88].

3.2.2.2 Attacks classified by goal

Adversarial attacks can also be classified based on the attacker's objectives. The two primary categories are:

- Targeted attacks: In targeted attacks, the attacker aims to manipulate the input in such a way that the model produces a specific, incorrect output. The goal is to force the model to misclassify the input into a predetermined target class. Targeted attacks are often more challenging to execute, as they require precise perturbations to achieve the desired misclassification. An example of a targeted attack is the Carlini & Wagner (C&W) attack, which generates adversarial examples that are specifically designed to be classified as a chosen target class [89].

- Untargeted attacks: In untargeted attacks, the attacker seeks to cause the model to misclassify the input into any class other than the correct one. The objective is to simply degrade the model's performance by increasing the overall error rate, without aiming for a specific incorrect output. Untargeted attacks are often easier to perform,

as they do not require the same level of precision as targeted attacks. An example of an untargeted attack is the Fast Gradient Sign Method (FGSM), which perturbs the input to maximize the loss function, leading to misclassification [14].

3.2.2.3 Attacks classified by timing

Adversarial attacks can also be categorized based on the timing of the attack in relation to the model's training and deployment phases. The two main categories are:

- Training-time (poisoning) attacks: Poisoning attacks occur during the training phase of a model, where the attacker manipulates the training data to introduce vulnerabilities or biases into the model. The goal of poisoning attacks is to degrade the model's performance or to create backdoors that can be exploited later. These attacks can involve injecting malicious samples into the training dataset or altering existing samples to mislead the learning process.
- Test-time (evasion) attacks: Evasion attacks occur during the deployment phase of a model, where the attacker crafts adversarial examples to evade detection or mislead the model into making incorrect predictions. These attacks are designed to exploit the model's vulnerabilities in real-time, often by adding small, imperceptible perturbations to the input data. Evasion attacks are particularly relevant in scenarios such as image recognition, where an attacker might modify an image to cause a misclassification.

3.2.2.4 Attacks classified by perturbation type

Adversarial attacks can also be classified based on the type of perturbations applied to the input data. The main categories include:

- Additive perturbations: Additive perturbations involve adding a small, carefully crafted noise vector to the original input data. The goal is to create an adversarial example that is visually similar to the original input but causes the model to misclassify it. Examples of additive perturbation methods include the Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD).
- Geometric perturbations: Geometric perturbations involve applying transformations to the input data, such as rotations, translations, or scaling. These perturbations

can alter the spatial structure of the input while maintaining its overall content. Geometric perturbations can be particularly effective in scenarios where the model is sensitive to changes in the input's orientation or position. An example of a geometric perturbation method is the Spatial Transformation Attack [90].

- Semantic perturbations: Semantic perturbations involve making changes to the input data that alter its meaning or context, rather than just its appearance. These perturbations can include modifications to the content of the input, such as changing words in a text or altering objects in an image. Semantic perturbations can be more challenging to detect, as they may not involve obvious visual changes. An example of a semantic perturbation method is the Textual Adversarial Attack [91], which modifies words in a sentence to change its meaning while preserving grammatical correctness.

3.2.2.5 Attack categories in reinforcement learning

In RL, the adversarial attacks can be classified according to the target of the attack. The main categories include:

- State perturbation attacks: These attacks involve manipulating the state observations perceived by the agent. The goal is to alter the agent's perception of the environment, leading to suboptimal decision-making. State perturbation attacks can be executed by adding noise to the state inputs or by crafting specific adversarial states that mislead the agent.

- Action perturbation attacks: These attacks focus on manipulating the actions taken by the agent. The attacker may interfere with the action selection process, causing the agent to execute suboptimal or harmful actions. Action perturbation attacks can be implemented by altering the action outputs or by injecting noise into the action space.

- Reward manipulation attacks: These attacks involve tampering with the reward signals received by the agent. By modifying the rewards, the attacker can mislead the agent's learning process, causing it to converge to suboptimal policies. Reward

manipulation attacks can be carried out by adding noise to the reward signals or by crafting specific reward structures that misguide the agent.

- Environment manipulation attacks: These attacks target the environment itself, altering its dynamics or properties to create challenges for the agent. Environment manipulation attacks can involve changing transition probabilities, modifying state dynamics, or introducing obstacles that hinder the ability of the agent to learn effective policies.

3.3 Robust Reinforcement Learning

Robust Reinforcement Learning is a subfield of RL that focuses on ensuring agents can maintain reliable performance even when conditions deviate from the training setup. In practice, this means preparing agents to handle uncertainties, parameter variations, and environmental perturbations that they might face outside controlled settings. Some approaches to robust RL rely on adversarial training, where the environment actively challenges the agent, while others use different strategies to promote stability. The overarching aim is to produce agents that are not just effective in ideal situations but also trustworthy and adaptable in unpredictable or changing environments.

3.3.1 Robust Markov Decision Process (RMDP)

Robust Markov Decision Process (RMDP) is an extension of the standard MDP framework that incorporates uncertainty in the model's dynamics and rewards [92]. In RMDP, the transition probabilities and reward functions are not fixed but belong to predefined uncertainty sets. This allows the agent to account for potential variations in the environment, leading to more robust decision-making strategies. The goal in RMDP is to find a policy that maximizes the expected cumulative reward under the worst-case scenarios within these uncertainty sets. This approach helps in developing policies that are resilient to model misspecifications and environmental changes, making RMDP a valuable tool for robust reinforcement learning applications. The RMDP can be formally defined as a tuple $< S, A, \mathscr{P}, \mathscr{R}, \gamma >$, similar to the standard MDP. However, in RMDP:

- $\mathscr{P}$ is a set of possible transition probability functions, representing the uncertainty in the environment's dynamics.
- $\mathscr{R}$ is a set of possible reward functions, capturing the uncertainty in the reward structure.

The objective in RMDP is to find a policy $\pi : S \to A$ that maximizes the expected cumulative reward under the worst-case transition probabilities and reward functions, as formalized in Equation 3.16.

$$\arg\max_{\pi} \min_{P \in \mathscr{P}, R \in \mathscr{R}} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right] \tag{3.16}$$

This formulation encourages the development of policies that are robust to uncertainties in the environment, leading to more reliable performance in varied and unpredictable scenarios.

3.3.2 Noisy Action Robust Markov Decision Process (NR-MDP)

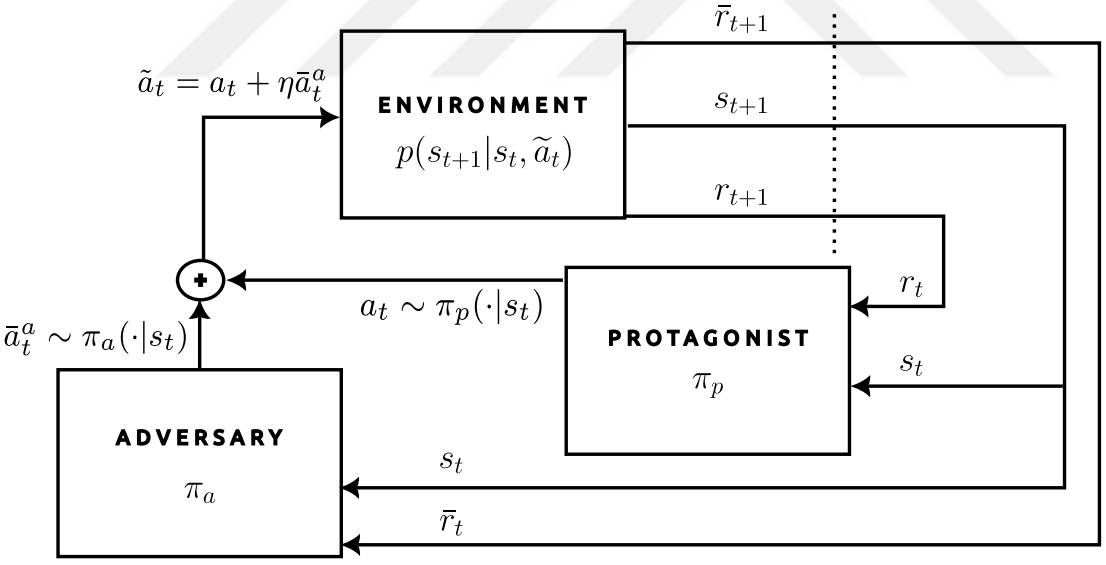

Figure 3.3: Structure of Noisy Action Robust Markov Decision Process (NR-MDP) framework [34].

The Noisy Action Robust Markov Decision Process (NR-MDP) [26] is a framework designed to enhance the robustness of RL agents against adversarial perturbations in the action space. NR-MDP introduces noise into the action selection process, simulating

potential disturbances that an agent may encounter in real-world scenarios. By training agents within this noisy framework, the goal is to improve their resilience to unexpected changes and ensure more stable performance across a range of conditions. The NR-MDP is modelled as a zero-sum game between the protagonist agent and an adversary, where the adversary aims to minimize the protagonist agent's performance by introducing noise to the actions. At each time step, the protagonist selects an action $a_t \sim \pi_p(\cdot|s_t)$, and the adversary selects a perturbation action $\bar{a}_t \sim \pi_a(\cdot|s_t)$. The applied action in the environment is a convex combination of these two actions, defined as $\tilde{a}_t = a_t + \eta\bar{a}_t$, where $\eta \in [0,1]$ controls the level/strength of noise introduced by the adversary. Figure 3.3 illustrates the structure of the NR-MDP framework.

3.3.3 State-Adversarial Markov Decision Process (SA-MDP)

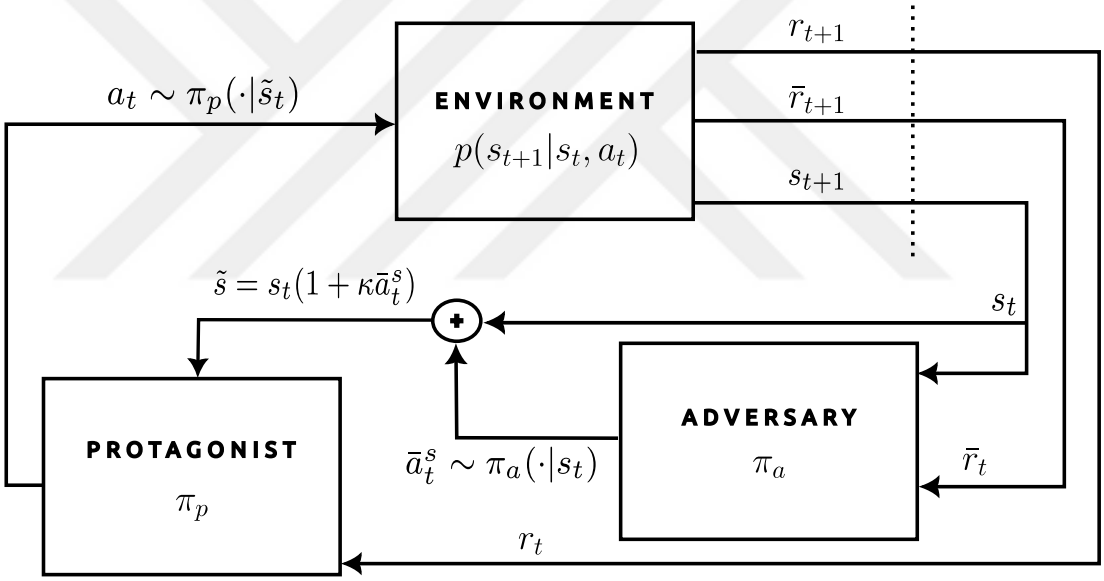

Figure 3.4: Structure of State-Adversarial Markov Decision Process (SA-MDP) framework [34].

The State-Adversarial Markov Decision Process (SA-MDP) [24] is a framework designed to enhance the robustness of RL agents against adversarial perturbations in the state observations. SA-MDP propose an adversary that manipulates the input state received by the agent, simulating potential disturbances that may occur in real-world scenarios. By training agents within this adversarial framework, the goal is to improve their resilience to unexpected changes in the environment and ensure more stable

performance across a range of conditions. The SA-MDP is modeled as a zero-sum game between the protagonist agent and an adversary, where the adversary aims to minimize the protagonist agent's performance by perturbing the state observations. At each time step, the adversary selects a perturbation action $\bar{a}_t^s \sim \pi_a(\cdot|s_t)$, and the perturbed state observed by the agent is defined as $\tilde{s}_t = s_t(1 + \kappa\bar{a}_t^s)$. The protagonist then selects an action $a_t \sim \pi_p(\cdot|\tilde{s}_t)$ based on this perturbed state. It is important to note that the true state s_t is not directly accessible to the protagonist and remains unchanged; only the agent's perception is altered. Figure 3.4 illustrates the structure of the SA-MDP framework.

3.3.4 Robustness evaluation methods

Assessing the robustness of RL agents is a fundamental step toward ensuring their reliability, stability, and performance in the presence of adversarial perturbations or environmental uncertainties. A range of methodologies has been proposed to evaluate the robustness of RL agents. In this thesis, the emphasis is placed on evaluation-based approaches, which include the following methods:

- Adversarial testing: This method involves subjecting the RL agent to adversarial attacks during the evaluation phase. By introducing perturbations to the state or action space, the agent's ability to maintain performance under challenging conditions can be assessed. Metrics such as attack success rate (ASR) and reward degradation can be used to quantify the impact of these attacks on the agent's performance.

- Sensitivity analysis: This method involves systematically varying the parameters of the environment or the agent's policy to assess how sensitive the agent's performance is to these changes. By analyzing the agent's response to different levels of perturbations, insights into its robustness can be gained.

- Cross-validation with different adversaries: Testing the RL agent against a variety of adversarial strategies can provide a comprehensive evaluation of its robustness. By exposing the agent to various forms of attacks, including state perturbations, action

perturbations, and reward manipulations, its ability to adapt and maintain performance can be thoroughly assessed.

4. RESEARCH METHODOLOGY

In this section, the methodological foundation of the proposed approach to enhancing the robustness of RL agents against sophisticated adversarial perturbations in dynamic environments is described. Motivated by the inherent vulnerabilities of traditional single-type adversarial attacks, which fail to capture the multi-modal nature of real-world threats, the discussion first elucidates the key limitations of such methods to underscore the need for a more comprehensive paradigm. Building upon a formal problem formulation and a set of practical assumptions, the concept of hybrid adversarial attacks is introduced—integrating perturbations across both action and state spaces. To rigorously model these interactions, the classical MDP is extended into an Action and State-Adversarial Markov Decision Process (ASA-MDP) framework. Finally, the section presents the core contribution of this study: the Adversarial Synthesis and Adaptation Framework, which enables systematic generation and adaptive mitigation of hybrid threats, paving the way for enhanced resilience in safety-critical applications.

4.1 Limitations of Single-Type Adversarial Attacks

Adversarial training has emerged as a prominent technique for enhancing the robustness of RL agents against adversarial attacks. The fundamental idea is to expose the agent to adversarial perturbations during training, allowing it to learn policies that can withstand such disruptions at test time. This approach has been shown to improve robustness significantly. However, most existing adversarial training methods focus on single-type attacks, such as perturbations to either the observation space or the action space, but not both simultaneously. This limitation is critical as it fails to account for the challenges of real-world scenarios where adversaries may exploit multiple vulnerabilities concurrently.

Intuitively, training against one type of perturbation may cause overfitting to that specific attack modality, resulting in a false sense of security. For example, an agent trained

to be robust against observation perturbations may still be vulnerable to action-space attacks, and vice versa. This gap in robustness can be exploited by adversaries that can adjust their strategies according to the known defenses of the agent. Furthermore, the interactions between different types of perturbations are not well understood. It is unclear whether robustness to one type of attack confers any resilience against another, or if the combined effect of multiple perturbations is simply additive. This uncertainty complicates the design of effective defense mechanisms.

All these limitations emphasize the necessity for a more holistic approach to adversarial training that considers the interplay between different types of attacks. By addressing these challenges, RL agents can be developed to exhibit robustness not only against individual perturbation types but also resilience to complex, multi-modal adversarial strategies. This motivation underpins the proposed methodology, which aims to bridge this gap through the introduction of the Action and State-Adversarial Markov Decision Process (ASA-MDP) and the Adversarial Synthesis and Adaptation Framework.

4.2 Assumptions and Problem Formulation

The proposed framework formulates hybrid adversarial attacks as a zero-sum game between a protagonist and an adversary. In this formulation, the adversary functions under a black-box assumption, lacking access to the protagonist's model architecture and parameters during attack generation. The adversary's sole feedback for assessing attack efficacy is the negation of the reward achieved by the protagonist within the environment. Consequently, while the protagonist optimizes to maximize its cumulative reward, the adversary concurrently optimizes to minimize it, leading to an adversarial optimization process over the reward function.

Both the protagonist and adversary are modeled as distinct policies, each parameterized by a separate neural network. The protagonist policy, denoted as π_p with parameters θ, maps states to actions, while the adversary policy, denoted as π_a with parameters ϕ, maps states to perturbations in both the state and action spaces. The adversary's goal is to learn perturbations that effectively degrade the protagonist's performance. The

ultimate goal is to find a robust policy for the protagonist that can withstand the (near) worst-case mixed perturbations generated by the adversary.

4.3 Action and State Adversarial Markov Decision Process (ASA-MDP)

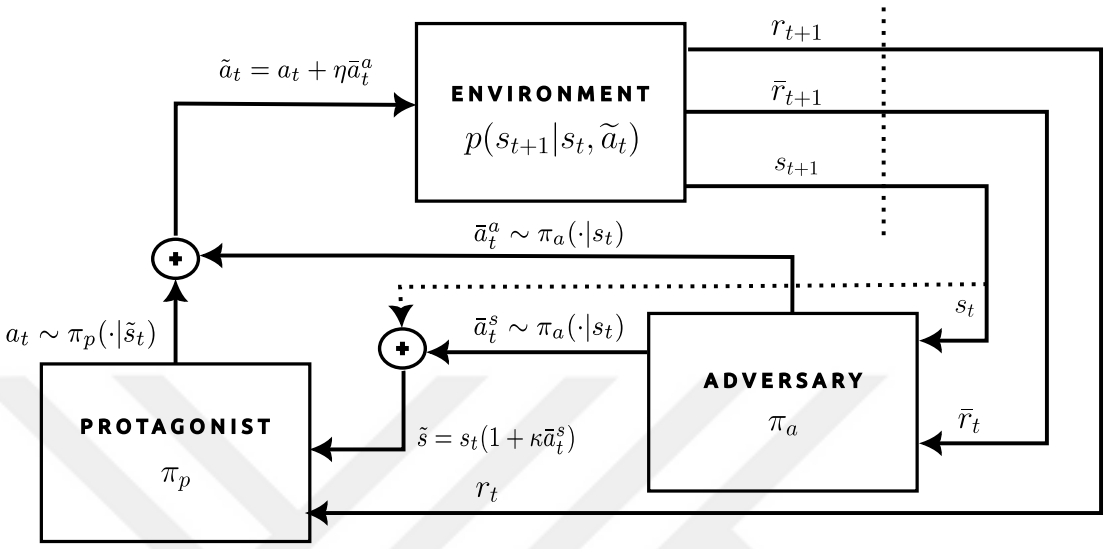

Figure 4.1: Structure of Action and State Adversarial MDP (ASA-MDP) framework [34].

To rigorously model the interaction between the protagonist and adversary in the presence of mixed perturbations, the classical Markov Decision Process (MDP) framework is extended to an Action and State-Adversarial MDP (ASA-MDP). An ASA-MDP is defined as a tuple $(S, A_p, A_a, P, R, \gamma, B_s, B_a)$, where:

- S is the set of states.
- A_p is the set of actions available to the protagonist.
- A_a is the set of perturbations available to the adversary.
- $P : S \times A_p \times A_a \times S \to [0,1]$ is the state transition probability function, which now depends on both the protagonist's action and the adversary's perturbation.
- $R : S \times A_p \times A_a \to \mathbb{R}$ is the reward function, which also depends on both the protagonist's action and the adversary's perturbation.
- $\gamma \in [0,1)$ is the discount factor.

- B_s is the set of allowable state perturbations.
- B_a is the set of allowable action perturbations.

ASA-MDP is a variant of RMDP in which the equilibrium properties established by Shapley [93] remain valid. The key difference between ASA-MDP and RMDP lies in the source of adversarial influence: in RMDP, the ground-truth transition dynamics are modified by the adversary, while the agent continues to observe the true states. In contrast, in ASA-MDP, the adversary manipulates the observed states (without changing the true states), leaving the underlying transition dynamics unchanged. The structure of ASA-MDP is illustrated in Figure 4.1.

$$\tilde{s}_t := s_t \cdot (1 + \kappa \bar{a}_t^s) \quad \text{where} \quad \tilde{s}_t \in B_s \tag{4.1}$$

$$\tilde{a}_t := a_t + \eta \bar{a}_t^a \quad \text{where} \quad \tilde{a}_t \in B_a \tag{4.2}$$

Equation 4.1 and 4.2 define the perturbation mechanism for states and actions, respectively. Here, $\tilde{s}_t$ and $\tilde{a}_t$ represent the perturbed state and action at time step t, while s_t and a_t denote the original state and action. The variables $\bar{a}_t^s$ and $\bar{a}_t^a$ are the adversarial perturbation actions generated by the adversary for the state and action spaces, respectively. $\bar{a}_t = (\bar{a}_t^s, \bar{a}_t^a)$ denotes the combined state and action perturbation generated by the adversary at time step t. The scalars κ and η control the magnitude of the perturbations, ensuring that they remain within the predefined bounds B_s and B_a. This formulation allows for a flexible and comprehensive modeling of mixed adversarial attacks, capturing the simultaneous influence of perturbations in both the state and action spaces.

Multiplicative perturbations are applied to states and additive perturbations to actions to capture their differing semantic roles and typical attack modalities. State observations correspond to environmental measurements that are naturally affected by proportional distortions (for example, sensor scaling or illumination changes); multiplicative perturbations therefore preserve relative structure while probing robustness to scale-like adversarial manipulations. By contrast, actions correspond to control commands that

are more plausibly compromised via direct signal injection or bias, making additive perturbations a more faithful model of actuator-level corruptions. Distinguishing perturbation types in this manner aligns the attack model with domain-specific noise characteristics and increases the realism of the adversarial scenarios. To formally specify the perturbation sets and enforce magnitude bounds $(\varepsilon_a, \varepsilon_s)$, clipping is applied to the adversarial policy's outputs, projecting perturbations onto the admissible range and thereby constraining their magnitudes as in Equations 4.3 and 4.4.

$$B_s(s) = \tilde{s} \in S : ||\tilde{s} - s||_\infty \leq \varepsilon_s \tag{4.3}$$

$$B_a(s) = \tilde{a} \in A : ||\tilde{a} - a||_\infty \leq \varepsilon_a \tag{4.4}$$

It is important to emphasize that the reward and transition functions depend directly on the perturbed action $\tilde{a}_t$ and the true state s_t, while their dependence on the perturbed state $\tilde{s}_t$ is indirect. This indirect dependence arises because the protagonist determines its action a_t based solely on the perturbed state $\tilde{s}_t$, and this action subsequently affects both the reward and transition dynamics through the perturbed action $\tilde{a}_t$. The exact mathematical definition of the reward and transition function are shared in Equations 4.5 and 4.6.

$$\begin{aligned} r(s_t, \tilde{a}_t) &:= r(s_t, a_t + \eta \bar{a}_t^a) \\ &:= r(s_t, \pi_p(a_t|\tilde{s}_t) + \eta \bar{a}_t^a) \\ &:= r(s_t, \pi_p(a_t|s_t \cdot (1 + \kappa \bar{a}_t^s)) + \eta \bar{a}_t^a) \end{aligned} \tag{4.5}$$

$$\begin{aligned} P(s_{t+1}|s_t, \tilde{a}_t) &:= P(s_{t+1}|s_t, a_t + \eta \bar{a}_t^a) \\ &:= P(s_{t+1}|s_t, \pi_p(a_t|\tilde{s}_t) + \eta \bar{a}_t^a) \\ &:= P(s_{t+1}|s_t, \pi_p(a_t|s_t \cdot (1 + \kappa \bar{a}_t^s)) + \eta \bar{a}_t^a) \end{aligned} \tag{4.6}$$

4.3.1 Relation to NR-MDP and SA-MDP

The proposed ASA-MDP framework generalizes and unifies existing robust MDP formulations, specifically the Noisy Action Robust MDP (NR-MDP) and State Adversarial MDP (SA-MDP). NR-MDP focuses solely on action-space perturbations, modeling scenarios where the agent's actions are corrupted by noise or adversarial

interference. In contrast, SA-MDP addresses state-space perturbations, capturing situations where the agent's observations in the environment are manipulated. ASA-MDP extends these frameworks by simultaneously incorporating both action and state perturbations, thereby providing a more comprehensive model of adversarial interactions. This dual-perturbation approach enables the analysis and development of robust policies that can withstand a broader range of adversarial strategies, reflecting the complex and multi-modal nature of real-world threats.

The ASA-MDP reduces to the NR-MDP when $\kappa = 0$, to the SA-MDP when $\eta = 0$, and to the standard MDP when both $\kappa = 0$ and $\eta = 0$. The ASA-MDP is formulated for tasks characterized by continuous state and action spaces. Rather than attempting to solve the inherently intractable minimax optimization problem exactly, the proposed method aims to approximate a Nash equilibrium by jointly learning stationary policies π_p^* and π_a^* [94].

4.4 Challenges in Balancing Mixed Perturbations

Designing robust RL agents capable of withstanding mixed adversarial attacks presents several significant challenges. First, the complexity of the attack space increases substantially when considering perturbations in both state and action spaces. This expansion results in a combinatorial increase in the number of potential attack strategies, making it difficult to anticipate and defend against all potential threats. Second, the interactions between state and action perturbations are not well understood. The attacks on action spaces are directly influence the transition dynamics and rewards of the environment, while state perturbations affect the agent's perception of the environment, which in turn influences its action choices. These two types of perturbations operate on different levels of the decision-making process. It is unclear how these different types of perturbations influence each other and the overall performance of the agent. This lack of understanding complicates the development of effective defense mechanisms. Third, training RL agents to be robust against mixed attacks requires significant computational resources. The need to simulate a wide range of adversarial scenarios during training can lead to increased training times and resource consumption. Finally, evaluating the robustness of RL agents against hybrid attacks is challenging. Standard evaluation

metrics may not adequately capture the agent's performance under complex adversarial conditions, making it difficult to assess the effectiveness of proposed defense strategies. Addressing these challenges is crucial for advancing the field of robust RL and ensuring the practical effectiveness of RL agents in real-world applications.

4.5 Adversarial Synthesis and Adaptation Framework

To address the challenges associated with mixed adversarial attacks, the Adversarial Synthesis and Adaptation Framework is proposed. This framework enables the systematic generation of hybrid perturbations and the adaptive training of RL agents to withstand such attacks. The core components of the framework include a dual-policy architecture, where separate neural networks are employed for the protagonist and adversary, and a training algorithm that iteratively updates both policies based on their interactions within the ASA-MDP environment. The framework also incorporates mechanisms for balancing the influence of state and action perturbations, allowing for the dynamic adjustment of perturbations during training. This adaptability is crucial for ensuring that the agent learns to cope with a diverse range of adversarial strategies. By integrating these components, the Adversarial Synthesis and Adaptation Framework provides a robust foundation for developing RL agents that can effectively navigate and perform in environments subject to complex, multi-modal adversarial threats.

As discussed in Section 2, adversarial training has been shown to be effective in improving the robustness of RL agents. However, existing approaches often focus on either state or action perturbations in isolation, neglecting the potential interactions between these two types of perturbations. The framework addresses this gap by explicitly modeling hybrid perturbations and their effects on the agent's performance. The min-max high-level objective of the framework is defined in Equation 4.7, which captures the nature of the problem in three steps: general formulation, including the protagonist and adversary policies and finally, the mixed perturbation modeling with explicit scaling factors.

$$
\begin{aligned}
&\max_{\theta}\min_{\phi}\mathbb{E}\left[\sum_{t=0}^{T}\gamma^{t} r\left(s_t, \tilde{a}_t\right)\right] \\
&\max_{\theta}\min_{\phi}\mathbb{E}\left[\sum_{t=0}^{T}\gamma^{t} r\left(s_t, \pi_p(a_t|\tilde{s};\theta) + \eta\,\pi_a(\bar{a}_t^{a}|s_t;\phi)\right)\right] \\
&\max_{\theta}\min_{\phi}\mathbb{E}\left[\sum_{t=0}^{T}\gamma^{t} r\left(s_t, \pi_p\big(a_t|s_t\cdot(1+\kappa\pi_a(\bar{a}_t^{s}|s_t;\phi));\theta\big)\right.\right. \\
&\qquad\qquad\left.\left. + \eta\,\pi_a(\bar{a}_t|s_t;\phi)\right)\right]
\end{aligned}
\tag{4.7}
$$

In this practical consideration of the Markov Game, the policies are represented as neural networks. Given the non-convex nature of deep neural network parameter spaces, the existence of a global minimax equilibrium cannot be guaranteed. Consequently, the adversarial training algorithm is designed to identify a local saddle point. This approach aligns with common practices in DRL and generative adversarial networks, where algorithms have been empirically shown to converge to high-quality local optima despite the absence of theoretical guarantees of global convergence.

Because a constant or rule-based adversary can be easily overpowered by the protagonist, ASA employs a learning-based adversary that adapts its strategy based on the protagonist's behavior, following approaches similar to [50,65]. This dynamic interaction encourages the protagonist to develop more robust policies that can withstand a wider range of adversarial strategies. The adversary is trained to identify and exploit the protagonist's weaknesses, while the protagonist learns to anticipate and counteract the adversary's tactics. This co-evolutionary process leads to the emergence of sophisticated strategies on both sides, ultimately resulting in a more resilient protagonist policy. The important thing to note is to not update the adversary and protagonist simultaneously, as this can lead to instability in training. Instead, updates are alternated between the two policies, allowing each to adapt to the latest strategy of the other. More specifically, during training, the parameters of the adversary are held fixed while the protagonist is updated for several iterations. Then, the roles are reversed, with the protagonist's parameters held fixed while the adversary is updated. The alternating process continues until either an approximate equilibrium is reached or a predefined maximum number of timesteps is exceeded, at which point the protagonist is considered

robust to the adversarial strategies learned during training. This approach maintains balance in the learning dynamics and promotes convergence to a stable solution.

At each time step, the adversary first observes the state s_t and generates the perturbation actions $\bar{a}_t^s$ and $\bar{a}_t^a$. Following Equation 4.1, the perturbed state $\tilde{s}_t$ is computed and presented to the protagonist. The protagonist then selects an action a_t based on this perturbed state, which is subsequently modified by $\bar{a}_t^s$ and $\bar{a}_t^a$ according to Equation 4.2. The resulting perturbed action $\tilde{a}_t$ is executed in the environment, yielding the corresponding reward r_t. Algorithm 1 summarizes the complete training procedure of the Adversarial Synthesis and Adaptation Framework.

Algorithm 1: Adversarial Synthesis and Adaptation Framework [34]

Input: environment ε, number of iterations N_{iter}, perturbation sets B_s, B_a, policies π_p, π_a, batch size B, buffers D_{π_p}, D_{π_a}
Initialize: policy parameters θ for π_p, ϕ for π_a, $n = 0$
while *not converged or* $n < N_{iter}$ **do**
  $s_0 \leftarrow$ reset environment ε
  $D_p \leftarrow \emptyset$, $D_a \leftarrow \emptyset$
  for $t = 0, 1, \ldots, B$ **do**
    $\bar{a}_t^a, \bar{a}_t^s := \pi_a(\cdot|s_t)$; `// compute adversarial attacks`
    compute $\tilde{s}_t$ according to Equation 4.1
    $a_t := \pi_p(\cdot|\tilde{s}_t)$; `// compute protagonist action`
    compute $\tilde{a}_t$ according to Equation 4.2
    $r_t, s_{t+1} \leftarrow \varepsilon(s_t, \tilde{a}_t)$; `// iterate environment`
    compute $\tilde{s}_{t+1}$ according to Equation 4.1
    $D_{\pi_p} \leftarrow D_{\pi_p} \cup \{(\tilde{s}_t, a_t, r_t, \tilde{s}_{t+1})\}$
    $D_{\pi_a} \leftarrow D_{\pi_a} \cup \{(s_t, \bar{a}_t, r_t, s_{t+1})\}$
  if *n is even* **then**
    $\theta \leftarrow PolicyOptimizer(D_{\pi_p}, \pi_p, \theta)$
  else
    $\phi \leftarrow PolicyOptimizer(D_{\pi_a}, \pi_a, \phi)$
  $n = n + 1$

4.5.1 Policy representations (protagonist vs adversary)

Both the protagonist and adversary policies are modeled using neural networks, allowing them to learn complex mappings from states to actions and perturbations, respectively. A straightforward approach would be to use an actor network for the adversary that outputs both state and action perturbations as a concatenated vector. However, this naive design implicitly assumes that the two types of perturbations operate with the same

objectives, which can lead to suboptimal performance due to their differing natures and objectives. In addition, in the absence of any regularization on the magnitude or frequency of the perturbation attacks, the adversary may learn to focus predominantly on one type of perturbation (either state or action), effectively ignoring the other. This imbalance can result in a protagonist that is robust to only one type of attack, leaving it vulnerable to the neglected perturbation type.

To address this limitation, ASA employs a multi-head architecture with connections linking the two heads in the actor network, as illustrated in Figure 4.2. In this design, shared base layers process the input state to extract common features, which are then passed to two separate output heads. One head is dedicated to generating state perturbations, while the other focuses on action perturbations. This separation enables each head to specialize in its respective task while still leveraging shared representations through the common base network. As a result, the adversary can more effectively model the interactions between state and action perturbations, leading to improved overall robustness.

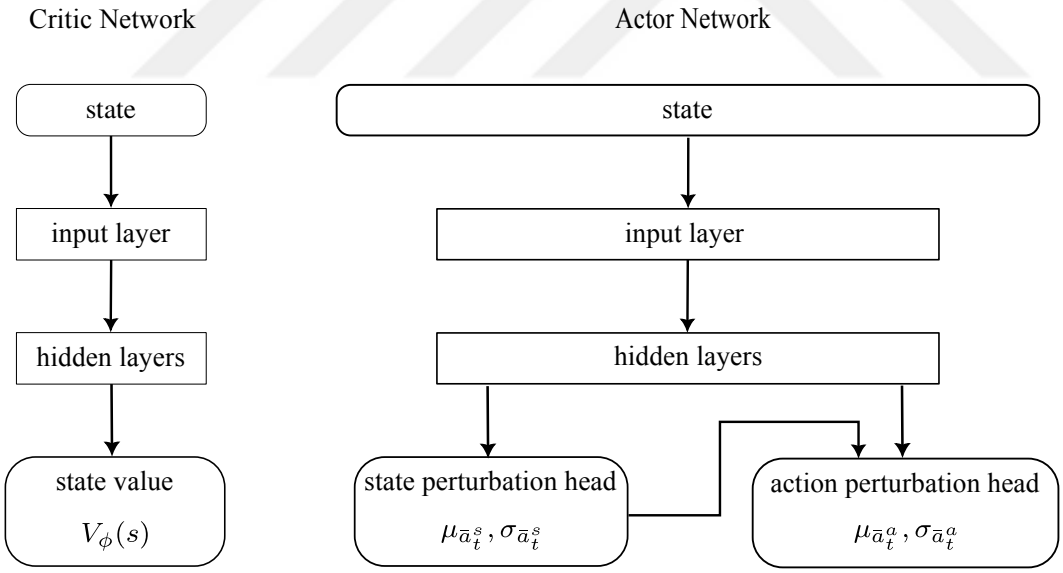

Figure 4.2: Adversarial multi-head PPO policy network structure [34].

Both the protagonist and adversary policies are trained using the Proximal Policy Optimization (PPO) algorithm [35], which is a state-of-the-art policy gradient method known for its stability and efficiency. The main challenge to achieve robustness

against mixed perturbations lies in the design of the adversary's objective function and the training procedure. The primary challenge in attaining robustness within the adversarial training framework lies in balancing the heterogeneous strengths of different attack modalities and in appropriately regularizing the adversarial policy to manage the resulting trade-offs during training. Owing to the modular nature of PPO, structural constraints or regularization terms can, in principle, be directly incorporated into the adversary's objective function, provided that the formulation remains differentiable and amenable to policy gradient optimization. However, the conducted preliminary experiments indicated that conventional regularization techniques developed for black-box settings were inadequate for producing well-balanced hybrid adversarial attacks.

The adversary's multi-head PPO policy architecture enforces balanced perturbations via an intrinsic regularization term that penalizes perturbation magnitude and promotes a desirable allocation between state and action perturbations. To capture the causal influence of the perturbed observation on the protagonist's subsequent control, the state-perturbation output is routed as an additional input to the action-perturbation head, enabling the adversary to model and exploit the dependency between $\tilde{s}_t$ and the resulting $\tilde{a}_t$.

The objective function for the adversary is defined similarly to the standard PPO objective, with modifications to account for the dual perturbation outputs and the intrinsic regularization. In practical implementation of the PPO, the policy ratio $r_t(\phi)$ between the current and old policies is computed in the log space, as shown in Equation 4.8.

$$r_t(\phi) = \frac{\pi_a(\bar{a}_t|s_t;\phi)}{\pi_a(\bar{a}_t|s_t;\phi_{old})} \approx log(\pi_a(\bar{a}_t|s_t;\phi)) - log(\pi_a(\bar{a}_t|s_t;\phi_{old})) \tag{4.8}$$

The log probability of the combined perturbation action is calculated as the average of the log probabilities from each head, as detailed in Equation 4.9. This averaging approach ensures that both state and action perturbations contribute equally to the overall policy update, promoting a balanced adversarial strategy. The actual advantage of the

logarithmic formulation is that it helps to stabilize the training process by mitigating the effects of extreme probability values that can arise in continuous action spaces.

$$log(\pi_a(\bar{a}_t|s_t)) = \frac{log(\pi_a(\bar{a}_t^s|s_t)) + log(\pi_a(\bar{a}_t^a|s_t))}{2} \tag{4.9}$$

In this formulation, the assumption of each head contributing equally to the overall perturbation is made for simplicity, but this can be adjusted based on the specific requirements of the task or environment. In the case of white-box attacks, where the adversary has access to the protagonist's policy, the adversary can directly observe the protagonist's actions and states. This access allows the adversary to tailor its perturbations more effectively via adjusting weights on the contribution of each head.

Similarly, the entropy term, which encourages exploration, is computed as the average entropy of both heads, as shown in Equation 4.10. This averaging ensures that the exploration incentives are balanced between state and action perturbations, preventing the adversary from overfitting to one type of perturbation.

$$H[\pi_a(\cdot \mid s_t)] = \frac{(-\sum_{\bar{a}^s} \pi_a(\bar{a}_t^s \mid s) \log \pi_a(\bar{a}_t^s \mid s_t), -\sum_{\bar{a}^a} \pi_a(\bar{a}_t^a \mid s) \log \pi_a(\bar{a}_t^a \mid s_t))}{2} \tag{4.10}$$

Because the value loss depends only on the current state, it is included in the overall objective once, as in standard PPO.

The advantage function $A^\pi(s_t, a_t)$ is estimated using Generalized Advantage Estimation (GAE) [95], which provides a balance between bias and variance in the advantage estimates, as outlined in Equation 4.11. In the formulation, T denotes the horizon, λ is the GAE parameter, and δ is the temporal difference error defined as in 4.12.

$$A^\pi(s_t, a_t) = \delta_t + \gamma\lambda\,\delta_{t+1} + \ldots + (\gamma\lambda)^{T-t+1}\delta_{T-1} = \delta_t + \gamma\lambda A^\pi(s_{t+1}, a_{t+1}) \tag{4.11}$$

$$\delta_t = r_t + \gamma V^\pi(s_{t+1}) - V^\pi(s_t) \tag{4.12}$$

In the ASA framework, the adversary functions as an end-to-end attacker, directly observing the protagonist's state (but not action) to generate perturbations. This setup

aligns with a black-box attack scenario, where the adversary has no access to the internal parameters or gradients of the protagonist's policy. The adversary's objective is to minimize the protagonist's expected return by crafting perturbations that exploit vulnerabilities in the protagonist's policy. The applied perturbations modify both the state that the protagonist observes and the action it executes, thereby influencing the environment's response and the resulting rewards. The adversary's learning process is driven by the feedback received from the environment, which reflects the impact of its perturbations on the protagonist's performance.

In practical applications, it is crucial to ensure that the adversary's perturbations remain within the predefined bounds B_s and B_a. This constraint is enforced through clipping mechanisms that project the adversarial outputs onto the admissible range, thereby maintaining the realism and feasibility of the perturbations.

In the case of excessive perturbations, the protagonist policy may become unstable or fail to learn effectively. To mitigate this risk, the perturbation magnitudes are carefully regulated during training, ensuring that they remain within a range that challenges the protagonist without overwhelming it. This balance is essential for fostering robust learning and preventing the adversary from dominating the training process.

Also, in the case of low perturbations, the adversary may struggle to find effective strategies to challenge the protagonist. This causes the learned protagonist policy to be less robust, as it may not be adequately exposed to challenging scenarios during training.

In the existing literature, it is common to determine the perturbation budgets $(\varepsilon_a, \varepsilon_s)$ beforehand. These budgets define the maximum allowable perturbations for both the state and action spaces, ensuring that the perturbations remain within realistic and feasible limits. By establishing these budgets in advance, researchers can better control the training dynamics and maintain a balanced challenge for the protagonist.

5. SIMULATIONS and DISCUSSIONS

This chapter presents the simulation setup and results of the proposed methods, along with a discussion of their implications. The experiments were designed to evaluate the performance of the proposed ASA framework and benchmark algorithms under various adversarial attack scenarios. The chapter begins with a description of the benchmark algorithms, environments, and evaluation metrics employed in the experiments. Implementation details are then provided, followed by a comprehensive analysis of the simulation results. Finally, the findings and their implications for the field of robust RL are discussed.

5.1 Simulations

This section presents a detailed examination of benchmark algorithms, simulation settings, environments (training and testing), evaluation metrics implementation details, and performance comparisons.

5.1.1 Benchmark algorithms

To evaluate the effectiveness of the proposed ASA framework, it is compared against several benchmark algorithms that represent different strategies for enhancing the robustness of RL agents. The selected benchmarks include:

- Non-Robust PPO: This is the standard Proximal Policy Optimization (PPO) [35] algorithm without any adversarial training or robustness enhancements. It serves as a baseline to assess the impact of adversarial training methods.
- Noise-Augmented PPO (NA-PPO): This algorithm introduces random mixed noise to the agent's action and observation during training. The noise is sampled from a uniform distribution within a specified range, aiming to improve the agent's resilience to action-space perturbations.

- Noisy Action Robust PPO (NR-PPO): This algorithm focuses on robustness against action-space perturbations. It incorporates adversarial training where the adversary perturbs the agent's actions during training, allowing the agent to learn policies that can withstand such disruptions. NA-PPO is the alternating training version of the method proposed by Tessler et al. [26].

- Alternating Training with Learned Adversaries PPO (ATLA-PPO) [65]: This method employs an alternating training strategy where the protagonist and adversary are updated in turns. The adversary applies perturbations to the agent's perceived state, affecting the observations received by the protagonist agent. This approach aims to improve robustness against single-type attacks.

- Naive Hybrid PPO (N-HYBRID-PPO): This benchmark extends the NR-PPO/ATLA-PPO approach by allowing the adversary to perturb both the state and action spaces during training. However, it does so without a specialized architecture or regularization to balance the two types of perturbations, potentially leading to suboptimal robustness.

These benchmark algorithms provide a comprehensive basis for evaluating the performance of ASA-PPO, which is designed to enhance robustness against hybrid adversarial attacks affecting both state and action spaces. All benchmarks— excluding NA-PPO—use learned perturbation strategies instead of fixed attack patterns, aligning with the methodology adopted in the proposed framework. By comparing ASA-PPO with these established methods, the effectiveness of the approach in improving the resilience of RL agents in dynamic and adversarial environments can be assessed.

5.1.2 Environments and evaluation metrics

The experiments were conducted in Mujoco Playground [96] framework which uses the Mujoco physics engine [97] for simulating continuous control tasks. The environments are based on DeepMind Control Suite [98], which provides a set of standardized tasks for evaluating RL algorithms. The selected environments are: PendulumSwingup, FingerSpin, and CheetahRun. These environments were chosen for their varying levels

of complexity and the challenges they present in terms of control and stability. The details of each environment are as follows:

- PendulumSwingup (Figure 5.1): This task involves swinging a pendulum from a downward position to an upright position and balancing it there. The state space consists of the pendulum's angle and angular velocity, while the action space is the torque applied to the pendulum. The reward is based on the angle of the pendulum, with higher rewards for being closer to the upright position.

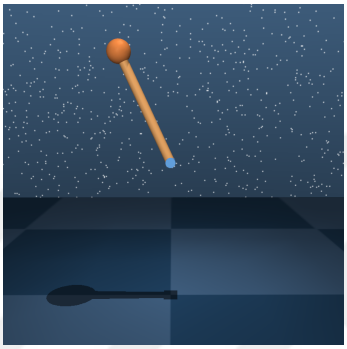

Figure 5.1: Illustration of PendulumSwingup environment [98].

- FingerSpin (Figure 5.2): In this task, a robotic finger with a 3-DoF (Degrees of Freedom) must rotate an unactuated hinge continuously. The state space includes the angles and angular velocities of the finger joints, while the action space consists of torques applied to these joints. The reward is given based on the angular velocity of the hinge, encouraging continuous rotation.

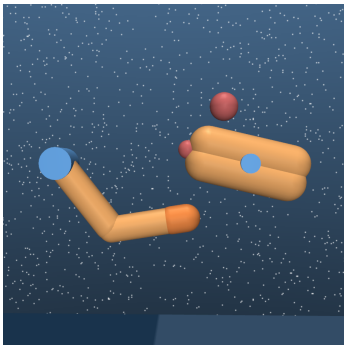

Figure 5.2: Illustration of FingerSpin environment [98].

- CheetahRun (Figure 5.3): This environment requires a simulated cheetah robot to run as fast as possible along a straight line. The state space includes the positions and velocities of various body parts, while the action space consists of torques applied to the joints. The reward is based on the forward velocity of the cheetah.

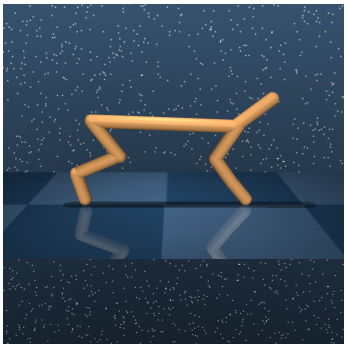

Figure 5.3: Illustration of CheetahRun environment [98].

5.1.3 Implementation details

In all training runs, running statistics of the observations and rewards are logged to normalize them to have zero mean and unit variance. The action space is also normalized to the range of $[-1.0, 1.0]$. The actor networks of the protagonist and adversary are both modeled using multi-layer perceptrons (MLPs) with two hidden layers, each containing 256 units and using the tanh activation function. The activation function is selected as tanh to ensure actions remain within the valid range. The critic networks are similarly structured but with no activation function at output layer, yielding a single output unit representing the state value.

Five perturbation scaling levels are determined for both action-space and observation-space attacks, denoted as $\eta, \kappa \in \{0.05, 0.10, 0.15, 0.20, 0.25\}$. These levels represent the maximum allowable perturbations that the adversary can apply to the protagonist's actions and observations, respectively. These scalings ensure that the perturbations are proportional to the magnitude of the actions and states, maintaining a consistent level of challenge across different environments.

The PPO implementation was adapted from the codebase presented in [99] to implement both the benchmark algorithms and the proposed method. The implementation supports vectorized PPO training on GPU using JAX [100]. The use of the JAX library and GPU

acceleration enables efficient numerical computation and automatic differentiation. Training was conducted on a system equipped with an NVIDIA V100 GPU and 256 GB of RAM. The hyperparameters were optimized for the non-robust setting and applied to all methods without modification, as summarized in Table 5.1. For certain environments, specific hyperparameters were adjusted to better align with task requirements, as indicated in the table.

Table 5.1: PPO hyperparameters for all methods [34].

Hyperparameters	Default	Environment-Specific
learning rate	1×10^{-3}	
discount factor	0.995	FingerSpin: 0.95
entropy cost coefficient	1×10^{-2}	
value cost coefficient	0.5	
max gradient norm	0.5	
normalize observation	True	
action repeat	1	PendulumSwingup: 4
unroll length	25	
number of minibatches	32	
number of updates per batch	4	PendulumSwingup: 8
number of environments	2048	
number of timesteps	2×10^{8}	FingerSpin: 1×10^{8}

5.1.4 Training

The training process involves alternating updates between the protagonist and the adversary. Each training iteration consists of collecting experience (s_t, a_t, r_t, s_{t+1}) by interacting with the environment, followed by updating the protagonist's or adversary's policy using PPO. The training continues until a predefined number of timesteps is reached or until the performance of the protagonist stabilizes. The training curves for each method under various attack scenarios are presented in Figure 5.4.

In the figure, the first column shows the training curves under no attack (Non-Robust PPO), the second column shows the curves under action-space attacks (NR-PPO), and the third column shows the curves under observation-space attacks (ATLA-PPO). Each row corresponds to a different environment: PendulumSwingup, FingerSpin, and CheetahRun from top to bottom, respectively. The solid curves represent the average return over training episodes, with shaded areas indicating the standard deviation across five different random seeds.

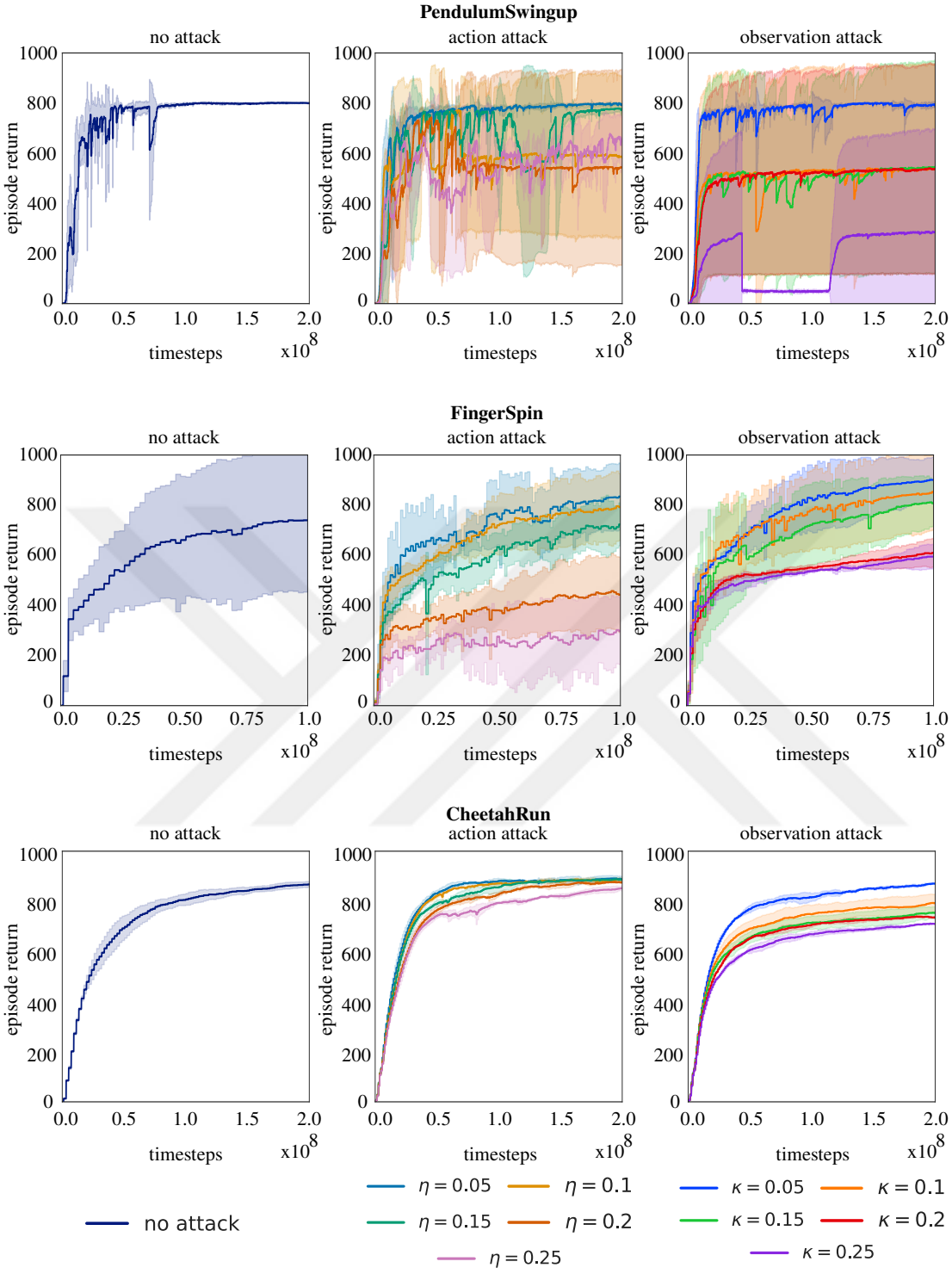

Figure 5.4: Learning curves of the agent under three conditions: no attack, action-space attacks, and observation-space attacks. Results are shown for varying attack budgets (η and κ) as mean ± standard deviation over 5 random seeds [34].

Similarly, Figure 5.5 illustrates the training curves for the proposed naive hybrid attack (N-HYBRID-PPO) and the proposed action and state hybrid attacks (ASA-PPO) under mixed adversarial attacks. In all benchmark algorithms and the hybrid attacking methods, the protagonist performance gets lower as the attack budget increases, which is expected as the adversary becomes more capable of disrupting the protagonist's actions and observations. Even though agents trained against naive hybrid attacks perform better than the asa-hybrid attacks in the training phase, the important aspect is their performance in the testing phase under various adversarial conditions, which will be discussed in the next section.

When comparing the training curves across different methods and environments, the protagonist's performance is observed to generally improve over time, albeit at different rates depending on the method and the complexity of the task. In some environments, the training curves exhibit significant variance, particularly in FingerSpin and PendulumSwingup. A key factor influencing robustness in these tasks is the nature of their control requirements. CheetahRun's dynamic stability and high-dimensional action space, as noted in previous works [25,101], afford the agent substantial redundancy. This allows it to absorb perturbations by leveraging alternative motor strategies, thereby minimizing performance variance. On the other hand, the success of PendulumSwingup and FingerSpin is contingent upon precise, time-sensitive control, a characteristic that explains their heightened sensitivity and greater performance variability when subjected to adversarial attacks.

In almost all settings, the protagonist trained against an adversary (through action-space perturbations, observation-space perturbations, or hybrid attacks) performs worse than the non-robust baseline during training. This outcome is expected, as adversarial training introduces additional challenges for the protagonist, making it more difficult to achieve high returns. However, robust agents are expected to perform comparably to the non-robust agent in the absence of perturbations and to outperform it under adversarial conditions, which will be evaluated during the evaluation phase.

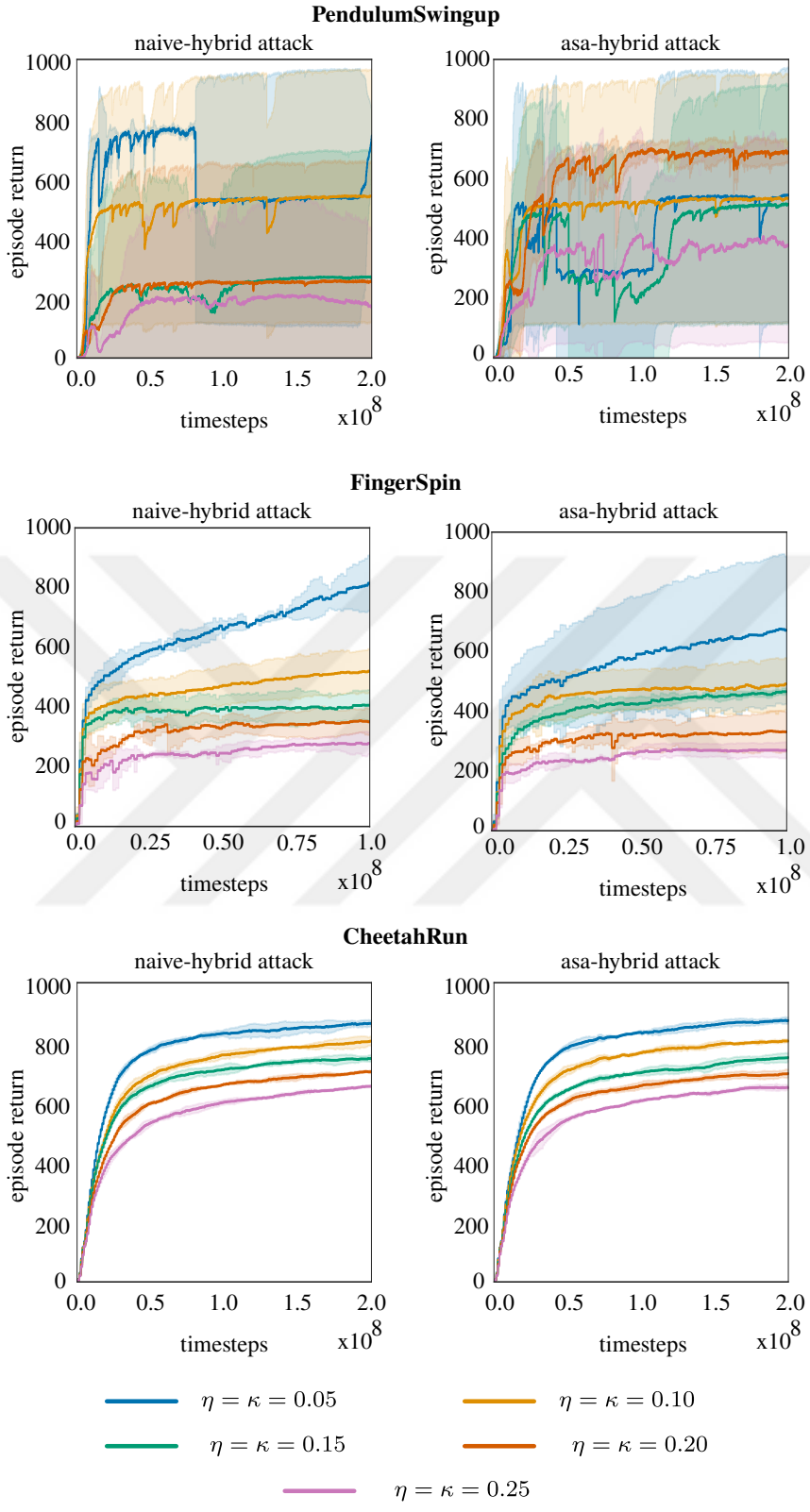

Figure 5.5: Learning curves of the agent under two conditions: naive-hybrid, and asa-hybrid attacks. Results are shown for varying attack budgets (η and κ) as mean ± standard deviation over 5 random seeds [34].

5.1.5 Evaluations

This section presents a comprehensive analysis of diverse adversarial attacks to evaluate their efficacy in fostering robust policy learning in RL. The investigation is guided by the following research questions:

- Question 1: To what extent does robustness against adversarial perturbations in the action space confer robustness against perturbations in the observation space, and vice versa?
- Question 2: Which adversarial attack modality is most effective at inducing (near) worst-case performance in an RL agent?
- Question 3: Can training that incorporates a mixture of adversarial attacks produce a protagonist agent with simultaneous robustness to action-space, observation-space, and combined perturbation attacks?
- Question 4: How well do agents trained with adversarial attacks generalize to environments with varying dynamics and parameters?

The evaluation phase assesses the robustness of the trained protagonist agents against various adversarial attack scenarios. The agents are tested in environments with different types and levels of perturbations, including action-space attacks, observation-space attacks, and mixed attacks. The performance of each agent is measured based on the average return over multiple evaluation episodes, with results presented in the following subsections.

5.1.5.1 Generalization across perturbation domains

To evaluate the generalization and performance changes of the protagonist agents trained under specific adversarial conditions, they are tested against a range of perturbation levels in both action and observation spaces. Figure 5.6 presents heatmaps illustrating the test performance of the agents across these different perturbation domains. The heatmaps display the average return of the protagonist agent under various attack

conditions. The horizontal axis represents the level of perturbation (η or κ), while the vertical axis indicates the strength of the adversarial attack applied during testing.

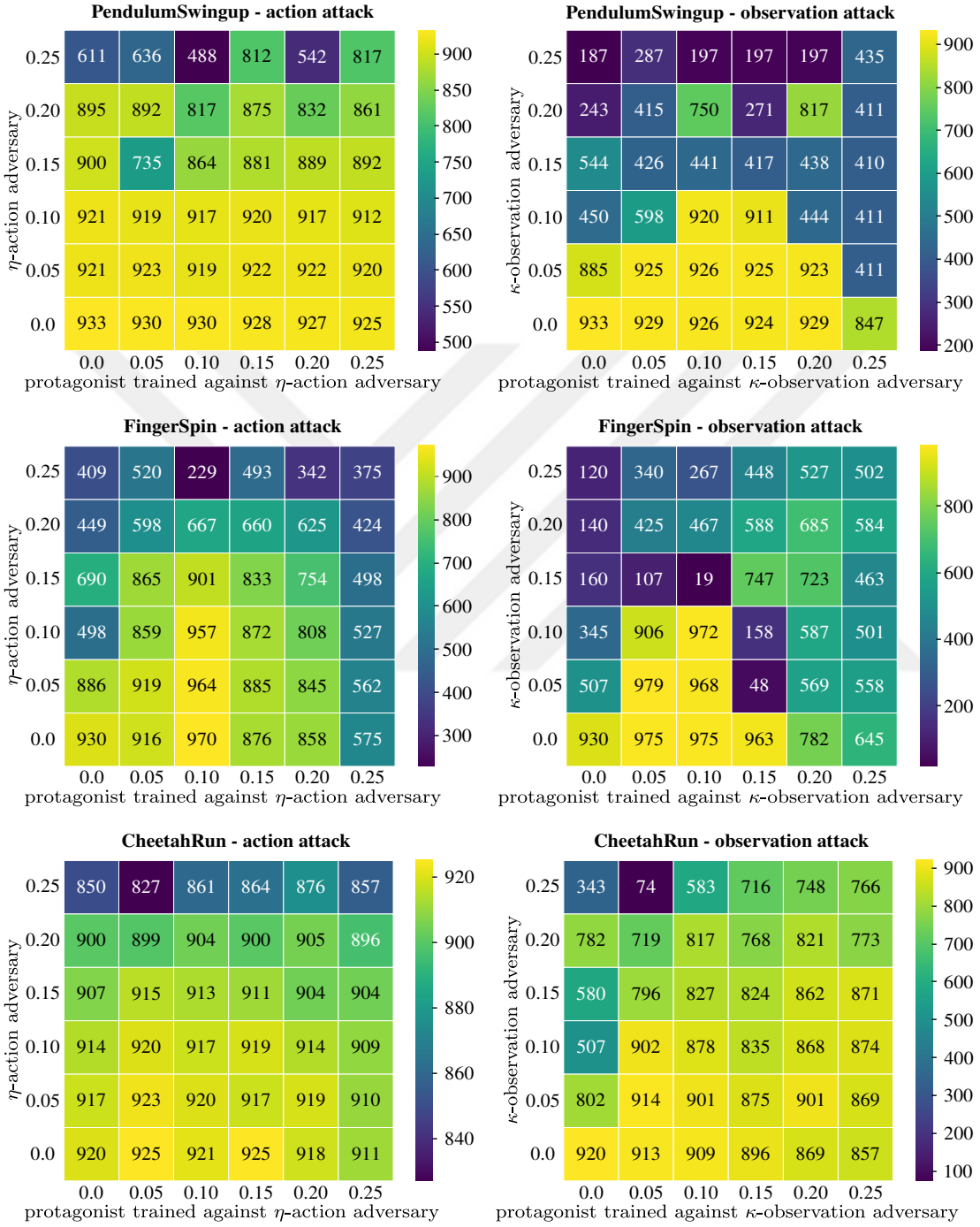

Figure 5.6: Test performance of trained agents under adversarial attacks of the same type with varying perturbation levels, reported as mean ± standard deviation over 5 random seeds [34].

The existing body of research in ARL underscores the critical role of perturbation magnitude in the development of robust agents. Excessive perturbations, while potentially beneficial in highly challenging environments, may destabilize the training process and impede convergence. Conversely, perturbations of insufficient magnitude may fail to produce meaningful gains in robustness. The analysis presented in Figure 5.6 substantiates these findings, revealing that the average performance of the protagonist agent declines when confronted with adversaries exhibiting greater strength than those encountered during training, although the observed degradation does not follow a strictly monotonic trend. Furthermore, the results demonstrate that tasks such as FingerSpin experience pronounced performance deterioration under strong perturbations. In contrast, agents trained with milder perturbations exhibit greater stability, maintaining performance levels that more closely approximate those achieved in unperturbed scenarios.

In order to address the first research question, the generalization capabilities of agents trained under specific adversarial conditions are evaluated. Specifically, the performance of agents trained against action-space attacks is assessed when subjected to observation-space attacks, and vice versa. Figure 5.7 presents heatmaps that illustrate the performance of the protagonist agent under these cross-attack conditions. The values in the heatmaps are expressed as a percentage relative to the performance of the agent trained to be robust against the corresponding attack. Since there is no one-to-one correspondence between the perturbation levels in action and observation spaces, the evaluations are conducted across all combinations of η and κ values.

The heatmaps presented in Figure 5.7 demonstrate that agents trained for robustness against a specific category of adversarial perturbation—either in the action space or the observation space—exhibit minimal generalization to the alternate type, with only a few exceptions. In particular, agents trained to withstand action-space attacks experience a pronounced decline in performance when evaluated under observation-space attacks, and vice versa. These results suggest that robustness is predominantly confined to the perturbation domain encountered during training, with limited cross-domain transferability. This finding underscores the importance of developing robust policies that can withstand both forms of adversarial interference, especially in real-world

settings where uncertainties and perturbations may concurrently affect both action and observation spaces.

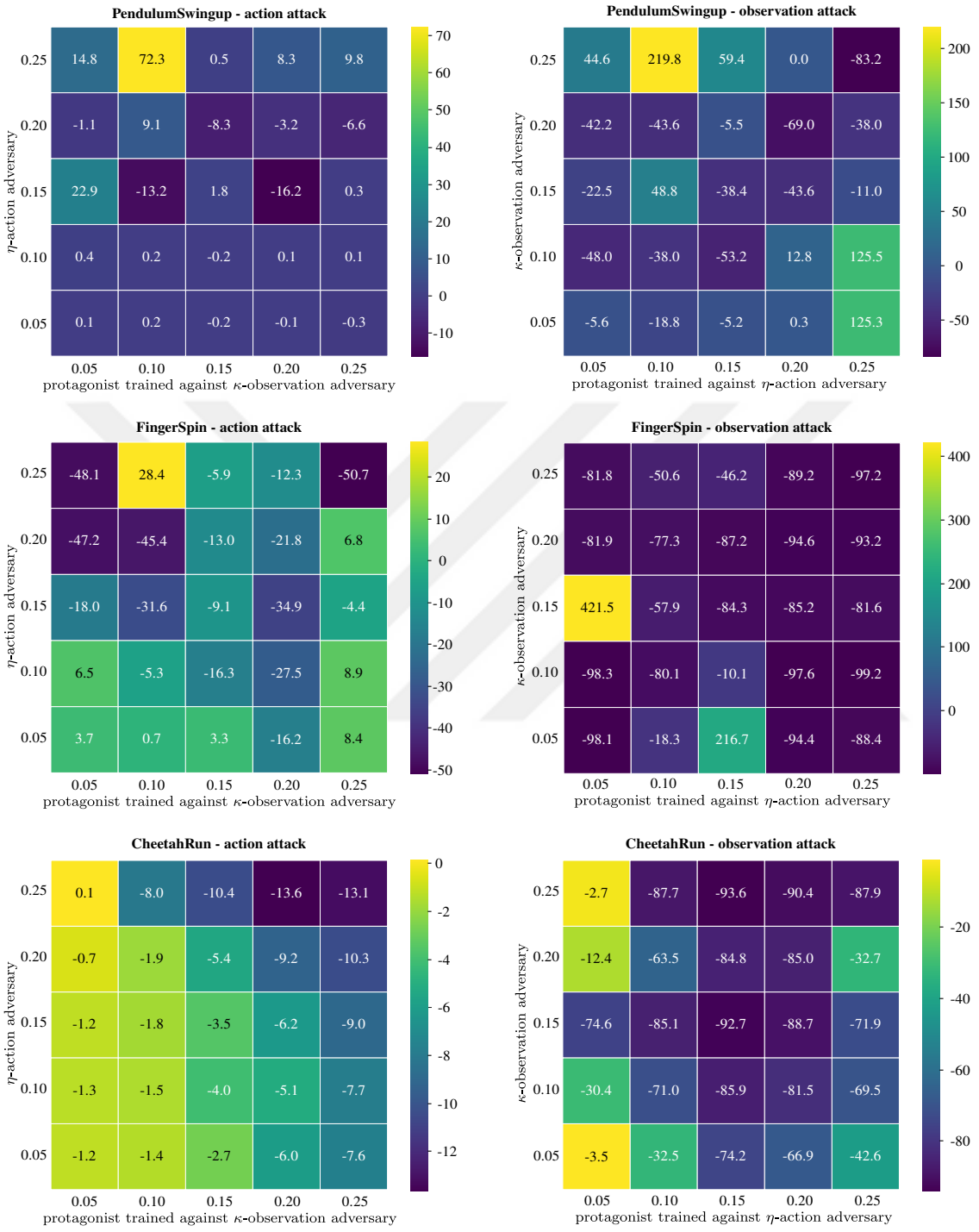

Figure 5.7: Change in performance of trained agents under different attack types, shown as a percentage relative to the performance of the agent trained to be robust to each corresponding attack, reported as mean ± standard deviation over 5 random seeds [34].

5.1.5.2 Effectiveness of adversarial attacks

To address the second research question, the effectiveness of different types of adversarial attacks is evaluated. This involves comparing the performance of the non-robust protagonist agent when subjected to various attack strategies, including all attack types. Figure 5.8 illustrates the test performance of the non-robust protagonist agent under different attack conditions and budgets. It should be noted that the hybrid attacking adversaries use more budget than the others since they perturb both the action and observation spaces.

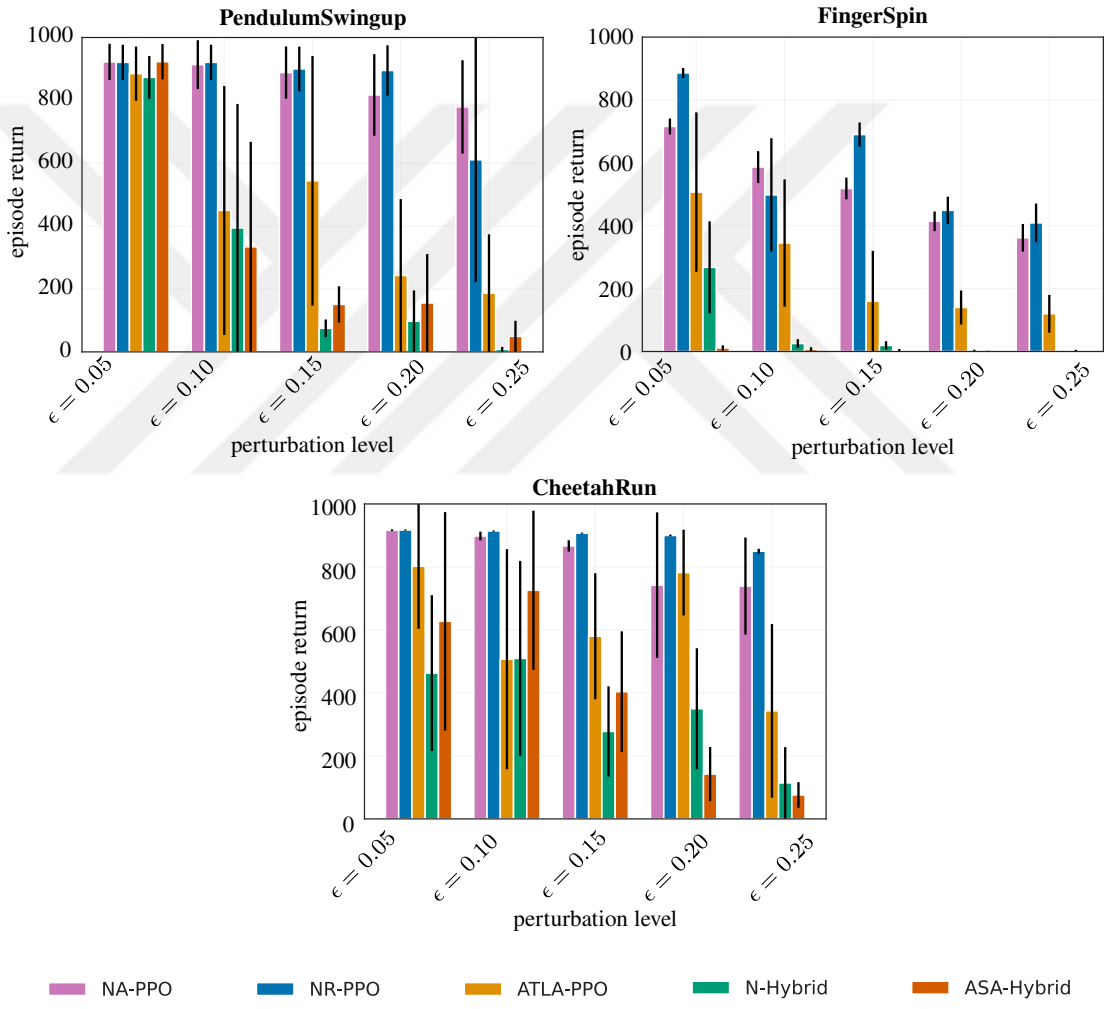

Figure 5.8: Vulnerability evaluation of non-robust protagonist policies under various adversarial attack types. Results are reported as mean ± standard deviation over 20 independent evaluation episodes (lower values indicate stronger attacks) [34].

The results indicate that the hybrid attacking adversaries are the most effective at degrading the performance of the non-robust protagonist agent, particularly under large perturbation budgets. This finding suggests that the combined effect of perturbations in both domains creates a more challenging adversarial environment, highlighting the need for robust policies that can withstand such comprehensive attacks.

Without loss of generality, specific perturbation levels were selected for each environment to facilitate a legitimate testing of the robustness performance among the baseline methods and their hybrid counterparts. The perturbation magnitudes for the experiments are determined as below:

- PendulumSwingup and FingerSpin: $\eta = 0.05$ and $\kappa = 0.05$
- CheetahRun: $\eta = 0.10$ and $\kappa = 0.10$

In Table 5.2, the performance of all protagonist agents trained with different methods is compared in a vanilla environment without any adversarial perturbations. This evaluation serves to assess the baseline performance of each method in the absence of attacks. The results show that all robust training methods achieve performance levels comparable to the non-robust baseline, indicating that the incorporation of adversarial training does not significantly compromise the agent's ability to perform well in standard conditions. Despite being trained under substantially stronger perturbations than the other benchmarks, both N-HYBRID-PPO and ASA-PPO maintain performance on par with the non-robust agent, demonstrating their effectiveness in learning robust policies without sacrificing baseline performance.

Table 5.2: Comparison of test performance between benchmark algorithms and hybrid attackers in the vanilla environment without adversarial perturbations. Results are reported as mean ± standard deviation over 20 independent evaluation episodes (higher values indicate better performance) [34].

Environment	PendulumSwingup	FingerSpin	CheetahRun
Non-Robust	932.5 ± 41.6	930.9 ± 8.9	919.2 ± 2.6
NA-PPO	759.6 ± 380.2	873.3 ± 13.3	826.7 ± 3.2
NR-PPO	930.2 ± 43.0	916.6 ± 32.1	921.1 ± 2.1
ATLA-PPO	929.1 ± 49.4	975.1 ± 10.9	909.1 ± 3.6
N-HYBRID-PPO	926.1 ± 51.0	961.0 ± 20.5	905.8 ± 5.0
ASA-PPO	931.8 ± 42.0	885.5 ± 15.9	907.6 ± 5.8

5.1.5.3 Unified robustness via hybrid attacks

To address the third research question, the robustness of the protagonist agents trained with hybrid adversarial attacks is evaluated. The performance of these agents is compared against the benchmark algorithms under hybrid attack scenarios. Tables 5.3 and 5.4 present the test performance of the agents under naive hybrid attacks and asa-hybrid attacks, respectively.

The results demonstrate that both N-HYBRID-PPO and ASA-PPO significantly outperform the benchmark algorithms under hybrid attack scenarios. Notably, N-HYBRID-PPO achieves the highest performance against naive hybrid attacks, whereas ASA-PPO excels against asa-hybrid attacks. These findings suggest that training with hybrid adversarial attacks effectively equips the protagonist agent with the capability to withstand a broader range of perturbations, thereby enhancing its overall robustness.

Table 5.3: Comparison of test performance between benchmark algorithms and the naive hybrid attacker. Results are reported as mean ± standard deviation over 20 independent evaluation episodes (higher values indicate better performance) [34].

Environment	PendulumSwingup	FingerSpin	CheetahRun
NA-PPO	733.9 ± 367.0	248.8 ± 14.6	731.4 ± 27.4
NR-PPO	196.8 ± 393.6	31.0 ± 46.4	592.9 ± 294.0
ATLA-PPO	913.2 ± 65.7	485.5 ± 26.2	786.5 ± 75.4
N-HYBRID-PPO	**920.4 ± 59.3**	**877.4 ± 14.7**	**906.1 ± 1.9**

Table 5.4: Comparison of test performance between benchmark algorithms and the asa-hybrid attacker. Results are reported as mean ± standard deviation over 20 independent evaluation episodes (higher values indicate better performance) [34].

Environment	PendulumSwingup	FingerSpin	CheetahRun
NA-PPO	193.0 ± 386.0	14.5 ± 20.6	518.0 ± 136.6
NR-PPO	792.6 ± 269.5	16.8 ± 15.9	880.1 ± 30.8
ATLA-PPO	440.6 ± 397.0	12.0 ± 11.5	720.2 ± 118.9
ASA-PPO	**922.4 ± 61.4**	**797.6 ± 32.0**	**915.2 ± 2.8**

It is worth noting that the naive attacker allocates more of its budget to perturbing the observation space, with less consideration given to the action space. This imbalance can be observed in the results presented in Table 5.3, where NR-PPO exhibits lower robustness compared to ATLA-PPO. In contrast, the ASA attacker balances its attack budget across both spaces, under which NR-PPO and ATLA-PPO perform similarly.

In order to better illustrate the advantage of the suggested ASA-PPO algorithm over the N-HYBRID-PPO, a detailed comparison is provided in Table 5.5. This table presents the test performance of both methods when the protagonist is tested against single-type attackers. The results indicate that ASA-PPO consistently outperforms N-HYBRID-PPO across all environments and attack types. This better performance can be credited to the ASA attacker's strategic allocation of its perturbation budget, which allows the protagonist to effectively challenge the adversary in both action and observation spaces.

These findings emphasize the significance of maintaining a balanced mixture of adversarial perturbations to obtain a more realistic and comprehensive assessment of robustness. Moreover, despite being trained under hybrid and more challenging adversarial conditions, the proposed ASA-PPO method sustains strong performance across natural (unperturbed), action-only, and observation-only attack settings.

Table 5.5: Comparison of test performance between protagonists trained with hybrid attackers and those trained against action-only or observation-only attackers. Results are reported as mean ± standard deviation over 20 independent evaluation episodes (higher values indicate better performance) [34].

Environment	PendulumSwingup		FingerSpin		CheetahRun	
Method - Attack	action	observation	action	observation	action	observation
N-HYBRID-PPO	921.3 ± 57.4	679.0 ± 401.2	816.7 ± 9.2	806.0 ± 23.1	854.5 ± 7.7	832.7 ± 10.0
ASA-PPO	$\mathbf{921.5 \pm 57.9}$	$\mathbf{932.2 \pm 43.0}$	$\mathbf{875.4 \pm 26.3}$	$\mathbf{833.1 \pm 24.1}$	$\mathbf{889.9 \pm 6.9}$	$\mathbf{868.1 \pm 6.3}$

5.1.5.4 Generalization to novel environment dynamics and parameters

To address the fourth research question, the robustness of the trained protagonist agents is evaluated in environments with varying dynamics and parameters. This assessment aims to determine the extent to which the agents can generalize their learned policies to new and unseen conditions. Figure 5.9 depicts the test results of the agents across all environments with modified parameters.

In this evaluation, the mass is altered for PendulumSwingup and CheetahRun, while the damping coefficient is modified for FingerSpin. Assuming the default scale for the parameters is 1.0, the parameters are varied within the range of 0.25 to 2.0 in increments of 0.25. This range allows for a comprehensive assessment of the adaptability of the agent to changes in the environment's physical properties.

At default parameters (scale = 1.0), all methods achieve comparable performance. However, as the environment parameters deviate from their default values, the performance of all methods tends to decline. The hybrid methods, developed to mitigate both action and observation perturbations, exhibit moderate resilience under mild deviations but show more pronounced performance degradation at extreme parameter values compared to baselines such as NR-PPO and ATLA-PPO.

Notably, the adversarial attack budgets employed during training were randomly predetermined to enhance agent robustness. The hybrid agents, having been trained against a stronger adversary with larger budgets for combined action and observation perturbations, prioritize defense against targeted adversarial threats, which may limit their effectiveness in addressing parametric variations. While ASA-HYBRID and N-HYBRID-PPO demonstrate strong performance against adversarial attacks, their sensitivity to shifts in environment parameters underscores the orthogonality of these robustness dimensions.

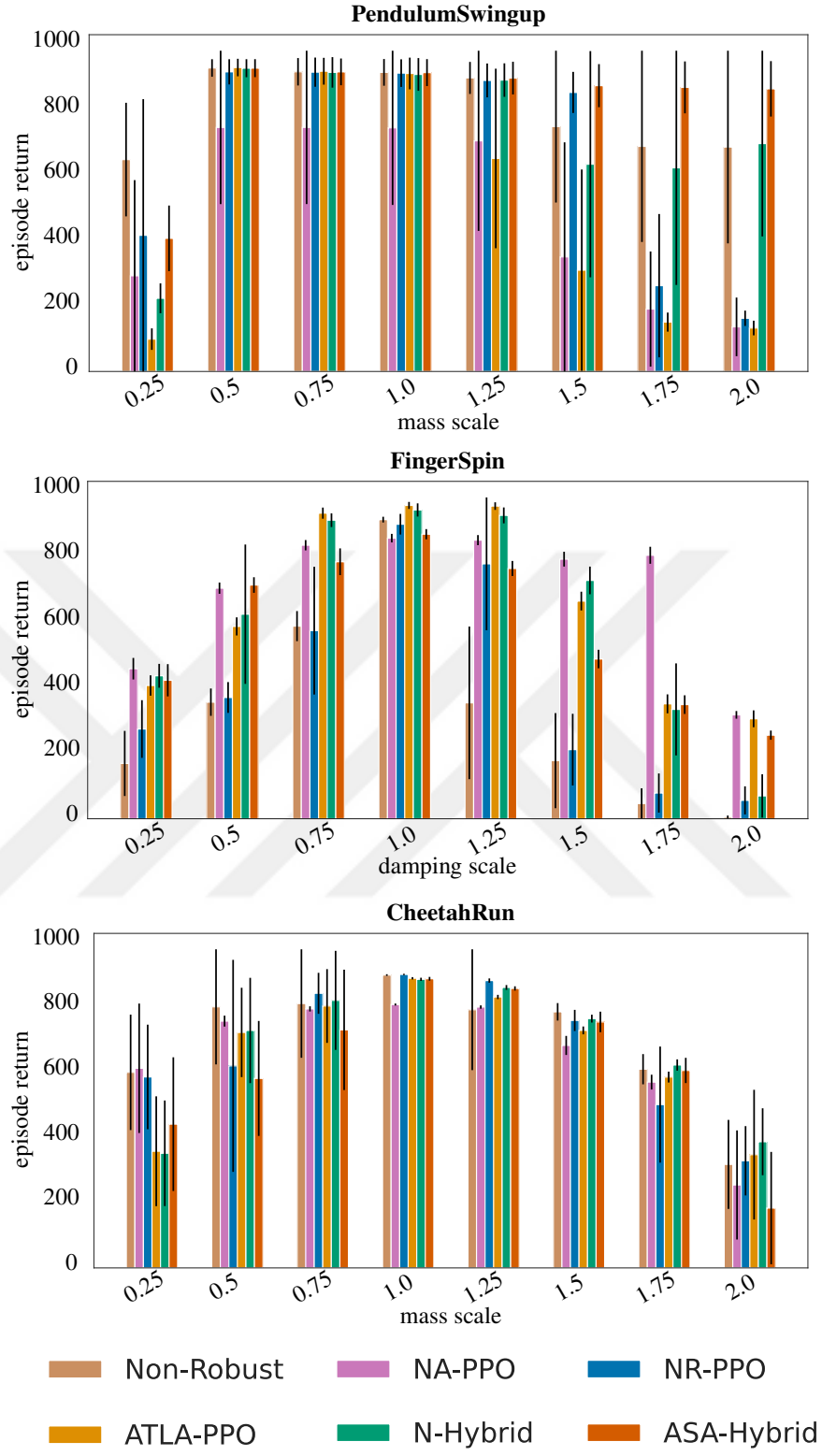

Figure 5.9: Comparison of test performance between benchmark algorithms and hybrid attackers across environments with varying parameters. Results are reported as mean ± standard deviation over 20 independent evaluation episodes (higher values indicate better performance) [34].

5.2 Discussions

The results presented in the simulations section offer meaningful perspectives on the effectiveness of adversarial training methods in enhancing the robustness of reinforcement learning agents. The findings highlight several key aspects of robustness, generalization, and the interplay between different types of adversarial attacks.

The simulations and evaluations lead to several key observations regarding the research questions:

- No transferability of robustness: The limited generalization observed between action-space and observation-space robustness underscores the need for targeted training approaches. Agents trained to withstand perturbations in one domain do not necessarily exhibit resilience in the other, indicating that robustness is highly domain-specific. This finding suggests that future research should explore methods that can simultaneously address both types of perturbations to develop more versatile agents.

- Hybrid attacks are most effective: The effectiveness of hybrid adversarial attacks in degrading the performance of non-robust agents highlights the importance of considering multiple perturbation domains during training. The superior performance of agents trained with hybrid attacks suggests that exposure to a diverse range of adversarial conditions could contribute to the creation of more robust policies.

- Robustness-baseline performance trade-off: The ability of robust training methods to maintain performance comparable to non-robust agents in unperturbed environments is crucial. It indicates that adversarial training does not necessarily compromise an agent's effectiveness under standard conditions, which is essential for practical applications. This balance between robustness and baseline performance is a key consideration for deploying RL agents in real-world scenarios. In our findings, although trained under stronger perturbations, both N-HYBRID-PPO and ASA-PPO performed slightly worse than the non-robust agent in some environments and better

in others, demonstrating their ability to maintain competitive performance in natural settings.

- Unified robustness via hybrid training: The success of hybrid-trained agents in maintaining strong performance across various attack scenarios demonstrates the potential of adversarial training in fostering unified robustness. The proposed ASA-HYBRID method, in particular, shows promise in achieving resilience against both action and observation perturbations, making it a valuable approach for real-world applications where agents may encounter a variety of adversarial conditions.

- Sensitivity to environment parameters: The observed sensitivity of hybrid-trained agents to changes in environment parameters indicates that robustness to adversarial attacks does not necessarily translate to robustness against parametric variations. This finding suggests that future research should investigate methods to integrate adversarial training with techniques that enhance adaptability to environmental changes, such as domain randomization or meta-learning approaches.

6. CONCLUSIONS AND RECOMMENDATIONS

This final chapter concludes the thesis with a summary of the main findings and outlines potential future research directions.

6.1 Conclusions

This thesis makes notable contributions to the field of RL, particularly in the areas of adversarial training and robustness. The proposed Action and State Adversarial Markov Decision Process (ASA-MDP) effectively integrates both action and state perturbations, addressing a critical gap in existing methodologies. Through mathematical formulations and analyses, the ASA-MDP framework has been shown to provide a comprehensive approach for analyzing the robustness of the agents in the presence of hybrid attacks.

Empirical evaluations of the robustified agent in the existence of single-type adversaries showed that robustness against one type of adversary does not guarantee robustness against another type. This finding underscores the importance of considering mixed/hybrid perturbations in the design of robust RL agents. The naive combination of action and state adversaries was found to be ineffective, leading to suboptimal performance. The problem with this naive approach was identified as the imbalance between the two types of perturbations.

To address this, Adversarial Synthesis and Adaptation Framework was proposed, which dynamically balances the influence of action and state adversaries during training thanks to a unique network structure of the adversary. The structure allows for a more nuanced interaction between the two perturbation types, leading to improved performance and robustness, highlighting its potential as a viable solution for training RL agents in adversarial environments. The development of the ASA-PPO algorithm further enhances the robustness of agents by incorporating adversarial training techniques that account for both action and state perturbations.

The effects of the proposed methods and the existing benchmark methods were evaluated in various simulation environments, in Mujoco Playground simulations. In the preliminary comparisons on the environments with no perturbations, the performance of all methods was comparable, indicating that the proposed approaches do not introduce any significant overhead in benign settings. Also, the adversaries were compared against a non-robust agent, and the results showed that the adversaries were effective in degrading the performance of the non-robust agent, validating their efficacy. Specifically, in almost all cases the hybrid attacks were found to be more effective than single-type attacks, emphasizing the need for robust training methods that can handle mixed perturbations.

In the additional evaluations, these methods were tested on their ability to improve the robustness of RL agents against different types of adversarial attacks. The ASA-PPO method, although trained under more severe hybrid adversarial conditions, demonstrated robust performance across non-adversarial environments, as well as under action-only and observation-only attack scenarios. This indicates that the ASA-PPO algorithm successfully balances the influence of both action and state perturbations, leading to a more resilient agent. The results suggest that the ASA-PPO method is a promising approach for training robust RL agents capable of withstanding a variety of adversarial attacks.

6.2 Recommendations

The outcomes of this thesis open various avenues for future work in the field of robust RL. The following are some potential directions for further investigation:

- Unified hybrid perturbations: Future research could focus on developing a more unified approach to hybrid perturbations, where the interaction between action and state perturbations is more deeply integrated into the learning process. This could involve exploring new architectures or training paradigms that inherently account for the complexities of mixed perturbations. In this thesis, multiplicative perturbations are applied to states, and additive perturbations are applied to actions. Although this separation yields stable and interpretable adversarial settings, the alignment of perturbation metrics between states and actions would facilitate more direct

comparisons and fairer benchmarking. The establishment of such a unified metric is considered a promising direction for future research.

- Robustness against parametric/dynamic variations: The current work primarily focuses on robustness against adversarial attacks. Future studies could explore the robustness of the trained agents to changes in environmental parameters and dynamics, which is critical in real-world scenarios where situations can change unpredictably. In this thesis, the generalization capabilities of the proposed ASA-PPO method were evaluated in environments with varying parameters. However, the robustness of the ASA-PPO method against changes in environment dynamics was not satisfactory. The observed sensitivity to parameter variations likely stems from the fact that adversarial training against action and observation perturbations does not inherently provide robustness to alterations in the underlying system dynamics. This observation underscores the necessity of complementary approaches that explicitly incorporate parametric uncertainty during training.

- Robustness in real-world scenarios: Although the approach demonstrates effectiveness, it depends on access to a simulator and presently lacks formal theoretical guarantees. The framework further presumes that the adversary has access to the actual state of the environment, an assumption that may not hold in partially observable environments. Future research could extend this framework to Partially Observable Markov Decision Processes (POMDPs) [102] by integrating history-dependent policy representations.

- Adaptive balancing between perturbation types: The current approach uses a fixed structure to balance the influence of action and state perturbations. Future research should investigate adaptive systems which modify the ratio of perturbation levels automatically according to the learning process or the specific characteristics of the environment, as suggested in [33].

REFERENCES

[1] **Arulkumaran, K.**, **Deisenroth, M.P.**, **Brundage, M. and Bharath, A.A.** (2017). Deep reinforcement learning: A brief survey, *IEEE Signal Processing Magazine*, *34*(6), 26–38.

[2] **Andrychowicz, O.M.**, **Baker, B.**, **Chociej, M.**, **Jozefowicz, R.**, **McGrew, B.**, **Pachocki, J.**, **Petron, A.**, **Plappert, M.**, **Powell, G.**, **Ray, A.** *et al.* (2020). Learning dexterous in-hand manipulation, *The International Journal of Robotics Research*, *39*(1), 3–20.

[3] **Sutton, R.S.**, **Barto, A.G.** *et al.* (1998). *Reinforcement Learning: An Introduction*, volume 1, MIT Press Cambridge.

[4] **Berner, C.**, **Brockman, G.**, **Chan, B.**, **Cheung, V.**, **Dębiak, P.**, **Dennison, C.**, **Farhi, D.**, **Fischer, Q.**, **Hashme, S.**, **Hesse, C.** *et al.* (2019). Dota 2 with large scale deep reinforcement learning, *arXiv preprint arXiv:1912.06680*.

[5] **Sarkhi, S.M.K. and Koyuncu, H.** (2024). Optimization Strategies for Atari Game Environments: Integrating Snake Optimization Algorithm and Energy Valley Optimization in Reinforcement Learning Models, *AI*, *5*(3), 1172–1191.

[6] **Silver, D.**, **Huang, A.**, **Maddison, C.J.**, **Guez, A.**, **Sifre, L.**, **Van Den Driessche, G.**, **Schrittwieser, J.**, **Antonoglou, I.**, **Panneershelvam, V.**, **Lanctot, M.** *et al.* (2016). Mastering the game of Go with deep neural networks and tree search, *Nature*, *529*(7587), 484–489.

[7] **Mnih, V.**, **Kavukcuoglu, K.**, **Silver, D.**, **Rusu, A.A.**, **Veness, J.**, **Bellemare, M.G.**, **Graves, A.**, **Riedmiller, M.**, **Fidjeland, A.K.**, **Ostrovski, G.** *et al.* (2015). Human-level control through deep reinforcement learning, *Nature*, *518*(7540), 529–533.

[8] **Kaufmann, E.**, **Gehrig, M.**, **Foehn, P.**, **Ranftl, R.**, **Dosovitskiy, A.**, **Koltun, V. and Scaramuzza, D.** (2019). Beauty and the beast: Optimal methods meet learning for drone racing, *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, pp.690–696.

[9] **Guo, Y.**, **Huang, C. and Liu, R.** (2024). Development of an Attention Mechanism for Task-Adaptive Heterogeneous Robot Teaming, *AI*, *5*(2), 555–575.

[10] **Hwangbo, J.**, **Sa, I.**, **Siegwart, R. and Hutter, M.** (2017). Control of a quadrotor with reinforcement learning, *IEEE Robotics and Automation Letters*, *2*(4), 2096–2103.

[11] **Levine, S.**, **Finn, C.**, **Darrell, T. and Abbeel, P.** (2016). End-to-end training of deep visuomotor policies, *Journal of Machine Learning Research*, *17*(39), 1–40.

[12] **Korkmaz, E.** (2021). Investigating vulnerabilities of deep neural policies, *Uncertainty in Artificial Intelligence*, PMLR, pp.1661–1670.

[13] **Szegedy, C.**, **Zaremba, W.**, **Sutskever, I.**, **Bruna, J.**, **Erhan, D.**, **Goodfellow, I. and Fergus, R.** (2013). Intriguing properties of neural networks, *arXiv preprint arXiv:1312.6199*.

[14] **Goodfellow, I.J.**, **Shlens, J. and Szegedy, C.** (2014). Explaining and harnessing adversarial examples, *arXiv preprint arXiv:1412.6572*.

[15] **Huang, S.**, **Papernot, N.**, **Goodfellow, I.**, **Duan, Y. and Abbeel, P.** (2017). Adversarial attacks on neural network policies, *arXiv preprint arXiv:1702.02284*.

[16] **Sun, J.**, **Zhang, T.**, **Xie, X.**, **Ma, L.**, **Zheng, Y.**, **Chen, K. and Liu, Y.** (2020). Stealthy and efficient adversarial attacks against deep reinforcement learning, *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp.5883–5891.

[17] **Moos, J.**, **Hansel, K.**, **Abdulsamad, H.**, **Stark, S.**, **Clever, D. and Peters, J.** (2022). Robust reinforcement learning: A review of foundations and recent advances, *Machine Learning and Knowledge Extraction*, *4*(1), 276–315.

[18] **Uther, W. and Veloso, M.** (1997). Adversarial reinforcement learning, **Technical Report**, Tech. rep., Carnegie Mellon University. Unpublished.

[19] **Kos, J. and Song, D.** (2017). Delving into adversarial attacks on deep policies, *arXiv preprint arXiv:1705.06452*.

[20] **Wang, Z.**, **Xiang, X.**, **Duan, Y. and Yang, S.** (2024). Adversarial deep reinforcement learning based robust depth tracking control for underactuated autonomous underwater vehicle, *Engineering Applications of Artificial Intelligence*, *130*, 107728.

[21] **Guo, W.**, **Liu, G.**, **Zhou, Z.**, **Wang, L. and Wang, J.** (2024). Enhancing the robustness of QMIX against state-adversarial attacks, *Neurocomputing*, *572*, 127191.

[22] **Standen, M.**, **Kim, J. and Szabo, C.** (2025). Adversarial Machine Learning Attacks and Defences in Multi-Agent Reinforcement Learning, *ACM Computing Surveys*, *57*(5), 1–35.

[23] **Wang, L.**, **Zheng, S.**, **Tai, S.**, **Liu, H. and Yue, T.** (2024). UAV air combat autonomous trajectory planning method based on robust adversarial reinforcement learning, *Aerospace Science and Technology*, *153*, 109402.

[24] **Zhang, H.**, **Chen, H.**, **Xiao, C.**, **Li, B.**, **Liu, M.**, **Boning, D. and Hsieh, C.J.** (2020). Robust deep reinforcement learning against adversarial perturbations on state observations, *Advances in Neural Information Processing Systems*, *33*, 21024–21037.

[25] **Oikarinen, T.**, **Zhang, W.**, **Megretski, A.**, **Daniel, L. and Weng, T.W.** (2021). Robust deep reinforcement learning through adversarial loss, *Advances in Neural Information Processing Systems*, *34*, 26156–26167.

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

[27] **Cai, K.**, **Zhu, X. and Hu, Z.** (2023). Reward poisoning attacks in deep reinforcement learning based on exploration strategies, *Neurocomputing*, *553*, 126578.

[28] **Rakhsha, A.**, **Radanovic, G.**, **Devidze, R.**, **Zhu, X. and Singla, A.** (2020). Policy teaching via environment poisoning: Training-time adversarial attacks against reinforcement learning, *International Conference on Machine Learning*, PMLR, pp.7974–7984.

[29] **Yu, K.**, **Fu, M.**, **Tian, X.**, **Yang, S. and Yang, Y.** (2024). Curriculum reinforcement learning-based drifting along a general path for autonomous vehicles, *Robotica*, *42*(10), 3263–3280.

[30] **Foehn, P.**, **Brescianini, D.**, **Kaufmann, E.**, **Cieslewski, T.**, **Gehrig, M.**, **Muglikar, M. and Scaramuzza, D.** (2022). Alphapilot: Autonomous drone racing, *Autonomous Robots*, *46*(1), 307–320.

[31] **Kaymaz, M.**, **Ayzit, R.**, **Akgün, O.**, **Atik, K.C.**, **Erdem, M.**, **Yalcin, B.**, **Cetin, G. and Ure, N.K.** (2024). Trading-Off Safety with Agility Using Deep Pose Error Estimation and Reinforcement Learning for Perception-Driven UAV Motion Planning, *Journal of Intelligent & Robotic Systems*, *110*(2), 55.

[32] **Liu, Y.**, **Xu, H.**, **Liu, D. and Wang, L.** (2022). A digital twin-based sim-to-real transfer for deep reinforcement learning-enabled industrial robot grasping, *Robotics and Computer-Integrated Manufacturing*, *78*, 102365.

[33] **Liu, Q.**, **Kuang, Y. and Wang, J.** (2024). Robust deep reinforcement learning with adaptive adversarial perturbations in action space, *2024 International Joint Conference on Neural Networks (IJCNN)*, IEEE, pp.1–8.

[34] **Erdem, M. and Üre, N.K.** (2025). Learning to Balance Mixed Adversarial Attacks for Robust Reinforcement Learning, *Machine Learning and Knowledge Extraction*, *7*(4), 108.

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

[36] **Kaelbling, L.P.**, **Littman, M.L. and Moore, A.W.** (1996). Reinforcement learning: A survey, *Journal of Artificial Intelligence Research*, *4*, 237–285.

[37] **Watkins, C.J. and Dayan, P.** (1992). Q-learning, *Machine Learning*, *8*(3), 279–292.

[38] **Sutton, R.S.**, **McAllester, D.**, **Singh, S. and Mansour, Y.** (1999). Policy gradient methods for reinforcement learning with function approximation, *Advances in Neural Information Processing Systems*, *12*.

[39] **Levine, S.**, **Kumar, A.**, **Tucker, G. and Fu, J.** (2020). Offline reinforcement learning: Tutorial, review, and perspectives on open problems, *arXiv preprint arXiv:2005.01643*.

[40] **Kaufmann, E.**, **Bauersfeld, L.**, **Loquercio, A.**, **Müller, M.**, **Koltun, V. and Scaramuzza, D.** (2023). Champion-level drone racing using deep reinforcement learning, *Nature*, *620*(7976), 982–987.

[41] **Littman, M.L.**, (1994). Markov games as a framework for multi-agent reinforcement learning, Machine Learning Proceedings 1994, Elsevier, pp.157–163.

[42] **Albrecht, S.V.**, **Christianos, F. and Schäfer, L.** (2024). *Multi-Agent Reinforcement Learning: Foundations and Modern Approaches*, MIT Press.

[43] **Tampuu, A.**, **Matiisen, T.**, **Kodelja, D.**, **Kuzovkin, I.**, **Korjus, K.**, **Aru, J.**, **Aru, J. and Vicente, R.** (2017). Multiagent cooperation and competition with deep reinforcement learning, *PloS One*, *12*(4), e0172395.

[44] **Tan, M.** (1993). Multi-agent reinforcement learning: Independent vs. cooperative agents, *Proceedings of the Tenth International Conference on Machine Learning*, pp.330–337.

[45] **Foerster, J.**, **Farquhar, G.**, **Afouras, T.**, **Nardelli, N. and Whiteson, S.** (2018). Counterfactual multi-agent policy gradients, *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

[46] **Amato, C.** (2024). An introduction to centralized training for decentralized execution in cooperative multi-agent reinforcement learning, *arXiv preprint arXiv:2409.03052*.

[47] **Li, X.**, **Ren, J. and Li, Y.** (2024). Multi-mode filter target tracking method for mobile robot using multi-agent reinforcement learning, *Engineering Applications of Artificial Intelligence*, *127*, 107398.

[48] **Dinneweth, J.**, **Boubezoul, A.**, **Mandiau, R. and Espié, S.** (2022). Multi-agent reinforcement learning for autonomous vehicles: A survey, *Autonomous Intelligent Systems*, *2*(1), 27.

[49] **Zhao, W.**, **Queralta, J.P. and Westerlund, T.** (2020). Sim-to-real transfer in deep reinforcement learning for robotics: a survey, *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, pp.737–744.

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

[51] **Gleave, A.**, **Dennis, M.**, **Wild, C.**, **Kant, N.**, **Levine, S. and Russell, S.** (2020). Adversarial Policies: Attacking Deep Reinforcement Learning, *International Conference on Learning Representations*.

[52] **Vinitsky, E.**, **Du, Y.**, **Parvate, K.**, **Jang, K.**, **Abbeel, P. and Bayen, A.** (2020). Robust reinforcement learning using adversarial populations, *arXiv preprint arXiv:2008.01825*.

[53] **Liu, G. and Lai, L.** (2023). Efficient adversarial attacks on online multi-agent reinforcement learning, *Advances in Neural Information Processing Systems*, *36*, 24401–24433.

[54] **Lee, X.Y.**, **Ghadai, S.**, **Tan, K.L.**, **Hegde, C. and Sarkar, S.** (2020). Spatiotemporally constrained action space attacks on deep reinforcement learning agents, *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp.4577–4584.

[55] **Chen, K.**, **Guo, S.**, **Zhang, T.**, **Xie, X. and Liu, Y.** (2021). Stealing deep reinforcement learning models for fun and profit, *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, pp.307–319.

[56] **Jackson, M.T.**, **Jiang, M.**, **Parker-Holder, J.**, **Vuorio, R.**, **Lu, C.**, **Farquhar, G.**, **Whiteson, S. and Foerster, J.** (2023). Discovering general reinforcement learning algorithms with adversarial environment design, *Advances in Neural Information Processing Systems*, *36*, 79980–79998.

[57] **Tan, K.**, **Wang, J. and Kantaros, Y.** (2023). Targeted adversarial attacks against neural network trajectory predictors, *Learning for Dynamics and Control Conference*, PMLR, pp.431–444.

[58] **Madry, A.**, **Makelov, A.**, **Schmidt, L.**, **Tsipras, D. and Vladu, A.** (2018). Towards Deep Learning Models Resistant to Adversarial Attacks, *International Conference on Learning Representations*.

[59] **Croce, F. and Hein, M.** (2020). Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, *International Conference on Machine Learning*, PMLR, pp.2206–2216.

[60] **Nilim, A. and El Ghaoui, L.** (2005). Robust control of Markov decision processes with uncertain transition matrices, *Operations Research*, *53*(5), 780–798.

[61] **Suilen, M.**, **Badings, T.**, **Bovy, E.M.**, **Parker, D. and Jansen, N.**, (2024). Robust markov decision processes: A place where AI and formal methods meet, Principles of Verification: Cycling the Probabilistic Landscape: Essays Dedicated to Joost-Pieter Katoen on the Occasion of His 60th Birthday, Part III, Springer, pp.126–154.

[62] **Başar, T. and Bernhard, P.** (2008). *H-Infinity Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*, Springer Science & Business Media.

[63] **Pan, X.**, **Seita, D.**, **Gao, Y. and Canny, J.** (2019). Risk averse robust adversarial reinforcement learning, *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, pp.8522–8528.

[64] **Liang, Y.**, **Sun, Y.**, **Zheng, R. and Huang, F.** (2022). Efficient adversarial training without attacking: Worst-case-aware robust reinforcement learning, *Advances in Neural Information Processing Systems*, *35*, 22547–22561.

[65] **Zhang, H.**, **Chen, H.**, **Boning, D. and Hsieh, C.J.** (2021). Robust Reinforcement Learning on State Observations with Learned Optimal Adversary, *International Conference on Learning Representation (ICLR)*.

[66] **Pattanaik, A.**, **Tang, Z.**, **Liu, S.**, **Bommannan, G. and Chowdhary, G.** (2018). Robust Deep Reinforcement Learning with Adversarial Attacks, *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp.2040–2042.

[67] **Li, Y.**, **Pan, Q. and Cambria, E.** (2022). Deep-attack over the deep reinforcement learning, *Knowledge-Based Systems*, *250*, 108965.

[68] **Behzadan, V. and Munir, A.** (2017). Vulnerability of deep reinforcement learning to policy induction attacks, *Machine Learning and Data Mining in Pattern Recognition: 13th International Conference, MLDM 2017, New York, NY, USA, July 15-20, 2017, Proceedings 13*, Springer, pp.262–275.

[69] **Sarkar, S.**, **Babu, A.R.**, **Mousavi, S.**, **Ghorbanpour, S.**, **Gundecha, V.**, **Gutierrez, R.L.**, **Guillen, A. and Naug, A.** (2023). Reinforcement learning based black-box adversarial attack for robustness improvement, *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, IEEE, pp.1–8.

[70] **Tramer, F.**, **Carlini, N.**, **Brendel, W. and Madry, A.** (2020). On adaptive attacks to adversarial example defenses, *Advances in Neural Information Processing Systems*, *33*, 1633–1645.

[71] **Tobin, J.**, **Fong, R.**, **Ray, A.**, **Schneider, J.**, **Zaremba, W. and Abbeel, P.** (2017). Domain randomization for transferring deep neural networks from simulation to the real world, *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, pp.23–30.

[72] **Tobin, J.**, **Biewald, L.**, **Duan, R.**, **Andrychowicz, M.**, **Handa, A.**, **Kumar, V.**, **McGrew, B.**, **Ray, A.**, **Schneider, J.**, **Welinder, P.** *et al.* (2018). Domain randomization and generative models for robotic grasping, *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, pp.3482–3489.

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

[74] **Muratore, F.**, **Eilers, C.**, **Gienger, M. and Peters, J.** (2021). Data-efficient domain randomization with bayesian optimization, *IEEE Robotics and Automation Letters*, *6*(2), 911–918.

[75] **Garcıa, J. and Fernández, F.** (2015). A comprehensive survey on safe reinforcement learning, *Journal of Machine Learning Research*, *16*(1), 1437–1480.

[76] **Chow, Y.**, **Nachum, O.**, **Duenez-Guzman, E. and Ghavamzadeh, M.** (2018). A lyapunov-based approach to safe reinforcement learning, *Advances in Neural Information Processing Systems*, *31*.

[77] **Altman, E.** (2021). *Constrained Markov Decision Processes*, Routledge.

[78] **Brunke, L.**, **Greeff, M.**, **Hall, A.W.**, **Yuan, Z.**, **Zhou, S.**, **Panerati, J. and Schoellig, A.P.** (2022). Safe learning in robotics: From learning-based control to safe reinforcement learning, *Annual Review of Control, Robotics, and Autonomous Systems*, *5*(1), 411–444.

[79] **Thananjeyan, B.**, **Balakrishna, A.**, **Nair, S.**, **Luo, M.**, **Srinivasan, K.**, **Hwang, M.**, **Gonzalez, J.E.**, **Ibarz, J.**, **Finn, C. and Goldberg, K.** (2021). Recovery rl: Safe reinforcement learning with learned recovery zones, *IEEE Robotics and Automation Letters*, *6*(3), 4915–4922.

[80] **Subramanian, A.**, **Chitlangia, S. and Baths, V.** (2022). Reinforcement learning and its connections with neuroscience and psychology, *Neural Networks*, *145*, 271–287.

[81] **Berger-Tal, O.**, **Nathan, J.**, **Meron, E. and Saltz, D.** (2014). The exploration-exploitation dilemma: a multidisciplinary framework, *PloS One*, *9*(4), e95693.

[82] **Schulman, J.**, **Levine, S.**, **Abbeel, P.**, **Jordan, M. and Moritz, P.** (2015). Trust region policy optimization, *International Conference on Machine Learning*, PMLR, pp.1889–1897.

[83] **Fudenberg, D. and Tirole, J.** (1991). *Game Theory*, MIT Press.

[84] **Nash Jr, J.F.** (1950). Equilibrium points in n-person games, *Proceedings of the National Academy of Sciences*, *36*(1), 48–49.

[85] **v. Neumann, J.** (1928). Zur theorie der gesellschaftsspiele, *Mathematische Annalen*, *100*(1), 295–320.

[86] **Tramèr, F.**, **Boneh, D.**, **Kurakin, A.**, **Goodfellow, I.**, **Papernot, N. and McDaniel, P.** (2018). Ensemble adversarial training: Attacks and defenses, *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*.

[87] **Chen, P.Y.**, **Zhang, H.**, **Sharma, Y.**, **Yi, J. and Hsieh, C.J.** (2017). Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models, *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp.15–26.

[88] **Cheng, M.**, **Le, T.**, **Chen, P.Y.**, **Zhang, H.**, **Yi, J. and Hsieh, C.J.** (2019). Query-Efficient Hard-Label Black-Box Attack: An Optimization-based Approach, *International Conference on Learning Representation (ICLR)*.

[89] **Carlini, N. and Wagner, D.** (2017). Towards evaluating the robustness of neural networks, *2017 IEEE Symposium on Security and Privacy (SP)*, IEEE, pp.39–57.

[90] **Xiao, C.**, **Zhu, J.Y.**, **Li, B.**, **He, W.**, **Liu, M. and Song, D.** (2018). Spatially transformed adversarial examples, *International Conference on Learning Representations*.

[91] **Qiu, S.**, **Liu, Q.**, **Zhou, S. and Huang, W.** (2022). Adversarial attack and defense technologies in natural language processing: A survey, *Neurocomputing*, *492*, 278–307.

[92] **Wiesemann, W.**, **Kuhn, D. and Rustem, B.** (2013). Robust Markov decision processes, *Mathematics of Operations Research*, *38*(1), 153–183.

[93] **Shapley, L.S.** (1953). Stochastic games, *Proceedings of the National Academy of Sciences*, *39*(10), 1095–1100.

[94] **Perolat, J.**, **Scherrer, B.**, **Piot, B. and Pietquin, O.** (2015). Approximate dynamic programming for two-player zero-sum Markov games, *International Conference on Machine Learning*, PMLR, pp.1321–1329.

[95] **Schulman, J.**, **Moritz, P.**, **Levine, S.**, **Jordan, M. and Abbeel, P.** (2015). High-dimensional continuous control using generalized advantage estimation, *arXiv preprint arXiv:1506.02438*.

[96] **Zakka, K.**, **Tabanpour, B.**, **Liao, Q.**, **Haiderbhai, M.**, **Holt, S.**, **Luo, J.Y.**, **Allshire, A.**, **Frey, E.**, **Sreenath, K.**, **Kahrs, L.A.** *et al.* (2025). MuJoCo Playground, *arXiv preprint arXiv:2502.08844*.

[97] **Todorov, E.**, **Erez, T. and Tassa, Y.** (2012). Mujoco: A physics engine for model-based control, *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, pp.5026–5033.

[98] **Tunyasuvunakool, S.**, **Muldal, A.**, **Doron, Y.**, **Liu, S.**, **Bohez, S.**, **Merel, J.**, **Erez, T.**, **Lillicrap, T.**, **Heess, N. and Tassa, Y.** (2020). dm_control: Software and tasks for continuous control, *Software Impacts*, *6*, 100022.

[99] **Lu, C.**, **Kuba, J.**, **Letcher, A.**, **Metz, L.**, **Schroeder de Witt, C. and Foerster, J.** (2022). Discovered policy optimisation, *Advances in Neural Information Processing Systems*, *35*, 16455–16468.

[100] **Bradbury, J.**, **Frostig, R.**, **Hawkins, P.**, **Johnson, M.J.**, **Leary, C.**, **Maclaurin, D.**, **Necula, G.**, **Paszke, A.**, **VanderPlas, J.**, **Wanderman-Milne, S.** *et al.* (2018). JAX: composable transformations of Python+ NumPy programs.

[101] **Islam, R.**, **Henderson, P.**, **Gomrokchi, M. and Precup, D.** (2017). Reproducibility of benchmarked deep reinforcement learning tasks for continuous control, *arXiv preprint arXiv:1708.04133*.

[102] **Xiang, X. and Foo, S.** (2021). Recent advances in deep reinforcement learning applications for solving partially observable markov decision processes (pomdp) problems: Part 1—fundamentals and applications in games, robotics and natural language processing, *Machine Learning and Knowledge Extraction*, *3*(3), 554–581.

CURRICULUM VITAE

Name Surname: Mustafa Erdem

EDUCATION:

- **B.Sc.:** 2014, Istanbul Okan University, Department of Mechatronics Engineering
- **M.Sc.:** 2019, Istanbul Technical University, Department of Mechatronics Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2016 - 2025, Research Assistant at Istanbul Technical University, Department of Mechatronics Engineering

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Erdem, M., Üre N. K.** (2025). Learning to Balance Mixed Adversarial Attacks for Robust Reinforcement Learning. *Machine Learning and Knowledge Extraction*, 7(4), 108.

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- **Kaymaz, M., Ayzit, R., Akgün, O., Atik, K. C., Erdem, M., Yalcin, B., ... & Ure, N. K.** (2024). Trading-Off Safety with Agility Using Deep Pose Error Estimation and Reinforcement Learning for Perception-Driven UAV Motion Planning. *Journal of Intelligent & Robotic Systems*, 110(2), 55.
- **Turgut, Y., Erdem, M.** (2020). Forecasting of Retail Produce Sales Based on XGBoost Algorithm. *Global joint conference on industrial engineering and its application areas*. (pp. 27-43), Cham: Springer International Publishing.