Adaptive Policy Smoothing in Reinforcement Learning: Applications to Wavefront Sensorless Adaptive Optics and Robotics

Payam Parvizi

Thesis submitted to the University of Ottawa
in partial fulfillment of the requirements for the
Doctor of Philosophy degree in Mechanical Engineering

Department of Mechanical Engineering
Faculty of Engineering
University of Ottawa

Abstract

Optical communication between low-Earth orbit (LEO) satellites and the ground is an emerging form of free-space data transmission. It offers significantly faster data transfer and supports higher bandwidths than radio frequency communication. However, atmospheric turbulence distorts the optical beam wavefront, leading to reduced data transfer rates. Adaptive Optics (AO) can correct these distortions by using real-time control commands, informed by data from a wavefront sensor, to adjust a deformable mirror.

Traditional AO systems, however, suffer from high complexity and cost, with a significant portion of the cost attributed to wavefront sensors. Additionally, the wavefront sensors are limited in dynamic range, consume a fraction of the incident beam intensity, and introduce latency between the measurement and actuation of the deformable mirror. These factors can cause discrepancies between the measured and actual atmospheric characteristics as the satellite traverses the sky.

This thesis demonstrates that reinforcement learning (RL) can serve as a viable, low-cost, and low-latency alternative by eliminating the wavefront sensor and its associated processing electronics. This can be accomplished by developing a control policy learned through interactions with a cost-effective and ultra-fast readout of a low-dimensional photodetector array rather than relying on a wavefront phase profiling camera.

Inspired by the application of an RL-based wavefront sensorless AO system for optical LEO satellite-to-ground communication downlinks, this thesis recognizes and addresses a general limitation in standard deep RL controllers. A significant limitation of policies trained using standard deep RL algorithms is the presence of high-frequency components in the control signal, which can lead to oscillations that increase actuator amplitudes. This limitation results in decreased correction speed and the system's inability to keep pace with rapidly changing wavefronts. Existing action regularization methods mitigate these oscillations but often reduce performance, particularly in fast-evolving dynamic environments, where they re-

strict the policy from quickly adjusting to account for large state changes.

To address this challenge, a novel State-Adaptive Proportional Policy Smoothing (SAPPS) method is proposed for RL. SAPPS reduces high-frequency components in the control signal in continuous environments through policy smoothing proportionally to the magnitude of state changes, ensuring the policy remains responsive to environmental changes without compromising performance.

The proposed SAPPS method is integrated with the on-policy deep RL algorithm Proximal Policy Optimization (PPO) and compared against standard PPO and the state-of-the-art smoothing methods CAPS and LipsNet, both implemented within the PPO framework for fair comparison. To assess the generality of the proposed approach, it is evaluated across standard MuJoCo continuous-control tasks, which serve as widely used benchmarks for RL, and a real-world quadcopter hovering experiment that demonstrates its hardware applicability. In addition, the method is evaluated on a complex wavefront sensorless AO system for optical satellite communication, highlighting its effectiveness in highly dynamic environments. To address the challenges of AO systems and facilitate the evaluation of RL algorithms, an RL environment for a simulated wavefront sensorless AO system is also developed.

The results show that PPO+SAPPS performs comparably to PPO+CAPS, and both outperform standard PPO in MuJoCo tasks. Specifically, PPO+SAPPS improves performance by 11% and policy smoothness by 28%. In the physical quadcopter experiment, PPO+SAPPS outperforms PPO+CAPS and standard PPO, achieving a 17% higher average return and a 29% improvement in policy smoothness. In the wavefront sensorless AO system, PPO+SAPPS achieves performance comparable to PPO+CAPS, reaching the maximum performance attained by the Shack-Hartmann wavefront sensor in a quasi-static atmosphere. Moreover, under dynamic conditions, PPO+SAPPS surpasses standard PPO, PPO+CAPS, and PPO (LipsNet) in high-velocity conditions.

These findings highlight the contribution of the proposed SAPPS method in reducing high-frequency control fluctuations in proportion to environmental changes, while enabling more responsive performance compared to state-of-the-art smoothing methods across multiple simulated and real-world systems. In particular, it demonstrates strong potential for optical satellite communication, which could effectively serve rural and remote communities by enabling high-speed connectivity with reduced costs.

Dedication

To my brother ***Poorya Parvizi***
& my mother ***Narges Memaryan***
& my father ***Ali Parvizi***

Acknowledgments

Completing this thesis has been a challenging yet deeply rewarding journey—one that has helped me grow both personally and professionally. I am profoundly grateful to all the people who supported me along the way.

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr. Davide Spinello. I feel incredibly fortunate that he believed in me and welcomed me into his research team. His guidance extended far beyond academic supervision—he was always available, patient, and empathetic, offering unwavering support through both research challenges and personal difficulties.

I am also deeply thankful to my co-supervisors, Dr. Ross Cheriton and Dr. Colin Bellinger. Dr. Cheriton guided me with warmth and patience. His steady encouragement and belief in me boosted my confidence like only a close friend could. He was never too busy to listen or offer thoughtful feedback. Dr. Bellinger consistently followed up on my progress, helping me navigate research obstacles and sharing insightful resources that significantly deepened my understanding. He brought clarity to complex ideas and always made time to support me.

I was fortunate to collaborate with the wonderful members of our research group—Abhishek Naik, Runnan Zou, Farzan Soleymani, and Nathaniel Mailhot. Whether directly or indirectly involved in my thesis, they were not only brilliant collaborators but also a source of constant academic and emotional support.

To my lifelong friends from around the world—Arif Badem, İrem Uçkunlar Badem, Yashar Badienia, Melisa Chuong, Nazila Farhangzad, Arvin Hosseinnejad, Safoora Kamjan, Mani Kazimi, Mohammad Khavassi, Golden Nadimi, Mahmoodreza Parvizi, Ashkan Soltani, and Taher Torabi—thank you for always being there for me. Your encouragement, even from afar, meant more than you know. Through your messages, calls, and thoughtful check-ins, you reminded me that distance means little when hearts are close. I am genuinely grateful to have you in my life.

I also cherish the friendships I formed on campus and beyond. Kazem Alam-

beigi, Shahin Sharafi, Uy Vo, Yuzhen Yan, and Amit Nayak—thank you for the countless conversations during our breaks that gave me strength. Whether we were sharing ideas, frustrations, or simply enjoying a few moments of laughter between long work sessions, your presence made the everyday challenges feel lighter.

My heartfelt thanks go to my new community in Canada: Shahla Amujani, Saeed Memarian, Fabian Basaldua, Elizabeth Bustos, Nazanin Hemmati, Sasan Norouzian, Amir Rezapour, Shadieh Taherzadeh, and Arash Akbari Tabrizi. Your kindness and support made me feel at home and surrounded by love. In a new country, you gave me a sense of belonging I will never forget. Whether it was sharing meals, celebrating milestones, or simply being there to listen, each of you brought warmth, care, and joy into my life.

I also want to express my gratitude to my uncles, Javad Parvizi and Josef Parvizi. They have been a constant source of inspiration and guidance throughout my life. Their achievements, integrity, and dedication to their fields have always motivated me. I feel fortunate to have them in my life and truly appreciate their support in helping shape my dreams.

Finally, I owe everything to my beloved family. To my parents, Narges and Ali—thank you for your boundless love, constant support, and belief in me, especially in moments of self-doubt. Your sacrifices and encouragement have been the foundation of everything I have achieved. To my dear grandmother, Robab—your gentle words and love have been a quiet source of strength throughout my life. And to my brother, Poorya—you have stood by me every step of the way, always selfless and endlessly supportive. You've been not just a sibling but a true friend. Your encouragement during tough times and your belief in me have meant more than words can ever express. I could never repay your kindness.

This achievement is as much yours as it is mine—thank you from the bottom of my heart.

This thesis was funded by the National Science and Engineering Research Council (NSERC) of Canada through Discovery grant RGPIN-2022-03921, and by the National Research Council Canada (NRC) through the AI4D grant AI4D-135-2.

Contents

List of Figures

List of Tables

List of Acronyms

AO	Adaptive Optics
ANN	Artificial Neural Network
CAPS	Conditioning for Action Policy Smoothness
CNN	Convolutional Neural Network
DDPG	Deep Deterministic Policy Gradient
DRL	Deep Reinforcement Learning
GEO	Geostationary Orbit
HCIPy	High Contrast Imaging for Python
LEO	Low-Earth Orbit
LQG	Linear Quadratic Gaussian
LSTM	Long Short-Term Memory
MDP	Markov Decision Process
MLP	Multilayer Perceptron
MPC	Model Predictive Control
PID	Proportional–Integral–Derivative
PPO	Proximal Policy Optimization
RL	Reinforcement Learning
RNN	Recurrent Neural Network
SAC	Soft Actor-Critic
SAPPS	State-Adaptive Proportional Policy Smoothing
SPGD	Stochastic Parallel Gradient Descent
SSIM	Structural Similarity Index Measure
WSL-AO	Wavefront Sensorless Adaptive Optics

Chapter 1

Introduction

The growing demand for high-speed data transmission in satellite communication has increased interest in optical satellite-to-ground downlinks, particularly for low-Earth orbit (LEO) satellites. However, atmospheric turbulence distorts the optical beam's wavefront, resulting in degraded signal quality. Adaptive optics (AO) systems mitigate these effects by controlling deformable mirrors to correct wavefront distortions in real time. Traditional AO systems, however, suffer from high complexity and cost, with a significant portion attributed to wavefront sensors.

Motivated by the challenges of LEO optical satellite communication and the need for AO, this thesis aims to improve beam quality and fiber coupling efficiency under atmospheric turbulence. It focuses on designing an RL-based controller offering a lower-cost and low-latency alternative to traditional AO. Deep RL policies can exhibit high-frequency action fluctuations due to the complexity of deep neural networks and their sensitivity to perturbation. This is the case in highly dynamic environments like the AO system, where the environment evolution is faster or comparable to the controller's sampling rate.

To address this issue, this thesis proposes a novel smoothing method that regulates actions based on the magnitude of state changes, aiming to reduce excessive actuator usage and improve energy efficiency and overall system performance. The method causes the policy to remain smooth while allowing actions to quickly adjust to large state transitions in highly dynamic environments. To demonstrate the generality and effectiveness of the proposed method, it is evaluated across multiple domains: standard MuJoCo continuous-control benchmarks, a real-world quadcopter experiment, and a complex wavefront sensorless AO system under dynamic turbulence. These evaluations collectively confirm the method's ability to achieve smooth and stable control across diverse settings.

The main contributions of this thesis are as follows: (1) the introduction of a novel State-Adaptive Proportional Policy Smoothing (SAPPS) method that enhances policy smoothness and improves actuator efficiency without compromising performance, (2) the development of an RL environment for wavefront sensorless AO designed for optical satellite-to-ground communication, and (3) the first demonstration of RL-based AO control for optical downlinks using low-pixel-count detectors.

The following sections outline in more detail the motivation behind this thesis, the key challenges involved, and the contributions made in response.

1.1 Motivation and Objectives

The increasing demand for high-speed wireless communication facilities and high-capacity data transmission with low latency has led to a significant increase in bandwidth and capacity requirements where physical fiber or copper links are not practical. Although radio frequency technology is widely used for wireless communication worldwide due to its reliability, it also faces some limitations. Radio frequency systems are restricted by a bandwidth bottleneck due to the carrier wavelength. Additionally, these systems require licensing in most frequency bands and suffer from interference. To achieve higher data rates, increasing the size of the antennas and the power of the transmitters is necessary [1].

Free-space optical communication is an alternative that can provide significantly higher data transmission rates [2]. In contrast to radio frequency signals, free-space optical communication relies on optical carriers to transmit information between points through an unguided channel. Using optical carriers, optical communication systems offer several advantages over radio frequency communication, including increased bandwidth, lower power consumption, easier deployment, unlicensed spectrum allocation, reduced transmitter and receiver sizes, and immunity to interference and jamming [3].

However, optical beams suffer from wavefront errors as they propagate through significant distances (kilometers) of atmosphere, as shown in Figure 1.1. Variations in atmospheric temperature, pressure, and humidity cause turbulent motion, resulting in random fluctuations in the refractive index. This inhomogeneity distorts the beam's wavefront [4], where a wavefront is defined as a surface on which the phase of the wave remains constant [5]. Such distortions increase with the propagation distance through the atmosphere between the transmitter and receiver and

degrade the beam's coherence over the optical path, leading to a reduction in the potential bandwidth and data transfer rate of the link [6, 7].

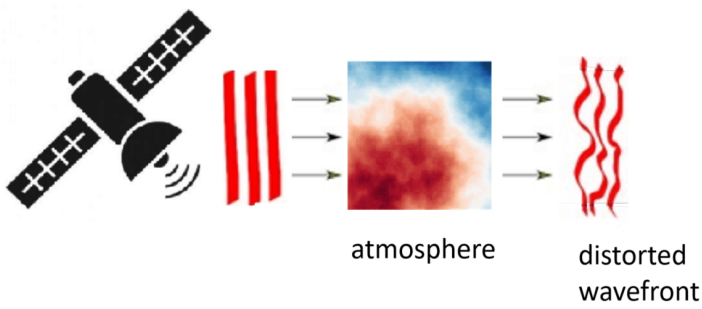

Figure 1.1: Distortion of an optical beam's wavefront caused by atmospheric turbulence.

In astronomy, adaptive optics (AO) systems have been effectively used to mitigate the issue of atmospheric turbulence [4, 8, 9, 10]. These systems correct wavefront distortions in real-time by adjusting a deformable mirror through a closed-loop feedback process based on measurements from wavefront sensors [11].

Traditional AO systems remain costly and complex, with a significant portion attributed to the wavefront sensor. Additionally, AO systems often consume a portion of the incident beam intensity to illuminate the camera pixels adequately. These systems also have a limited dynamic range and introduce latency. In fast-moving LEO satellites, this latency can lead to outdated wavefront corrections, which result in errors at spatial and temporal scales of optical coherence [12].

Recently, research has demonstrated the potential of reinforcement learning (RL) to address complex control problems in various AO applications, such as astronomy [13, 14, 15] and microscopy [16]. RL algorithms have been applied in both wavefront sensor-based AO systems [17, 18, 19, 20] and wavefront sensorless AO systems [21, 22, 23] to improve performance and reduce latency. However, existing RL implementations in AO systems are primarily designed for applications focused on optimizing image sharpness rather than improving the reliability of optical data links in optical satellite communications.

Applying RL to LEO optical communication downlinks presents challenges due to rapidly changing atmospheric turbulence and the short communication window as satellites traverse the sky within minutes. These conditions require rapid and stable correction, making standard deep RL algorithms insufficient. Such algorithms can produce high-frequency action fluctuations, which can overload actuators, slow the effective correction speed of the deformable mirror, and com-

promise link stability. To address this issue, a novel policy smoothing approach is introduced that regulates the policy's responsiveness based on the magnitude of state changes.

While increasing image sharpness can improve fiber-coupling efficiency, it typically has expensive, high-resolution, and high-latency readout circuits (e.g., infrared cameras) that gather excessive information that can compromise loop speed. In optical communication, the focus is on ensuring a stable and reliable optical link to achieve consistent and uninterrupted high-data transmission with lower latency and cost, where low-pixel detectors (e.g., photodetectors) are more effective.

Moreover, most research addresses AO challenges in applications that operate at lower speeds with long coherence times, such as astronomical observation, geostationary orbit (GEO) satellite-to-ground communication [24], microscopy, and ophthalmology [16]. In these applications, atmospheric turbulence varies slowly. In contrast, Low Earth Orbit (LEO) satellite-to-ground communication involves rapid satellite movement, causing high-altitude atmospheric variations to change much faster, which presents a significantly greater challenge.

The factors that make AO for LEO satellite-to-ground optical communication unique include:

- Rapid satellite movement causes high-altitude atmospheric variations to change quickly.
- The optical beam travels through the lower atmosphere at low elevation angles where atmospheric turbulence is strong.
- Since LEO satellites cross the sky in a matter of minutes, any delays or operational instabilities can interrupt the data stream. Therefore, AO correction must rapidly achieve a high degree of correction and maintain stability.
- There is latency in real-time AO correction as high-speed updates are required to keep pace with turbulence dynamics.

The wavefront sensorless RL approach has the potential to overcome the limitations of traditional AO, offering a more cost-effective, low-latency solution for optical LEO satellite-to-ground communication by using photodetectors in place of wavefront sensors and high-resolution cameras. In AO, where atmospheric turbulence is highly complex and difficult to predict, model-based approaches are often impractical. This makes model-free RL a more suitable choice for developing wavefront sensorless AO systems.

However, applying model-free RL to such systems presents significant challenges. Recent studies have shown that deep RL-based controllers can generate oscillatory control responses due to the high complexity of deep neural networks. This issue is particularly pronounced in applications where control responses vary substantially within acceptable output limits [25]. Moreover, neural network outputs (actions) in deep RL are highly sensitive to small variations in inputs (states/observations) [26], causing oscillatory behavior that leads to high-frequency fluctuations, excessive actuator movements, and insufficient wavefront correction. These effects increase stabilization time after each action, reducing correction speed, power efficiency, and overall control performance [27, 28].

To overcome these challenges, it is essential to develop a model-free RL controller for a wavefront sensorless AO system that operates without an explicit system model or predefined turbulence model while remaining robust under dynamically varying conditions. Addressing these limitations requires advancements in standard RL methods focused on improving stability, responsiveness, and control efficiency to ensure reliable optical communication.

The factors that make deformable mirror control in RL-based wavefront sensorless AO systems uniquely challenging are as follows:

- Low-pixel-count detectors for observation measurements reduce system latency but introduce partial observability, which provides limited information about the environment for the controller.
- The highly dynamic nature of the atmosphere demands an RL controller capable of real-time correction. Rather than relying on frame-by-frame measurements, the controller must enable predictive wavefront correction.
- Atmospheric turbulence exhibits local and fast time-scale variability, requiring the RL controller to generalize across a wide range of atmospheric conditions.
- Standard RL controllers often generate oscillatory control responses, leading to excessive actuator usage and large control actions. This results in long stabilization times and reduced correction speed. The controller must minimize high-frequency fluctuations in the control signal to maintain efficiency.

Overall, the objective of this thesis is to develop an RL–based control method that produces smooth control responses while remaining responsive in highly dynamic environments. In addition, it aims to develop an RL environment for a

simulated low-latency and low-cost wavefront sensorless AO system designed for LEO satellite-to-ground communication, enabling effective training and testing of RL algorithms under controlled simulation conditions.

1.2 Contributions

Inspired by the application of an RL-based wavefront sensorless AO system for optical LEO satellite-to-ground communication downlinks, this research recognizes and addresses a general limitation in standard deep RL controllers.

As mentioned in Section 1.1, deep RL algorithms face a significant challenge due to high-frequency components in the control signal. This leads to oscillation in the control signal, which results in excessive actuator usage and increased power consumption. To address this challenge, a novel State-Adaptive Proportional Policy Smoothing (SAPPS) method is proposed for an RL algorithm that improves policy smoothness in static and dynamic environments without compromising performance. Leveraging Lipschitz continuity [29], this method accounts for the differences between subsequent observations without directly minimizing actions. With this approach, if the change in time-contiguous observations is small, there is a penalty for significant changes in subsequent actions. In contrast, if the change in time-contiguous observations is large, there is a penalty for small subsequent action changes. Thus, this method reduces high-frequency action fluctuations through policy smoothing but does so in proportion to the magnitude of state changes, ensuring the policy remains responsive to environmental changes.

This adaptive behavior is particularly critical in satellite-to-ground optical communication. For example, once the beam is successfully coupled into the fiber, a rapidly changing atmosphere may require larger corrective actions to maintain coupling. In such cases, small action changes may be inadequate and lead to a loss in coupling efficiency. In contrast, in quasi-static or low-dynamic atmospheric conditions, large action changes may be unnecessary and even slow down the correction speed due to mirror inertia. Therefore, SAPPS modulates the policy response to the magnitude of observation changes to ensure that action responses remain proportional. This prevents overreaction in quasi-static or low-dynamic conditions and underreaction in highly dynamic environments.

Beyond AO, the proposed SAPPS method integrated with a standard deep RL algorithm is used to demonstrate smooth and stable control performance across standard MuJoCo continuous-control benchmarks and quadcopter hovering en-

vironments. This generalization highlights the method's potential for improving optical communication systems and general robotics control applications.

The underlying idea of SAPPS is general and can be integrated with common deep RL algorithms, such as DDPG [30], TD3 [31], SAC [32], and PPO [33]. In this thesis, I focus on applying SAPPS with PPO due to PPO's relative simplicity, stability, and ease of implementation, which make it well-suited for rapid experimentation and prototyping. Moreover, PPO is widely regarded as a preferred method for sim-to-real robotic control applications (e.g., [34, 35, 36, 37]), making it a strong candidate for evaluating the effectiveness of SAPPS and comparing it against existing methods across a variety of problems.

SAPPS integrated with PPO is evaluated against standard (vanilla) PPO and two state-of-the-art policy smoothness methods, CAPS and the LipsNet neural network, both implemented with PPO. The evaluations are conducted across multiple domains: (1) standard MuJoCo continuous-control tasks, which serve as standardized benchmarks for RL; (2) a real-world quadcopter hovering experiment, which demonstrates hardware applicability and shows that SAPPS remains effective not only in simulation but also in physical systems; and (3) a complex wavefront sensorless AO system for optical satellite communication under varying turbulence and drift-velocity conditions, which highlights the method's capability to operate effectively in dynamic environments. To address the challenges of AO systems outlined in Section 1.1 and to facilitate the evaluation of RL algorithms, an RL environment for a simulated wavefront sensorless AO system is developed, as detailed in Chapter 5.

To summarize, the contributions of this work are:

- Introduction of a novel State-Adaptive Proportional Policy Smoothing (SAPPS) method for RL algorithms, which improves policy smoothness in static and dynamic environments. The algorithmic framework is presented in Chapter 3.

- Development of the first open-source RL environment for a simulated wavefront sensorless AO for training and testing RL algorithms (details in Section 5.1). This is the first AO-RL environment implemented according to the OpenAI Gymnasium framework standards, simplifying the analysis of RL algorithms. The design and implementation details are provided in Chapter 5.

- The first demonstration of RL's potential for wavefront sensorless AO satellite data downlink. The corresponding experimental results are presented in

Chapter 6.

Research outputs associated with this thesis include:

- [38] Parvizi, P., Zou, R., Bellinger, C., Cheriton, R., & Spinello, D. (2023). Reinforcement learning environment for wavefront sensorless adaptive optics in single-mode fiber coupled optical satellite communications downlink. *Photonics*, 10(12), 1371. MDPI.
- [39] Zou, R., Parvizi, P., Cheriton, R., Bellinger, C., & Spinello, D. (2023). Wavefront sensorless adaptive optics for free-space satellite-to-ground communication using reinforcement learning. In *COAT 2023*.
- [40] Parvizi, P., Bellinger, C., Cheriton, R., Naik, A., & Spinello, D. (2025). Action-Regularized Reinforcement Learning for Adaptive Optics in Optical Satellite Communication. In *Optica Open*.
- Parvizi, P., Naik, A., Bellinger, C., Cheriton, R., & Spinello, D. (2025). Adaptive Policy Regularization for Smooth Control in Reinforcement Learning. Manuscript in preparation for submission to *IEEE Transactions on Automation Science and Engineering*.

1.3 Outline

The thesis is organized as follows:

- *Chapter 2*: This chapter provides a comprehensive examination of the foundational concepts and principles underlying traditional AO systems. It includes an in-depth review of the literature on AO systems with and without wavefront sensors and provides a broad overview of the current state of knowledge in the field. Additionally, it introduces the foundational principles of RL and commonly used RL algorithms, and reviews prior research focused on enhancing policy smoothness in RL-based control.
- *Chapter 3*: This chapter provides a detailed explanation of the challenges associated with standard deep RL algorithms in generating smooth actions. It then presents the necessary preliminaries, followed by a detailed discussion of the proposed State-Adaptive Proportional Policy Smoothing (SAPPS) method, which aims to optimize RL controllers for smoother policies in both static and dynamic continuous control environments.

- *Chapter 4*: This chapter evaluates the generalization capability of the proposed SAPPS method in robotic control tasks. It presents results from standardized MuJoCo continuous-control benchmarks and a real-world quadcopter hovering experiment, demonstrating that SAPPS serves as a general-purpose RL approach that remains effective in both simulated and real-world systems.

- *Chapter 5*: This chapter introduces the RL environment developed for training and evaluating RL algorithms in a wavefront sensorless AO system. It also discusses the three fundamental components of the RL formulation: the action space, the observation space, and the reward function.

- *Chapter 6*: This chapter presents a more detailed analysis of experiments and results obtained from both quasi-static and dynamic atmospheres under various atmospheric conditions in a wavefront sensorless AO system. It compares different policy smoothness methods, including the proposed SAPPS method, with a Shack-Hartmann wavefront sensor-based AO system and a flat mirror condition.

- *Chapter 7*: This chapter summarizes the overall significance of the research findings and the main outcomes of the thesis. Additionally, potential areas for future research are suggested to provide directions for further exploration.

Chapter 2

Background and Literature Review

This chapter is organized around the two overarching themes of this thesis. Sections 2.1 and 2.2 focus on adaptive optics (AO), providing essential background on AO principles, including both wavefront sensor-based and wavefront sensorless approaches (e.g., RL-based methods). Sections 2.3, 2.4 and 2.5 then shift focus to RL, outlining the fundamental concepts and commonly used RL algorithms, followed by a discussion of the limitations of standard deep RL methods, particularly their tendency to generate high-frequency control oscillations in dynamic environments. Recent advances aimed at improving policy smoothness are also reviewed. The chapter concludes by identifying current research gaps and outlining how this thesis addresses them.

2.1 Adaptive Optics

Optics science focuses on the study of light and vision [41]. Optical systems are widely used across various fields, including astronomy, biomedical imaging, and optical communication. These systems are designed to direct and manage the propagation of light to form clear images or transmit data efficiently. Achieving high-resolution imaging or reliable optical communication depends on maintaining the coherence and structure of the light wavefront during propagation. However, disturbances along the optical path (e.g., atmospheric turbulence) introduce distortions that can significantly degrade system performance.

These distortions, also known as aberrations, occur when the phase of a light wavefront deviates from its ideal shape, leading to imperfect image formation or intermittent communication. A wavefront is a surface where the phase of the light

wave remains constant [5]. Various factors, including lens manufacturing defects or misalignment within the optical system, can cause aberrations. More importantly, aberrations can occur when light passes through a medium with a constant or variable index of refraction, such as the Earth's turbulent atmosphere or biological specimens [4].

Currently, most optical satellite communication ground stations direct optical beams onto photodetectors or couple them into multi-mode fiber to improve coupling efficiency [42]. However, this approach has limitations, such as modal noise and the inability of multi-mode fiber to preserve phase coherence, which restricts the use of high order phase modulation schemes and fiber amplifiers for long-haul data transmission. Coupling into single-mode fiber enables data rates >10 Gbps per wavelength channel but requires significantly higher optical power from the satellite to compensate for reduced coupling efficiency. Larger telescope apertures provide the benefit of collecting more light; however, they also experience greater wavefront distortions due to atmospheric turbulence, reducing effective coupling efficiency and leading to diminishing returns on the signal.

Adaptive optics (AO) has emerged as an effective technique for mitigating these distortions in real time [43]. Initially developed for astronomical imaging, traditional AO systems dynamically correct such distortions by adjusting a deformable mirror based on feedback from wavefront sensors. A traditional AO system consists of three main components: the wavefront sensor, the controller, and the deformable mirror. The wavefront sensor measures wavefront distortion in real-time and provides this information to the controller. The controller then uses an internal algorithm to generate control signals that adjust the actuators of the deformable mirror.

This section begins by explaining optical aberrations and their sources, followed by an overview of how the wavefront is represented. It then introduces AO as a solution for correcting these aberrations, starting with a brief history of AO and continuing with a detailed explanation of the wavefront correction procedures and the operation of wavefront sensors.

2.1.1 Source of Aberration

One of the primary sources of aberration is ***atmospheric turbulence***. In satellite-to-ground communication, atmospheric turbulence induces the distortion of light waves as they pass through the Earth's atmosphere. Atmospheric turbulence can

also occur between the microscope aperture and a biological specimen [44], although the turbulence in this case is much lower than that of the Earth's atmosphere.

Temperature, humidity, and pressure variations in the atmosphere can cause random fluctuations in wind velocity, which is referred to as turbulent motion in the atmosphere. These fluctuations lead to changes in atmospheric density (density distribution), which alter the refractive index. This inhomogeneity in the atmospheric index profile can cause the wavefront of a beam to change as it propagates through the turbulent atmosphere. These changes result in intensity fluctuations and beam spreading [4]. In satellite-to-ground communication, atmospheric turbulence can lead to intermittent and unreliable communication links. Similarly, in astronomical telescopes, this phenomenon causes visual distortions such as blurring, twinkling, and shimmering of astronomical objects, making it challenging to obtain clear images. Figure 2.1 shows the impact of atmospheric turbulence on image quality by comparing images of a nearby galaxy taken with and without AO [45].

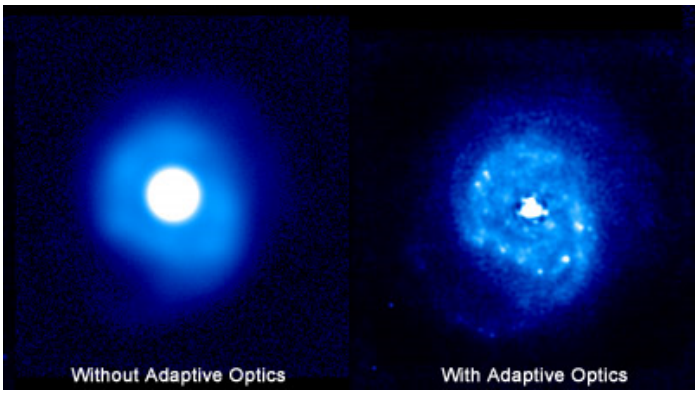

Figure 2.1: The nuclear region of the nearby galaxy NGC 7469, without and with AO from Canada France Hawaii Telescope.

Atmospheric turbulence arises from various phenomena, including convection, wind shear, and wind flow over objects. Convection occurs when the lower atmosphere is heated, causing convective gas bubbles to rise. This process can lead to the formation of cumulus clouds and lightning storms. Wind shear, on the other hand, results from differences in horizontal velocity between atmospheric layers. Additionally, when wind flows over objects such as mountains or telescope domes, it can induce turbulence [46]. This phenomenon has been extensively studied to develop theoretical explanations for the effects of atmospheric turbulence [47, 48] and to investigate it experimentally in AO [49, 50].

Aberrations may also have ***non-atmospheric sources***. One primary source of non-atmospheric aberrations is optical misalignment, which can negatively impact system performance. For instance, tip/tilt aberrations can shift the image position in imaging systems along the horizontal and vertical axes without affecting its quality. However, even a slight misalignment in optical data links, especially in single-mode fiber coupling, can significantly reduce performance. To mitigate these effects, various studies have focused on detecting and correcting tip/tilt aberrations in AO systems [51, 52] and developing high-performance AO systems with precise tip/tilt control [53]. Non-atmospheric aberrations can also be attributed to limitations in the technology employed during manufacturing, leading to microerrors. These microerrors can be addressed within the controller of the AO system [4].

Beyond atmospheric turbulence, misalignments, and manufacturing errors, there are ***other aberration sources*** that can impact optical systems. One such example is in ophthalmology, particularly in retinal imaging. In this case, turbulence in the vitreous humor of the eye can distort imagery. AO has been used to correct these distortions and has improved vision for patients with conditions such as keratoconus or scarring [54, 55, 56, 57]. Another source of aberration occurs in light transmission through seawater. With a refractive index greater than 1.3, seawater efficiently transmits light in the blue-green spectral region, which makes it an ideal medium for surface-to-submarine communications. However, substantial thermal gradients in seawater produce turbulence, which degrades optical signal quality. AO can be employed to mitigate these distortions and improve underwater optical communication [58, 59].

2.1.2 Wavefront Representation

The wavefront is a 2D map of the phase at a plane perpendicular to the line of sight from the beam's origin to the target. Wavefront aberrations are commonly expressed using a set of polynomials, including the Power-Series representation [4] and the widely used Zernike series [60, 5, 61]. One drawback of the Power-Series representation is that it is not an orthonormal set over a circular domain. In contrast, the Zernike series consists of orthonormal polynomials over a circle, making them very useful for working with circular apertures and convenient for serving as a set of basis functions. Another advantage of the Zernike series over the Power-Series representation is that these polynomials are mutually orthogonal

and linearly independent. Thus, each has an independent effect on the optical system [62].

Zernike Series

Zernike [60] introduced a set of polynomials that are orthonormal to the circular pupil, known as the Zernike series. Orthogonal polynomials can also be derived for other pupil geometries, such as hexagonal pupils, as described by Mahajan [63]. They are constructed as sums of power series terms with appropriate normalizing factors. One effective method for analyzing phase aberrations is by decomposing the wavefront phase function, denoted $\Phi(\rho, \theta)$, into the Zernike series [43]:

$$\Phi(\rho, \theta) = \sum_{n,m} \alpha_n^m Z_n^m(\rho, \theta) \tag{2.1}$$

where ρ represents the normalized radius, ranging from 0 to 1, and θ represents the angle measured clockwise from the y-axis in polar coordinates. The coefficients α_n^m are used for aberration balancing. Z_n^m denotes each mode of the Zernike polynomial with radial (n) and azimuthal (m) orders, as illustrated in Figure 2.2. The first three Zernike polynomials are the piston Z_0^0, x-tilt Z_1^1, and y-tilt Z_1^{-1}. The tilt term refers to the angle at which a wavefront is oriented with respect to a reference plane. The piston term refers to a constant retardation or advancement of the phase across the entire beam [4]. In simpler terms, the piston corresponds to a uniform expansion or displacement of the wavefront without any changes in its shape.

2.1.3 History of Adaptive Optics

Horace Babcock first proposed the AO technique in 1953 [64] to improve astronomical images. He suggested using a deformable optical element controlled by a wavefront sensor to correct for atmospheric distortions. In 1957, Vladimir P. Linnik built upon this idea by describing how a beacon placed in the atmosphere could probe disturbances and improve astronomical images. Linnik's paper provided the first reference to what are now termed "guide stars" [4]. A guide star is a reference star near the observed astronomical object as a secondary light source. The guide star is selected based on its brightness to ensure it is easily detectable by the wavefront sensor. Because the guide star is close to the main target, measuring the distortions in its wavefront allows for the inference of distortions in the wavefront

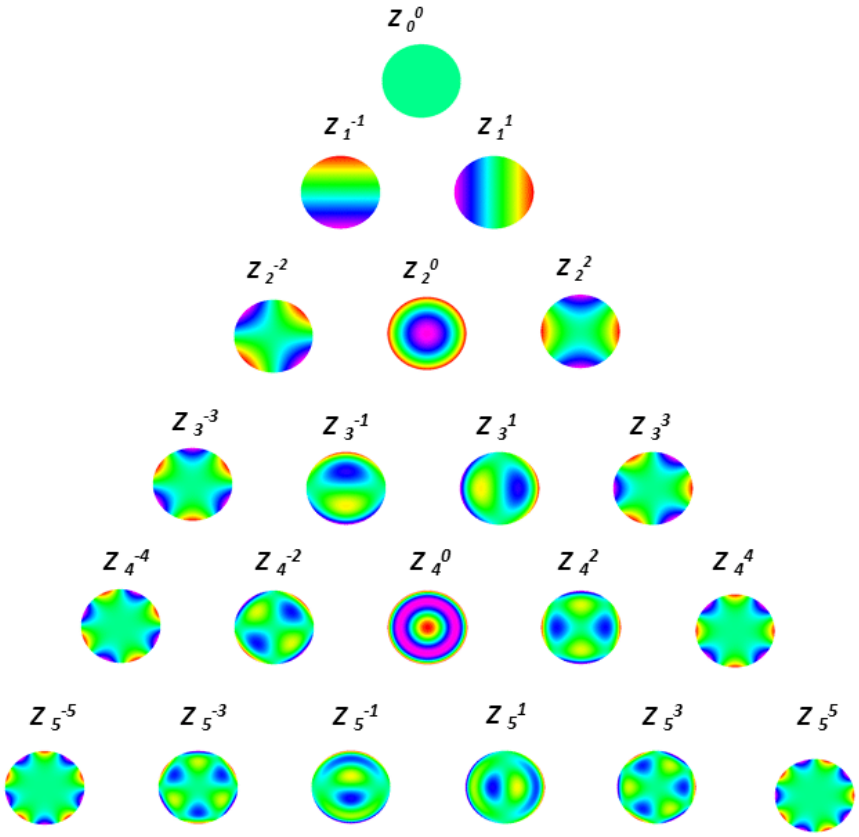

Figure 2.2: Representation of the Zernike pyramid formed by the first 21 modes

of the main target. Thus, the wavefront sensor can observe perturbations in the main target's wavefront with the help of the guide star.

Although Babcock and Linnik proposed methods that could have improved astronomical images, the high development cost at the time made building an AO system for astronomy impractical. The invention of the laser sparked interest in studying optical propagation through turbulent atmospheres, experimentally and theoretically. Labeyrie's laser speckle studies led to the proposal of using Fourier transforms to analyze speckle patterns in images of stars, aiming to improve the resolution of large telescopes. In this context, 'speckle' refers to the grainy structure observed when a laser beam is reflected from a diffusing surface [65].

Meanwhile, defense-oriented research began using segmented mirrors to mitigate atmospheric effects, aiming to concentrate laser beams on remote targets through trial and error using the multidither technique. As satellites were launched into orbit, the need for surveillance imaging emerged, leading to efforts to adapt these methods for capturing images of orbital objects [66].

The first AO system capable of sharpening two-dimensional images was de-

veloped by Hardy and his colleagues at Itek Corporation in 1977. This system featured an AC shearing interferometer, a parallel analog data processor, and a monolithic piezoelectric active mirror, all configured in a closed-loop arrangement, as shown in Figure 2.3 [67]. In the figure, the reference source refers to the guide star. A review article by Hardy in 1978 [68] provides a history of active and adaptive optics, detailing the state-of-the-art advancements at that time. The developments from the first three decades are further reviewed by Babcock [69], Hardy [70], and Greenwood [71].

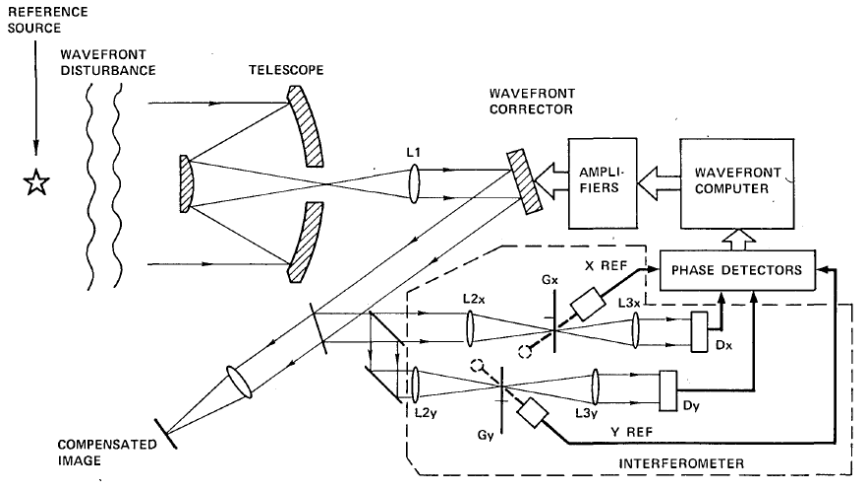

Figure 2.3: Block diagram of a real-time atmospheric compensation system proposed by Hardy et al. [67].

New techniques and findings are constantly being published in scientific journals and presented by technical societies. Although many problems still need to be addressed, new techniques can lead to discoveries. For example, while telescopes still require complex and expensive designs to image extrasolar planets, improvements in the AO technique have enabled many breakthroughs. The 2020 Nobel Prize in Physics was awarded to Reinhard Genzel and Andrea Ghez for their observations of the supermassive black hole at the center of the Milky Way, an achievement made possible with the vital assistance of AO [72].

2.1.4 Wavefront Correction in Traditional Adaptive Optics

The objective of AO is to minimize the phase aberration function, represented by $\Phi(\rho, \theta)$ (Eqn. 2.1), over the pupil of an optical system. This can be formalized as a reduction of the Zernike coefficients, α_n^m, which are mathematical terms that describe aberrations in an optical system. In other words, AO seeks to minimize

phase distortions and Zernike coefficients in the incoming light, ideally eliminating distortions on a plane. The closer the phase aberration function approaches zero, the fewer distortions remain in the optical system.

As schematized in Figure 2.4, the architecture of the traditional AO system used in telescopes is considered. Light from a satellite travels through space and enters Earth's atmosphere with the assumption of no phase aberration. However, as the light passes through the atmosphere and reaches the telescope's aperture, it encounters atmospheric turbulence, which varies as a function of time and space. This turbulence introduces a phase aberration Φ_{ab} at the telescope's entrance pupil. Afterward, the light at the entrance pupil is projected onto a deformable mirror comprising N_a actuators [73, 74, 75], which introduces an additional phase aberration Φ_{DM}. Thus, the overall phase aberration after passing through the deformable mirror is given by $\Phi_r = \Phi_{ab} + \Phi_{DM}$.

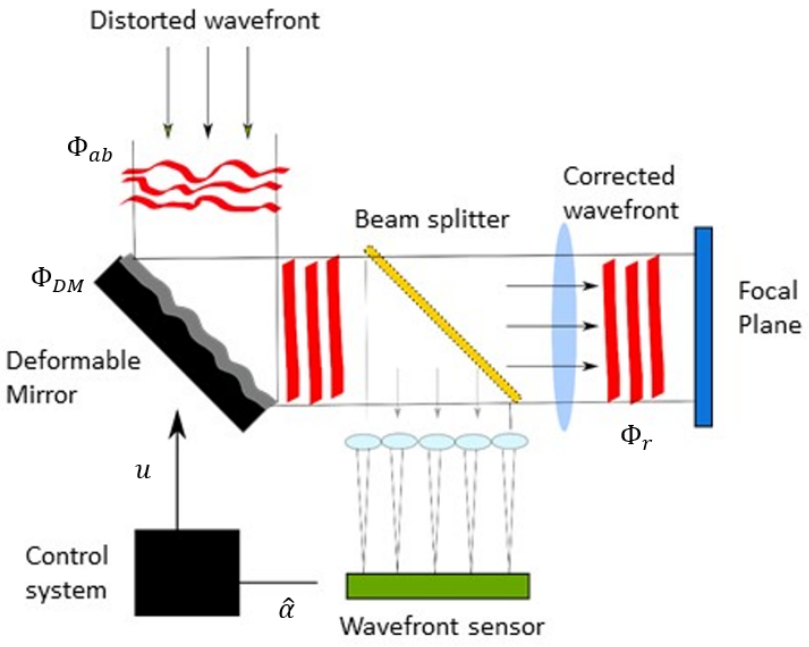

Figure 2.4: Example of a traditional AO system

After being projected from the deformable mirror, the light is split by a beam splitter into two paths: one directed towards the exit pupil and the other towards the wavefront sensor [76, 77, 78, 79]. Along the path leading to the exit pupil, the light is focused onto a focal plane, forming the final image with phase aberration Φ_r. The other path leads to the wavefront sensor, which estimates the phase aberration Φ_r as a set of Zernike coefficients N_α collected into a vector $\hat{\boldsymbol{\alpha}} \in \mathbb{R}^{N_\alpha}$. Finally,

a controller receives these coefficients, computes a vector $\boldsymbol{u} \in \mathbb{R}^{N_a}$, and generates a control signal for the actuators of the deformable mirror. The goal of the controller is to minimize the Zernike coefficient set $\hat{\boldsymbol{\alpha}}$, thereby minimizing the phase aberration Φ_r [43].

2.1.5 Wavefront Sensors

Wavefront sensors play a crucial role in quantifying aberrations in optical systems. Commonly used types include Shack-Hartmann, Pyramid, and Holographic Modal wavefront sensors. Although all these sensors are designed to measure and correct optical wavefront distortions, they differ significantly in their operating principles.

The Shack-Hartmann wavefront sensor uses a lenslet array to divide the wavefront into smaller segments and measures local slopes by detecting focal spot displacements [80, 81]. In contrast, the Pyramid wavefront sensor modulates the wavefront using a pyramidal optical element, converting phase distortions into measurable intensity patterns [82, 83]. Lastly, the Holographic Modal wavefront sensor compares a test wavefront against a holographically recorded reference wavefront and extracts aberrations by decomposing their difference into a set of basis functions called modes [84].

Among these, the Shack-Hartmann sensor is the most widely used due to its robustness, simplicity, and broad applicability in AO. Therefore, its functionality is explained in detail to provide a deeper understanding of its working principles.

Shack-Hartmann Wavefront Sensor

The Shack-Hartmann sensor consists of an aperture equipped with microlenses known as a microlens array. As the wavefront passes through the microlens array, each lens focuses a portion of the incoming wavefront onto a specific spot on a detector array. Each lens in the array represents a sub-aperture, sampling a distinct region of the wavefront as it enters the optical system [79]. In the absence of aberrations, the detector captures an evenly distributed array of spots, as shown in Figure 2.5 (top). However, when aberrations are present, the focal spots displace from their ideal positions, leading to an uneven distribution on the detector, as shown in Figure 2.5 (bottom) [85]. The Shack-Hartmann sensor analyzes these spot displacements to estimate wavefront distortions across the aperture.

The displacement vector of each sub-aperture is illustrated in Figure 2.6. It is

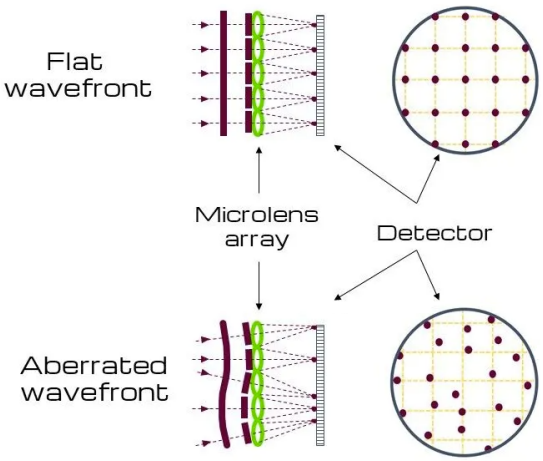

Figure 2.5: Shack-Hartmann Wavefront Sensor [85]

assumed that the Shack-Hartmann wavefront sensor has a circular aperture. For each sub-aperture (j), the coordinates of the focal spot's center $\left(x_c^{(j)}, y_c^{(j)}\right)$ are determined by calculating the centroids [43]:

$$x_c^{(j)} = \frac{\sum_i x_i I_i}{\sum_i I_i}, \qquad y_c^{(j)} = \frac{\sum_i y_i I_i}{\sum_i I_i} \tag{2.2}$$

The intensity, denoted as I_i, is measured by a pixel located at coordinates (x_i, y_i) on the detector's focal spot surface within a global reference frame centered on the aperture of the Shack-Hartmann wavefront sensor. The displacement of the focal spots $\left(s_x^{(j)}, s_y^{(j)}\right)$ is determined using the centroids, $\left(x_c^{(j)}, y_c^{(j)}\right)$, relative to the reference position $O^{(j)}$ in each sub-aperture:

$$s_x^{(j)} = \frac{\sum_i \left(x_i - x_c^{(j)}\right) I_i}{\sum_i I_i}, \qquad s_y^{(j)} = \frac{\sum_i \left(y_i - y_c^{(j)}\right) I_i}{\sum_i I_i} \tag{2.3}$$

Primot and Ellerbroek [86, 87] proposed a continuous form of Equation 2.3 in polar coordinates. Additionally, Dai [88] established a direct relationship between the coefficient of the series (α) and the displacement of the focal spots. A detailed discussion of the coefficient α is provided in Section 2.1.2.

Numerous wavefront reconstruction methods have been proposed and analyzed in the literature. One significant study is by Luke [89], who provided a comprehensive review and comparison of various wavefront reconstruction techniques.

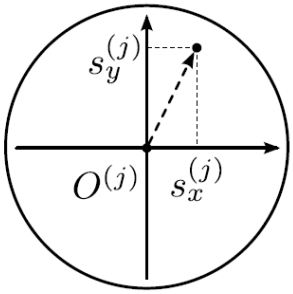

Figure 2.6: Illustration of the displacement vector $\left(s_x^{(j)}, s_y^{(j)}\right)$ of the focal spot centroid relative to the sub-aperture center $O^{(j)}$ in a Shack-Hartmann wavefront sensor.

2.2 Apdaptive Optics Applications

AO control can be categorized into two main approaches: Wavefront Sensor-Based AO and Wavefront Sensorless AO. The primary distinction between these methods lies in the use of a wavefront sensor. Wavefront sensors provide stability and high accuracy, but they also introduce significant costs and latency, which influence the choice of control method.

Given the critical role of AO across various fields, its application in astronomy is particularly noteworthy. Astronomical communication occurs in two forms: satellite-to-satellite and satellite-to-ground communication. In satellite-to-satellite communication, atmospheric turbulence is not a concern due to the absence of an atmosphere. However, in satellite-to-ground communication, atmospheric turbulence is the primary source of distortion, which affects the quality and precision of optical signals.

This section provides a literature review of AO applications in satellite-to-ground communication, with a focus on improving astronomical imaging and optical communication using wavefront sensor-based and sensorless control methods.

2.2.1 Wavefront Sensor-Based Adaptive Optics

Significant research and development have been conducted on wavefront sensor-based AO. Several researchers have provided comprehensive overviews and foundational principles to facilitate an in-depth understanding of this technique. Tyson has detailed the history and principles of AO, including the sources of aberrations, wavefront sensing, correction methods, and reconstruction techniques [4,

90]. Hardy [91] discussed the foundational principles for AO in astronomy, including optical image structures and methods for wavefront sensor correction and reconstruction. Additionally, Becker and Davies [11, 92] contributed to the field by presenting an extensive overview of AO principles and their various astronomical applications. Roddier [9] provided a thorough background on AO in astronomy and the design of AO systems, including detailed descriptions of AO using natural guide stars and laser beams.

In wavefront sensor-based AO, the primary components include wavefront sensors, deformable mirrors, and controllers [93, 94]. The selection of these components in an AO system for astronomy highly depends on the intended application.

Deformable Mirror Selection

One important consideration is the type and number of deformable mirrors used. Morgan [95] presented a study on Micro-Electro-Mechanical Systems (MEMS) deformable mirrors, highlighting their cost-effective and precise wavefront control capabilities, which make them suitable for high-contrast imaging of exoplanets with coronagraph instruments. Cahoy [96] conducted experiments to evaluate the ability of MEMS deformable mirrors to perform wavefront control for exoplanet direct imaging using Shack-Hartmann and Michelson interferometer wavefront sensors. Piezoelectric deformable mirrors have also been used in AO systems due to their relatively fast response and low cost. Kanno [97] presented a study on developing piezoelectric deformable mirrors for low-voltage AO applications. Similarly, Guan [98] provided a comprehensive overview of the historical development and performance of piezoelectric deformable mirrors, including test results from their implementation in 4m telescopes, 1.8m telescopes, and 1m solar telescopes.

In addition, using multiple deformable mirrors offers advantages over traditional single deformable mirrors in AO systems. A multi-conjugate deformable mirror system can correct a larger amount of atmospheric distortion because the mirrors can be independently adjusted, allowing for the correction of aberrations at various layers within the atmosphere. Johnston et al. [99] investigated methods for expanding an adaptive optical telescope's compensated field of view using multiple deformable mirrors. Their analysis included measurement noise, wavefront sensor sampling, and wavefront reconstruction from slope measurements. Conversely, Rigaut [100] examined the principles, limitations, and performance of multi-conjugate AO and concluded that the ultimate limitation of multi-conjugate deformable mirror systems arises from the discretization of the correction process.

Wavefront Sensor Selection

The selection of the type of wavefront sensor is a critical consideration in AO systems. As discussed in Section 2.1.5, the Shack-Hartmann wavefront sensor is popular and widely used due to its robustness and simplicity. It has been implemented in various astronomical applications [80, 81, 101]. However, other types of wavefront sensors can also be helpful. One such example is the Pyramid wavefront sensor. A study by Chew et al. [102] compared the performance of the Shack-Hartmann and Pyramid wavefront sensors in closed-loop wavefront compensation scenarios. The results indicated that the Pyramid wavefront sensor achieved a higher Strehl ratio than the Shack-Hartmann sensor. This sensor utilizes a pyramid structure to measure the slope of the wavefront, enabling it to measure the wavefront with fewer sub-apertures than the Shack-Hartmann sensor, as shown in Figure 2.7 [103]. The Pyramid wavefront sensor has been implemented in various systems, such as the Keck II AO system [104], the LBT telescope [105], and extremely large telescopes [106].

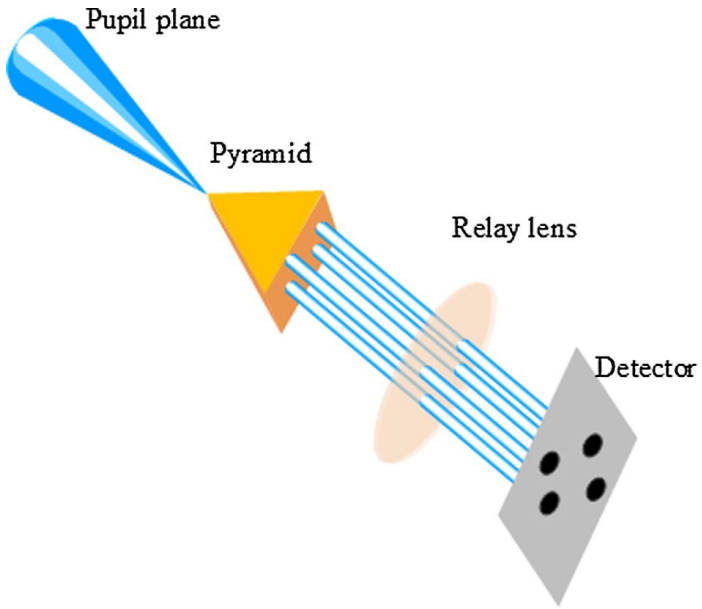

Figure 2.7: The schematic diagram of the Pyramid wavefront sensor

Controller Selection

The controller is another critical component of the AO system. It processes the recovered slopes from the wavefront sensor and applies them to the deformable mirror actuators using the command matrix [93]. Among the various controllers used

in the industry, the Proportional–Integral–Derivative (PID) controller is the most widely implemented and has played a significant role in control systems across various fields [107]. Wu et al. introduced a PID controller based on linear matrix inequalities, which was evaluated in an AO system to regulate the shape of a magnetic fluid deformable mirror [108].

Although the PID controller is commonly used in various practical applications, tuning its parameters can be complex, especially in AO. This process requires accounting for factors such as the intensity of atmospheric turbulence, detector noise, and other changing environmental conditions. These complexities make parameter tuning time-consuming and inefficient. To address this challenge, Ke et al. [109] proposed using fuzzy control to self-tune the PID controller parameters in AO. Their experiments demonstrated that this method improves system response speed and simplifies parameter adjustment for laser communication and wavefront correction.

However, PID control remains not ideal for AO systems due to its poor performance in correcting dynamic turbulence and disturbances with narrow bandwidths [110]. Adaptive control offers a solution for complex and dynamic environments by continuously adjusting their parameters. One such method is Linear Quadratic Gaussian (LQG) control based on the Kalman filter. Wang et al. [111] demonstrated that the energy of residual disturbances in LQG control based on the Kalman filter is nearly half that of PI control in AO. This finding verifies the advantage of LQG in mitigating disturbances. Petit et al. [112] proposed using LQG control based on the Kalman filter in most AO systems, with a particular focus on extreme systems. They demonstrated improved correction performance in a laboratory environment. Gibson et al. [113] claimed that the solution to phase distortion caused by atmospheric turbulence is an adaptive feedforward control loop. This loop, which is built around a multichannel lattice filter, drives the mirror to adapt to changing atmospheric conditions in real-time. Simulations for a 1-meter telescope demonstrated that the adaptive control loop significantly improves imaging resolution compared to a classical linear time-invariant control loop.

Tesch and Gibson [114] compared an adaptive controller based on a recursive least-squares lattice filter with an optimal linear time-invariant controller. This optimal controller was constructed using an identified state-space model of the turbulence flow within a classical AO loop. The predictive capabilities of both the adaptive and optimal controllers significantly improved performance, resulting in

a decrease in residual wavefront error and an increase in intensity on a target camera. Le Roux et al. [115] proposed an optimal closed-loop control law through multi-conjugate AO numerical simulations designed to represent astronomical observation using an 8-meter-class telescope in the near-infrared. Their approach expresses prior information on the turbulence and wavefront sensing noise into a state-space model. A Kalman filter then provided the optimal phase estimation. Numerical simulation demonstrated that their approach outperformed conventional techniques. Despite the high performance of adaptive and optimal control methods in AO systems, real-time distortion caused by dynamic atmospheric conditions can lead to instability in the controller's output, resulting in relatively high computational costs [116].

Model Predictive Control can be used to predict future wavefront effects. By predicting the wavefront effects, the latency introduced by measurement and computation of wavefront correction can be mitigated [117, 118]. Furthermore, Glück et al. [119] proposed a predictive control approach for multi-mirror AO systems to compensate for vibrations in Extremely Large Telescopes. Also, Guyon and Males [120] proposed a predictive control with an Empirical Orthogonal Functions framework for atmospheric predictions based on previous wavefront sensor measurements. Beyond predicting wavefront and atmospheric effects, research has also focused on predicting wind motion. Johnson et al. [121] presented a predictive controller that estimates the wind motion and velocity in the atmospheric turbulence layer using Gauss-Newton minimization.

The stability of predictive control can be improved by using neural networks. Neural networks are flexible, universal, and often used when no accurate mathematical model of the process exists [122]. To overcome the time delay between wavefront measurement and corrections in AO systems, Wong et al. [123] compared predictive control using empirical orthogonal functions (EOF) with predictive control using artificial neural networks (ANNs). They observed several advantages of ANNs over the EOF framework, notably the ability of the ANN framework to accommodate extended model complexity and the inclusion of nonlinearities.

In high-contrast imaging, temporal delay error often dominates the wavefront sensor error budget [124]. Also, systems often experience dynamic misalignment between the deformable mirror and the wavefront sensor, referred to as misregistration [125]. Nousiainen et al. proposed a model-based RL approach to address these limitations of AO systems for astronomy [18, 126]. They developed a closed-

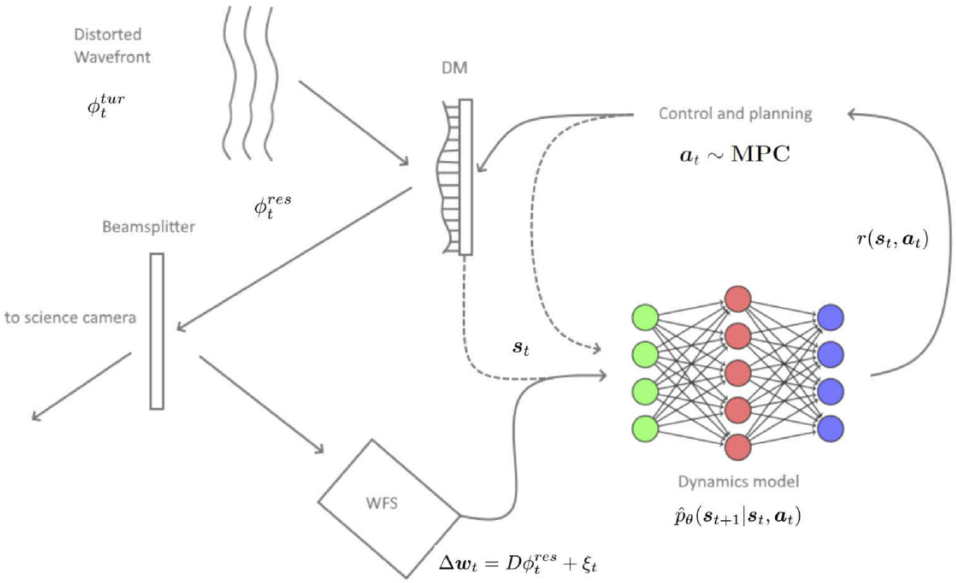

Figure 2.8: Example of AO control using model-based reinforcement learning [18]

loop AO system modeled as a Markov decision process, which is the mathematical framework for RL [127]. Their setup incorporates model predictive control, a wavefront sensor, and ANNs, as shown in Figure 2.8.

ANNs are composed of simple yet nonlinear units. Increasing the number of layers in these networks enables them to provide more detailed data representations and reduce unwanted variability [128]. However, as the number of layers increases, optimization difficulties arise due to the nonlinearity present at each layer [129]. This problem can be addressed by using neural networks with recurrent connections, known as Recurrent Neural Networks (RNNs). RNNs are designed to model sequential data for sequence recognition and prediction tasks [130]. They are composed of hidden states with nonlinear dynamics that serve as the memory of the network [131, 132]. This structure enables RNNs to store and memorize information, and map input sequences to output sequences at each timestep to predict the next sequences [133].

The Long Short-Term Memory (LSTM) network is a type of recurrent neural network designed to ensure that the gradient of the objective function concerning the state signal does not vanish [134, 135]. LSTM networks can effectively manage long-term dependencies [136]. To address the delay problem in AO systems, Chen [137] proposed a predictive model based on an LSTM Recurrent Neural Network that uses Zernike modal coefficients. This control model processes wavefront slope data from the wavefront sensor to compute the voltage adjustments required for

the deformable mirror by using Zernike modal coefficients. In addition, Liu et al. [138] claimed that control systems in AO use parametric techniques that require identifying and tracking turbulence parameters. This approach can complicate implementations and cause instability when facing variable conditions. To address this, they presented a wavefront predictor that uses an LSTM neural network to reduce latency while assuming no prior knowledge of the atmosphere, thus requiring no user input.

A Convolutional Neural Network (CNN) is another type of neural network specifically designed to capture spatial features and patterns in grid-structured data, such as images. This is achieved through a hierarchical architecture of layers that perform convolutional operations [139]. Swanson et al. [93] evaluated the performance of predictive control in AO systems for wavefront prediction and reconstruction by using convolutional LSTMs and dense CNNs. The aim was to reduce servo-lag error. The key difference between these two network types lies in their accounting for temporal information: Convolutional LSTMs account for temporal information, whereas dense CNNs do not. To improve the robustness of the AO systems under different seeing conditions and to demonstrate predictive capabilities, they proposed predictive controls based on an adversarial prior. Their findings revealed that convolutional LSTMs outperform dense CNNs when servo-lag is the dominant issue, while dense CNNs perform better when noise is the primary source of error [140].

The critical step in successfully implementing a model predictive control strategy is to develop a model capable of accurately predicting system behavior. Although standard models demonstrate strong predictive capabilities, achieving high accuracy remains challenging due to uncertainties in physical parameters and unmeasured states [141]. To better capture the dynamic behavior of complex nonlinear systems, different RL methods are used for system identification [142]. In this approach, a suitable model structure is first selected, followed by the identification of model parameters using available historical data. However, implementing predictive control with these techniques requires a large dataset, and obtaining accurate data is costly. This challenge is particularly significant in cost-effective AO systems, where limitations in precise components complicate accurate data collection [143].

AO control can also be achieved with model-free RL, which enables the controller to learn a nonlinear policy without requiring priori information on atmospheric dynamics. Pou et al. [17] proposed a model-free RL approach with an

autoencoder neural network to mitigate the effects of noise and bandwidth error in AO systems. Experimental results on an 8 m telescope equipped with a Shack-Hartmann wavefront sensor demonstrated that their method outperformed model-based predictive approaches.

Challenges of wavefront sensors

All the studies discussed in this section (Section 2.2.1) include wavefront sensors, either in their experimental setups or as part of data collection. However, wavefront sensor-based AO systems are costly and complex, with a significant portion attributed to the wavefront sensor, especially in systems that operate in the infrared for optical satellite-to-ground links. In such applications, infrared wavefront sensors suffer from high read noise and require cooling to reduce dark noise, unlike silicon-based sensors [104].

In addition, wavefront sensors introduce several other limitations. They consume a portion of the incident beam's intensity, have a limited dynamic range, and introduce latency between measurements and the actuation of the deformable mirror. This latency can result in outdated wavefront measurements as the satellite rapidly moves across the sky, leading to significant errors at the characteristic space-time scales. Furthermore, in LEO satellite downlinks, wavefront sensors may fail to guarantee stable fiber coupling under intense scintillation and phase wavefront singularities caused by high turbulence [144, 145, 146].

I aim to facilitate fast, reliable, and lower-cost satellite-to-ground communications to improve data transmission efficiency. Optical satellite-to-ground communication using wavefront sensorless AO offers a promising solution to these issues, particularly through improved light coupling with reduced latency and lower system cost. For instance, it is estimated that for every 1% increase in light coupling, the system cost decreases by approximately 2% compared to the original cost [147, 148, 149]. In a traditional AO system, opting for a 40 cm telescope instead of a 60 cm one leads to substantial savings in dome and telescope size that can be more than USD 100,000. The costs of these components are on the order of: USD 20,000–50,000 for the mount and telescope system [149], a minimum of USD 50,000–100,000 for the wavefront sensor, and a minimum of USD 100,000 for the dome enclosure. In contrast to traditional AO, incorporating an RL model into the system would only require a compact processor, such as a field-programmable gate array (FPGA), and a simpler detection unit (e.g., a multi-channel photodetector), which would be significantly more cost-effective and offer minimal latency.

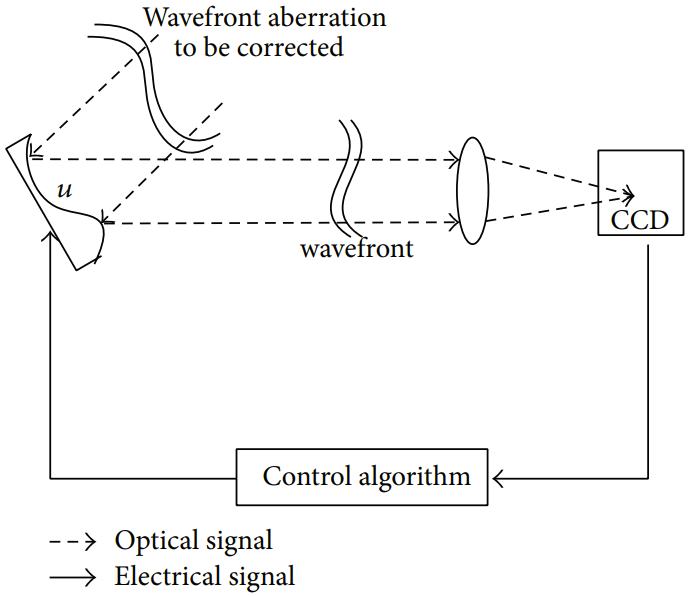

Figure 2.9: Wavefront sensorless AO system [152]

2.2.2 Wavefront Sensorless Adaptive Optics

The primary motivation for developing wavefront sensorless adaptive optics (WSL-AO) systems is to eliminate the need for wavefront sensors to reduce costs and latency. However, the complexity of optical systems presents a significant challenge in replacing wavefront sensors with a reliable algorithm for predicting the required distortion correction. Therefore, developing an effective algorithm to overcome this challenge is essential.

WSL-AO systems are broadly categorized into model-based and model-free approaches [150, 151]. A simplified example of a WSL-AO system is shown in Figure 2.9 [152].

Model-Based WSL-AO

In the model-based WSL-AO approach, wavefront aberrations are estimated using a predefined system model that establishes a relationship between optical aberrations and a metric function. This metric function is typically derived based on the basis modes of aberrations. An advantage of this approach is its flexibility in selecting arbitrary basis modes while accounting for their non-orthogonality in the linear least squares optimization process [153].

One commonly used model-based WSL-AO method for aberration correction relies on the approximately linear relationship between the mean square of aberra-

tion gradients and the second moment of the far-field intensity distribution. In this approach, the AO system utilizes singular value decomposition of the correlation matrix of aberration mode gradients to reconstruct new Zernike aberration modes [154, 155, 156, 157].

Other model-based WSL-AO methods have also been proposed. One such approach involves using holographic modal wavefront sensing with a large dynamic range to establish a relationship between aberrations and intensity. This approach represents aberrations using Lukosz modes, where gradients are orthogonal and modal coefficients can be estimated independently [158]. Additionally, a fast model-based wavefront correction based on the linear phase diversity method is presented [159, 160]. Although it is claimed that this method is suitable for various turbulence conditions, its application is limited to static or slowly changing wavefronts. This means that significant changes during the iterative process can diminish the effectiveness of this technique. Furthermore, a self-calibration procedure has been proposed to obtain the Gram matrix without requiring a wavefront sensor. The Gram matrix represents inner products between vectors that encode image features or wavefront distortions. When properly calibrated, it can aid in simultaneous wavefront distortion correction by using the influence functions of a deformable mirror as basis modes [161].

Model-based WSL-AO methods aim to improve correction by utilizing structural information about the performance metric function. However, these methods provide only limited improvements in convergence speed and come with high implementation costs. On the other hand, model-free WSL-AO methods offer a more cost-effective implementation and provide faster and more efficient convergence [14].

Model-Free WSL-AO

In contrast to model-based approaches, model-free WSL-AO methods optimize system performance without relying on an explicit system model. Instead, they use direct optimization techniques that iteratively adjust wavefront to maximize a performance. This approach simplifies the system's complexity, making it well-suited for a wide range of wavefront correction scenarios.

One simple model-free algorithm used in WSL-AO is PID control. To improve the dynamic behavior of the control system, Zheng et al. [162] proposed a PID control method that uses a tracking differentiator to arrange the transient process. In this work, a quadrant photodetector is used to measure the tip/tilt errors intro-

duced by the deformable mirror. However, tuning PID parameters in AO can be challenging due to the system's complexity.

The general stochastic search algorithm is a widely used optimization approach for correcting wavefront distortions in WSL-AO systems. Three common general stochastic search algorithms applied are Simulated Annealing [163], the Genetic Algorithm [164], and Stochastic Parallel Gradient Descent (SPGD) [165, 166, 167, 168]. Among these, SPGD is the most popular choice due to its simple implementation and robust correction capabilities [151, 169]. However, the SPGD algorithm tends to converge slowly and is prone to local optima, which can degrade correction performance. This challenge becomes more significant as the search space for control variables expands due to an increased number of correction units or aberration modes [170].

To address these challenges, significant efforts have been made to improve the SPGD algorithm by integrating it with other techniques. Examples include the decoupled SPGD approach, which combines SPGD with pattern recognition [171], and SPGD integrated with the Adam or AMSGrad optimizers from deep learning [172, 151]. In addition, He et al. [150] proposed a hybrid approach for free space optical communication that combines model-based and model-free WSL-AO. The model-based method utilizes geometric optics theory with sphere packing [173] to compensate for low-order aberrations. Meanwhile, the model-free method utilizes SPGD to address high-order aberrations.

RL is a promising approach for reducing temporal latency and improving performance across various atmospheric conditions without requiring a system model. One commonly used RL algorithm is the Deep Deterministic Policy Gradient (DDPG) algorithm. Hu et al. [22, 174] applied the DDPG algorithm in their AO system and demonstrated that its correction performance was comparable to that of the SPGD algorithm. However, the correction speed achieved with DDPG was approximately 2.5 to 9 times faster than that of SPGD.

To achieve high-contrast imaging for exoplanets observations (or for communication in severe atmospheric turbulence), AO systems are required to meet the necessary contrast levels. However, the performance of these systems is often limited by telescope vibrations and temporal errors caused by the latency of the control loop. These challenges cannot be adequately addressed using the standard DDPG algorithm. To overcome these limitations, a DDPG algorithm integrated with a Recurrent Neural Network (RNN) is proposed [175, 176]. This approach enables the controller to learn the sequential dynamics of system observations and predict the

wavefront correction in real-time.

Wavefront distortion compensation in AO using a Convolutional Neural Network (CNN) has also been developed to improve image quality when a laser is propagated through the atmosphere. This approach uses a CNN to process input images and predict the corresponding Zernike coefficients. Results demonstrated that this approach improves the performance and reduces wavefront residual variance compared to no compensation [177, 14]. Further research investigated various CNN architectures for turbulence wavefront detection, including a standard CNN, a ResNet network [178], and an EfficientNet-B0 network [179]. Among these, the EfficientNet-B0 CNN outperformed the others regarding wavefront detection accuracy and speed under various turbulence conditions [180].

2.2.3 Limitations of Current Control Strategies

While the discussed literature provides valuable insights into AO control strategies, the majority of research focuses on optimizing image sharpness and uses cameras as feedback sensors. Cameras have high resolution and data acquisition and processing times, which introduce delays that are incompatible with real-time communication requirements. These approaches are not suitable for satellite-to-ground optical communication, which requires providing reliable optical data links while minimizing latency and cost.

Among the remaining studies, model-based wavefront sensorless approaches have been applied to optical data links. However, these methods require an accurate model of the system or the atmosphere, which is difficult to obtain and maintain in the highly dynamic turbulence conditions of LEO satellite-to-ground communication. As such, model-based solutions are not well-suited for practical use in these environments.

Some model-free approaches, including SPGD and RL-based controllers, have shown promise. However, many of these still rely on image-based metrics or require long convergence times, which limits their applicability to scenarios with short coherence times. Therefore, despite ongoing progress, a model-free solution for optical communication that can operate with minimal latency and low-pixel-count feedback remains an unmet need in the current literature.

Motivated by the limitations of existing methods and the need for a wavefront sensorless AO system using a low-pixel-count detector, RL-based AO systems present a compelling alternative. An advantage of RL-based control is its

ability to address complex dynamical systems, especially in highly uncertain environments where designing optimal controllers is difficult [181]. Additionally, RL-based controllers are model-free, enabling them to learn effective control policies directly from interaction without requiring an explicit system model. This characteristic makes them particularly useful for applications where system dynamics are difficult to model accurately.

To support this approach, it is essential to provide a background on RL.

2.3 Background on Reinforcement Learning

Machine learning algorithms can be broadly categorized into supervised learning, unsupervised learning, and RL [182]. Supervised learning involves learning from a labeled dataset provided by a knowledgeable external supervisor. On the other hand, unsupervised learning focuses on finding structure hidden in collections of unlabeled data. RL differs significantly from both supervised and unsupervised learning. RL involves learning how to map environmental states to actions in a way that maximizes the expected cumulative future rewards [127]. Also, RL enables a controller to be trained online in dynamic environments that evolve over time.

Markov Decision Process is a mathematically idealized form of an RL problem for which precise theoretical statements can be made [127].

2.3.1 Markov Decision Process

The Markov Decision Process (MDP) is a discrete-time stochastic control process that provides a framework for describing the environment with which an RL controller interacts [183, 184]. The graphical representation of the agent and environment interaction in an MDP is shown in Figure 2.10 [127].

An MDP is defined by the tuple (S, A, P, r, γ), where an RL controller interacts with an environment in discrete time steps. In this formulation, $s_t \in S$ denotes the state of the environment at time t, with S representing the continuous state space. The RL-controller, also called the "Agent", has the policy, which is the strategy it uses to determine actions $a_t \in A$ at each time step. Here, A represents the action space. $r : S \times A \rightarrow R$ is the reward function, and γ is the discount factor. The reward function and discount factor are explained in more detail in Section 2.3.2.

In the MDP framework, P represents the transition probability function, which

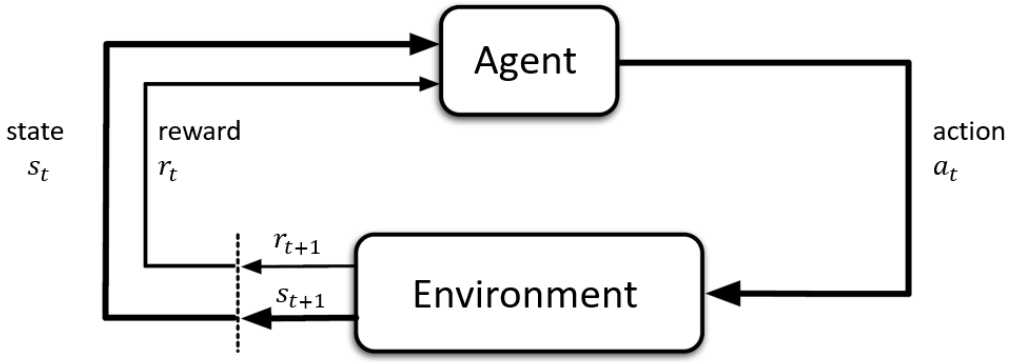

Figure 2.10: The agent–environment interaction in a Markov decision process

determines the probability of the environment transitioning to the next state $s_{t+1} \in S$. Generally, the state of the process can be influenced by actions and states from past history, as described below:

$$P(s_t, a_t, s_{t+1}) = P(s_{t+1}|s_t, \cdots, s_0, a_t, \cdots, a_0) \tag{2.4}$$

The transition probability function $P(s_t, a_t, s_{t+1})$ represents the probability of the environment transitioning from s_t to s_{t+1} based on the action a_t. This probability depends on the history of previous states $s_0, \cdots, s_t \in S$ and actions $a_0, \cdots, a_t \in A$.

The MDP is based on the Markov property, which describes the memoryless nature of a stochastic process. This property implies that the future state depends only on the current state and action, as these contain all the necessary information from the past. The Markov Property can be expressed using the following equation:

$$P(s_t, a_t, s_{t+1}) = P(s_{t+1}|s_t, a_t) \tag{2.5}$$

According to this equation, $P : S \times A \rightarrow S$ represents that the probability of transitioning to the next state $s_{t+1} \in S$ depends on the current state $s_t \in S$ with the current action taken $a_t \in A$.

As shown in Figure 2.10, the RL model operates as a sequential loop. At each time step t, the agent observes the current state of the environment $s_t \in S$. Also, the environment provides feedback to the agent in the form of a reward $r_t = r(s_t, a_t)$. The reward indicates how good the action was in the immediate step. Based on its policy, the agent generates an action $a_t \in A$, where a_t is the action taken in

the environment at time step t. The environment then transitions to the next state $s_{t+1} \in S$ according to the transition probability function $P(s_{t+1}|s_t, a_t)$.

While the MDP framework provides a mathematical foundation for modeling decision-making problems, it assumes the transition probability function, P, is known. However, in RL, this transition is usually unknown. For this reason, RL algorithms learn optimal policies directly through interaction with the environment using model-free methods that do not require knowledge of P or model-based methods that estimate P from experience.

2.3.2 Reward and Return

In RL, the *reward* is the feedback an agent receives from the environment at a given time step. This scalar value represents the immediate performance of the action taken by the agent. In contrast, the *return* refers to the cumulative rewards the agent receives over the long run. The main objective of the agent is to find an optimal policy that maximizes the *expected return*.

The relationship between reward and return at time t is defined as follows [127]:

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{2.6}$$

The variable G_t is the discounted sum of future rewards, with a discount factor $0 \leq \gamma \leq 1$. If $\gamma = 0$, the agent is *myopic*, focusing only on maximizing immediate reward, r_{t+1}. As γ approaches 1, the return objective takes future rewards into account more strongly, and the agent becomes more farsighted.

Returns at successive time steps are recursively related:

$$\begin{aligned} G_t &= r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} + \cdots \\ &= r_{t+1} + \gamma\left(r_{t+2} + \gamma r_{t+3} + \gamma^2 r_{t+4} + \cdots\right) \\ &= r_{t+1} + \gamma G_{t+1} \end{aligned} \tag{2.7}$$

This equation indicates that the return at the current time step, G_t, is equal to the immediate reward, r_{t+1}, plus the discounted return from the next time step, γG_{t+1}.

2.3.3 Policy and Value Function

The agent contains a *policy*, which maps states to the probabilities of possible actions. If the agent follows the policy π at time t, then $\pi(a|s)$ represents the probability of taking action $a_t = a$ given that the state $s_t = s$. In other words, the policy π defines a probability distribution over $a \in A(s)$ for each $s \in S$.

The *value function* for a state s under a policy π represents the expected return when starting in state s and following policy π. In MDP, the value function can be defined as:

$$V_\pi(s) = \mathbb{E}_\pi \left[G_t | s_t = s \right] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s \right], \forall s \in S \tag{2.8}$$

where $\mathbb{E}_\pi [.]$ denotes the expected value of a random variable under the policy π. The function V_π is called *state-value function for policy* π [127].

Additionally, the value of taking action a in state s under a policy π is defined as *action-value function for policy* π, denoted by $Q_\pi(s, a)$. The action-value function is defined as the expected return starting from state s, taking the action a, and following policy π:

$$Q_\pi(s, a) = \mathbb{E}_\pi \left[G_t | s_t = s, a_t = a \right] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a \right] \tag{2.9}$$

A key property of the state-value function, as described in Equation 2.8, is that it satisfies a recursive relationship similar to the one established for the return in Equation 2.7.

$$\begin{aligned} V_\pi(s) &= \mathbb{E}_\pi \left[G_t | s_t = s \right] \\ &= \mathbb{E}_\pi \left[r_{t+1} + \gamma G_{t+1} | s_t = s \right] \\ &= \sum_a \pi(a|s) \sum_{s'} P(s'|s, a) \left[r + \gamma \mathbb{E}_\pi \left[G_{t+1} | s_{t+1} = s' \right] \right] \\ &= \sum_a \pi(a|s) \sum_{s'} P(s'|s, a) \left[r + \gamma V_\pi(s') \right], \forall s \in S \end{aligned} \tag{2.10}$$

where $r = r(s, a, s')$ is the immediate reward the agent receives when it transitions from a state s to a next state s' after taking an action a. This equation is known as *Bellman equation* for V_π. It establishes a relationship between the value of a state and the values of its successor states. Following the same steps as in equation 2.10, the Bellman equation for the action-value function $Q_\pi(s, a)$, defined in Equation

2.9, can be derived as follows:

$$Q_\pi(s,a) = \sum_{s'} P(s'|s,a) \left[r + \gamma \sum_{a'} \pi(a'|s') Q_\pi(s',a') \right], \forall s \in S, \forall a \in A \tag{2.11}$$

A relationship exists between the state-value function $V_\pi(s)$ and the action-value function $Q_\pi(s,a)$, as illustrated in Figure 2.11 [127].

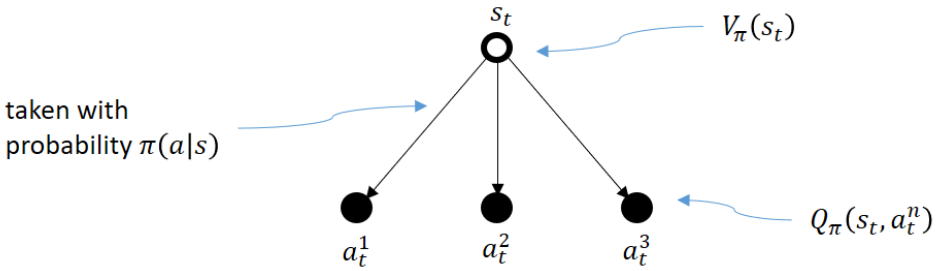

Figure 2.11: Relation between state-value and action-value functions

At time t, for a given state s_t, the state-value function $V_\pi(s_t)$ is computed. Using the current policy $\pi(a|s)$ and the state s_t, the possible actions a_t^n, where $n \in (1,2,3,\cdots)$, are generated. For each action a_t^n, the action-value function $Q_\pi(s_t,a_t^n)$ is calculated, which represents the expected return for taking action a_t^n in state s_t and then following the policy π. This establishes the relationship between the state-value function $V_\pi(s)$ and action-value function $Q_\pi(s,a)$, which can be expressed as:

$$V_\pi(s) = \sum_a \pi(a|s) Q_\pi(s,a) \tag{2.12}$$

The above equation indicates that the state-value function is the weighted sum of the action-value function for all possible actions in state s, where the weights are the probabilities of choosing each action according to the policy.

In addition, the action-value function $Q_\pi(s,a)$ can be expressed in terms of the state-value function, as shown in Figure 2.12 [127].
At time t, for a given state s_t and action a_t^n, the action-value function $Q_\pi(s_t,a_t^n)$ is computed. Using the transition probability function $P(s_{t+1}|s_t,a_t^n)$, the possible next states s_{t+1}^m, where $m \in (1,2,3,\cdots)$ and their immediate reward value $r(s_t,a_t^n,s_{t+1}^m)$ are generated. For each state s_{t+1}^m, the state-value function $V_\pi(s_{t+1}^m)$ is

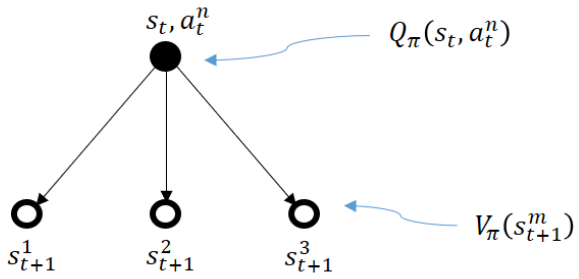

Figure 2.12: Relation between action-value and state-value functions

calculated, which represents the expected return in state s_{t+1}^m. This establishes the relationship between the action-value function $Q_\pi(s,a)$ and state-value function $V_\pi(s)$, which can be expressed as:

$$Q_\pi(s,a) = \sum_{s'} P(s'|s,a) \left[r(s,a,s') + \gamma V_\pi(s') \right] \tag{2.13}$$

The above equation indicates that the action-value function is the weighted sum of the discounted state-value of the next states and the immediate reward. The weights are the transition probability function, which defines the probability of transitioning to each possible next state from state s after taking action a.

2.3.4 Optimal Policy and Value Function

A policy π is considered better than or equal to another policy π' if and only if its expected return is greater than or equal to that of π' for all states [127].

$$\pi \geq \pi' \iff V_\pi(s) \geq V_{\pi'}(s), \ \forall s \in S \tag{2.14}$$

The objective of an optimal control problem is to determine the *optimal policy* that maximizes the cumulative return. *Optimal state-value function* represents the maximum expected return that can be achieved when the agent follows the optimal policy.

$$V_*(s) = \max_\pi V_\pi(s), \ \forall s \in S \tag{2.15}$$

Similarly, *optimal action-value function* is defined as:

$$Q_*(s,a) = \max_\pi Q_\pi(s,a), \ \forall s \in S, \forall a \in A \tag{2.16}$$

The function $Q_*(s,a)$ represents the expected return for taking action a in state s and following the optimal policy.

The optimal state-value function can be expressed as the *Bellman optimality equation*, independent of any specific policy. This equation states that the value of a state under an optimal policy is equal to the expected return achieved by taking the best possible action from that state [127].

$$\begin{aligned} V_*(s) &= \max_{a \in A} Q_{\pi_*}(s,a) \\ &= \max_a \mathbb{E}_{\pi_*}\left[G_t | s_t = s, a_t = a\right] \\ &= \max_a \mathbb{E}_{\pi_*}\left[r_{t+1} + \gamma G_{t+1} | s_t = s, a_t = a\right] \\ &= \max_a \mathbb{E}_{\pi_*}\left[r_{t+1} + \gamma V_*(s_{t+1}) | s_t = s, a_t = a\right] \\ &= \max_a \sum_{s'} P(s'|s,a)\left[r(s,a,s') + \gamma V_*(s')\right] \end{aligned} \tag{2.17}$$

The last two equations of (2.17) represent the Bellman optimality equation for the state-value function.

Also, following similar steps, the Bellman optimality equation for the action-value function is:

$$Q_*(s,a) = \sum_{s'} P(s'|s,a)\left[r(s,a,s') + \gamma \max_{a'} Q_*(s',a')\right] \tag{2.18}$$

2.3.5 Policy Iteration and Value Iteration

Policy iteration is an iterative method that alternates between *policy evaluation* and *policy improvement*, as shown in Figure 2.13 [127]. In the policy evaluation, the state-value function V_π is computed for the current policy π. In policy improvement, the policy π is updated by making it greedy with respect to the current state-value function V_π.

The policy evaluation process is represented by Equations 2.12 and 2.13, as expressed below:

$$V_\pi(s) = \sum_a \pi(a|s) \sum_{s'} P(s'|s,a)\left[r(s,a,s') + \gamma V_\pi(s')\right] \tag{2.19}$$

Following the policy evaluation using the above equation, the policy is improved using Equation 2.17:

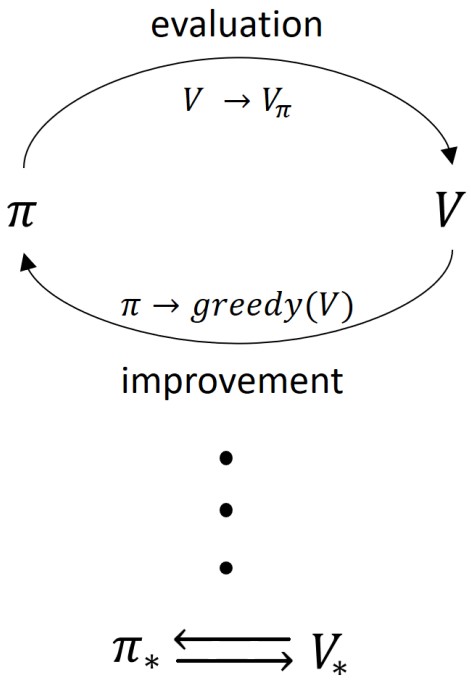

Figure 2.13: Generalized Policy Iteration

$$\pi'(s) = \arg\max_a \sum_{s'} P(s'|s,a) \left[r(s,a,s') + \gamma V_\pi(s') \right] \tag{2.20}$$

As illustrated in Figure 2.14, the policy iteration process terminates when the policy evaluation converges, resulting in the optimal policy and optimal state-value function [127].

Unlike policy iteration, which alternates between policy evaluation and policy improvement, *value iteration* uses a single iterative process. It updates the state-value function using the Bellman optimality equation directly and derives the optimal policy once the state-value function converges to the optimal state-value function $V_*(s)$. The value iteration equation is derived from the Bellman optimality equation (2.17) and is defined as:

$$V_{k+1}(s) = \max_a \sum_{s'} P(s'|s,a) \left[r(s,a,s') + \gamma V_k(s') \right] \tag{2.21}$$

where k represents the iteration number in the value iteration process. Once the value function $V(s)$ has converged, the optimal policy $\pi_*(s)$ can be determined using the following equation:

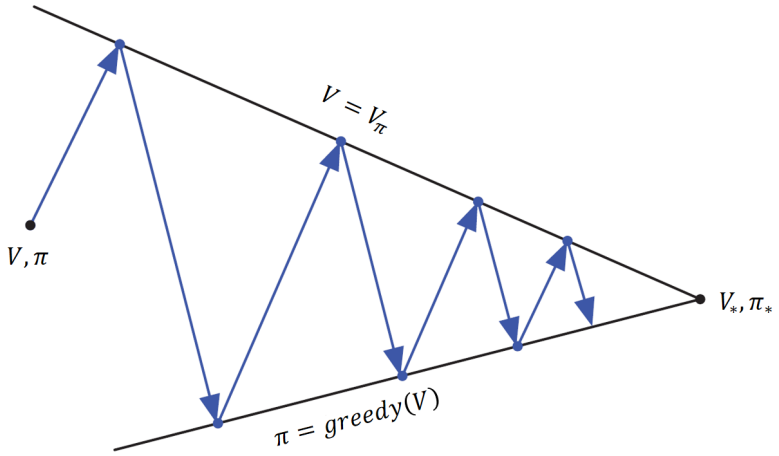

Figure 2.14: Visualization of the Generalized Policy Iteration Process

$$\pi_*(s) = \arg\max_a \sum_{s'} P(s'|s,a) \left[r(s,a,s') + \gamma V_*(s') \right] \tag{2.22}$$

for all $s \in S$.

2.3.6 Model-Free Learning

Policy Iteration and Value Iteration are dynamic programming methods that solve the Bellman equations exactly, but only when a model of the environment is available. Dynamic programming requires access to the transition probability $P(s'|s,a)$, which makes it a model-based approach. However, this transition function is often unknown or intractable to model accurately in many real-world cases. In such cases, model-free learning becomes essential. Model-free learning does not require access to this transition probability function. Instead, the agent interacts directly with the environment and learns from sampled transitions collected through experience. Although P is not explicitly known, its effects are indirectly observed in the collected experience along trajectories [127].

Model-free methods are commonly categorized into three types: (1) value-based methods, which learn a value function and derive a policy from it; (2) policy-based methods, which learn the policy directly; and (3) actor-critic methods, which combine both approaches by using value functions to guide policy updates.

Value-Based Methods

Value-based methods are a category of model-free learning in which the agent learns value functions directly from experience and then derives a policy based on value estimates. Two common approaches within this category are the Monte Carlo (MC) method, which updates value estimates only after complete episodes (i.e., without bootstrapping), and Temporal-Difference (TD) learning, which updates value estimates using the current estimate of the value of the next state, also known as bootstrapping [127]. Since the problem is formulated as an infinite-horizon task, TD learning is the method of choice in this context.

The simplest TD method, known as $TD(0)$, where the 0 denotes a one-step return, is formulated as follows [127]:

$$V(s_t) \leftarrow V(s_t) + \alpha \left[r_{t+1} + \gamma V(s_{t+1}) - V(s_t) \right] \tag{2.23}$$

where α is a learning rate. The term $r_{t+1} + \gamma V(s_{t+1})$ is the bootstrapped target used to update $V(s_t)$. The difference between the bootstrapped target and the current estimate, $V(s_t)$, is known as the TD error, which is defined as:

$$\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t) \tag{2.24}$$

As the TD error approaches zero, it typically indicates that the value function converges toward the expected return under the current policy, which is a sign of learning progress.

Policy-Based Methods

Policy-based methods are a category of model-free learning in which the agent learns the policy directly, rather than deriving it from a value function. A common class of these methods is the policy gradient method, which optimizes the policy by maximizing the expected return using gradient ascent. Gradient ascent is a general optimization technique that updates the policy parameters by moving in the direction of the gradient of the expected return [33].

A detailed explanation of the computation involved in policy gradient methods is provided in Section 2.4.3.

Actor-Critic Methods

Actor-critic methods are a category of model-free learning that combines value-based and policy-based approaches. These methods consist of two components: an actor, which represents the policy, and a critic, which estimates a value function and provides feedback to improve the actor's policy [185].

Actor-critic methods can be understood as a model-free analog of policy iteration in dynamic programming, but without requiring access to the transition probability function $P(s'|s,a)$. The critic evaluates the current policy by estimating value functions (similar to policy evaluation), while the actor updates the policy using the critic's feedback (similar to policy improvement) [185]. This iterative process between the actor and critic forms a loop that resembles policy iteration.

Typically, the actor uses a parametrized policy $\pi_\phi(a|s)$, while the critic estimates a state-value function $V_\theta(s)$ or an action-value function $Q_\theta(s,a)$. The symbols ϕ and θ denote the parameters of the actor's policy network and the critic's value function network, respectively. The critic learns by minimizing the TD error (δ_t; Equation 2.24) to improve its value estimates. The actor then applies the policy gradient method (Equation 2.42) to adjust the policy parameters in a direction that increases the expected return based on the critic's evaluation:

$$\phi \leftarrow \phi + \alpha\, \delta_t\, \nabla_\phi \log \pi_\phi(a_t|s_t) \tag{2.25}$$

This equation illustrates how actor-critic methods combine value estimation and policy gradients to improve the policy based on the experiences collected through interaction with the environment.

2.4 Deep Reinforcement Learning Algorithms

I review three families of RL algorithms: Soft Actor-Critic (SAC), Deep Deterministic Policy Gradient (DDPG), and Proximal Policy Optimization (PPO). These algorithms cover a range of approaches, including on-policy and off-policy learning, stochastic and deterministic policies, and entropy-based methods. In on-policy algorithms, like PPO, the policy is updated based on experiences collected while following the current policy. In contrast, off-policy algorithms, such as SAC and DDPG, can learn from past experiences collected using different policies.

These variations are essential when considering their applicability in wavefront sensorless AO, as each approach has its strengths and weaknesses. Off-policy algo-

rithms like SAC and DDPG are generally more sample-efficient than on-policy algorithms. Also, the entropy regularization in SAC enables rich exploration, which makes it effective in high-dimensional action spaces. On the other hand, PPO, an on-policy algorithm, is often known for its stability and ease of training. Given sufficient training time, on-policy algorithms can provide high-performance results. Each algorithm is discussed in more detail in the following subsections.

2.4.1 Soft Actor-Critic (SAC)

SAC is an off-policy actor-critic algorithm based on the maximum entropy RL framework. It is particularly useful in complex and stochastic environments, such as wavefront sensorless AO systems. SAC uses deep neural networks to approximate the actor and the critic. The actor is responsible for maximizing the expected return while maintaining high entropy, which encourages exploration and helps avoid getting stuck in suboptimal policies. Meanwhile, the critic estimates the Q-function for a given state-action pair. The Q-function provides feedback for policy improvement by guiding the agent toward actions that maximize cumulative expected future rewards [32].

To describe SAC effectively, it is crucial first to provide an overview of the entropy-regularized RL framework. Entropy is a measure of average uncertainty or information in a probability distribution. In RL, it quantifies the uncertainty of a policy when taking action for a given state. For example, the entropy of a coin toss can be determined by the probability distribution of its outcomes. A coin that is heavily biased toward heads will have low entropy because the outcome is more predictable, while a fair coin, with equal probabilities for heads and tails, will have high entropy because the outcome is maximally uncertain.

Let the action a_t at time t be sampled from the policy $\pi(a_t|s_t)$ at state s_t. The entropy $\mathcal{H}$ of the policy's action distribution can be defined as [186]:

$$\mathcal{H}\left(\pi\left(\cdot|s_t\right)\right) = \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\left[-\log \pi(a_t|s_t)\right] \tag{2.26}$$

where ρ_π is the distribution of trajectories induced by policy π. The notation $(s_t, a_t) \sim \rho_\pi$ indicates that the state-action pair (s_t, a_t) is sampled from ($\sim$) the distribution ρ_π.

In standard RL, the objective is to learn a policy $\pi(a_t|s_t)$ that maximizes the expected cumulative reward, expressed as $\sum_t \mathbb{E}_{a_t\sim\pi}\left[r(s_t, a_t)\right]$. In SAC, the objective is extended with a maximum entropy term, referred to as the *maximum entropy*

objective. This objective ensures that the optimal policy not only maximizes the expected cumulative reward but also aims to maximize its entropy at each visited state:

$$\pi_* = \arg\max_{\pi} \sum_t \mathbb{E}_{(s_t,a_t)\sim\rho_\pi} \left[r(s_t,a_t) + \alpha \mathcal{H}\left(\pi\left(\cdot|s_t\right)\right)\right] \tag{2.27}$$

The relative importance of reward and entropy is determined by the temperature parameter α, which controls the stochasticity of the optimal policy. Higher values of α encourage greater stochasticity, which leads the agent to explore the action space more extensively.

To maximize the objective, the *soft policy iteration* can be employed. This method alternates between policy evaluation and policy improvement within the maximum entropy framework [187]. In the policy evaluation step, the value of the policy π is computed using *soft state-value function*, expressed as:

$$V(s_t) = \mathbb{E}_{a_t\sim\pi}\left[Q(s_t,a_t) - \alpha \log \pi(a_t|s_t)\right] \tag{2.28}$$

In addition to the soft state-value function, the *soft action-value function Q(s,a)* can be computed starting from a randomly initialized function $Q : S \times A \rightarrow \mathbb{R}$, and repeatedly applying a modified Bellman backup operator $\mathcal{T}^\pi$ given by:

$$\mathcal{T}^\pi Q(s_t,a_t) \triangleq r(s_t,a_t) + \gamma \mathbb{E}_{s_{t+1}\sim p}\left[V(s_{t+1})\right] \tag{2.29}$$

where $p : S \times A \rightarrow S$ represents the distribution over the next state s_{t+1} given the current state s_t and action a_t. The parameters of the soft Q-function can be trained by minimizing the soft Bellman residual [32]:

$$\begin{aligned} J_Q(\theta) &= \mathbb{E}_{(s_t,a_t)\sim D}\left[\frac{1}{2}\left(Q_\theta(s_t,a_t) - \mathcal{T}^\pi Q_{\bar{\theta}}(s_t,a_t)\right)^2\right] \\ &= \mathbb{E}_{(s_t,a_t)\sim D}\left[\frac{1}{2}\left(Q_\theta(s_t,a_t) - \left(r(s_t,a_t) + \gamma \mathbb{E}_{s_{t+1}\sim p}\left[V_{\bar{\theta}}(s_{t+1})\right]\right)\right)^2\right] \end{aligned} \tag{2.30}$$

where D is the replay buffer of the collected experiences and $V_{\bar{\theta}}(s_{t+1})$ is estimated using a target network for the Q-function, as defined in Equation (2.28).

The policy improvement step updates the policy toward a Boltzmann distribution over actions, defined by the current soft Q-function. A Boltzmann distribution assigns higher probabilities to actions with higher Q-values while maintaining stochasticity by assigning non-zero probabilities to all actions. The expression captures this behavior:

$$\pi(a|s) \propto \exp\left(\frac{1}{\alpha}Q(s,a)\right)$$
$$\pi(a|s) = \frac{\exp\left(\frac{1}{\alpha}Q(s,a)\right)}{\sum_{a'}\exp\left(\frac{1}{\alpha}Q(s,a')\right)}$$

where the denominator, also denoted as $Z(s)$, normalizes the distribution over actions at state s. However, since this Boltzmann distribution may not lie within the desired family of parameterized policies Π (e.g., Gaussian distributions), the updated policy is obtained by projecting the Boltzmann distribution onto π using an information projection defined by the Kullback-Leibler (KL) divergence [32]. The KL divergence quantifies the discrepancy between two probability distributions [188]. The policy improvement step is defined as:

$$\pi_{new} = \arg\min_{\pi\in\Pi} D_{KL}\left(\pi(\cdot|s_t) \,\middle\|\, \frac{\exp\left(\frac{1}{\alpha}Q^{\pi_{old}}(s_t,\cdot)\right)}{Z^{\pi_{old}}(s_t)}\right) \tag{2.31}$$

The policy parameters can be learned by directly minimizing the expected KL-divergence mentioned in Equation (2.31). The temperature parameter α scales the entropy of the updated policy in the objective function, which leads to the following formulation:

$$J_\pi(\phi) = \mathbb{E}_{s_t\sim D}\left[\mathbb{E}_{a_t\sim\pi_\phi}\left[\alpha\log\left(\pi_\phi(a_t|s_t)\right) - Q_\theta(s_t,a_t)\right]\right] \tag{2.32}$$

This objective involves an expectation over the policy's output distribution, where actions a_t are sampled from $\pi(a_t|s_t)$. As a result, errors cannot be directly backpropagated using standard backpropagation techniques due to the stochasticity of the sampling process. For this reason, reparameterization techniques are used to train the policy. The policy is reparameterized using a neural network transformation:

$$a_t = f_\phi(\epsilon_t; s_t) \tag{2.33}$$

where $\epsilon_t \sim \mathcal{N}(0, I)$ is an input noise vector sampled from a fixed distribution (e.g., a spherical Gaussian). Equation (2.32) can be rewritten as:

$$J_\pi(\phi) = \mathbb{E}_{s_t\sim D,\epsilon_t\sim\mathcal{N}}\left[\alpha\log\left(\pi_\phi(f_\phi(\epsilon_t;s_t)|s_t)\right) - Q_\theta(s_t, f_\phi(\epsilon_t;s_t))\right] \tag{2.34}$$

The parameters of the Q-function and the policy can be updated iteratively by

applying Equations (2.30) and (2.34). Since the simulated environment enforces action bounds externally, after the agent outputs actions, I do not apply the squashing function typically used in the Gaussian policy of SAC.

In addition, the temperature parameter α can be learned rather than being fixed at a constant value. The objective for learning α is defined as follows [186]:

$$J(\alpha) = \mathbb{E}_{a_t \sim \pi_t} \left[-\alpha \log \pi_t(a_t|s_t) - \alpha \bar{\mathcal{H}} \right] \tag{2.35}$$

where $\bar{\mathcal{H}}$ is the constant vector equal to the target entropy.

The methodology of the SAC algorithm is summarized in the pseudocode provided in Appendix A.1.

2.4.2 Deep Deterministic Policy Gradient (DDPG)

DDPG is an off-policy actor-critic algorithm designed to learn optimal control policies in continuous action spaces. Unlike algorithms that rely on stochastic policies, DDPG employs a deterministic policy to map states to actions directly. This characteristic makes it particularly well-suited for problems involving continuous action spaces. By using deep neural networks to approximate the value function and policy, DDPG can effectively handle high-dimensional observation spaces, which enables it to tackle complex environments [30].

DDPG is an algorithm for learning both a Q-function and a policy through interaction with the environment E. It uses off-policy data and the Bellman equation to train the Q-function, which is then used to optimize the policy. DDPG shares significant similarities with the SAC algorithm. The key difference between the two lies in their approaches to policy learning. SAC uses a soft Q-function and entropy regularization to encourage exploration, whereas DDPG uses a deterministic policy to stabilize the learning process.

The Bellman equation for the action-value function $Q_\pi(s, a)$, as defined in Equation (2.11), is expressed as follows:

$$Q_\pi(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim E} \left[r(s_t, a_t) + \gamma \mathbb{E}_{a_{t+1} \sim \pi} \left[Q_\pi(s_{t+1}, a_{t+1}) \right] \right] \tag{2.36}$$

Since the target policy is deterministic, the policy can be expressed as a function $\mu : S \rightarrow A$, and the expectation over the policy, $\mathbb{E}_{a_{t+1} \sim \pi} [\cdot]$ corresponds to the single

outcome. Equation (2.36) can be rewritten as:

$$Q_\mu(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim E} \left[r(s_t, a_t) + \gamma Q_\mu(s_{t+1}, \mu(s_{t+1})) \right] \tag{2.37}$$

In this form, the expectation depends only on the environment E. This implies that it is possible to learn Q_μ off-policy, using transitions generated from a different stochastic behavior policy β, with these transitions stored in a replay buffer D [30]. The Q-function can then be learned by minimizing the temporal difference (TD) between the estimated Q-value and a target value y_t. The loss function is as follows:

$$L(\theta, D) = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1}) \sim D} \left[\left(Q_\theta(s_t, a_t) - y_t \right)^2 \right] \tag{2.38}$$

The experiences (s_t, a_t, r_t, s_{t+1}) are sampled from the replay buffer D. The target value y_t, used to compute the TD error in Equation (2.38), is defined as:

$$y_t = r(s_t, a_t) + \gamma \, Q_\theta(s_{t+1}, \mu(s_{t+1})) \tag{2.39}$$

Q-learning is a commonly used off-policy algorithm that uses a greedy policy defined as $\mu(s) = \text{argmax}_a \, Q(s, a)$ [30]. However, applying Q-learning to continuous action spaces is not straightforward. This is because, in continuous spaces, finding the greedy policy requires optimizing a_t at every time step. This optimization process can become too slow when using large, unconstrained function approximators and non-trivial action spaces. To address this issue, an actor-critic approach based on the Deterministic Policy Gradient (DPG) algorithm [189] is proposed.

The DPG algorithm uses a deterministic policy function $\mu_\phi(s)$ parameterized by ϕ to map states to specific actions. The critic function $Q(s, a)$ is learned using Equation (2.38), where θ represents the parameters of the critic. The actor is updated iteratively to maximize the expected return from the start distribution J with respect to the actor's parameters:

$$\nabla_\phi J = \mathbb{E}_{s_t \sim D} \left[\nabla_a \, Q_\theta(s_t, \mu(s_t)) \, \nabla_\phi \, \mu_\phi(s_t) \right] \tag{2.40}$$

One of the key challenges in continuous action space learning is the problem of exploration. Off-policy algorithms, such as DDPG, address this challenge by decoupling exploration from the learning process. An exploration policy μ' is developed to address the exploration challenge by adding noise sampled from a noise

process $\mathcal{N}$ to the actor policy:

$$\mu'(\boldsymbol{s}_t) = \mu_\phi(\boldsymbol{s}_t) + \mathcal{N} \tag{2.41}$$

The noise added to the deterministic policy is called "Ornstein-Uhlenbeck" noise. This type of noise is designed to generate temporally correlated perturbations for efficient exploration while maintaining consistency in the policy [190]. Over time, this noise tends to converge toward a mean value when further exploration is no longer necessary.

The methodology of the DDPG algorithm is summarized in the pseudocode provided in Appendix A.2.

2.4.3 Proximal Policy Optimization (PPO)

PPO is an on-policy, policy gradient algorithm that alternates between collecting experience through environmental interactions and optimizing a surrogate objective function using stochastic gradient-based methods [33]. It employs a deep neural network to approximate the policy and the value function. To ensure stable learning, PPO utilizes a "clipped surrogate objective", which restricts large updates to the policy.

Policy gradient methods compute an estimator of the policy gradient and then use it in a stochastic gradient ascent algorithm. The gradient estimator can be expressed as:

$$\hat{g} = \hat{\mathbb{E}}_t \left[\nabla_\phi \log \pi_\phi(a_t|s_t) \hat{A}_t \right] \tag{2.42}$$

where π_ϕ represents a stochastic policy and $\hat{A}_t$ denotes an estimator of the advantage function at time step t. The expectation $\hat{\mathbb{E}}_t$ indicates the empirical average over a finite batch of samples in an algorithm that alternates between sampling and optimization [33]. This iterative process ensures the policy is updated based on the latest environment interactions. The estimator $\hat{g}$ can be obtained by differentiating the objective:

$$L(\phi) = \hat{\mathbb{E}}_t \left[\log \pi_\phi(a_t|s_t) \hat{A}_t \right] \tag{2.43}$$

While performing multiple optimization steps on the loss function $L(\phi)$ may seem advantageous, it can often result in destructively large policy updates and increase the risk of model collapse. To mitigate this, PPO uses a clipped surrogate

objective function, which limits the extent of policy updates during optimization:

$$L^{CLIP}(\phi) = \hat{\mathbb{E}}_t \left[\min \left(\rho_t(\phi) \hat{\mathbb{A}}_t, \text{clip} \left(\rho_t(\phi), 1-\epsilon, 1+\epsilon \right) \hat{\mathbb{A}}_t \right) \right] \tag{2.44}$$

where ϕ represents the vector of policy parameters, and $\rho_t(\phi)$ is the probability ratio between the current policy and the old policy at time t, defined as:

$$\rho_t(\phi) = \frac{\pi_\phi(a_t|s_t)}{\pi_{\phi_{\text{old}}}(a_t|s_t)} \tag{2.45}$$

Also, the advantage estimator $\hat{\mathbb{A}}_t$ is defined as:

$$\hat{\mathbb{A}}(s,a) = r_t + \gamma V(s_{t+1}) - V(s_t) \tag{2.46}$$

This formulation limits drastic changes in the policy during updates. Its effectiveness depends on the hyperparameter ϵ, which controls the magnitude of policy updates to prevent instability. If ϵ is set too small, convergence slows down; if it is too large, the risk of model collapse increases. In the objective function, the second term, $\text{clip} \left(\rho_t(\phi), 1-\epsilon, 1+\epsilon \right) \hat{\mathbb{A}}_t$, modifies the surrogate objective by clipping the probability ratio within the interval $[1-\epsilon, 1+\epsilon]$. The final objective function is then computed as the minimum of the clipped and unclipped objectives to establish a lower bound on the unclipped objective.

The methodology of the PPO algorithm is summarized in the pseudocode provided in Appendix A.3.

2.4.4 High Frequency Oscillations in Deep RL Control

Recent studies demonstrate that deep RL-based controllers may exhibit oscillatory control responses due to the high complexity and sensitivity of deep neural networks. This issue is especially pronounced in applications such as robotics, where the control responses can exhibit significant variations within the acceptable output limits [25]. In deep RL, it has been observed that learned policies can be highly sensitive to small perturbations in inputs (states/observations), often resulting in significant variations in outputs (actions) [26]. These output oscillations can result in high-frequency fluctuations and excessive actuator usage, leading to increased power consumption and reduced energy efficiency, and potentially damaging or hazardous operation [27, 28].

This issue is especially critical in AO systems, where such oscillations can lead

to excessive actuator usage and large control actions, which slow down correction speed due to mirror inertia. To address this issue, prior research has proposed policy smoothness techniques.

2.5 Policy Smoothness in Reinforcement Learning

Policy regularization methods have been widely explored in RL to improve exploration, stability, and generalization during policy learning. While these methods address different aspects of policy optimization, they generally do not enforce smooth control behavior. Classical approaches, such as entropy bonuses, KL penalties, and weight decay exemplify this limitation. Entropy regularization promotes exploration during training [32], but it does not ensure temporally smooth actions. KL penalties, including PPO's clipping mechanism, stabilize learning by constraining policy updates between iterations [33], but they do not regulate the continuity of actions within a trajectory. Weight decay improves generalization by penalizing large parameter magnitudes [191], though its effect is restricted to model complexity rather than temporal smoothness.

An alternative class of methods addresses policy smoothness at inference time, aiming to restrict high-frequency action fluctuations. A simple approach is to apply filters to the neural network controller. However, this can alter the system's dynamic response and disrupt the Markov assumption, leading to inconsistent behavior in neural network controllers [181, 25].

Alternatively, the neural network can be regularized by enforcing Lipschitz continuity [29]. The Lipschitz constant is the smallest value that satisfies the Lipschitz continuity condition for a given function. It measures the sensitivity of the function to variations of its argument. A smaller Lipschitz constant indicates that the function varies more smoothly with respect to changes in its argument [192]. Gouk et al. [29] claimed that neural network models regularized using Lipschitz continuity outperform those employing other common regularizers.

Leveraging on Lipschitz continuity, Spectral Normalization was proposed to constrain the Lipschitz constant of the actor network [193]. This network enhancement method was introduced to stabilize the training of generative adversarial networks [194] in image generation tasks. Takase et al. [195] extended the application of Spectral Normalization to RL by implementing it in each layer of a multilayer perceptron (MLP). This layer-wise approach ensures that the actor network is globally Lipschitz continuous. This method achieves action smoothness

by directly modifying the neural network instead of altering the RL algorithm. However, applying Spectral Normalization to all layers often results in significant performance loss [181]. In contrast, network-wise constraints do not limit the capacities of layers, making them more effective than layer-wise constraints [196]. For this reason, a neural network structure called LipsNet was proposed, which constrains the Lipschitz constant network-wise [197]. This approach ensures that the neural network is both globally and locally Lipschitz continuous, which prevents significant performance loss. An advantage of the network enhancement approach is that it does not complicate the RL algorithm. However, constraining the Lipschitz constant within the network structure may limit its ability to represent complex functions. Such limitations can result in degraded control performance or even failure in learning [198], especially in dynamic environments.

Another method for mitigating action fluctuation is to use DRL with hierarchical network techniques. Hierarchical DRL decomposes a long-horizon RL task into a hierarchy of subtasks. In this framework, a higher-level policy learns to accomplish the overall task by selecting the optimal subtasks, which serve as higher-level actions. Each subtask is treated as an independent RL problem, with a lower-level policy trained to solve it [199]. Chen et al. [200] introduced a nested policy iteration method where one network outputs the action distribution while another outputs a policy inertia scalar. While this approach facilitates the decomposition of complex decision-making processes into subproblems, it also increases the computational and architectural complexity of RL algorithms by introducing additional neural networks and policies [181].

Reward shaping or engineering is another technique for conditioning RL agents to achieve desired behavior [25]. Reward engineering is based on prior knowledge of the tasks to design specific reward functions that penalize non-smooth trajectories [28]. In robotics, a common approach is to include additional cost terms such as $-\alpha|a_t|$ or $-\alpha|a_t - a_{t-1}|$ in the reward function to encourage smoother control actions. While shaping the reward function can significantly influence learning efficiency, it necessitates networks to learn through indirect feedback, which may not always guide the policy toward the desired behavior. Furthermore, reward engineering approaches are often highly task-specific, which limits their generalizability across different tasks.

Another approach to mitigating action fluctuation is the direct optimization of the value function or optimizing the value function and actor policy simultaneously. For value function optimization, a temporal regularization method is

proposed, which modifies the target of the value function by adding the moving average of past value functions [201]. This approach has been shown to improve performance in high-dimensional Atari games. Additionally, the Locally Lipschitz Continuous Constraint (L2C2) is introduced for simultaneous policy and value function optimization [202]. This constraint penalizes the similarity between actions under close states and the similarity of critic values under close states. As a result, both the policy and value functions are regularized to satisfy the L2C2. While optimizing the value function can improve stability, discrepancies between the surrogate objective learned by policy gradient algorithms and the true value function can degrade policy performance. Therefore, focusing on direct policy optimization can be more effective in achieving smoother policies [203].

Mysore et al. [25] proposed the Conditioning for Action Policy Smoothness (CAPS) regularization constraint to improve temporal and spatial smoothness by directly regularizing the policy for high-performance quadrotor drones. Temporal smoothness encourages consecutive actions to be similar, while spatial smoothness, inspired by Shen et al. [204], prevents significant changes in the policy's output when small perturbations are injected into its input. This approach helps mitigate the effects of measurement noise and modeling uncertainties. Additionally, a number of modifications to CAPS have been introduced for different applications. For instance, Cao et al. [205] proposed Image-based Regularization for Action Smoothness (IRAS) for scenarios with high-dimensional inputs. Also, Lee et al. [28] introduced Gradient-based CAPS (Grad-CAPS), which reduces differences in the gradients of actions.

While CAPS and its variants improve policy smoothness and performance compared to Vanilla RL algorithms, they can be less effective in highly dynamic environments. These methods are designed to minimize changes in subsequent actions regardless of the magnitude of changes in time-contiguous observations. However, in highly dynamic environments, where changes in time-contiguous observations are significant, enforcing minimal action changes can overly constrain the policy and hinder exploration.

Overall, various approaches have been proposed to improve policy smoothness, including filtering, reward engineering, network regularization, hierarchical architectures, and regularization of policy and/or value functions. While these methods can improve policy smoothness, they often have drawbacks such as reduced performance, increased architectural complexity, reliance on task-specific tuning, limited generalizability, or insufficiency in highly dynamic environments.

However, CAPS offers simplicity in implementation and effectiveness by directly regularizing the policy without modifying the underlying RL architecture, and demonstrates strong performance in MuJoCo environments and physical drone experiment [25]. Due to these advantages, CAPS is selected as a benchmark for evaluation.

In addition, LipsNet is included as a benchmark for the wavefront sensorless AO environment under dynamic conditions. As discussed in this section, LipsNet constrains the Lipschitz constant at the network level to promote smoother behavior while avoiding the severe performance degradation often observed with layer-wise constraints. This makes it a suitable benchmark for comparison in AO scenarios where a rapidly evolving atmosphere plays a dominant role.

2.6 Summary

This chapter first provided an overview of AO, including the definition and sources of aberrations, wavefront representation, traditional wavefront correction procedures, and the operation of wavefront sensors. The application of wavefront sensor-based AO systems in astronomy was then reviewed.

From the literature, it is observed that wavefront sensor-based AO systems face challenges related to cost and complexity, mainly due to their reliance on wavefront sensors. To address these limitations, wavefront sensorless AO approaches have been proposed. However, the majority of the existing wavefront sensorless AO studies focus on optimizing image sharpness using cameras as a feedback sensor. While cameras have high resolution, their data acquisition and processing times introduce delays that are incompatible with real-time communication requirements, particularly in LEO satellite-to-ground communication. Furthermore, although other model-based and model-free approaches have been applied to optical data links, they often require accurate models or exhibit slow convergence, limiting their robustness in dynamic turbulence. Therefore, a model-free solution for optical communication that can operate with minimal latency and low-pixel-count feedback remains an unmet need in the current literature.

Motivated by these studies and the need for a wavefront sensorless AO system with low-pixel-count detectors, RL-based AO controllers have emerged as a promising solution. RL-based control is particularly well-suited for complex and uncertain environments where designing optimal controllers is difficult. To demonstrate the challenges and potential of RL controllers, this thesis presents

the development of an RL environment for a simulated wavefront sensorless AO system, designed for training and testing various RL algorithms. This environment enables the investigation of various scenarios, including quasi-static, semi-dynamic, and dynamic conditions, while operating with a low-pixel-count photodetector. Leveraging this environment, the thesis provides the first demonstration of RL's potential for wavefront sensorless AO in satellite optical downlink.

However, studies show that deep RL controllers often produce oscillatory control responses due to the high complexity and sensitivity to small variations in deep neural networks. In AO systems, such oscillations can lead to excessive actuator movements and large control actions, which reduce correction speed due to the inertia of the mirror.

To mitigate these issues, various methods have been proposed to improve policy smoothness, including filtering, reward engineering, network enhancements, hierarchical architectures, adversarial disturbance handling, and policy and/or value regularization. While these approaches can improve smoothness, they often suffer drawbacks such as reduced performance, increased complexity, limited generalizability, or insufficiency in highly dynamic environments.

To address these shortcomings, this thesis introduces State-Adaptive Proportional Policy Smoothing (SAPPS), a novel regularization framework that promotes proportional rather than saturated policy responses. While existing policy smoothness methods apply fixed penalties or impose local or global constraints on action changes, SAPPS regulates the policy to produce action changes in proportion to the magnitude of state changes. In doing so, SAPPS encourages proportional action adjustments while suppressing fluctuations around these proportional responses. This adaptive mechanism enables the policy to remain smooth while maintaining responsiveness and performance across static and dynamic conditions.

Chapter 3

Policy Regularization for Smooth Control in Dynamic Environments

Designing control systems is a fundamental challenge in robotics. These systems are deployed across various domains, including industrial robotics [206, 207], autonomous vehicles [208, 209], medical robotics [210], and AO systems [117, 38]. RL is increasingly explored as a promising approach for control, especially in scenarios where developing accurate system models or designing optimal controllers is challenging [181].

A key advantage of RL-based control is its capability to address complex dynamical systems, especially in highly uncertain environments where designing optimal controllers is challenging [181]. Additionally, RL-based controllers are model-free, meaning they can learn control policies without requiring an explicit system's model. This characteristic makes them particularly useful for applications where a system's dynamics are difficult to model or may suffer from parameter uncertainty and sensitivity to variations in the environment and input. Deep RL has become increasingly popular due to its effectiveness in addressing high-dimensional, real-world applications with continuous action and/or state spaces [211, 33].

This chapter focuses on deep RL for continuous control tasks. The focus is primarily on dynamic environments, where the evolution of the environment occurs at a rate comparable to or faster than the agent's sampling rate. Under such conditions, rapid state variations can occur, making it more difficult for the policy to learn effective behavior from experience. In contrast, static environments are characterized by environmental evolution that occurs slower than the agent's sampling rate, such that the environment remains approximately constant between consec-

utive actions. In this work, both static and dynamic environments are analyzed.

Recent studies demonstrate that a significant challenge in deep RL for continuous control is the presence of high-frequency oscillations in learned policies. These oscillations can lead to excessive actuator usage, increased power consumption, reduced energy efficiency, and instability in real-world systems [27, 28]. This issue arises from the complexity of deep neural networks and their sensitivity to small perturbations in input observations, which can result in significant variations in output actions [26, 25]. While such issues are challenging in static environments, where state changes are agent-driven, they become even more challenging in dynamic environments, where states evolve independently of the agent's actions. In such settings, the agent must adapt rapidly to maintain performance, which often leads to more aggressive actions that compromise smooth control.

While existing action regularization approaches mitigate these oscillations, they often introduce policy performance degradation, increased architectural complexity, reliance on task-specific tuning, and limited generalizability, as discussed in Section 2.5. Traditional RL regularization and smoothing methods restrict the policy from quickly adjusting actions to account for large state changes in dynamics environments, as further demonstrated in the experiments in Section 6.2.2.

To address this, a novel *State-Adaptive Proportional Policy Smoothing (SAPPS)* method is proposed for RL. SAPPS improves policy smoothness while maintaining high performance across static and dynamic environments. It adaptively adjusts smoothness constraints to suppress high-frequency components in control signals without compromising performance in continuous control tasks. This method leverages Lipschitz continuity to optimize the policy based on differences between time-contiguous observations without directly minimizing actions. With this approach:

- when the change between time-contiguous observations is small, the method penalizes significant changes in subsequent actions.
- when the change between time-contiguous observations is significant, it penalizes small subsequent action changes.

Thus, SAPPS ensures policy smoothness by regulating control responses in proportion to the magnitude of state changes, enabling the policy to remain responsive to environmental changes without compromising performance.

For example, in AO systems, two primary concerns arise: achieving accurate wavefront correction and maintaining that correction over time. Suppose an RL

controller can achieve near-optimal correction regardless of the smoothness of its policy. The challenge is to maintain this correction while accounting for the drift velocity of the atmosphere. When the drift velocity is low, the environment changes slowly. In such cases, the policy should respond gradually to avoid excessive actuator usage (overaction), which slows down correction speed due to mirror inertia. In contrast, when the drift velocity is high, a slow-responding or overly smooth policy becomes ineffective (underaction). In this scenario, the system requires quicker and more aggressive actuator adjustments to maintain the quality of the wavefront correction. Experimental results demonstrating the RL controller's performance under varying turbulence velocities are presented in Section 6.2.2.

The core idea of SAPPS is general and can be integrated with a variety of deep RL algorithms. In this thesis, the Proximal Policy Optimization (PPO) algorithm is selected from the wide range of deep RL methods because it provides a stable training framework through its clipping mechanism, which limits the magnitude of parameter updates between policy iterations [33]. This clipping improves optimization stability during training but does not regulate the smoothness of actions over time within a trajectory. For this reason, while PPO ensures stable learning, it does not prevent high-frequency oscillations in the control signals generated by the policy. A detailed description of PPO is provided in Section 2.4.3. Soft Actor-Critic (SAC) [32] often outperforms PPO in certain MuJoCo environments. Its optimization objective includes a maximum entropy term that encourages diverse behaviors to improve exploration and generalization. However, this increased stochasticity can also introduce higher action oscillations, potentially resulting in undesirable fluctuations in the learned policy [200].

For this reason, PPO is adopted as the baseline algorithm because of its relative simplicity and training stability, which provide a more reliable experimental foundation. Moreover, PPO is widely regarded as a strong baseline for sim-to-real robotic control applications (e.g., [34, 35, 36, 37]), making it well-suited for evaluating SAPPS against existing methods across multiple domains. Additionally, preliminary experiments on the wavefront sensorless AO system using Vanilla PPO are reported in Appendix D. PPO has demonstrated high effectiveness while being easy to implement [212], making it a suitable choice for this study.

Therefore, the proposed SAPPS method is integrated with the PPO algorithm, hereafter referred to as PPO + SAPPS. The policy smoothness and performance of PPO + SAPPS are evaluated against two baselines: vanilla PPO and PPO combined with the Conditioning for Action Policy Smoothness (CAPS) method [25]

(PPO + CAPS), a well-established approach for improving policy smoothness. Additionally, for the dynamic scenarios of the wavefront sensorless AO system, PPO + SAPPS is also compared with the LipsNet neural network [197], which is implemented using PPO.

The major evaluation focuses on continuous control of a simulated deformable mirror in a wavefront sensorless AO system for optical satellite communication. The task is to concentrate a distorted light beam from a satellite laser transmitter into a ground-based single-mode fiber while minimizing action magnitudes to reduce inertial penalties associated with significant mirror adjustments, see Fig. 5.1. This AO problem is highly dynamic due to the characteristic timescale of atmospheric turbulence $\sim$1 ms, which causes stochastic variations in the captured wavefront even in the absence of any action. These variations occur on a timescale comparable to or faster than the sampling frequency of the control policy.

Beyond the AO system, the methods are also evaluated on several MuJoCo continuous control benchmarks from the OpenAI Gym suite [213], including Walker, Reacher, HalfCheetah, Swimmer, and Ant. These environments differ in both state and action space dimensions, allowing assessment of the methods across a diverse range of continuous control tasks. Unlike the AO scenario, these environments are static, where observations remain unchanged unless actions are applied. Despite their static nature, evaluating SAPPS in these benchmarks is important to verify that the proposed method generalizes beyond dynamic environments. Moreover, smooth policy behavior remains desirable even in static environments, as excessive action oscillations can degrade control performance, increase energy consumption, and cause actuator wear. These evaluations aim to examine its effectiveness as a general-purpose RL method rather than one tailored specifically to AO. Additionally, to evaluate the real-world applicability, the methods are deployed on a Bitcraze Crazyflie 2.1 nano quadcopter [214]. In this setting, the control objective is to maintain stable hovering at a fixed altitude, thereby demonstrating that SAPPS remains effective not only in simulation but also in physical systems.

The experimental results presented in Chapters 6 and 4 indicate that SAPPS outperforms Vanilla PPO in both average reward and policy smoothness. The results are organized in order of increasing level of system complexity and the extent of real-world implementation:

- *MuJoCo environments*: SAPPS achieved a 11% higher average return and a 28% improvement in policy smoothness compared to Vanilla PPO. Moreover, SAPPS performed comparably with CAPS.

- *Wavefront sensorless AO-RL environment*: In the preliminary experiments, SAPPS outperformed Vanilla PPO, achieving a 4% increase in fiber coupling efficiency and a 10% improvement in policy smoothness. While CAPS demonstrated higher fiber coupling efficiency in low-velocity conditions and improved policy smoothness across all drift velocities, SAPPS outperformed CAPS in high-velocity conditions, achieving approximately 9% improved performance. Also, compared to the LipsNet neural network, SAPPS demonstrated a faster initial response, higher fiber coupling efficiency in high-velocity conditions, and improved policy smoothness across all drift velocities.

- *Crazyflie 2.1 nano quadcopter*: In the simulated environment, SAPPS demonstrated performance comparable to Vanilla PPO while achieving an average 29% improvement in policy smoothness. In the real-world environment, SAPPS outperformed Vanilla PPO with a 17% higher average return and a 29% improvement in policy smoothness.

In the remainder of this chapter, Section 3.1.1 outlines the necessary preliminaries, followed by a detailed discussion of the proposed approach in Section 3.1.2. Section 3.2 provides a summary of the chapter.

3.1 State-Adaptive Proportional Policy Smoothing Method

3.1.1 Preliminaries

Lipschitz Continuity and CAPS

Consider two metric spaces, (X, d_X) and (Y, d_Y), where d_X and d_Y denote the metrics on the sets X and Y, respectively. A function $f : X \to Y$ is called Lipschitz continuous if there exists a constant $K \geq 0$ such that, $\forall x_1, x_2 \in X$, the following inequality holds:

$$d_Y\left(f(x_1), f(x_2)\right) \leq K d_X(x_1, x_2) \tag{3.1}$$

where K is the Lipschitz constant. A smaller K indicates that f varies more smoothly with respect to changes in its argument, whereas a larger K means f is more sensitive to variations of its argument [192].

Motivated by the concept of Lipschitz continuity, the metric spaces (S, d_S) and (A, d_A) in RL are considered, where d_S and d_A are metrics induced by norms on the state space S and action space A, respectively. The policy function $\pi : S \to A$ maps

states to actions. The policy π is Lipschitz continuous if there exists a constant $K \geq 0$ such that:

$$d_A(\pi_\theta(s_t), \pi_\theta(s_{t+1})) \leq K d_S(s_t, s_{t+1}) \tag{3.2}$$

Mysore et al. [25] proposed a method called "Conditioning for Action Policy Smoothness" (CAPS) to improve smoothness in policies. To formulate their approach in terms of Lipschitz continuity, they considered the 2-norm $d_A(\pi_\theta(s_1), \pi_\theta(s_2))$ and assumed a fixed state change $d_S(s_1, s_2) = 1$. By setting the state change to 1, they assumed that state changes are uniformly spaced within the metric space. While the authors do not explicitly justify this choice, it is interpreted as a practical simplification to avoid the challenges associated with computing distances in complex or high-dimensional state spaces. In Lipschitz continuity, the ratio between action change and state change can have high variance, especially when state changes vary significantly across samples, which can make the regularization term unreliable during training. By treating all state changes as uniform, they eliminate the need for environment-specific metrics. As a result, the Lipschitz continuity condition (Eq. 3.2) is simplified to $\| \pi_\theta(s_1) - \pi_\theta(s_2) \|_2 \leq K$.

The CAPS objective function, $J^{CAPS}_{\pi_\theta}$, extends the Vanilla PPO objective function, J_{π_θ} in (2.44), by including additional regularization terms. It includes a temporal smoothness regularization term, L_T, and a spatial smoothness term, L_S, each weighted by their respective regularization coefficients, λ_T and λ_S:

$$J^{CAPS}_{\pi_\theta} = J_{\pi_\theta} - \lambda_T L_T - \lambda_S L_S \tag{3.3}$$

where L_T and L_S are defined as follows:

$$\begin{aligned} L_T &= \| \pi_\theta(s_t) - \pi_\theta(s_{t+1}) \|_2 \\ L_S &= \| \pi_\theta(s_t) - \pi_\theta(\bar{s}_t) \|_2 \end{aligned} \tag{3.4}$$

Here, the term L_T penalizes policies when the actions taken in the next state differ from those in the current state. The term L_S encourages taking similar actions in similar states. It promotes spatial smoothness by encouraging small perturbations injected into input states to have minimal effect on the policy's output [204]. This regularization mitigates the effects of measurement noise and modeling uncertainties, which improves policy robustness. To model state perturbations, the noisy state $\bar{s}$ is sampled from a distribution ϕ around the original state: $\bar{s}_t \sim \phi(s_t)$. The resulting L_S term is defined as the difference between the actions produced by

the policy π_θ when applied to the original state and its noisy version.

Although CAPS significantly improves policy smoothness, it may lead to reduced performance in highly dynamic environments due to its underlying assumption that state changes are uniformly spaced. Specifically, the temporal smoothness regularization term is designed to minimize changes in subsequent actions. However, in dynamic environments, the magnitude of change in subsequent actions should be proportional to the degree of change in time-contiguous observations, as it depends on the complexity and drift velocity of the environment. As a result, this assumption can limit the effectiveness of CAPS in such environments.

LipsNet Neural Network

As a benchmark for comparison with the proposed method in dynamic scenarios of a wavefront sensorless AO system, the LipsNet neural network is considered, which improves policy smoothness in RL by constraining the Lipschitz constant of the network. The core idea behind LipsNet is the Multi-dimensional Gradient Normalization (MGN) technique, which enforces a global Lipschitz condition at the network level [197]. Given a neural network f, the MGN formulation defines a normalized network:

$$f_{MGN}(x) = K\frac{f(x)}{\|\nabla_x f(x)\| + \epsilon} \tag{3.5}$$

Here, K is a positive scalar representing the desired Lipschitz constant, $\nabla_x f(x)$ denotes the Jacobian matrix of f with respect to x, and ϵ is a small positive constant added for numerical stability. The Jacobian norm is used to ensure that the entire network satisfies the Lipschitz constraint.

LipsNet achieves Lipschitz continuity explicitly through the normalization in (3.5), which bounds the Jacobian norm of the network output. This ensures Lipschitz continuity, since a differentiable function f whose Jacobian norm is bounded by a constant K for all inputs is Lipschitz continuous [215].

To automatically adjust the Lipschitz constant, LipsNet introduces two variants: *LipsNet-G* and *LipsNet-L*.

LipsNet-G Uses a global Lipschitz constant K, a learnable scalar parameter, optimized with the network weights. The loss function includes a regularization term

to adjust the value of K:

$$\begin{aligned} L^{\text{LipsNet-G}}(\theta) &= L(\theta) + \lambda K^2 \\ K &\leftarrow K - \eta_k \nabla_K L^{\text{LipsNet-G}} \end{aligned} \tag{3.6}$$

where, λ is a regularization term, and η_K is the learning rate for K.

However, enforcing a global Lipschitz constant can limit model expressiveness, as the optimal Lipschitz constant may vary across different neighborhoods of the x value. To address this, *LipsNet-L* introduces a separate neural network $K(x)$ to learn a local Lipschitz constant to ensure local Lipschitz continuity, as shown in Figure 3.1. The objective function is augmented with a regularization term based on $K(x)$, and the Lipschitz adjustment for is given by:

$$\begin{aligned} L^{\text{LipsNet-L}}(\theta, x) &= L(\theta) + \lambda K(x)^2 \\ \phi &\leftarrow \phi - \eta_k \nabla_\phi L^{\text{LipsNet-L}} \end{aligned} \tag{3.7}$$

Here, ϕ is the vector of $K(x)$ parameters.

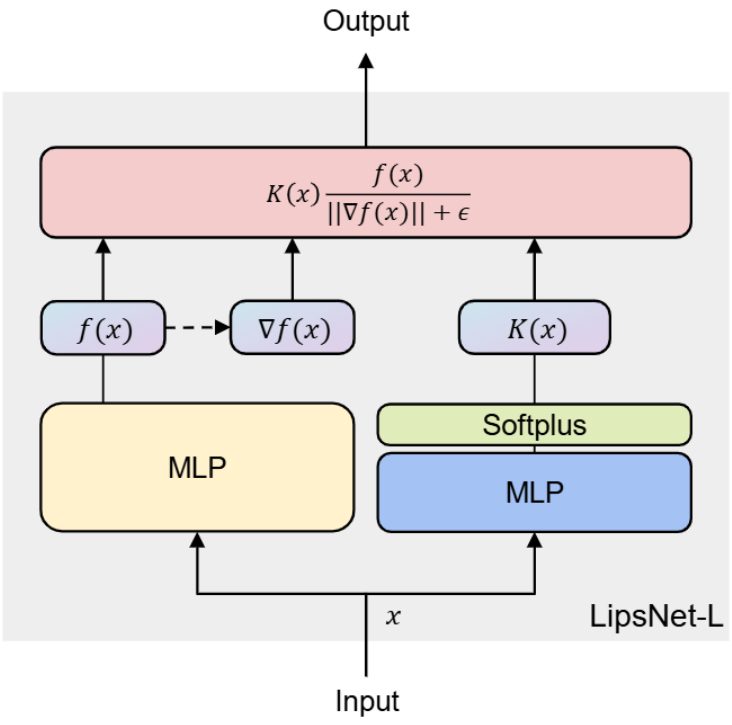

Figure 3.1: Structure of LipsNet-L [197]

Although LipsNet effectively improves policy smoothness and demonstrates strong performance across various control tasks, it introduces additional complex-

ity by incorporating an auxiliary neural network to estimate a local Lipschitz constant (see Eq. 3.7). Moreover, its output is structurally modified by normalizing the original output with the Jacobian norm and scaling it by the Lipschitz constant (see Eq. 3.5). While this architecture promotes smoothness without altering the underlying RL algorithm, it may modify the policy's representation and complicate its integration with existing models.

Smoothness Measurement

Policy smoothness can be evaluated using various approaches. One approach is action fluctuation, which measures the norm of the difference between subsequent actions. Another approach involves computing the Lipschitz constant, where a lower value indicates a smoother policy. Additionally, smoothness can be evaluated using the smoothness measure Sm introduced by Mysore et al. [25]. enabling the detection of high-frequency fluctuations that may be less apparent in the time domain.

The smoothness metric Sm is derived from the Fast Fourier Transform (FFT) frequency spectrum and is defined as follows [25]:

$$Sm = \frac{2}{nf_s} \sum_{i=1}^{n} M_i f_i \tag{3.8}$$

where M_i represents the amplitude of the i^{th} frequency component, f_i for $i \in [1, n]$, and f_s represents the sampling frequency. This metric provides the mean weighted normalized frequency by considering the frequencies and amplitudes of the control signal components. A higher Sm value indicates the presence of larger high-frequency components, which correlates with increased actuation consumption. In contrast, a lower Sm value indicates a smoother policy response with reduced high-frequency fluctuations.

While Sm is a meaningful indicator of policy smoothness within a given environment, its values are not directly comparable across different environments or even across MuJoCo tasks. This limitation arises because the sampling frequency (or timestep) can differ across domains, which changes the frequency resolution of the FFT. In addition, the dimensionality of the action space varies—for example, two actuators in Swimmer versus six in HalfCheetah—which changes the distribution of amplitudes across frequencies. Finally, the physical interpretation of smoothness is task-dependent. A smooth sequence of joint torques in HalfCheetah

corresponds to different dynamics than in Walker. For these reasons, all smoothness comparisons with *Sm* in this thesis are reported within the same environment.

3.1.2 Proposed Method: State-Adaptive Proportional Policy Smoothing

The State-Adaptive Proportional Policy Smoothing (SAPPS) method is proposed for RL to improve policy smoothness in both static and dynamic environments. SAPPS optimizes actuator consumption by minimizing unnecessary action fluctuations while improving performance in real-world continuous control environments. By reducing abrupt policy shifts, SAPPS has the potential to lower latency and improve exploration.

To achieve smoothness in policies, a regularization term applicable to both static and dynamic continuous control environments is proposed. This term operates as follows:

- **Small State Change Scenario**: When perturbations in observations occur or the drift velocity of the dynamic environment is low, the difference between time-contiguous observations remains small. As a result, the difference between the corresponding subsequent actions is penalized to be proportionally small (see Figure 3.2, top).
- **Large State Change Scenario**: When the drift velocity of the environment is high, the difference between time-contiguous observations is large. As a result, the difference between the corresponding subsequent actions is penalized to be proportionally large (see Figure 3.2, bottom).

Thus, the regularization term enforces policy smoothness by ensuring that changes in successive actions are proportionally scaled to changes in their corresponding time-contiguous observations, allowing the policy to remain responsive to environmental changes.

The objective function

$$J_{\pi_\theta}^{SAPPS} = J_{\pi_\theta} - \lambda_T L_T \tag{3.9}$$

is proposed which integrates the Vanilla PPO objective function J_{π_θ} defined in Eq. 2.44, with the smoothness regularization term

$$L_T = \left| \log \left(c_{hm} \frac{\|\mu_\theta(s_{t+1}) - \mu_\theta(s_t)\|_2 + \epsilon_1}{\|s_{t+1} - s_t\|_2 + \epsilon_2} \right) \right| \tag{3.10}$$

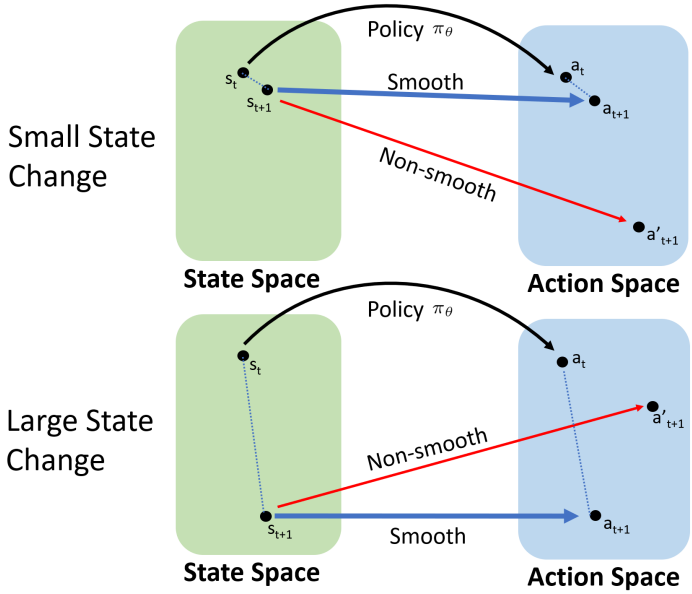

Figure 3.2: Illustration of policy smoothness for small and large state changes from s_t to s_{t+1} under policy π_θ. A regularization term ensures that changes in subsequent actions are proportionally scaled to the corresponding time-contiguous state changes.

and its regularization coefficient λ_T. In (3.10), c_{hm} is a homogeneous ratio that accounts for differences in state and action space scales. ϵ_1 and ϵ_2 are positive noise terms added to prevent division by zero and facilitate policy learning through perturbation. The coefficients of these noise parameters are identical. In the smoothness regularization term, the deterministic policy μ_θ is used, which represents the mean action of the stochastic policy π_θ. This substitution reduces sampling noise and variance during training by removing the stochasticity in the action selection process. The log function is applied to mitigate fluctuations in the loss values and ensure that as the difference between the deterministic action and the state approaches zero, the loss converges to zero. The absolute operator $|\cdot|$ is used to treat positive and negative deviations equally. This is particularly useful when the goal is to measure the magnitude of action changes relative to state changes without considering the direction of the deviation. As a result, the regularization term becomes symmetric, which helps avoid bias toward increasing or decreasing action changes and promotes smoothness in both directions.

The formulation of the smoothness regularization term L_T is motivated by the Lipschitz continuity condition, as described in Eq. (3.2). Under this condition, the ratio between changes in actions and the corresponding changes in states is considered as an empirical estimate of the policy's smoothness. A smaller ratio indicates

a smaller Lipschitz constant, which promotes smoother policy behavior. However, directly constraining this ratio using a Lipschitz constant can hinder the policy's ability to explore and lead to conservative behavior. To preserve exploration while encouraging smoothness, this ratio can be used as a regularization term in the loss function. Minimizing this term encourages the policy to respond smoothly to variations in observed states. However, the raw ratio can have high variance, particularly when the difference between state and action changes is significant, making it unstable and unreliable for training. To address this, a modified formulation is introduced that reduces variance and stabilizes the ratio, which enables more robust learning.

The pseudocode for this method is as follows:

Algorithm 1 PPO with SAPPS method

for iteration = 1, 2, … **do**
  for actor = 1, 2, …, N **do**
    Run policy $\pi_{\theta_{\text{old}}}$ in environment for T timesteps
    Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$
  end for
  Compute regularization term L_T
  Compute standard PPO objective J_{π_θ}
  Compute SAPPS objective $J^{SAPPS}_{\pi_\theta} = J_{\pi_\theta} - \lambda_T L_T$
  Optimize $J^{SAPPS}_{\pi_\theta}$ w.r.t. θ
  $\theta_{\text{old}} \leftarrow \theta$
end for

3.2 Summary

In this chapter, the State-Adaptive Proportional Policy Smoothing (SAPPS) method is introduced. SAPPS is designed to reduce high-frequency components in the control signal of RL controllers by applying policy smoothing in proportion to the magnitude of time-contiguous observation changes. This proportional smoothing ensures that action changes scale consistently with state changes: (i) when observation changes are large, corresponding action changes are proportionally large, and (ii) when observation changes are small, corresponding action changes are proportionally small. In this way, SAPPS achieves smoother policies while maintaining responsiveness to environmental changes and avoiding the performance degradation that arises from overly rigid smoothing.

SAPPS is evaluated in a wavefront sensorless AO system for optical satellite communication to examine its capability and responsiveness in a dynamic environment. Beyond AO, SAPPS is further evaluated on multiple MuJoCo continuous-control tasks, which serve as standardized RL benchmarks and enable assessment of its general-purpose applicability across diverse continuous-control domains. Finally, to assess its real-world applicability, SAPPS is evaluated in a quadcopter experiment, providing evidence of its effectiveness beyond simulation in a physical system where smooth and responsive control is critical.

The experimental results, presented in Chapters 4 and 6, show that SAPPS consistently outperforms Vanilla PPO in both average reward and policy smoothness. SAPPS achieved significant improvements in policy smoothness and demonstrated competitive or superior performance compared to CAPS and LipsNet neural network, with clear advantages under high drift velocity AO conditions.

Chapter 4

Generalization of SAPPS to Robotics

This chapter investigates multiple domains to assess the generality and real-world applicability of the proposed approach. Specifically, the SAPPS-regularized PPO framework is applied to standard MuJoCo continuous-control tasks and a real-world quadcopter experiment. This extension from an optical communication system to robotic control tasks enables a comprehensive evaluation of the method's adaptability, stability, and smoothness across environments with distinct physical and dynamic characteristics. Through this multi-domain evaluation, the method's effectiveness is demonstrated across a broad range of RL control applications.

4.1 MuJoCo Environments

The general applicability of the proposed SAPPS method was evaluated on five MuJoCo-simulated robotics tasks from the OpenAI Gym environments: Walker, Reacher, Half Cheetah, Swimmer, and Ant. These environments differ in their state and action space dimensions, enabling evaluation across a diverse range of continuous control problems.

Before presenting the experimental results for the MuJoCo and quadcopter environments, it is important to clarify that policy smoothness is evaluated using the Smoothness measure (Sm), as described in Section 3.1.1. A lower value of Sm indicates a smoother policy. However, since the term "smoothness" may be misleading, this metric is referred to as "Action Fluctuation (Sm)" in the results.

4.1.1 Experimental Setup

Five widely used MuJoCo environments—Walker2D, Reacher, Half Cheetah, Swimmer, and Ant—are selected, as shown in Figure 4.1 and detailed in Table 4.1. These environments vary in state and action space dimensions, enabling the evaluation of methods on different continuous control tasks.

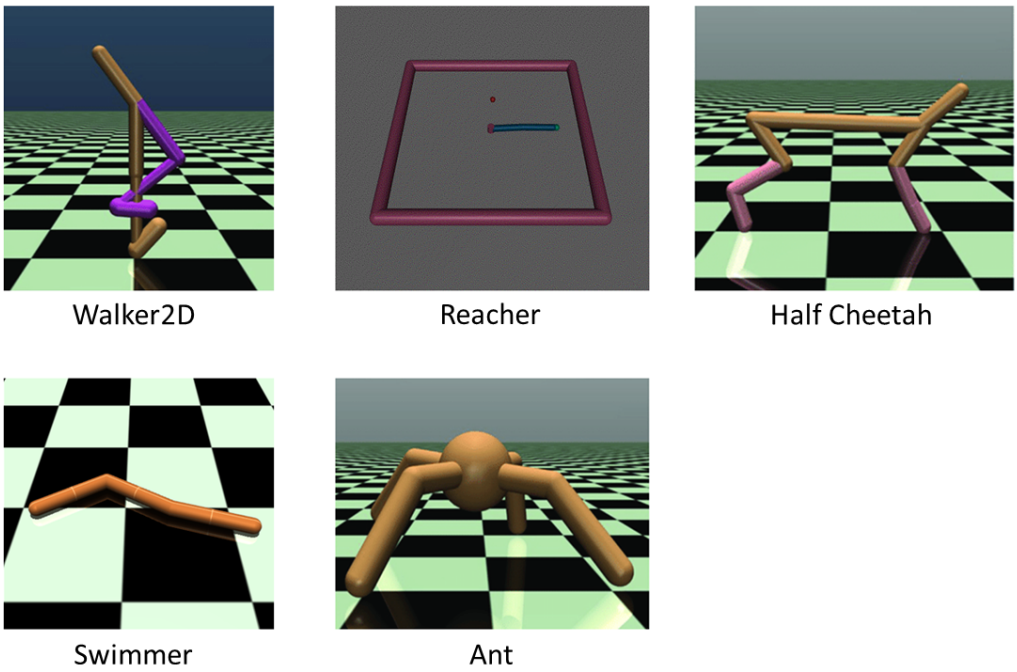

Figure 4.1: MuJoCo simulated robotics tasks from the OpenAI Gym environments: Walker, Reacher, Half Cheetah, Swimmer, and Ant.

Environment	Observation dim.	Action dim.	Description
Walker2d-v4	17	6	2D bipedal locomotion
Reacher-v4	10	2	Two-link planar reaching task
HalfCheetah-v4	17	6	2D planar cheetah locomotion
Swimmer-v4	8	2	Three-link swimming in a viscous medium
Ant-v4	105	8	3D quadruped locomotion

Table 4.1: Summary of the MuJoCo continuous-control tasks selected

The key distinction between the stated MuJoCo environments and the wavefront sensorless AO-RL environment lies in the dynamic nature of the latter. In the AO environment, observations evolve continuously over time, even in the absence of actions. This dynamic behavior is primarily driven by atmospheric drift, which plays a critical role. As the atmospheric speed increases, the temporal cor-

relation between time-contiguous observations decreases, which results in greater variability in the environment.

4.1.2 Results and Discussion

The tuned hyperparameters and performance plots of the examined MuJoCo environments are shown in Appendix B, with the corresponding results summarized in Table 4.2. Note that in this table, a lower smoothness measure (Sm) indicates reduced action fluctuations and smoother policy, as defined by the Fast Fourier Transform-based metric in Section 3.1.1.

Table 4.2: Comparison of average rewards and smoothness measures (Sm) for methods across MuJoCo environments

Env. / Method	Walker2D (v4)		Reacher (v4)		Half Cheetah (v4)		Swimmer (v4)		Ant (v4)	
	Reward	$Sm \times 10^3$	Reward	$Sm \times 10^3$	Reward	$Sm \times 10^3$	Reward	$Sm \times 10^3$	Reward	$Sm \times 10^3$
Vanilla PPO	4002.04 ± 537.74	9.33 ± 2.44	-3.92 ± 0.47	4.47 ± 1.73	4134.74 ± 1738.91	13.14 ± 5.66	122.72 ± 22.32	6.07 ± 1.56	5520.00 ± 408.01	7.50 ± 2.19
PPO+CAPS	4109.52 ± 1262.70	2.67 ± 0.47	-4.06 ± 0.51	3.75 ± 1.61	6184.86 ± 568.03	6.35 ± 0.15	121.98 ± 25.74	3.32 ± 0.69	5551.74 ± 425.42	5.40 ± 1.49
PPO+SAPPS	4186.74 ± 801.38	3.71 ± 0.63	-3.94 ± 0.53	4.20 ± 1.85	6296.60 ± 439.64	9.49 ± 0.17	116.26 ± 30.36	4.87 ± 1.39	5735.32 ± 408.16	5.48 ± 1.85

As shown in Table 4.2, SAPPS outperforms Vanilla PPO in terms of average reward, achieving improvements of 5% in Walker2D, 52% in Half Cheetah, and 4% in Ant. Furthermore, SAPPS significantly improves policy smoothness, with an improvement of 60% in Walker2D, 28% in Half Cheetah, and 27% in Ant. Even in environments where SAPPS does not outperform Vanilla PPO in average reward, such as Reacher and Swimmer, it remains competitive while improving policy smoothness by 6% in Reacher and 20% in Swimmer.

SAPPS is also compared with CAPS, both of which outperformed the Vanilla PPO approach. SAPPS achieved a higher average reward than CAPS, whereas CAPS generated smoother policies. On average, across the five MuJoCo environments, SAPPS increased the average reward by 11% while improving policy smoothness by 28%. In contrast, CAPS achieved a 10% increase in average reward and a 42% improvement in policy smoothness. These results indicate that both SAPPS and CAPS perform comparably across the five MuJoCo environments.

SAPPS generally achieves higher rewards than CAPS and smoother policies than Vanilla PPO, which makes it a robust and adaptable method for continuous control tasks where both overall performance and policy smoothness are essential. The smooth control achieved by SAPPS is valuable not only for improving performance but also for reducing energy consumption, preventing dangerous actuation, and minimizing wear and tear on actuators.

4.2 Quadcopter Environment

To evaluate the hardware applicability of the proposed SAPPS method, the SAPPS-regularized PPO algorithm was implemented on a nano-quadcopter platform for a hovering task, and its performance was compared with that of the Vanilla PPO and PPO+CAPS methods.

4.2.1 Experimental Setup

The Crazyflie 2.1 is a versatile, open-source flying development platform with a mass of 27 grams and a motor-to-motor distance of 9 cm, as shown in Figure 4.2. It features a low-latency, long-range radio system and performs onboard state estimation updates at 1 kHz [214]. While the state estimation runs at a high frequency, the Crazyflie is designed to accept control commands at lower or variable rates, maintaining the most recently received command between updates.

Figure 4.2: Crazyflie 2.1 Nano Quadcopter

A Flow Deck sensor board was mounted on the Crazyflie to enable positioning [216]. It uses a Time-of-Flight sensor to measure the vertical distance to the ground and an optical flow sensor to detect horizontal motion relative to the ground. The detailed specifications of the hardware platform and onboard sensors are summarized in Table 4.3. In addition, the specifications of the individual sensors, along with other potential disturbances and their typical noise levels, are summarized in Table 4.4.

Component	Specification
Platform	Crazyflie 2.1 nano-quadcopter
Mass	27 g
Motor-to-motor distance	9 cm
Main processor	STM32F405 (Cortex-M4, 168 MHz)
Radio processor	nRF51822 (Cortex-M0, 32 MHz)
Sensors	IMU, barometer, and ToF sensors
Communication	2.4 GHz Crazyradio PA
State-estimation rate	1 kHz

Table 4.3: Summary of the Crazyflie 2.1 hardware platform and onboard sensors used in the quadcopter experiment [214].

Source	Specification	Typical Noise (1σ)
IMU Accelerometer	BMI088	$(1.6\text{–}1.9) \times 10^{-3}\ \mathrm{m/s^2/\sqrt{Hz}}$ [217]
IMU Gyroscope	BMI088	$0.014\ ^\circ/\mathrm{s}/\sqrt{\mathrm{Hz}}$ [217]
Barometer	BMP388	(0.1–0.4) Pa RMS [218]
Time-of-Flight sensor	VL53L1X	≈ 0.02 m RMS [219]
Radio link latency	2.4 GHz Crazyradio PA	≈ 4 ms (round-trip) [220, 221]

Table 4.4: Specifications of the onboard sensors and disturbance sources, including their typical noise or fluctuation levels.

The primary objective for this experiment is for the quadcopter to achieve and maintain a stable hover at 1 m above the ground. Before conducting experiments with the physical quadcopter, a simulation environment was developed to closely reproduce real-world conditions. In this simulation, the observation represents the quadcopter's altitude above the ground, and the action corresponds to its vertical speed, which is constrained to a maximum of $\pm 2\,\mathrm{m\,s^{-1}}$.

The training process consists of three stages:

- *Stage 1*: The policy is initially trained in a simulated environment for 1000 epochs (with 10 episodes per epoch) across 15 different random seeds.
- *Stage 2*: The trained policy is then deployed on the physical quadcopter, which undergoes further training for 10 epochs (100 episodes). This stage allows the policy to adapt to real-world challenges like sensor noise, actuation delays, and policy inaccuracies.
- *Stage 3*: Finally, the refined real-world policy is transferred back to simula-

tion for another 1000 training epochs across 15 different random seeds. This stage evaluates how well a policy trained under real-world conditions can be reintegrated into the simulated environment.

To address the effects of measurement uncertainty and timing perturbations (some of which are detailed in Table 4.4), two strategies were employed during training and deployment. First, in the simulation (stages 1 and 3), Gaussian noise with $\mu = 0$ and $\sigma = 1\,\text{cm}$ is added during training as a form of domain randomization [222], which exposes the policy to variability during training to improve its robustness to perturbations and prepare it for sim-to-real transfer to a physical quadcopter. Second, to prevent timing perturbations and ensure deterministic control update, the sampling frequency was fixed at 10 Hz for both simulated and real-world implementations. This fixed-rate control ensures stable command execution.

Each episode consists of 100 timesteps in both simulated and real-world experiments. Each timestep in the physical setup lasts 100 ms, so the quadcopter has 10 s per episode to reach and maintain the target altitude. In the experiments, PPO control actions are sampled and applied every 100 ms (10 Hz). The action remains constant during each control interval. The average policy network inference time is approximately 312 µs, which is significantly faster than the control update interval and enables meeting the real-time control requirements for stable Crazyflie operation.

The reward function encourages the quadcopter to reach and remain at the target altitude. The reward at each timestep is given by $r = -\left|h - h_{\text{target}}\right|$, where h represents the quadcopter's current altitude and h_{target} represents the target altitude. The total reward increases as the quadcopter approaches the target altitude and remains there for a longer period.

The hardware platform, onboard sensors, and their corresponding noise characteristics are summarized in Tables 4.3 and 4.4, while Table 4.5 lists hyperparameters used for training and deployment of the Vanilla PPO and policy smoothness methods.

The complete source code, including both the simulation and real-world implementations, is available at the following GitHub repository: https://github.com/payamparvizi/Crazyflie_RL.

Method	Hyperparameter	Value
PPO	Optimizer	Adam
	Activation function	ReLU
	Policy learning rate	5×10^{-4}
	Value learning rate	1×10^{-3}
	Policy hidden layer	$[64, 64]$
	Value hidden layer	$[64, 64]$
	Clipping ϵ	0.20
	Discount factor γ	1.0
	Value function loss coefficient	0.1
	Entropy coefficient	0
	Steps per episode	100
	Random seeds per configuration	10
CAPS	Regularization coefficient λ_T	5×10^{-3}
	Regularization coefficient λ_S	1×10^{-4}
SAPPS	Regularization coefficient λ_T	5×10^{-3}
	Homogeneous ratio c_{hm}	1.0

Table 4.5: Hyperparameters and their corresponding values in the quadcopter environment.

4.2.2 Results and Discussion

The results of the total reward and action fluctuation (Sm) for all the stages of the Quadcopter environment are summarized in Figures 4.3 and 4.4, respectively. As shown in Figure 4.3, although the total reward in stage 1 is comparable across all methods, SAPPS achieves a higher value in stages 2 and 3 compared to CAPS and Vanilla PPO. Additionally, Figure 4.4 shows that in stage 1, SAPPS initially reduces the action fluctuation (Sm) faster than other methods, and its final value is comparable to CAPS. In stages 2 and 3, SAPPS maintains a lower action fluctuation (Sm) than CAPS and Vanilla PPO, with a relative reduction of approximately 63% and 72%, respectively.

The results demonstrate the sim-to-real transferability of the SAPPS method. Despite being trained in simulation, SAPPS was successfully applied to the physical quadcopter, achieving both higher performance and a smoother policy compared to other methods. Importantly, as described in Chapter 3, the complexity of deep neural networks and their sensitivity to small perturbations in input observations can result in high-frequency oscillations in output actions. These oscillations can severely degrade real-world performance [25]. The successful deployment

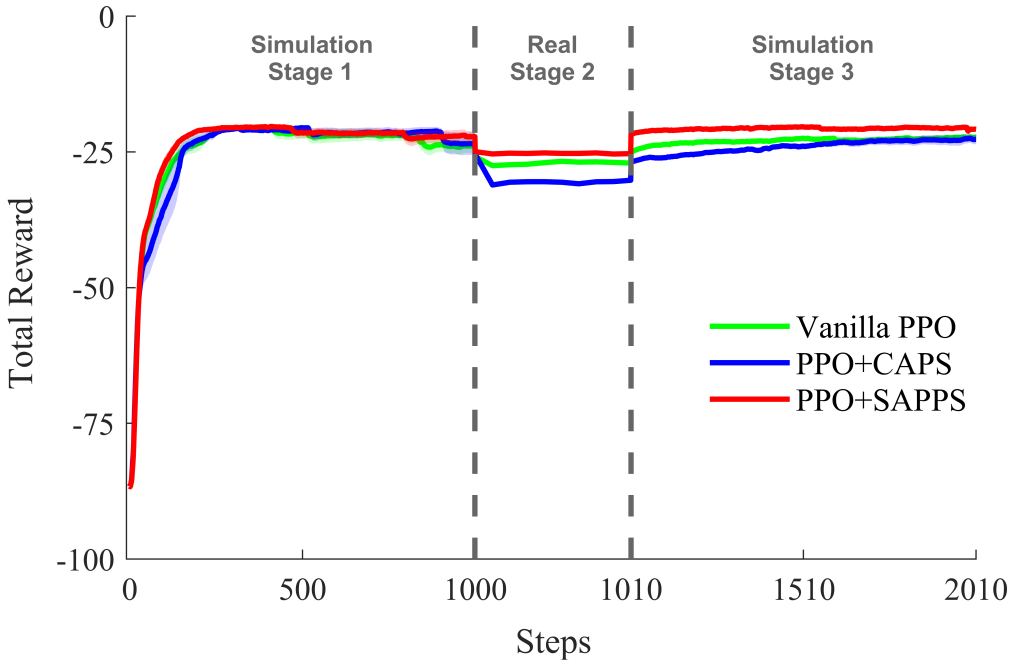

Figure 4.3: Total reward comparison of policy smoothness methods and Vanilla PPO across training stages in the Quadcopter environment

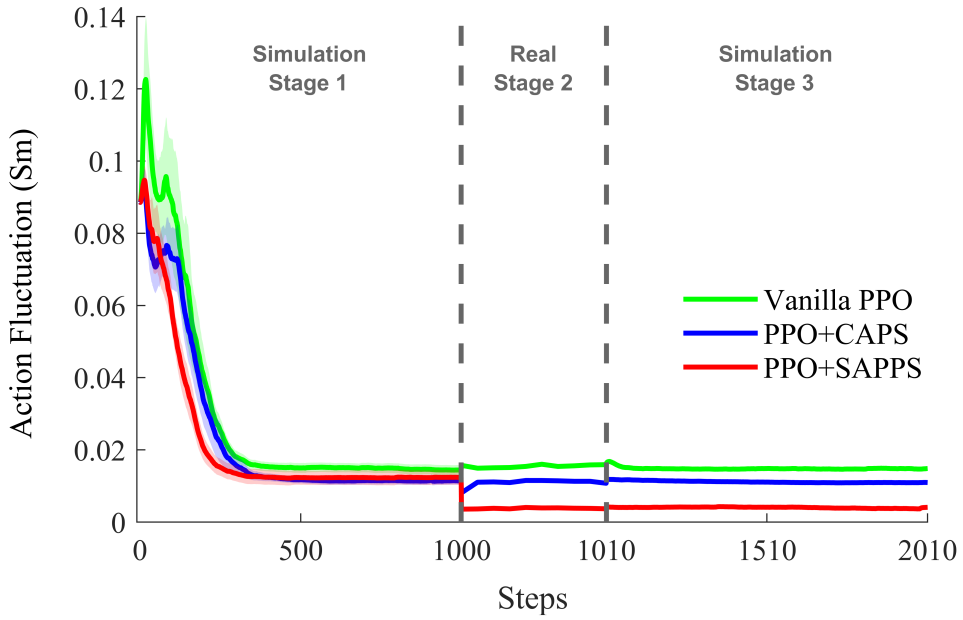

Figure 4.4: Smoothness measure comparison of policy smoothness methods and Vanilla PPO across training stages in the Quadcopter environment

of SAPPS on the physical system highlights its ability to mitigate these effects, demonstrating adaptability to dynamic changes and robustness to input perturbations, such as sensor noise and actuation delays. Therefore, SAPPS proves to be a strong candidate for sim-to-real transfer in continuous control problems.

To better evaluate the smoothness of the policy, the altitude and velocity changes of the real-world quadcopter in the final episode of Stage 2 are shown in Figures 4.5 and 4.6. As illustrated in Figure 4.5, all methods perform similarly during the initial timesteps of the episode as the quadcopter ascends. However, once it reaches the target altitude, oscillations occur. Among the evaluated methods, SAPPS is the only one that generates a trajectory around the target altitude, whereas the other two methods settle around the incorrect altitude with respect to the reference.

The oscillations observed arise from several factors, including actuation delays, the simplicity of the reward function, and sensor noise, which does not necessarily reflect actual changes in the quadcopter's state. Additional contributions may arise from disturbances that are not adequately compensated by the controller, and from control command saturation, which limits the controller's ability to apply corrective actions. Finally, a smaller portion of the oscillations may result from the policy not being fully optimal, and further training could help mitigate these effects.

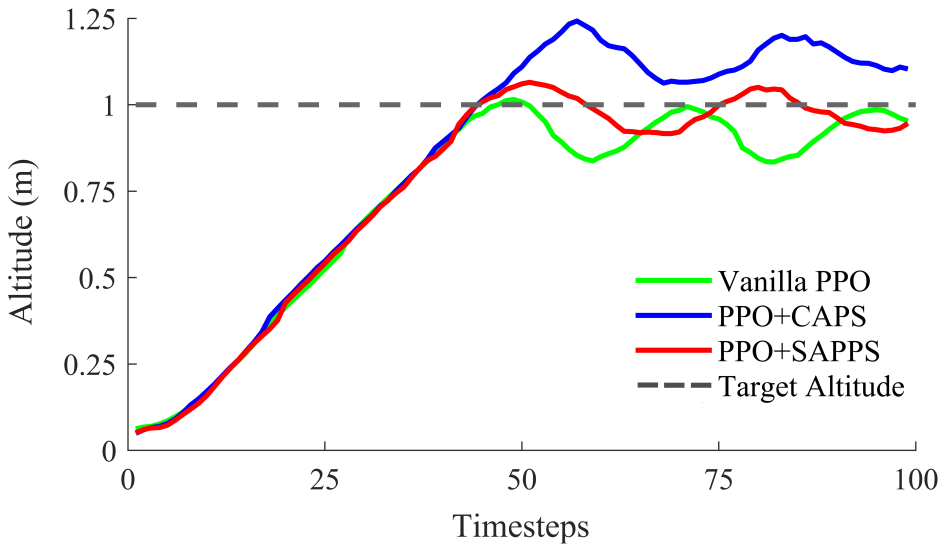

Figure 4.5: Comparison of altitude changes among policy smoothness methods and Vanilla PPO during the final episode of the real-world quadcopter experiment

In addition, Figure 4.6 presents the time history of the velocity. Given that the quadcopter's velocity is constrained between $\pm 0.2\,\mathrm{m\,s^{-1}}$, Vanilla PPO frequently reaches these boundary limits, which results in sharp velocity fluctuations and

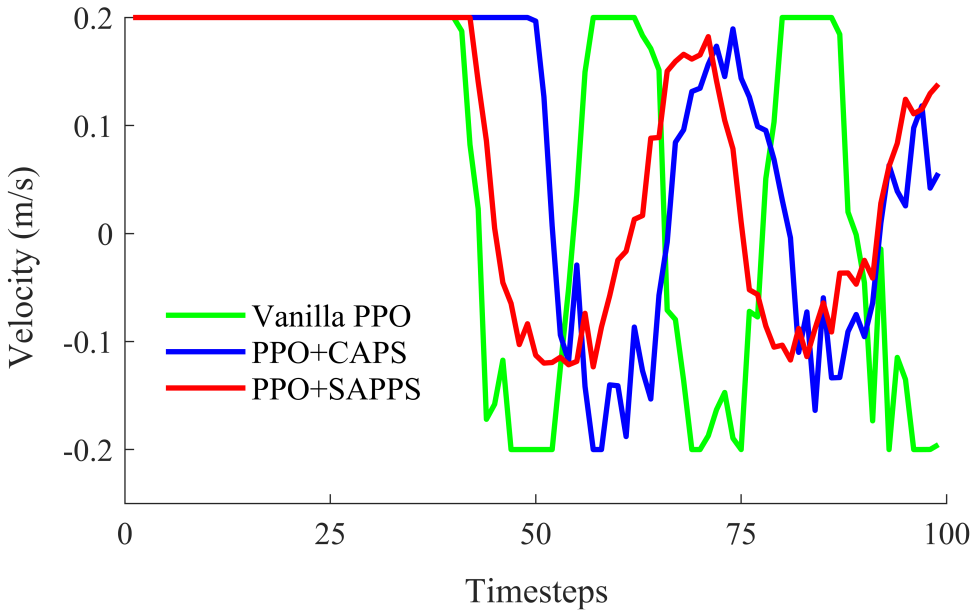

Figure 4.6: Comparison of velocity changes among policy smoothness methods and Vanilla PPO during the final episode of the real-world quadcopter experiment

sudden corrections. However, SAPPS demonstrates smoother velocity transitions by reducing abrupt changes. The CAPS method also achieves relatively smooth velocity changes, similar to SAPPS, but SAPPS further reduces unnecessary fluctuations. These results indicate that SAPPS effectively reduces velocity fluctuations while maintaining overall performance, which leads to more stable hovering.

SAPPS method achieves better performance and lower action fluctuation than other methods in real-world quadcopter experiments, which demonstrates its robustness and adaptability to real-world non-ideal conditions such as sensor noise and actuation delays. By generating smoother control actions, SAPPS not only improves performance but also improves operational efficiency by mitigating excessive energy consumption, abrupt actuation, and actuator wear and tear. These results confirm that SAPPS is a practical and effective solution for sim-to-real transfer in continuous control tasks.

4.3 Summary

To assess the general applicability of the proposed method, the SAPPS-regularized PPO algorithm was evaluated on a set of MuJoCo continuous control tasks, including Walker, Reacher, Half-Cheetah, Swimmer, and Ant. The results showed that

PPO+SAPPS consistently outperformed Vanilla PPO and achieved performance comparable to PPO+CAPS, demonstrating improved policy smoothness and overall control performance across diverse state and action space dimensions. In addition, the practical applicability of SAPPS was evaluated through deployment on a nano quadcopter for a hovering task aimed at maintaining a stable altitude, where PPO+SAPPS outperformed both Vanilla PPO and PPO+CAPS in simulation and real-world experiments.

Overall, the results demonstrated SAPPS's adaptability, smooth control, and practical utility, confirming it as a viable solution for both simulated and real-world continuous control applications.

Chapter 5

RL-Based Environment for Wavefront Sensorless Adaptive Optics

A simulated RL environment has been developed to train and evaluate RL algorithms in a wavefront sensorless AO system, as illustrated in Figure 5.1. In this setup, the RL controller adjusts the actuator configuration of a deformable mirror to correct wavefront distortions and improve data transmission rates. The agent receives observations from a low-pixel-count photodetector, which reflects the power distribution resulting from the current mirror configuration.

To achieve a high data rate in optical communication, the system employs a small active area detector, typically a single-mode fiber-coupled receiver with high bandwidth, positioned at the focus of the ground station telescope. Accordingly, the reward signal is derived from both the optical power coupled into the single-mode fiber and the corresponding power distribution on the photodetector. This reward provides feedback on the effectiveness of the agent's actions, guiding it to learn mirror configurations that mitigate wavefront distortions and optimize data transmission.

This chapter is organized as follows: Section 5.1 introduces the setup of the proposed AO-RL environment, developed for training and evaluating RL algorithms. Section 5.2 describes the environment configuration in detail. Furthermore, the three fundamental components of an RL formulation—action space, observation space, and reward function—are discussed in Sections 5.3, 5.4, and 5.5, respectively.

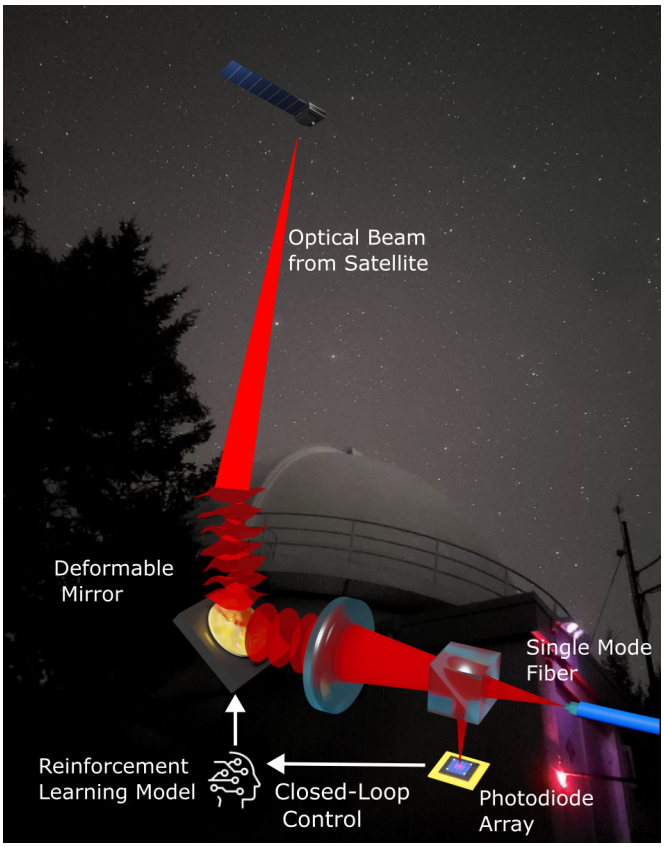

Figure 5.1: An illustration of the concept of a wavefront sensorless AO system for optical satellite communications using RL

5.1 Environment Setup

The RL environment is implemented following the standards of the OpenAI Gymnasium framework [213]. It is built on the HCIPy (High Contrast Imaging for Python) package [223], which provides a comprehensive set of libraries for AO. HCIPy includes libraries for wavefront generation, atmospheric turbulence modeling, propagation simulation, fiber coupling, and the implementation of deformable mirrors and wavefront sensors. Developing a simulated AO-RL environment is a crucial initial step toward developing RL-based wavefront sensorless AO systems for satellite communication downlinks. This simulation framework enables the development and refinement of RL algorithms to meet the strict requirements of this domain before undergoing costly evaluations in physical simulations and real-world implementations.

The AO system simulated in this environment is designed to couple 1550 nm light into a single-mode fiber under various turbulence conditions, characterized

by the parameter D/r_0, as illustrated in Figure 5.2. This parameter denotes the ratio of the telescope's diameter D to the Fried parameter r_0, which is a fundamental coherence length that quantifies the spatial correlation of atmospheric phase distortions across the wavefront [224, 225]. In other words, r_0 serves as a measure of the optical transmission quality through the atmosphere. A smaller r_0 indicates stronger turbulence and more severe wavefront distortions. Also, as the telescope diameter D increases, it collects more light and samples a larger region of the distorted wavefront, resulting in greater phase variance across the aperture and, thus, increased demands on the correction system.

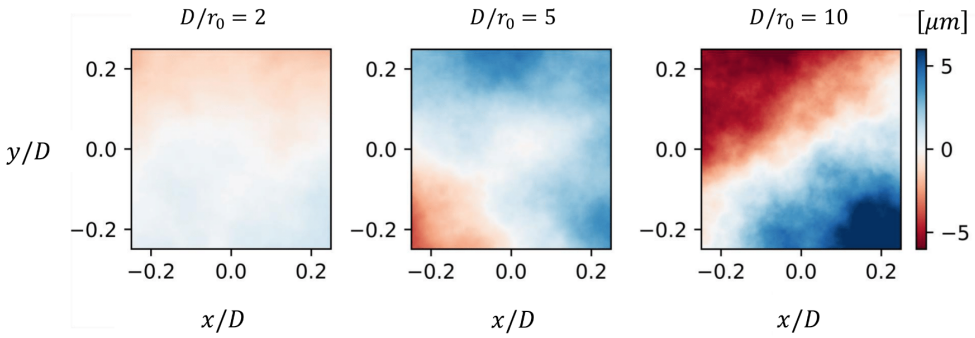

Figure 5.2: Atmospheric phase screen with respect to D/r_0

In this work, given the characteristics of the physical setup requirements, the telescope diameter D is fixed at 0.5 m, while the r_0 value is adjusted to evaluate the system's performance under varying levels of atmospheric turbulence.

It is assumed that the atmospheric turbulence is either quasi-static or dynamic and that the satellite remains stationary throughout the experiments. In quasi-static and dynamic settings, the atmosphere evolves independently of the agent's actions. The difference lies in the rate of atmospheric evolution relative to the agent's sampling frequency. In a quasi-static atmosphere, the temporal variation of turbulence is much slower than the agent's control frequency, which allows the environment to be assumed as constant (static). In contrast, in a dynamic atmosphere, the environment evolves on a time scale comparable to or faster than the agent's sampling frequency.

5.2 Environment Configuration

The RL problem can be formulated as a finite-horizon or an infinite-horizon problem. In the finite-horizon case, each episode corresponds to the duration of the

satellite's communication with the receiver, which provides a terminal state for the task. In contrast, the infinite-horizon case does not have terminal states and is more suitable for continuous control tasks. In this study, the infinite-horizon case is adopted to better reflect the continuous nature of the problem. However, an episodic structure is imposed for computational practicality. Each episode provides a bounded time window for experience collection and enables policy updates based on finite samples. This approach reduces memory overhead and simplifies evaluation without altering the underlying problem formulation.

The effectiveness of an RL policy is evaluated in mapping the deformable mirror from its neutral (flat) position to a shape that focuses the beam onto the single-mode fiber.

In the generated AO-RL environment, users can choose between quasi-static, semi-dynamic, and dynamic atmosphere settings. In these settings, the initial state is defined by the power distribution on the focal plane when the deformable mirror is in its flat configuration. In the preliminary experiment reported in Appendix D, the initial atmospheric conditions are the same to facilitate a better comparison of approaches. However, in the extended experiments reported in Section 6.2, the initial atmospheric conditions are randomly sampled from a distribution to demonstrate the generalizability of the approaches across diverse environments.

In a quasi-static environment, it is assumed that the atmosphere remains in a quasi-static state, and a static turbulence profile is considered valid for training purposes. In other words, it is assumed that the deformable mirror operates at a sufficiently high speed to approximate the atmosphere as quasi-static. Most deformable mirrors are capable of corrections at speeds of up to 1 – 2 kHz [226]. Faster deformable mirrors can be achieved by using smaller mirrors. For a 50 cm telescope and this deformable mirror choice, the system is expected to achieve reasonable correction under turbulence conditions where r_0 is less than 6.25 cm. r_0 conditions are anticipated to range from 5 cm to 15 cm for satellite elevation angles above 15°.

In a semi-dynamic environment, the turbulence profile remains constant (static) throughout each episode but changes randomly between episodes. This setting is designed to evaluate the ability of RL controllers to generalize across different quasi-static atmospheric conditions and to identify potential challenges before advancing to a fully dynamic environment that more closely reflects real-world scenarios.

In a dynamic environment, the evolution of atmospheric turbulence is influ-

enced by a velocity field that describes the motion of the atmosphere over space and time. This velocity field induces the temporal variation of the atmospheric phase screen, as illustrated in Figure 5.3. To maintain temporal continuity between episodes, the final time step of episode i is used as the initial condition for episode $i+1$.

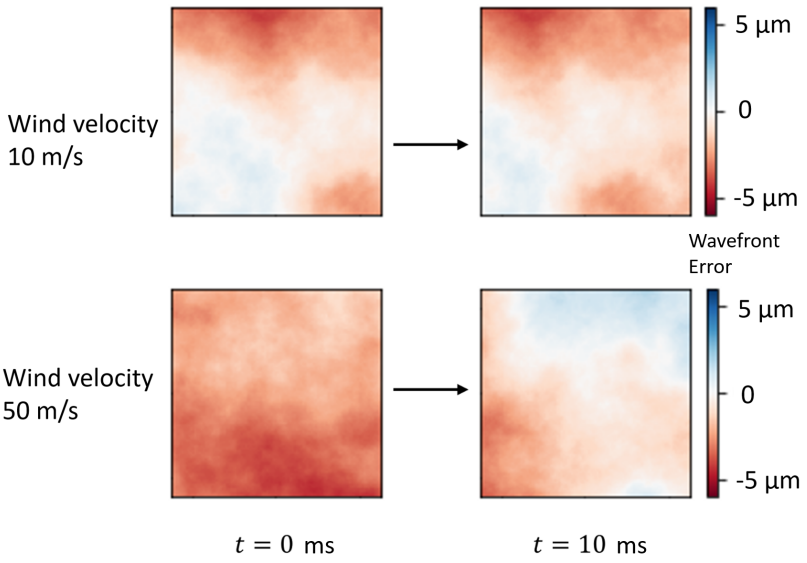

Figure 5.3: Effect of atmospheric velocity on the temporal evolution of the atmospheric phase screen. The color scale indicates phase variations.

In this RL problem, although training is organized into episodes for computational purposes (e.g., policy updates), the underlying problem is an infinite-horizon since the environment does not reset between episodes.

5.3 Action Space

In AO, the actions of the RL agent correspond to the displacements of actuators positioned beneath the deformable mirror, as illustrated in Figure 5.4. The number of actuators defines the degrees of freedom for shaping the mirror's surface. These actuators are responsible for spatially shaping the continuous reflective surface of the deformable mirror. Each actuator has a movement range of $\pm 5\,\mu$m, providing high precision in shaping and positioning the mirror's surface.

While it is possible to control individual actuators directly, doing so results in a high-dimensional action space. High-dimensional action spaces can pose challenges for RL algorithms in dynamic environments due to the curse of dimensionality. This issue occurs because computational complexity increases exponentially

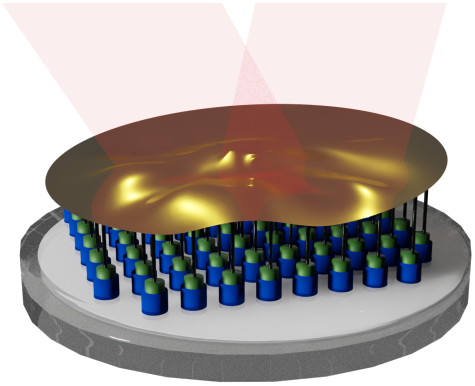

Figure 5.4: Illustration of a deformable mirror surface and incident beam

with the number of dimensions [227]. The policy may struggle to learn effective mappings from states to actions when the action space has significantly more dimensions than the state space.

To address this, the RL environment offers the option of parameterizing the action space using a projection along a lower dimensional set of basis functions. These mode bases can be chosen either from Zernike polynomials [60] or Disk Harmonic basis functions [228], with the latter serving as the default in the HCIPy framework. The choice of n mode bases corresponds to an n-dimensional continuous action space.

The analysis indicates that lower-order Zernike polynomials improve the policy's ability to effectively map states to actions. The decision to use low-order Zernike polynomials was based on an experiment conducted in a semi-dynamic environment characterized by atmospheric turbulence with $D/r_0 = 3.33$. In this environment, the turbulence profile remains static within each episode but changes randomly from one episode to the next. This setting is designed to evaluate the generalization capability of RL controllers across different quasi-static atmospheric conditions. Details of this experiment are provided in Appendix D.2.2.

5.4 Observation Space

The power distribution of the wavefront propagating through the focal plane is utilized to form the observation of the state of the environment. The focal plane profile is shown in Figure 5.5 (left). The white circle within the figure indicates the entrance of the single-mode fiber core.

The observable states of the system are directly and efficiently linked to the

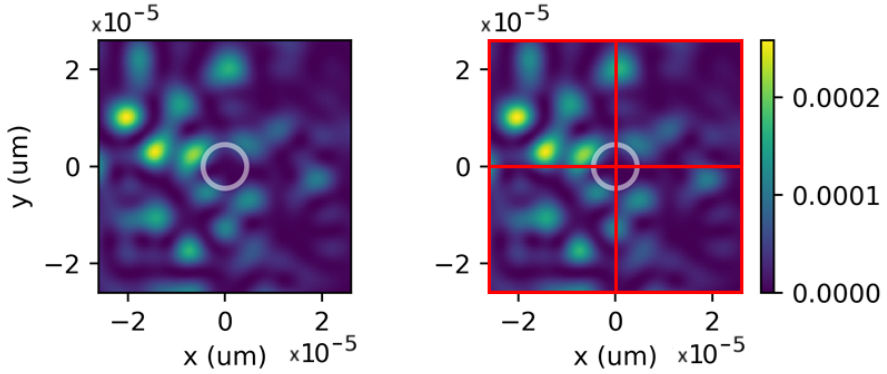

Figure 5.5: Focal plane profile, (left) continuous, (right) discretized into a sub-aperture array of 2×2 pixels

amount of light coupled into the fiber, as calculated by the reward function described in Section 5.5. This approach is required over using full Markovian states, which would require detailed information about the environment, such as the satellite's angle and atmospheric conditions—data that is often unavailable or difficult to measure accurately. This is necessary because it is infeasible to collect and utilize the information required for a fully Markovian representation. In other words, the observations rely on directly measured information about the environment's state after the wavefront has propagated through the focal plane and been discretized into a sub-aperture array of $n \times n$ pixels. In contrast, a fully Markovian representation would require complete knowledge of all system variables, which introduces significant uncertainty due to unmeasured or unmeasurable factors. The trade-off in this approach is a loss of full observability, as the state is compressed to optimize for cost and latency. However, RL provides tools to manage partial observability [229]. The agents can still learn effective policies using the photodetector as shown in Figure 5.5 (right), where each red square represents an individual pixel of the detector.

In the RL environment, the dimension of the observation space is n^2. Increasing the value of n increases the dimension of the observation space, which in turn reduces partial observability.

In the preliminary experiments (reported in Appendix D), conducted under quasi-static atmospheric conditions, the focal plane is discretized into a sub-aperture array consisting of 2×2 pixels. This setup was implemented to illustrate the possibility of using a fast and relatively cost-effective quadrant photodetector. A low-pixel-count detector mitigates the need for slower and more expensive read-out circuits used in infrared cameras. This approach allows for more light per pixel,

reduces noise, and improves the training speed of the RL algorithm. However, under dynamic atmospheric conditions, the low-pixel sub-aperture array may be insufficient due to low partial observability. In such cases, the dimension of the observation space can be increased by increasing the value of n. The corresponding experiment on a semi-dynamic environment is provided in Appendix D.2.2.

5.5 Reward Function

In the RL environment, two reward function options are available: the Strehl ratio (Equation 5.1) and the proposed fiber-coupling reward function (Equation 5.2).

The Strehl ratio-based reward function is calculated using the Strehl ratio of the optical system, which is defined as the ratio of the normalized peak intensity of the point spread function (PSF) to the peak intensity of an ideal, non-aberrated PSF as shown in Figure 5.6 [230]. PSF describes the response of an optical system to an infinitesimal point source of light. It represents the intensity distribution in the focal plane resulting from light diffraction and optical aberrations. In other words, the PSF characterizes how an optical system spreads light from a point source [231].

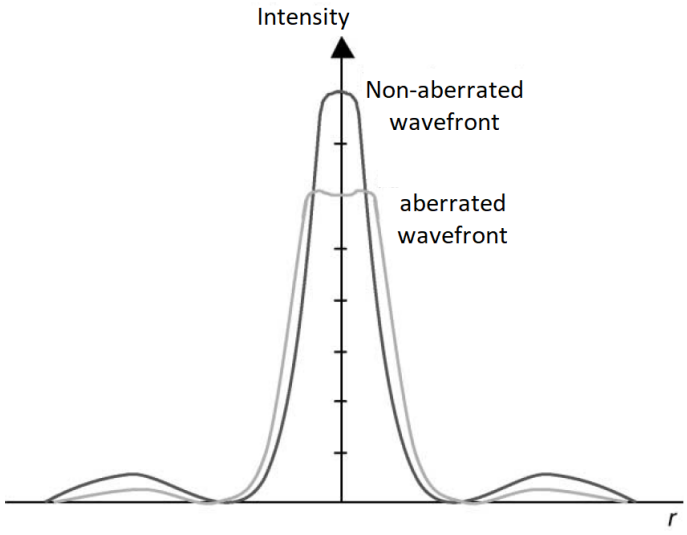

Figure 5.6: Strehl ratio criterion

A high Strehl ratio indicates a high degree of wavefront correction, where the focused light beam resembles an Airy disk and is approximately proportional to the amount of light that can be coupled into a fiber [232]. It is considered an approximation because the focused beam should resemble a Gaussian profile for op-

timal coupling into an optical fiber. According to Mahajan [233], the Strehl ratio r_{SR} for systems with a circular pupil can be expressed in terms of the variance of the phase aberration across the pupil as follows:

$$r_{SR} = e^{-\sigma_{\Phi}^2} \tag{5.1}$$

where σ_{Φ}^2 represents the variance of the phase aberration.

Although the Strehl ratio is commonly used in AO, it is not well-suited for the application discussed in this work. Since its calculation requires the measurement of the PSF, which typically necessitates a high-resolution detector, such as a high-resolution camera, to accurately capture the peak intensity of the PSF. However, the system explicitly avoids reliance on such high-resolution detectors (specifically, an InGaAs camera) in favor of a more cost-effective, compact, and faster solution: a low-pixel-count photodetector, as described in Section 5.4. Additionally, the calculation of the Strehl ratio requires sufficiently long exposure time to account for speckle variation, which is impractical for the system's objectives. Moreover, the Strehl ratio is not an ideal metric for coupling into a single-mode fiber. At high Strehl ratios, its correlation with single-mode fiber coupling efficiency weakens. This limitation occurs because, at higher Strehl ratios, the coupling efficiency is constrained by mode mismatch between a focused flat-top beam and the Gaussian mode of the single-mode fiber [232, 234]. As a result, the Strehl ratio's effectiveness as a reward function is limited to lower values.

The fiber-coupling reward function

$$r_N = \omega_1 r_{SMF} + \omega_2 r_{SSIM} - \omega_3 r_{TE} \tag{5.2}$$

is proposed for the RL environment to overcome these limitations. This reward function integrates three key components: the total power of the single-mode fiber r_{SMF}, the Structural Similarity Index Measure r_{SSIM}, and the tilt error r_{TE}, where ω_1, ω_2, and ω_3 are weights assigned to the individual terms to balance their contributions in the overall reward. The individual components of the reward function are defined as follows:

- r_{SMF}: This term represents the total power coupled into the single-mode fiber core, which is the primary optimization objective. The definition of r_{SMF} is:

$$r_{SMF} = \left(\frac{P_{curr}}{P_{ref}} \right)_{SMF} \tag{5.3}$$

where P_{curr} denotes the current total power coupled into the single-mode fiber, while P_{ref} is the reference value for the total power coupled when no aberrations are present. The ratio between these values represents the percentage of fiber coupling efficiency.

The total power coupled into a single-mode fiber is a critical indicator of light concentration and an effective reward signal for guiding the RL agent. In contrast to the Strehl ratio, which requires a high-resolution detector and computationally intensive image processing, r_{SMF} can be measured directly using a low-pixel-count photodetector placed at the fiber output, with less computation. However, r_{SMF} may suffer from reward sparsity during the early stages of training due to the small core diameter of the single-mode fiber. This sparsity may lead to slow convergence. Therefore, using the total power of a single-mode fiber as the sole reward function is inadequate.

- r_{SSIM}: The Structural Similarity Index Measure (SSIM) is a metric that quantifies the similarity between two images by comparing their luminance (local mean intensity), contrast (local standard deviation), and structural information (local covariance) [235]. In the context of wavefront correction, SSIM evaluates the similarity between the spatial structure of the power distribution with and without aberrations. However, the Strehl ratio compares only the peak intensity of the PSF under aberrated and ideal conditions. The SSIM combines luminance, contrast, and structural information equations into a single equation:

$$r_{SSIM} = \frac{(2\mu_{curr}\mu_{ref} + c_1)(2\sigma_{curr,ref} + c_2)}{(\mu_{curr}^2 + \mu_{ref}^2 + c_1)(\sigma_{curr}^2 + \sigma_{ref}^2 + c_2)} \tag{5.4}$$

  where μ_{ref}, σ_{ref} are the mean and standard deviation of the power distribution when there is no aberration, and $\mu_{curr}, \sigma_{curr}$ as the mean and standard deviation of the current power distribution. c_1 and c_2 are two constants to stabilize the division when the denominator is small, avoiding numerical singularities.

- r_{TE}: The cost associated with tilt error is included to improve the reward function. The tilt term refers to the angle at which a wavefront is oriented with respect to a reference plane [4]. Based on this definition, the orientation along the x and y axes can be calculated, referred to as x-tilt and y-tilt,

respectively [236, 237]. A photodetector is assumed to be placed at the focal plane of an optical system, where the power distribution is discretized into an $n \times n$ pixel array, as illustrated by the 2×2 pixel example shown in Figure 5.5. The power distribution on the focal plane can be expressed as the following matrix:

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix}$$

where p_{ij} denotes the power incident on the $(i,j)^{\text{th}}$ pixel of the photodetector.

x-tilt can be defined as the difference in the total power between the left and right halves of the photodetector, where the left half corresponds to the columns $j = 1$ to m, and the right half corresponds to the columns $j = m'$ to n. The equation for calculating x-tilt is given by:

$$x_{tilt} = \frac{\left| \sum_{i=1}^{n} \sum_{j=1}^{m} p_{ij} - \sum_{i=1}^{n} \sum_{j=m'}^{n} p_{ij} \right|}{\sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij}} \tag{5.5}$$

Similarly, the y-tilt represents the difference in the total power between the upper and lower halves of the photodetector, where the upper half corresponds to the rows $i = 1$ to m, and the lower half corresponds to the rows $i = m'$ to n. The y-tilt can be computed using the equation:

$$y_{tilt} = \frac{\left| \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij} - \sum_{i=m'}^{n} \sum_{j=1}^{n} p_{ij} \right|}{\sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij}} \tag{5.6}$$

Here, the values of m and m' are determined as follows:

$$\text{if} \begin{cases} n/2 & \text{is even,} \quad m = n/2 \quad \text{and} \quad m' = n/2 + 1, \\ n/2 & \text{is odd,} \quad m = \lfloor n/2 \rfloor \quad \text{and} \quad m' = \lceil n/2 + 1 \rceil, \end{cases}$$

where $\lfloor \cdot \rfloor$ denotes the floor function, and $\lceil \cdot \rceil$ denotes the ceiling function.

Finally, the reward for the total tilt error, r_{TE}, is defined as the sum of the x-tilt (Equation 5.5) and y-tilt (Equation 5.6), expressed as:

$$r_{TE} = x_{tilt} + y_{tilt} \tag{5.7}$$

In simpler terms, the total power coupled into the single-mode fiber core r_{SMF} reflects the performance at the fiber core's exit, while r_{SSIM} and tilt error r_{TE} reflect the performance at the fiber core's entrance. These metrics were chosen because they capture the optical system's direct observations in a sensorless setting.

Other turbulence-related parameters, such as the Fried parameter r_0, could in principle be incorporated into the reward function. However, because r_0 is an extrapolated rather than directly measurable quantity, including such parameters would introduce parameter uncertainty into the RL environment, which could compromise learning stability and policy reliability in a sensorless setting. For this reason, the reward is kept agnostic to r_0, and the RL algorithm is instead evaluated under multiple turbulence regimes (different D/r_0 values). This approach demonstrates robustness across varying conditions without requiring explicit knowledge of turbulence.

5.6 Discussion

The proposed RL environment leverages the HCIPy package, which models key physical processes in adaptive optics with reasonably high realism, making it a strong platform for developing controllers intended for real-world applications. However, several limitations constrain complete sim-to-real transferability. For example, atmospheric turbulence is generated using Kolmogorov theory, which provides a close approximation but does not fully model the complexity of real atmospheric conditions. Additionally, while the HCIPy deformable mirror model is physically based, it assumes idealized system behavior and neglects detailed actuator dynamics. Moreover, in real-world dynamic conditions, the Fried parameter r_0 and the atmospheric drift velocity are time varying, whereas in the RL environment, they are assumed fixed. Similarly, the satellite is not modeled as continuously traversing the sky. Sensor characteristics are also simplified, with realistic noise models and measurement delays neglected, and other optical aberrations beyond turbulence excluded. Therefore, while the proposed RL environment offers a physically plausible and valuable testbed for controller development, further validation would be necessary for robust real-world deployment.

5.7 Summary

This chapter presented the development of a simulated RL environment for wavefront sensorless AO designed for training and evaluating RL algorithms. The environment is implemented according to the OpenAI Gymnasium framework standards to simplify the integration and analysis of RL methods. Built using the HCIPy library, it enables simulations of atmospheric turbulence and optical propagation.

While training is organized into episodes for computational purposes, the underlying problem is modeled as an infinite-horizon, as the environment does not reset between episodes.

The key components of the environment include:

- **Action Space:** Defined by the movements of deformable mirror actuators, parameterized using mode bases (e.g., Zernike polynomials) to reduce dimensionality and improve learning efficiency.
- **Observation Space:** Based on discretized power distributions in the focal plane, enabling low-latency feedback through low-pixel-count photodetectors.
- **Reward Function:** Two reward types are considered. The traditional Strehl ratio is shown to be limited in practical scenarios. A fiber-coupling reward function is proposed, integrating single-mode fiber coupling efficiency, structural similarity, and wavefront tilt error to improve policy performance.

The environment includes both quasi-static and dynamic atmospheric conditions. Overall, it provides an efficient platform for exploring model-free control strategies in wavefront sensorless AO systems for satellite-to-ground optical communication.

Chapter 6

Results and Discussion on Wavefront Sensorless Adaptive Optics

This chapter presents a detailed evaluation of RL-based controllers within the RL environment developed for a simulated wavefront sensorless AO system, as discussed in Chapter 5. The objective is to maximize and maintain consistent fiber coupling efficiency while keeping the system low-cost by utilizing low-pixel-count photodetectors for LEO satellite-to-ground optical communication downlink.

6.1 Preliminary Analysis and Control Framework Selection

6.1.1 Initial Evaluation

As detailed in Appendix D, preliminary evaluations of RL controllers for the wavefront sensorless AO system were conducted to gain deeper insight into their potential challenges. Specifically, a diverse set of deep RL algorithms, namely SAC (described in Section 2.4.1), DDPG (described in Section 2.4.2), and PPO (described in Section 2.4.3), was evaluated under two distinct environmental scenarios with varying turbulence severity: quasi-static and semi-dynamic. In the quasi-static environment, the turbulence profile is assumed to remain static throughout training, corresponding to the condition where RL model convergence is significantly faster than the atmospheric coherence time. In contrast, the semi-dynamic environment maintains a static turbulence profile within each episode, but the profile randomly changes between episodes. This setting is designed to evaluate the generalization

capability of RL controllers across different quasi-static atmospheric conditions and to identify challenges before transitioning to a dynamic environment, which more closely represents real-world operational conditions. Additionally, the analyses include comparisons between the RL-based controllers, a Shack–Hartmann wavefront sensor-based AO system, and a flat mirror scenario.

These experiments (reported in Appendix D.2) demonstrated the potential of RL algorithms in wavefront sensorless AO systems, with the PPO algorithm outperforming the others. Additionally, they provided insight into the dimensions of the action space and observation space, as well as the definition of the reward function. However, in the quasi-static experiments, each policy was trained for a specific atmospheric condition. This approach is impractical, as the learned policy lacks generalization and may not adapt to other atmospheric conditions. Moreover, the experiments do not reflect real-world scenarios where atmospheric turbulence evolves over time. Assuming quasi-static or semi-dynamic environments limits the practical applicability of the results in real-world settings.

6.1.2 PPO-Based Controller Configuration Analysis

As detailed in Appendix E, additional experiments were conducted using the Vanilla PPO algorithm to further evaluate the RL-based controller for the simulated wavefront sensorless AO system under quasi-static environments. Since PPO demonstrated superior performance compared to SAC and DDPG in the preliminary experiments (shown in Figure D.1), and given its relative simplicity, training stability, and effectiveness in implementation [212], it has become widely regarded as a strong baseline for sim-to-real robotic control applications (e.g., [34, 35, 36, 37]), providing a reliable foundation for comparative evaluation. For these reasons, the subsequent analysis focused exclusively on the Vanilla PPO algorithm. Unlike the preliminary experiments (reported in Appendix D), in these experiments, each policy was trained and tested across a broader range of atmospheric conditions.

Specifically, in the initial stages of the evaluation (reported in Appendix E.2), the policy was trained and tested across 10 different atmospheric conditions. This enabled faster iteration and reduced computational cost when selecting effective configurations for action space normalization, environment setup, and reward function design. Although the primary interest lies in the dynamic environment, the quasi-static environment plays a critical role in the development process. It enables evaluation of RL algorithms under controlled conditions and demonstrates

that a policy can be effectively learned within a limited number of interaction timesteps. This not only validates the feasibility of RL in AO but also provides a foundation for policy adaptation in more complex dynamic scenarios. Once a suitable configuration was determined, the number of atmospheric conditions was increased to 50 in further experiments to improve the generality and robustness of the learned policy. The performance of the Vanilla PPO algorithm was also compared with that of a Shack–Hartmann wavefront sensor-based AO system and a flat mirror scenario in a quasi-static environment.

6.1.3 Policy Smoothness via SAPPS Method

As detailed in Appendix E, the results shown in Figures E.4 and E.5 demonstrate that the performance of the Vanilla PPO algorithm in the wavefront sensorless AO system under quasi-static conditions with varying turbulence severity is unsatisfactory. In these cases, its performance falls significantly short compared to the Shack–Hartmann wavefront sensor-based AO system. It is hypothesized that this poor performance results from the PPO policy's inability to maintain stable coupling within the single-mode fiber core, likely due to oscillations around the center of the fiber core. This hypothesis is supported by the results shown in Figure D.3, where it is evident that the learned PPO controller continues to generate varying actions even when the beam is centered on the fiber core. This instability may be caused by the policy's sensitivity to small perturbations in the observations, which causes fluctuations in the actions even when the beam is already well centered. Moreover, these oscillations pose a challenge for real-world implementation, as they reduce the correction speed due to the inertia of the deformable mirror.

To mitigate these oscillations and enforce smoother control while maintaining high fiber coupling efficiency, Vanilla PPO was integrated with the proposed SAPPS method formulated in Chapter 3. The resulting performance and policy smoothness (or action fluctuation) were compared with both the Vanilla PPO and Vanilla PPO integrated with the CAPS method. Additionally, the performance of all methods is compared with the Shack–Hartmann wavefront sensor-based AO system and a flat mirror scenario. These experiments are conducted in dynamic environments with varying turbulence drift velocities.

6.2 RL Controller with Policy Smoothing in Dynamic Environments

The evaluation of the RL-based controller for the simulated wavefront sensorless AO system was conducted using the Vanilla PPO algorithm in dynamic environments with varying drift velocities. The challenge of action oscillations was also examined, and the SAPPS method (presented in Section 3.1.2) was proposed to mitigate these oscillations and improve the performance of the Vanilla PPO algorithm. Additionally, Vanilla PPO, integrated with the CAPS method (discussed in Section 3.1.1), was evaluated as a benchmark for comparison with Vanilla PPO and the proposed PPO+SAPPS approach. In addition, the performance of LipsNet neural network (discussed in Section 3.1.1), implemented within PPO, was also compared against the proposed method under non-zero drift velocity conditions.

Before presenting the experimental setup and results for the wavefront sensorless AO system, it is important to clarify that policy smoothness is evaluated based on action fluctuation, which is defined as the norm of the difference between subsequent actions. Action fluctuation is used instead of the Smoothness measure (Sm) because Sm requires additional computations, such as performing a Fast Fourier Transform (FFT), which increases computational complexity and evaluation time. In contrast, action fluctuation is simpler and faster to compute, making it a more practical choice for evaluating wavefront sensorless AO systems.

6.2.1 Experimental Setup

The RL environment developed for the simulated wavefront sensorless AO system is described in detail in Chapter 5. It is implemented in accordance with the OpenAI Gymnasium framework standards (version 0.29.1) [213] and utilizes the HCIPy package (version 0.6.0) [223] and the PyTorch framework (version 2.0.1). The experiments were performed on a high-performance computing (HPC) platform provided by Compute Canada and on the National Research Council (NRC)'s computing cluster, each equipped with 5 GB of RAM per CPU, one GPU (model depending on availability), and Python 3.9.6.

The Vanilla PPO algorithm, along with its integrations with State-Adaptive Proportional Policy Smoothing (SAPPS) and Constrained Action Policy Smoothness (CAPS), is implemented using the Tianshou RL library (version 0.5.1) [238]. This RL library is used because it helps to minimize implementation errors and reduce

development time.

The simulated AO system aims to optimize light coupling into a single-mode fiber at a wavelength of 1550 nm under varying levels of atmospheric turbulence drift velocity. Given the requirements of the physical setup, the experimental configurations are summarized in Table 6.1.

Component	Description
Telescope Aperture	0.5 m
Wavelength	1550 nm
Wavefront Resolution	128×128 grid for wavefront generation
Deformable Mirror	Approximated by first 10 Zernike modes (no explicit actuator dynamics)
Atmospheric Model	Kolmogorov turbulence
Fried Parameter r_0	0.25 m and 0.10 m
Drift Velocity v	$0\,\mathrm{m\,s^{-1}}$ to $500\,\mathrm{m\,s^{-1}}$
Action Space	10-dimensional action space based on the first 10 Zernike polynomial modes
Observation Space	5×5 pixel photodetector array
Reward Function	Fiber-coupling reward function defined in Eq. 5.2
Simulation Time	1×10^{-4} s per step (10 kHz control rate)

Table 6.1: Summary of environment and optical configuration for the wavefront sensorless AO environment.

The simulation operates within a continuous environment where a velocity field describes the atmospheric drift dynamics. A minimum acceptable atmospheric speed is defined as $5\,\mathrm{m\,s^{-1}}$, while $500\,\mathrm{m\,s^{-1}}$ is regarded as extremely high. These thresholds were chosen based on the speeds typically encountered in real-world scenarios. A zero drift velocity case, which represents a quasi-static environment, is also considered.

For the simulations, a timestep of $\delta t = 1 \times 10^{-4}$ seconds was used. While the change in the atmospheric phase screen within a single timestep appears negligible, its cumulative effect over multiple steps leads to significant changes in observations, which, in turn, influence the objective function. The effect of drift velocity on the atmospheric phase screen is illustrated in Figure 5.3.

Additionally, $D/r_0 = 5$ is defined, which corresponds to a strongly turbulent atmospheric condition. The ratio D/r_0 represents the severity of turbulence in AO systems, as discussed in Section 5.1.

In addition to the environment and optical configuration summarized in Table 6.1, the hyperparameters of the training configuration are also tuned and documented for clarity. These hyperparameters, which are common across all experiments presented in this chapter, are listed in Table 6.2. Additional scenario-specific hyperparameters (e.g., clipping parameter ϵ, batch size, and hidden layer size) are reported alongside the respective experiment descriptions.

Parameters	Value
Optimizer	Adam
Activation function	ReLU
Max gradient norm	0.5
Number of epochs	900
Steps per epoch	3200
Steps per iteration	400
Network updates per iteration	20
Discount factor γ	0.95
GAE lambda λ	0.95
Value function loss coefficient	0.25
Entropy coefficient	0.01
Advantage normalization	True
Reward normalization	True
parallel training environments	50
Parallel testing environments	50
Frame stacking	States and actions
Random seeds per configuration	10

Table 6.2: Summary of common PPO hyperparameters applied across scenarios in the wavefront sensorless AO environment.

The common hyperparameters in Table 6.2 were selected through tuning to ensure stable training and reliable convergence. Key parameters such as the discount factor γ and GAE parameter λ were tuned to balance bias and variance in return and advantage estimation, while the entropy and value-loss coefficients were tuned to encourage exploration and stabilize critic learning. The number of parallel environments and random seeds was fixed to provide statistically robust evaluations. By keeping these hyperparameters constant across scenarios, the comparison between methods remains fair and unbiased.

To further improve training stability and performance, the remaining tunable hyperparameters were optimized according to each environment scenario. The corresponding search spaces for the PPO and policy smoothness methods (SAPPS and CAPS) are summarized in Table 6.3. These tuning processes were performed

under both quasi-static and dynamic conditions to ensure controlled and consistent parameter selection across scenarios.

Hyperparameter	Search Space
Learning rate (lr)	$1 \times 10^{-7} \rightarrow 1 \times 10^{-1}$
Clipping parameter (ϵ)	$0.01 \rightarrow 0.40$
Buffer size	$10{,}000 \rightarrow 30{,}000$
Batch size	$64 \rightarrow 512$
Hidden layer size	$80 \rightarrow 640$
Frame stacking number	$1 \rightarrow 10$
Flat scenario in episode configuration	Yes / No
Temporal smoothness coefficient ($\lambda_{T_{CAPS}}$)	$1 \times 10^{-3} \rightarrow 1$
Spatial smoothness coefficient ($\lambda_{S_{CAPS}}$)	$1 \times 10^{-3} \rightarrow 1$
Proposed smoothness coefficient ($\lambda_{T_{SAPPS}}$)	$1 \times 10^{-3} \rightarrow 1$
Homogeneous ratio (c_{hm})	$1 \times 10^{-2} \rightarrow 1 \times 10^{2}$

Table 6.3: Search space of hyperparameters explored for PPO and policy smoothness methods (SAPPS and CAPS).

6.2.2 Results

Zero Drift Velocity Condition

The zero drift velocity condition in a dynamic environment refers to a *quasi-static environment* in which the velocity field describing the atmosphere dynamics is zero, resulting in a static turbulence profile. In the experiments, the same static turbulence profile is maintained across all episodes. This setting provides a controlled environment for evaluation. If a configuration does not lead to satisfactory performance in the quasi-static setting, it is unlikely to succeed under the greater challenges posed by a dynamic setting. Therefore, it is logical to first determine an effective configuration in the quasi-static environment before transitioning to the dynamic environment.

The evaluation in the quasi-static setting investigates the issue of action oscillations by examining the performance of the PPO algorithm, along with benchmark approaches such as Vanilla PPO and PPO integrated with the CAPS method, under varying levels of turbulence severity.

This section compares the Vanilla PPO algorithm with policy smoothness methods integrated into the PPO algorithm in terms of fiber coupling efficiency and action fluctuation in a quasi-static environment under low ($D/r_0 = 2$) and high

($D/r_0 = 5$) turbulence severity. The policy smoothness methods considered are PPO+SAPPS (also denoted as SAPPS) and PPO+CAPS (also denoted as CAPS). Additionally, these approaches are also compared with a Shack-Hartmann wavefront sensor-based AO system and a flat mirror scenario.

The hyperparameters for the Vanilla PPO algorithm in the quasi-static environment are tuned based on the hyperparameter search space provided in Table 6.3 and the experiments conducted in Appendix E, and these hyperparameters are used for further tuning SAPPS and CAPS methods. These hyperparameters are summarized in Table 6.4.

Method	Hyperparameter	$D/r_0 = 5$	$D/r_0 = 2$
PPO	lr	5×10^{-5}	5×10^{-5}
	Clipping ϵ	0.15	0.15
	Buffer size	25,000	25,000
	Batch size	256	256
	Hidden layer size	320	320
	Frame stacking No.	5	5
	Flat scenario in episode config.	Yes	Yes
CAPS	Regularization coefficient λ_T	0.5	0.5
	Regularization coefficient λ_S	0.05	0.05
SAPPS	Regularization coefficient λ_T	0.5	1.0
	Homogeneous ratio c_{hm}	1.0	3.0

Table 6.4: Hyperparameters and their corresponding values in the quasi-static environment for the AO system.

The fiber coupling efficiency comparison under turbulence conditions with $D/r_0 = 5$ is shown in Figure 6.1, while Figure 6.2 presents the action fluctuation comparison.

As shown in Fig. 6.1, both CAPS and SAPPS perform comparably in a quasi-static environment with $D/r_0 = 5$. They outperform Vanilla PPO by approximately 56% and achieve 97% of the performance of the Shack-Hartmann wavefront sensor. Furthermore, Figure 6.2 highlights that these methods produce smoother policies, demonstrating 95% less action fluctuation compared to Vanilla PPO. This indicates that improved fiber coupling efficiency can be achieved with reduced actuator usage.

Also, a similar comparison was conducted under turbulence conditions with $D/r_0 = 2$. The comparison of fiber coupling efficiency among Vanilla PPO, SAPPS, and CAPS, as well as the Shack-Hartmann wavefront sensor and a flat mirror

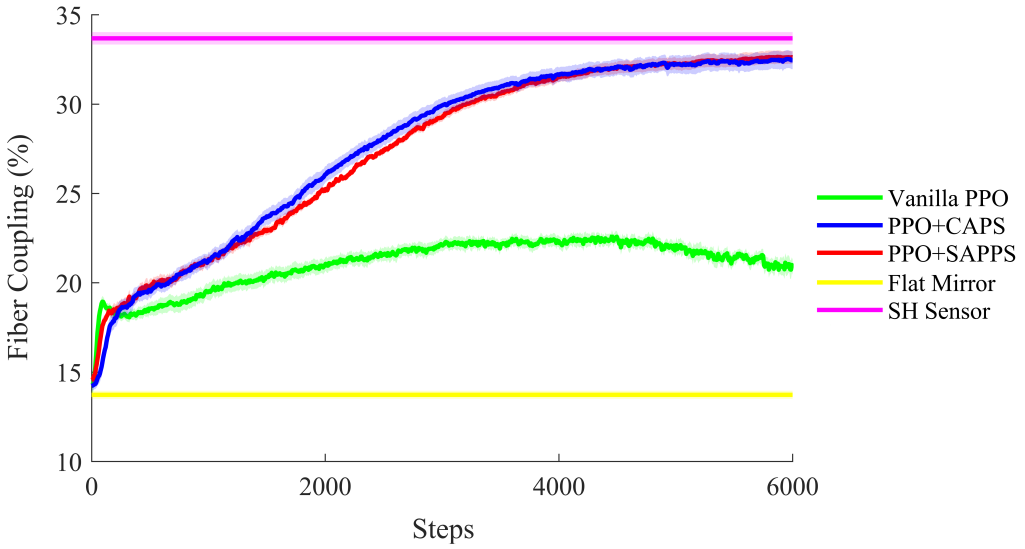

Figure 6.1: Comparison of the average fiber coupling (%) for Vanilla PPO, PPO+SAPPS, PPO+CAPS, Shack-Hartmann sensor, and flat mirror scenario, evaluated over 10 seeds from 50 distinct testing environments with $D/r_0 = 5$

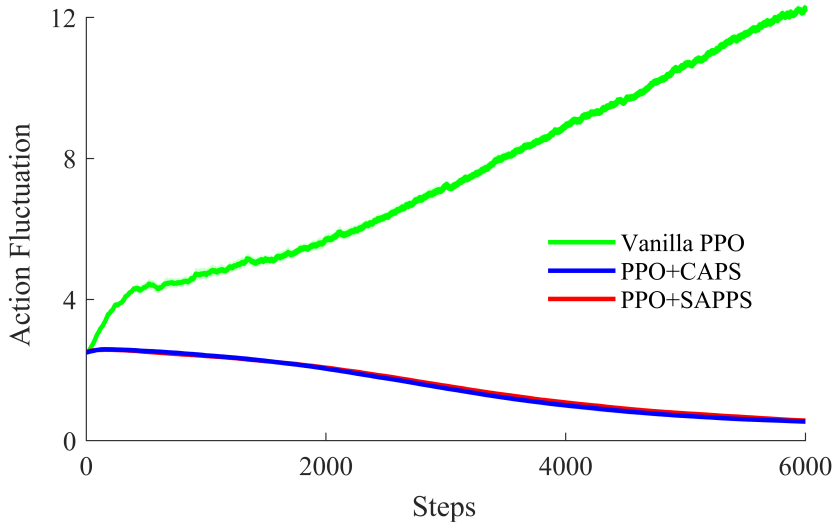

Figure 6.2: Comparison of the average action fluctuation for Vanilla PPO, PPO+SAPPS, PPO+CAPS, evaluated over 10 seeds from 50 distinct testing environments with $D/r_0 = 5$

scenario, is presented in Figure 6.3, while the comparison of action fluctuation is shown in Figure 6.4.

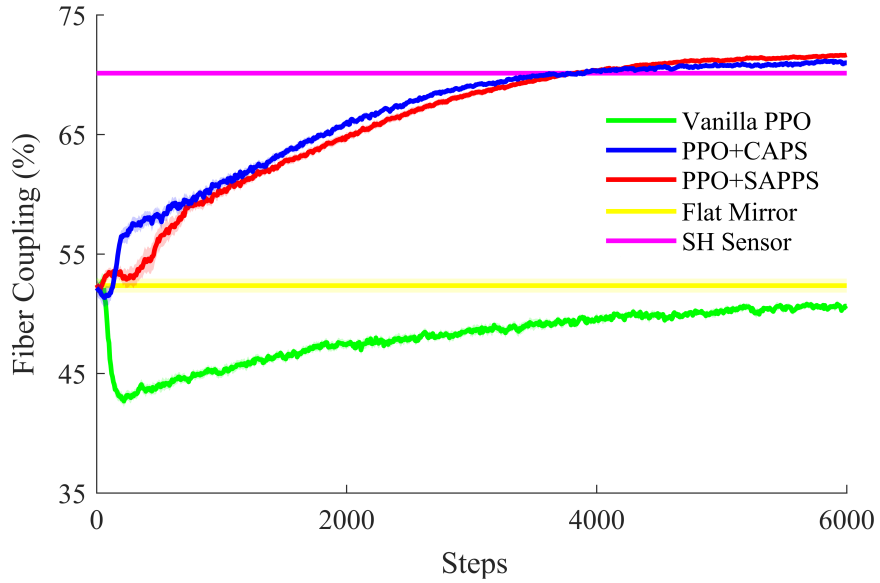

Figure 6.3: Comparison of the average fiber coupling (%) for Vanilla PPO, PPO+CAPS, PPO+SAPPS, Shack-Hartmann wavefront sensor, and a flat mirror, evaluated over 10 seeds from 50 distinct testing environments with $D/r_0 = 2$

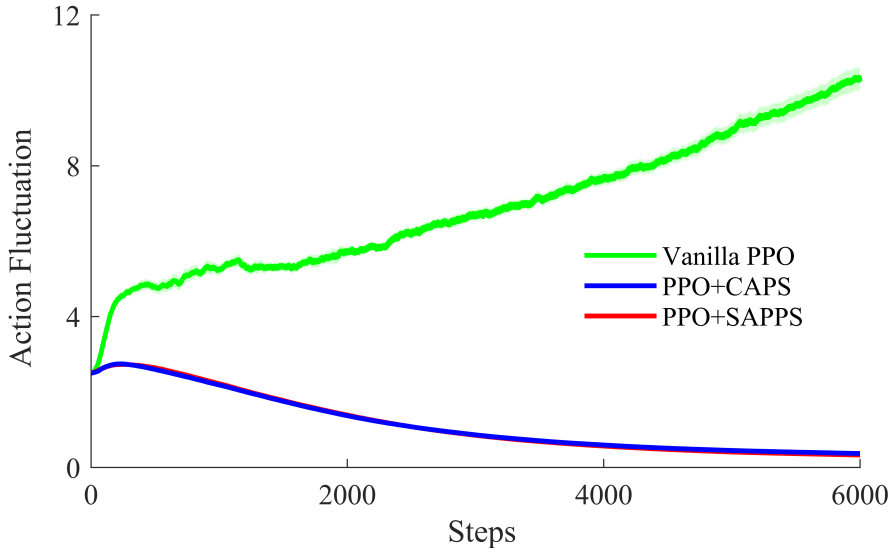

Figure 6.4: Comparison of the average action fluctuation for Vanilla PPO, PPO+CAPS, PPO+SAPPS, evaluated over 10 seeds from 50 distinct testing environments with $D/r_0 = 2$

These figures indicate that SAPPS and CAPS perform comparably, with SAPPS demonstrating slightly better results. As shown in Figure 6.3, both SAPPS and

CAPS outperform Vanilla PPO by approximately 41% and exceed the performance of the Shack-Hartmann sensor by approximately 2%. Additionally, Figure 6.4 shows that these methods produce smoother policies (lower action fluctuation) compared to Vanilla PPO, with an improvement of 97%.

In the quasi-static environment scenario, the similar performance of SAPPS and CAPS at $D/r_0 = 5$ and $D/r_0 = 2$ can be attributed to the fact that the environment remains unchanged when no action is taken. CAPS consistently operates by minimizing changes in subsequent actions, regardless of the setting. Similarly, in this setting, SAPPS reduces changes in subsequent actions because its decisions depend on time-contiguous states, which undergo minimal changes in a quasi-static environment. As a result, both approaches demonstrate almost similar performance under these conditions.

Since the SAPPS method integrated with the PPO algorithm outperformed the Vanilla PPO in both fiber coupling efficiency and policy smoothness (reduced action fluctuations) in the quasi-static environment—and performed comparably with the Shack-Hartmann wavefront sensor-based AO system—it is now appropriate to evaluate the proposed method in a dynamic environment that more closely reflects real-world conditions.

Non-Zero Drift Velocity Conditions

In a dynamic environment, the turbulence profile evolves over time during training, which simulates real-world operational conditions where a velocity field defines atmospheric motion. This setting reflects a scenario where the environment's state evolves over time even in the absence of actions, and the convergence of the RL model is slower than the atmospheric coherence time. Since $D/r_0 = 5$ corresponds to a stronger atmospheric turbulence than $D/r_0 = 2$, the value $D/r_0 = 5$ is selected for the experiments in the dynamic environment. The performance of the proposed SAPPS method integrated with the PPO algorithm is evaluated and compared against other benchmark approaches, including Vanilla PPO, PPO+CAPS, and PPO with the LipsNet neural network, across a range of atmospheric drift velocities from $5\,\mathrm{m\,s^{-1}}$ to $500\,\mathrm{m\,s^{-1}}$, as detailed in Section 6.2.1.

To ensure fair comparison across dynamic conditions, the hyperparameters of the Vanilla PPO algorithm, as well as those of the SAPPS, CAPS, and LipsNet policy smoothness methods, are retuned for different turbulence drift velocities. The search spaces used for these methods are summarized in Tables 6.5 and 6.3.

Hyperparameter	Search Space
Initial Lipschitz constant K_{init}	$0.1 \rightarrow 100$
Small constant ϵ	1×10^{-4}
Regularization coefficient λ	$1 \times 10^{-7} \rightarrow 1 \times 10^{-1}$
Activation function in $K(x)$	ReLU / Tanh
Hidden layers in $K(x)$	$[384, 256] \rightarrow [128, 64]$
Learning rate in $K(x)$	$1 \times 10^{-7} \rightarrow 1 \times 10^{-5}$

Table 6.5: Search space of hyperparameters explored for LipsNet neural network.

Table 6.2 presents the parameters that are common across all methods, while Table 6.6 lists the tuned hyperparameters for the dynamic scenarios.

Method	Hyperparameter	$v = 5\,\mathrm{m\,s^{-1}}$	$v = 50\,\mathrm{m\,s^{-1}}$	$v = 500\,\mathrm{m\,s^{-1}}$
PPO	lr	1×10^{-6}	1×10^{-6}	1×10^{-6}
	Clipping ϵ	0.10	0.05	0.05
	Buffer size	25,000	15,000	15,000
	Batch size	256	128	128
	Hidden layer size	480	320	320
	Frame stacking No.	9	7	7
	Flat scenario in episode config.	No	No	No
LipsNet	Initial Lipschitz constant K_{init}	3.0	0.3	3.0
	Small constant ϵ	1×10^{-4}	1×10^{-4}	1×10^{-4}
	Regularization coefficient λ	1×10^{-5}	1×10^{-7}	1×10^{-3}
	Activation function in $K(x)$	Tanh	Tanh	Tanh
	Hidden layers in $K(x)$	$[256, 128]$	$[192, 96]$	$[192, 96]$
	Learning rate in $K(x)$	1×10^{-7}	1×10^{-7}	1×10^{-7}
CAPS	Regularization coefficient λ_T	0.10	0.50	0.50
	Regularization coefficient λ_S	0.05	0.05	0.05
SAPPS	Regularization coefficient λ_T	0.05	0.075	0.05
	Homogeneous ratio c_{hm}	7.0	3.0	1.0

Table 6.6: Hyperparameters and their corresponding values in the dynamic environment for the AO system.

The methods were first evaluated under ***low-velocity conditions*** ($v = 5\,\mathrm{m\,s^{-1}}$). This drift velocity is chosen because it represents the lowest velocity commonly observed in real-world scenarios. These methods are compared based on fiber coupling efficiency, Shack–Hartmann-based fiber coupling efficiency, and action fluctuation. The Shack–Hartmann-based fiber coupling efficiency is defined as the ratio of the methods' fiber coupling efficiency to that achieved by the Shack-Hartmann wavefront sensor-based AO system. The corresponding comparison

plots are provided in Figures 6.5, 6.6, and 6.7, respectively.

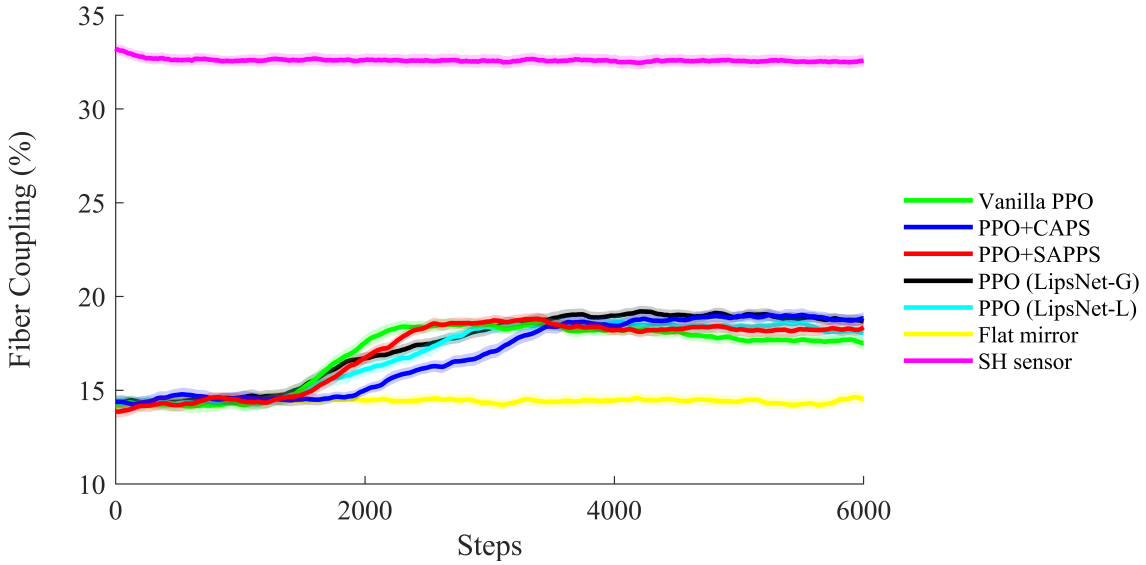

Figure 6.5: Comparison of the average fiber coupling (%) for Vanilla PPO, policy smoothness methods, Shack-Hartmann wavefront sensor, and flat mirror scenario, evaluated over 10 seeds from 50 distinct testing environments of $v = 5\,\mathrm{m\,s^{-1}}$

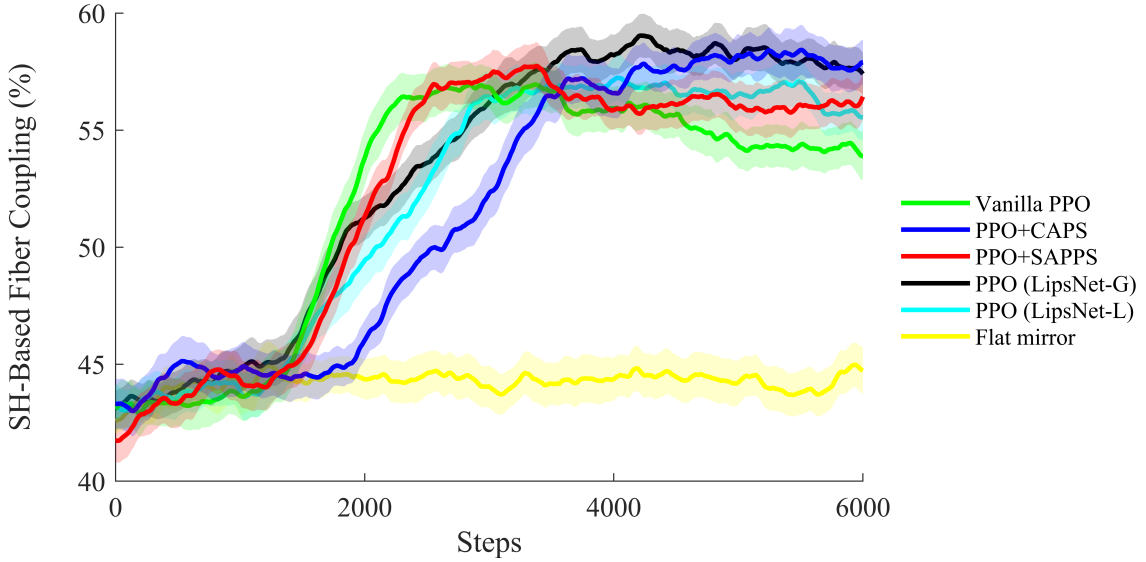

Figure 6.6: Comparison of the average Shack-Hartmann sensor-based fiber coupling for Vanilla PPO, policy smoothness methods, and flat mirror scenario, evaluated over 10 seeds from 50 distinct testing environments of $v = 5\,\mathrm{m\,s^{-1}}$

As shown in Figures 6.5 and 6.6, the policy smoothness methods outperform Vanilla PPO in maintaining fiber coupling efficiency and stability over time. In

contrast, Vanilla PPO (green) exhibits a gradual decline after initial improvement, indicating reduced robustness and a partial loss of coupling in the AO system. Maintaining stable coupling performance is essential for reliable optical communication, and the policy smoothness methods achieve this more effectively. Specifically, PPO+SAPPS (red) exhibits a rapid initial response and converges to a relatively stable value, maintaining consistent performance throughout the training period. PPO+CAPS (blue) shows a slower early response but achieves a higher final coupling efficiency. PPO (LipsNet-G) (black) and PPO (LipsNet-L) (cyan) also converge rapidly, with LipsNet-G achieving coupling efficiency slightly higher than SAPPS and maintaining stable performance thereafter. As shown in Figure 6.5, the proposed RL-based method achieves a fiber coupling efficiency of approximately 19% and outperforms the flat-mirror scenario (yellow). The Shack–Hartmann wavefront sensor remains the upper performance limit under the same conditions. The reasons for this performance gap are discussed in Section 6.3.

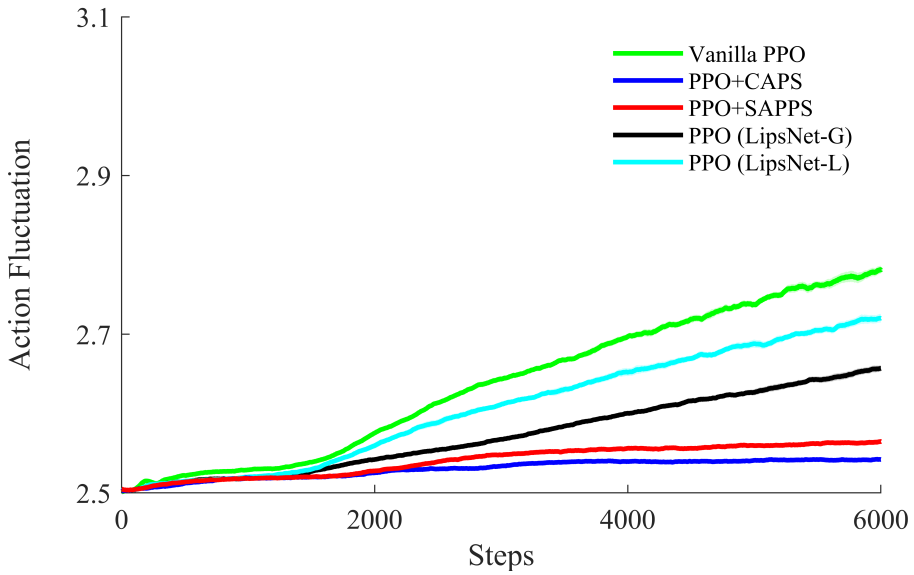

Figure 6.7: Comparison of the average action fluctuation for Vanilla PPO and policy smoothness methods, evaluated over 10 seeds from 50 distinct testing environments with $v = 5\,\mathrm{m\,s^{-1}}$

Also, as shown in Figure 6.7, all policy smoothness methods reduce action fluctuations relative to Vanilla PPO. Among them, SAPPS and CAPS achieve the most significant reductions, with CAPS exhibiting the lowest fluctuations overall. This suggests that at low atmospheric velocities, time-contiguous observations remain relatively close, which aligns with the CAPS assumption. Given that CAPS outper-

forms SAPPS under these conditions, it can be considered effective in low-dynamic environments where changes in observation are minimal.

Subsequently, the methods are evaluated under ***high-velocity conditions*** ($v = 50\,\mathrm{m\,s^{-1}}$). This drift velocity is chosen because it represents the high velocity commonly observed in real-world scenarios. The mentioned methods are compared with the Shack-Hartmann wavefront sensor-based AO and a flat mirror scenario. The comparison focuses on fiber coupling efficiency, SH-based fiber coupling efficiency, and action fluctuation, as shown in Figures 6.8, 6.9, and 6.10, respectively.

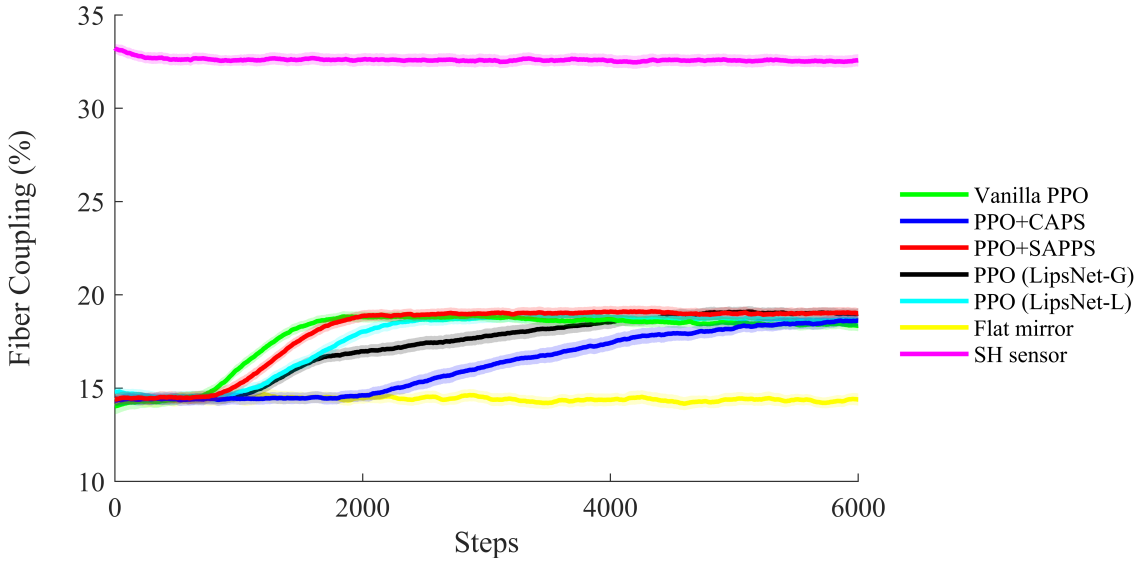

Figure 6.8: Comparison of the average fiber coupling (%) for Vanilla PPO, policy smoothness methods, Shack-Hartmann wavefront sensor, and flat mirror scenario, evaluated over 10 seeds from 50 distinct testing environments of $v = 50\,\mathrm{m\,s^{-1}}$

Figures 6.8 and 6.9 demonstrate that PPO+SAPPS (red) and PPO (LipsNet-G) (black) outperform Vanilla PPO and the other policy smoothness approaches in fiber coupling efficiency. PPO+SAPPS exhibits a rapid initial response and converges to the highest overall efficiency, demonstrating strong adaptability in a high drift velocity dynamic environment. PPO (LipsNet-G) achieves comparable performance with slightly slower convergence but maintains stable coupling thereafter. In contrast, PPO+CAPS (blue) achieves a coupling efficiency similar to Vanilla PPO but responds more slowly in the early steps. Meanwhile, Vanilla PPO (green) and PPO (LipsNet-L) show initial improvement followed by a gradual decline, indicating reduced stability under higher velocity dynamics. Additionally, Figure 6.10 shows that action fluctuations in PPO+CAPS and PPO+SAPPS are significantly lower than in Vanilla PPO and other policy smoothness methods, with

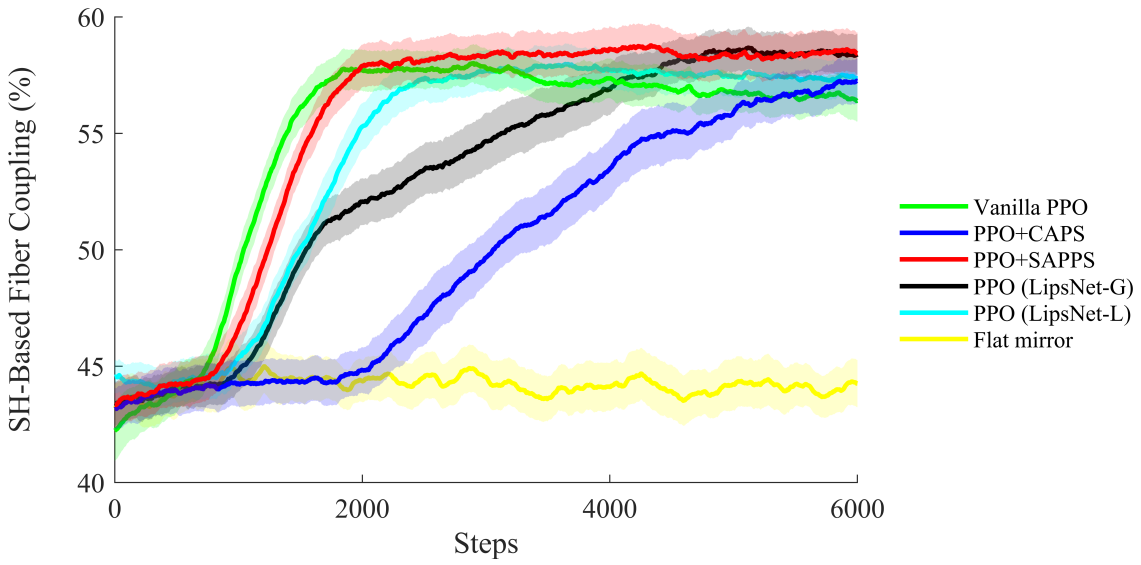

Figure 6.9: Comparison of the average Shack-Hartmann sensor-based fiber coupling (%) for Vanilla PPO, policy smoothness methods, and flat mirror scenario, evaluated over 10 seeds from 50 distinct testing environments of $v = 50\,\mathrm{m\,s^{-1}}$

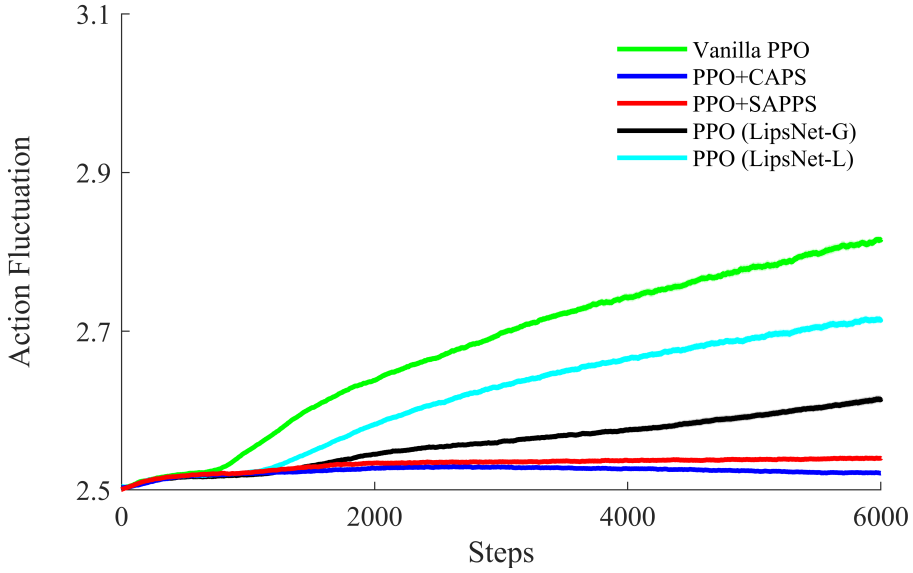

Figure 6.10: Comparison of the average action fluctuation for Vanilla PPO and policy smoothness methods, evaluated over 10 seeds from 50 distinct testing environments with $v = 50\,\mathrm{m\,s^{-1}}$

PPO+CAPS performing slightly better.

The results from PPO+CAPS indicate that while it maintains the smoothest policy, its performance in achieving high fiber coupling efficiency remains low. This is because, as atmospheric velocity increases, the differences between time-contiguous observations become more significant. Therefore, minimizing subsequent action changes is not beneficial.

Lastly, the methods are evaluated under ***extremely high-velocity conditions*** ($v = 500\,\mathrm{m\,s^{-1}}$), a drift velocity that is uncommon in real-world scenarios. The primary objective of this evaluation is to determine whether the methods can handle such extreme conditions or if they fail. In this evaluation, Vanilla PPO, along with policy smoothness methods, is compared against the Shack-Hartmann wavefront sensor-based AO and a flat mirror scenario. The comparison focuses on fiber coupling efficiency, Shack-Hartmann-based fiber coupling efficiency, and action fluctuation, as shown in Figures 6.11, 6.12, and 6.13, respectively.

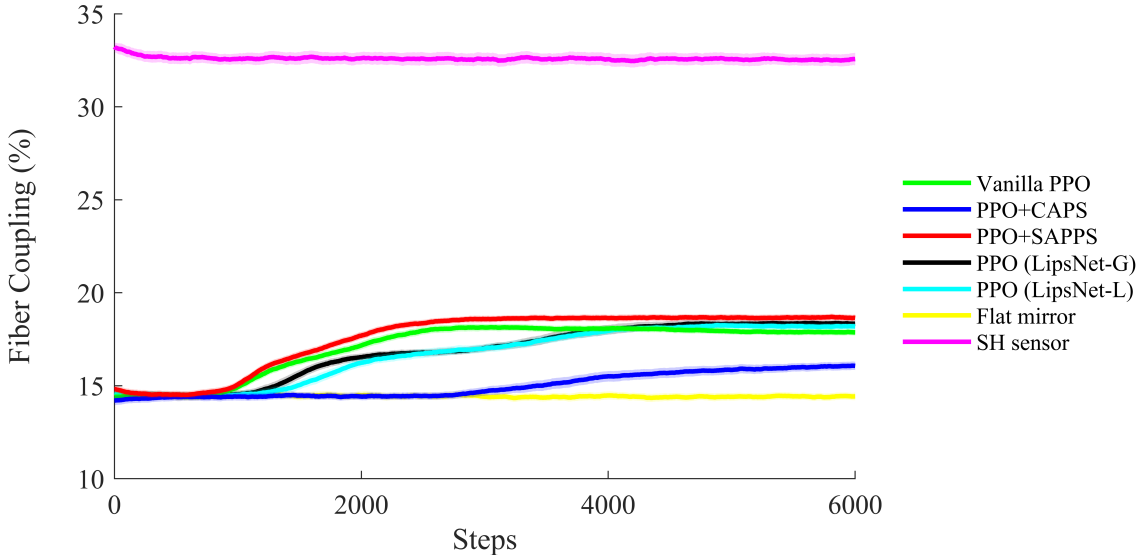

Figure 6.11: Comparison of the average fiber coupling (%) for Vanilla PPO, policy smoothness methods, Shack-Hartmann sensor, and flat mirror scenario, evaluated over 10 seeds from 50 distinct testing environments of $v = 500\,\mathrm{m\,s^{-1}}$

It can be seen from Figures 6.11 and 6.12 that even under extreme velocity conditions, PPO+SAPPS demonstrates a rapid initial response, converges to a stable fiber coupling efficiency, and outperforms both Vanilla PPO and other policy smoothness methods at all steps. While Vanilla PPO (green) initially improves, its performance degrades over time. Additionally, both PPO (LipsNet-G) (black) and PPO (LipsNet-L) (cyan) performed comparably, with slower initial responses

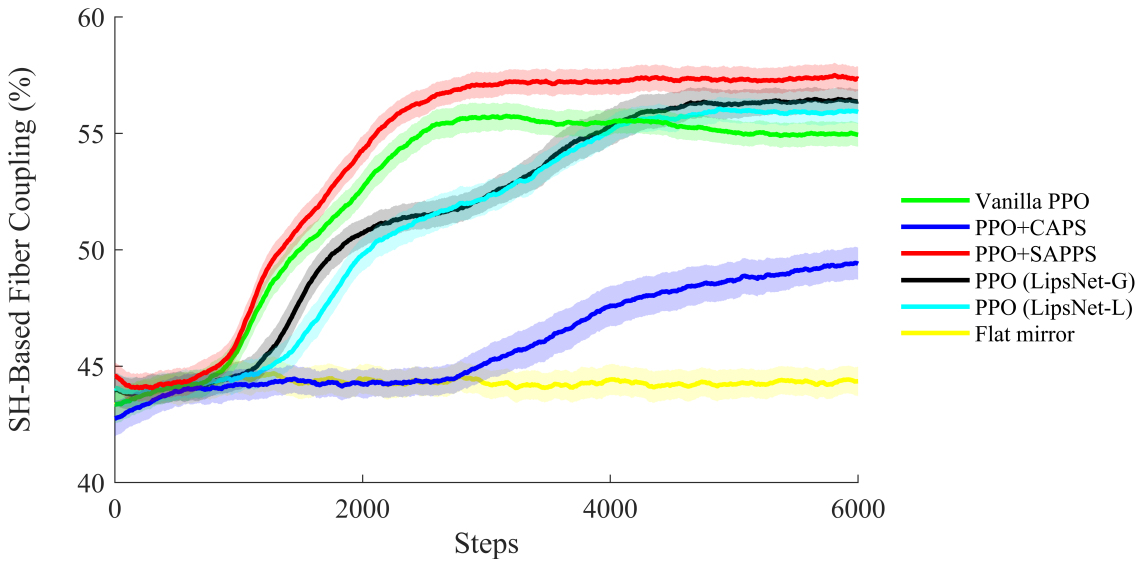

Figure 6.12: Comparison of the average Shack-Hartmann sensor-based fiber coupling (%) for Vanilla PPO, policy smoothness methods, and flat mirror scenario, evaluated over 10 seeds from 50 distinct testing environments of $v = 500\,\mathrm{m\,s^{-1}}$

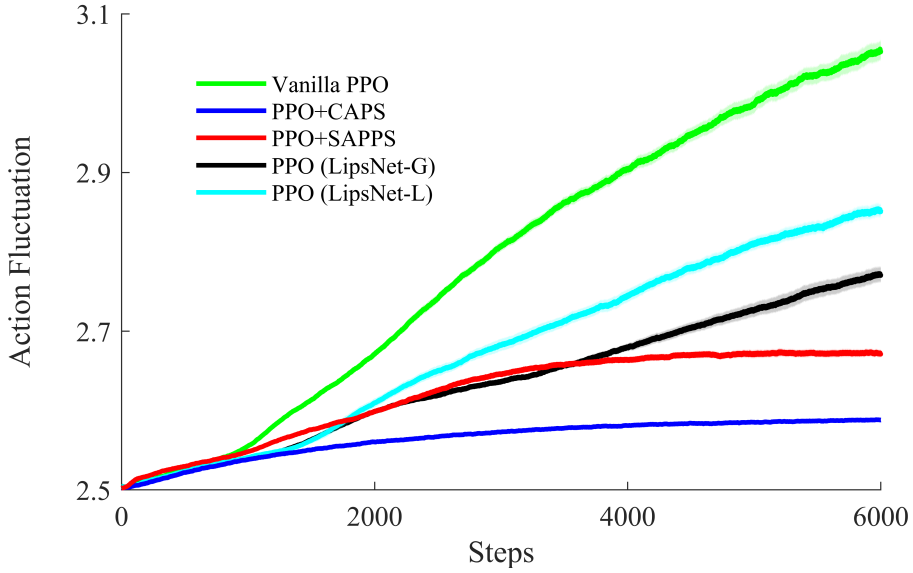

Figure 6.13: Comparison of the average action fluctuation for Vanilla PPO and policy smoothness methods, evaluated over 10 seeds from 50 distinct testing environments with $v = 500\,\mathrm{m\,s^{-1}}$

and lower coupling efficiencies than the PPO+SAPPS method. Meanwhile, CAPS (blue) shows a slower increase in efficiency, ultimately achieving a lower final performance compared to other methods.

Furthermore, Figure 6.13 shows that action fluctuations in policy smoothness methods are lower compared to Vanilla PPO, with CAPS achieving the smoothest policy. However, this increased policy smoothness comes at the cost of reduced coupling efficiency, as excessive minimization of changes in subsequent actions hinders adaptation to the dynamic environment. In contrast, SAPPS balances adaptation, which leads to superior performance.

Overall, as shown in Table 6.7, this section concludes that although PPO+CAPS outperforms Vanilla PPO at drift velocities of $5\,\mathrm{m\,s^{-1}}$ and $50\,\mathrm{m\,s^{-1}}$ and achieves the highest policy smoothness among all methods, it is still less effective than PPO+SAPPS. This limitation arises from the design of CAPS, which penalizes changes in consecutive actions without accounting for the magnitude of changes in temporally consecutive observations. When observation differences are large, restricting action changes—as CAPS does—can hinder the policy's responsiveness. As a result, the effectiveness of PPO+CAPS in maintaining coupling efficiency diminishes as atmospheric drift velocity increases. The LipsNet methods achieve better coupling efficiency than the CAPS method across different atmospheric velocities. This improvement arises from LipsNet's enforcement of Lipschitz continuity directly within the neural network architecture, which bounds the sensitivity of actions to changes in observations. However, the increased architectural complexity of LipsNet and the slower convergence of its Lipschitz module, due to a deliberately smaller learning rate, can result in a degraded initial response. Despite its advantage over CAPS in coupling efficiency, PPO (LipsNet) performs worse than PPO+SAPPS in policy smoothness across all velocities and coupling efficiency in high-velocity. This is because LipsNet constrains input-output sensitivity, rather than directly penalizing temporal action differences.

The advantage of the SAPPS method lies in its adaptability to the magnitude of changes in time-contiguous observations, which indirectly reflects the atmospheric velocity. This adaptive behavior prevents the method from penalizing large changes in consecutive actions when observation changes are large, and similarly, avoids enforcing unnecessary action changes when observation changes are minimal. As a result, SAPPS is well-suited for dynamic environments where observations may change either slowly or rapidly.

Env. / Method	$v = 5\,\mathrm{m\,s^{-1}}$		$v = 50\,\mathrm{m\,s^{-1}}$		$v = 500\,\mathrm{m\,s^{-1}}$	
	Coupling (%)	**Act. Fluc.**	**Coupling (%)**	**Act. Fluc.**	**Coupling (%)**	**Act. Fluc.**
Vanilla PPO	17.53 ± 1.06	2.78 ± 0.015	18.35 ± 0.92	2.82 ± 0.014	17.87 ± 0.53	3.05 ± 0.037
PPO+CAPS	18.83 ± 0.92	2.54 ± 0.003	18.58 ± 0.96	2.52 ± 0.008	16.07 ± 0.71	2.59 ± 0.009
PPO+SAPPS	18.35 ± 0.95	2.56 ± 0.005	18.99 ± 0.89	2.54 ± 0.009	18.65 ± 0.54	2.67 ± 0.014
PPO (LipsNet-G)	18.67 ± 0.93	2.65 ± 0.013	18.98 ± 0.89	2.61 ± 0.015	18.34 ± 0.53	2.77 ± 0.029
PPO (LipsNet-L)	18.07 ± 0.97	2.72 ± 0.017	18.65 ± 0.84	2.71 ± 0.014	18.19 ± 0.52	2.85 ± 0.025

Table 6.7: Comparison of average fiber coupling (%) and action fluctuation for methods across WSL-AO dynamic environment.

6.2.3 Sensitivity Analysis of the Proposed Method

This subsection examines the sensitivity of the SAPPS method to its two parameters: the regularization coefficient (λ_T) and the homogeneous ratio (c_{hm}). The analysis is performed in the dynamic environment with a drift velocity of $50\,\mathrm{m\,s^{-1}}$, using the tuned hyperparameters summarized in Tables 6.2 and 6.6 (column corresponding to $50\,\mathrm{m\,s^{-1}}$). The tuned values of the SAPPS parameters are c_{hm}=3.0 and λ_T=0.075.

For the sensitivity analysis of c_{hm}, the regularization coefficient is fixed at its tuned value of λ_T=0.075, while c_{hm} is varied as $\{1.5, 3.0, 6.0\}$. The results of this analysis are presented in Figure 6.14, where the left subplot illustrates the fiber coupling efficiency and the right subplot illustrates the corresponding action fluctuation.

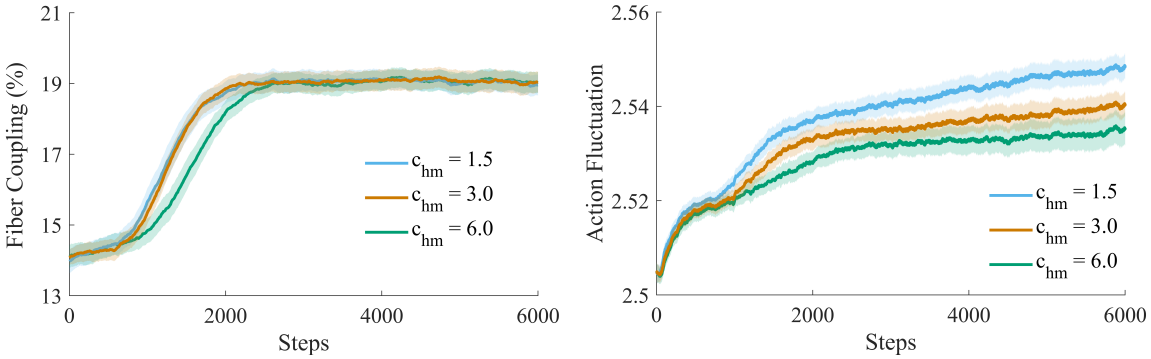

Figure 6.14: Sensitivity of the SAPPS method to the homogeneous ratio parameter c_{hm} at a fixed regularization coefficient of λ_T=0.075, evaluated across dynamic environments with a drift velocity of $v = 50\,\mathrm{m\,s^{-1}}$.

From Figure 6.14 and the definition of the SAPPS regularization term, which penalizes the ratio between action and observation changes, the influence of the

homogeneous ratio parameter (c_{hm}) becomes clear. As c_{hm} increases (green), the penalty on action changes strengthens while the relative emphasis on observation changes diminishes. Therefore, the controller becomes less responsive to changes in observation, leading to a slower initial response (green, left). However, this stronger penalization of action changes suppresses action fluctuations, resulting in smoother control behavior (green, right). In contrast, smaller c_{hm} values (blue) assign greater weight to observation changes and weaker penalty to action changes, making the controller more reactive and producing faster initial response. However, the reduced penalty on action changes comes at the cost of larger action fluctuations (blue, right). Overall, the intermediate value ($c_{\text{hm}} = 3.0$) (orange) provides a balanced trade-off, combining a relatively fast initial response with a moderate reduction in action fluctuations.

In addition, for the sensitivity analysis of λ_T, the homogeneous ratio is fixed at its tuned value of c_{hm}=3.0, while λ_T is varied as $\{0.0375, 0.075, 0.15\}$. The results of this analysis are presented in Figure 6.15, where the left subplot illustrates the fiber coupling efficiency and the right subplot illustrates the corresponding action fluctuation.

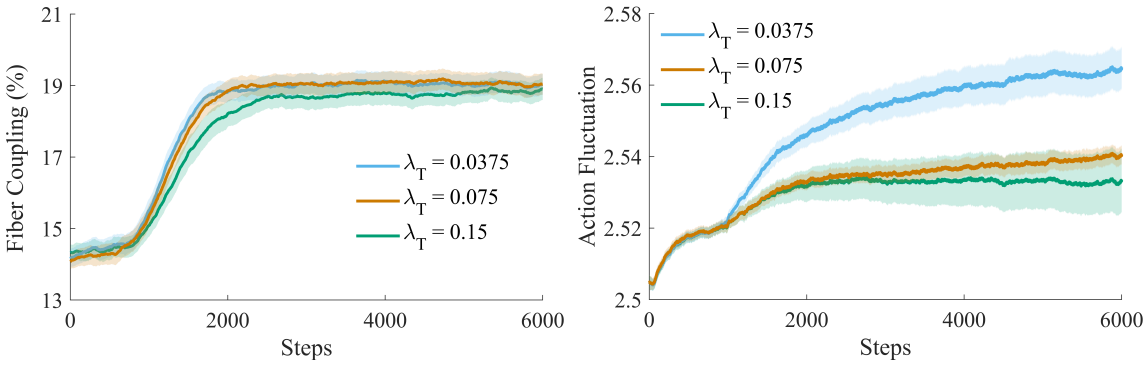

Figure 6.15: Sensitivity of the SAPPS method to the regularization coefficient λ_T at a fixed homogeneous ratio of c_{hm}=3.0, evaluated across dynamic environments with a drift velocity of $v = 50\,\text{m}\,\text{s}^{-1}$.

As shown in Figure 6.15, increasing λ_T strengthens the regularization effect, enforcing smoother actions and resulting in greater reduction of action fluctuations (green, right). However, stronger regularization also makes the controller less responsive to observation changes, leading to slightly lower fiber coupling efficiency (green, left). In contrast, smaller λ_T values (blue) apply weaker regularization, allowing the controller to react more aggressively to observation changes. Although this results in higher coupling efficiency, it comes at the cost of larger action fluctu-

ations. Overall, the intermediate setting of λ_T=0.075 (orange) achieves the highest coupling efficiency while maintaining moderate smoothness, representing the best trade-off between responsiveness and policy smoothness.

In summary, the sensitivity analyses of c_{hm} and λ_T demonstrate that the SAPPS method maintains stable performance across a reasonable range of parameter settings. Increasing either parameter improves policy smoothness by reducing action fluctuations, but it also decreases responsiveness to observation changes, leading to a reduction in fiber coupling efficiency. In contrast, smaller values result in weaker regularization and less effective reduction of action fluctuations. Although the method is not highly sensitive to parameter variations within the examined range, identifying the optimal pair (c_{hm}=3.0, λ_T=0.075) requires careful tuning to achieve the desired balance between smoothness and responsiveness.

6.3 Discussion

The proposed wavefront sensorless AO-RL environment is highly adaptable and easy to implement, which enables its applicability across various applications. The results of the experiments performed in this RL environment indicate that while the standard (Vanilla) PPO algorithm underperforms compared to the Shack-Hartmann wavefront sensor, the policy smoothness methods achieve performance comparable to the Shack-Hartmann wavefront sensor while maintaining high policy smoothness. This demonstrates that an RL-based wavefront sensorless AO system has significant potential in applications where the environment has a quasi-static condition. Particularly, in applications such as microscopy, ophthalmology, satellite-to-satellite communication, and GEO satellite-to-ground communication, where the atmosphere exhibits quasi-static behavior, this RL controller can be highly effective.

Policy smoothness methods integrated with the PPO algorithm perform comparably to the Shack-Hartmann wavefront sensor in a quasi-static atmosphere. Under dynamic atmospheric conditions, the proposed method a fiber coupling efficiency of about 19% and outperformed the flat-mirror scenario, although its performance remains below that of the Shack–Hartmann wavefront sensor. However, this comparison should be interpreted with caution, as the Shack-Hartmann sensor benefits from significantly more observational information, utilizing a high-resolution 128×128 pixel camera, whereas the RL-based controller relies on a 5×5 pixel photodetector. Despite having over 99% fewer pixels, the RL method consid-

ered here achieves over 50% of the Shack-Hartmann sensor's performance and outperforms a flat mirror scenario, which demonstrates satisfactory performance. 19% fiber coupling performance in a dynamic environment is still significantly below what is possible, which is over 70%, however this system delivers stable performance as a solid first step as this scenario represents a very challenging environment with very little environmental state information. In a quasi-static atmosphere, high performance can still be achieved over extended periods, even with lower-resolution sensors. This is because the atmospheric conditions remain static long enough for the corrections applied by the RL controller to take effect. As a result, the controller does not need to continuously adapt to rapidly changing turbulence. However, in a dynamic atmosphere, temporal variability adds complexity for RL controllers, which requires improved prediction and exploration.

It is important to emphasize that achieving high fiber coupling is not the only objective; maintaining stable coupling and achieving smooth control are equally critical. Smoothness is valuable not only for improving system performance but also as an important objective in itself. It is particularly valuable for real-world deployment, as it reduces excessive actuator usage, prevents large and abrupt control actions that the mirror's inertia cannot effectively follow, and minimizes wear and tear on actuators. As demonstrated in the dynamic environment evaluations, SAPPS exhibits a fast initial response and converges to a relatively stable coupling value with low action fluctuation, indicating the generation of an effectively smooth policy. Therefore, beyond improving fiber coupling performance, SAPPS contributes to safer, more efficient, and more reliable system operation than other approaches.

It can be observed from the results that as the RL controller learns, action fluctuation increases in the Vanilla PPO algorithm over time. This behavior may arise because, as the policy learns, it becomes more confident in selecting specific actions given observations, which leads to sharper transitions between actions. As a result, even minor variations in observations can lead to significant shifts in the selected actions, which result in noticeable fluctuations. This demonstrates a limitation of the Vanilla PPO algorithm in maintaining smooth policy updates. Therefore, policy smoothness methods are valuable as they mitigate such fluctuations by promoting lower variability in subsequent actions and improving overall policy smoothness.

As mentioned in the experimental setup (Section 6.2.1), in our simulations, we assumed a timestep of $\delta t = 1 \times 10^{-4}$ seconds to simulate the evolution of the

atmosphere under different drift speeds: $0\,\mathrm{m\,s^{-1}}$, $5\,\mathrm{m\,s^{-1}}$, $50\,\mathrm{m\,s^{-1}}$, and $500\,\mathrm{m\,s^{-1}}$. The policy network inference time is approximately 2.7×10^{-2} seconds, which is larger than the simulation timestep. This higher inference time is primarily due to the design of the Tianshou RL library, which prioritizes flexibility and modularity for algorithm development over real-time performance. As evident from its source code, Tianshou introduces additional computational overhead during inference by processing observations and actions through general-purpose interfaces intended for batch operations and experimentation. As a result, although Tianshou is highly effective for training, it leads to slower action evaluation times. To validate this, a PPO algorithm implemented from scratch for experiments described in Appendix D achieved a policy network inference time of approximately 1×10^{-3} seconds, confirming that the slowness is primarily due to the Tianshou library rather than the network architecture itself. Nevertheless, since the problem is formulated as an infinite-horizon task (with an episodic structure for computational practicality), the displacement of the atmosphere is of primary concern rather than the temporal control rate. For example, a timestep of $\delta t = 1 \times 10^{-4}$ seconds at $500\,\mathrm{m\,s^{-1}}$ results in the same atmospheric displacement as a timestep of $\delta t = 1 \times 10^{-3}$ seconds at $50\,\mathrm{m\,s^{-1}}$. Therefore, in practical deployments, the policy network could feasibly operate within relatively high drift velocities. Also, the inference delay could be mitigated through a more efficient Python implementation or by adopting JAX.

As discussed above, the policy network inference time is approximately 2.7×10^{-2} seconds, and each episode consists of 20 timesteps. Therefore, the expected total episode duration is approximately 0.54 seconds. However, the actual duration calculated per episode is approximately 1.9 seconds, which is 1.36 seconds higher than expected. This discrepancy mainly arises from the Tianshou RL library for reasons similar to those previously discussed, and the HCIPy package used in the proposed RL environment. First, the atmospheric models are generated based on Kolmogorov turbulence, which introduces additional computational overhead. Second, these models have a high spatial resolution, further increasing the computational load. Additionally, the deformable mirror and propagation libraries within HCIPy perform computationally intensive operations. Finally, the reward function calculations in the proposed RL environment may also contribute to the increased episode time.

The PPO+SAPPS method demonstrated satisfactory performance compared to the standard (Vanilla) PPO algorithm in both quasi-static and dynamic atmospheres. Also, in high-velocity dynamic atmospheres, PPO+SAPPS outperformed

other policy smoothness methods. However, as shown in Tables 6.4 and 6.6, the SAPPS method is more sensitive to its hyperparameters compared to the CAPS method. This sensitivity occurs because of the method's adaptability to variations in observations, which requires careful tuning when turbulence severity and velocity change. As discussed in Section 6.2.3, the two interdependent parameters, c_{hm} and λ_T, must be tuned simultaneously to achieve optimal performance.

6.4 Summary

This chapter presented a detailed evaluation of RL-based controllers developed for a simulated wavefront sensorless AO system. The objective was to achieve high and consistent fiber coupling efficiency while maintaining a low-cost system by using low-pixel-count photodetectors for LEO satellite-to-ground optical downlinks. The evaluation showed that the Vanilla PPO algorithm performed poorly under both quasi-static and dynamic atmospheric conditions, falling significantly short of the performance achieved by the Shack–Hartmann wavefront sensor-based system. This limitation was linked to the PPO policy's inability to maintain stable fiber coupling due to high-frequency action oscillations.

To mitigate action fluctuation, the State-Adaptive Proportional Policy Smoothing (SAPPS) method was proposed and integrated with the PPO algorithm. In quasi-static conditions, SAPPS outperformed Vanilla PPO in fiber coupling efficiency and policy smoothness and achieved performance comparable to the Shack–Hartmann wavefront sensor-based AO system. Additionally, both SAPPS and CAPS methods demonstrated superior performance over Vanilla PPO in experiments across various turbulence severities.

In dynamic environments, SAPPS consistently outperformed other methods across turbulence velocities ranging from $50\,\text{m}\,\text{s}^{-1}$ to $500\,\text{m}\,\text{s}^{-1}$, maintaining higher coupling efficiency while preserving smooth control. CAPS achieved the smoothest control signals but was less adaptable to rapidly changing conditions. SAPPS's strength lies in its ability to adapt to the magnitude of observation changes, avoiding the penalization of necessary action changes under dynamic conditions. This adaptive behavior enables SAPPS to remain responsive to environmental changes while reducing fluctuations in action.

Although the coupling efficiency remains below that of Shack–Hartmann based systems, primarily due to limitations in the RL system's photodetector, PPO+SAPPS achieved a consistant fiber coupling efficiency of about 19% while reducing action

fluctuation and outperformed the flat-mirror scenario. The results validate the potential of RL-based wavefront sensorless AO systems for real-world applications. Beyond coupling performance, achieving smooth control is also an important objective. PPO+SAPPS contributes to more stable and reliable system operation by maintaining stable fiber coupling with low action fluctuation. This reduces excessive actuator usage, prevents abrupt control actions, and minimizes wear and tear, making it particularly valuable for real-world deployment.

Chapter 7

Conclusion and Future Work

This thesis presents a cost-effective reinforcement learning (RL)-based approach for wavefront sensorless adaptive optics (AO) in optical satellite communication downlinks. To systematically evaluate RL algorithms, the first RL environment for optical satellite communication downlinks was developed. Using this environment, off-policy algorithms such as Soft Actor-Critic (SAC) and Deep Deterministic Policy Gradient (DDPG), as well as the on-policy Proximal Policy Optimization (PPO) algorithm, were implemented to optimize the coupling of 1550 nm light into a single-mode fiber under varying turbulence conditions. The RL controller learns an optimal policy through interaction with the environment, generating control signals using only limited photodetector data, thus eliminating the need for costly wavefront sensors and a complex AO system.

In preliminary experiments, among the tested RL algorithms, PPO demonstrated superior performance in a quasi-static atmosphere across various turbulence severities. However, like other RL-based controllers, PPO exhibited oscillatory control responses due to the high complexity of deep neural networks and their sensitivity to small perturbations in input observations. These high-frequency oscillations can lead to increased actuator usage and power consumption, which limits the practicality of RL-based AO for real-world applications.

Inspired by the application of an RL-based wavefront sensorless AO system, the State-Adaptive Proportional Policy Smoothing (SAPPS) method was introduced to address the challenge of high-frequency oscillations in standard RL algorithms. SAPPS mitigates these oscillations in RL controllers while improving overall performance. Integrated with Proximal Policy Optimization (PPO), SAPPS demonstrated superior results across multiple continuous control environments, including MuJoCo tasks, a real-world quadcopter hovering experiment, and a complex

wavefront sensorless AO system. The results show that SAPPS consistently outperformed vanilla PPO in both average reward and policy smoothness across the domains. Moreover, SAPPS performed comparably to the CAPS method and surpassed it in the quadcopter experiment. In the wavefront sensorless AO system, SAPPS achieved more stable and higher fiber coupling efficiency than the state-of-the-art CAPS and LipsNet methods (both integrated with PPO) at high drift velocities, which demonstrates its effectiveness in dynamic environments.

The applicability and performance of the proposed SAPPS method can be further examined and developed under broader and more challenging conditions, including integration with off-policy RL algorithms, abrupt environmental changes, and extreme or rare disturbances.

- The underlying concept of SAPPS is general and can be integrated with common deep RL algorithms. In this thesis, SAPPS is integrated into the on-policy PPO framework. However, on-policy algorithms are known to suffer from low sample efficiency. Although off-policy methods are typically more sample-efficient, in dynamic environments with rapid environmental changes, the experiences stored in the replay buffer can quickly become outdated. With a large buffer, the policy risks training on stale data, while with a small buffer, it may fail to capture sufficient variability. Therefore, future investigations should also examine the performance of the SAPPS method within off-policy algorithms.

- The robustness of the SAPPS method should be further examined under rare and abnormal fluctuations of large amplitude, such as extreme turbulence or sudden disturbances. While this thesis primarily focused on statistically typical models of environmental variability, future work could incorporate non-Gaussian or large amplitude random variations, as well as domain randomization with rare events, to evaluate the limits of SAPPS's robustness under such conditions.

- The SAPPS method deliberately avoids enforcing strict Lipschitz continuity constraints on the policy, as rigid bounds can hinder exploration and adaptability in continuous-control tasks. Instead, SAPPS promotes smooth policy behavior through adaptive proportional regularization rather than hard constraints. Nevertheless, future work could investigate hybrid formulations that integrate SAPPS with locally adaptive Lipschitz continuity bounds or

saturation limits, particularly for applications requiring stronger safety or stability guarantees.

Additionally, while this thesis has demonstrated the feasibility and advantages of RL-based wavefront sensorless AO for optical satellite communication downlinks, several key areas require further exploration to enhance performance, adaptability, and real-world applicability:

- One crucial direction is to refine the RL controller through more effective exploration strategies and reward functions to improve fiber coupling efficiency under dynamic turbulence conditions, thereby moving closer to the practical limit of 70%.

- Another important focus is on integrating imitation learning from Shack-Hartmann wavefront sensors. Although the RL-based AO system has been developed without direct supervision from traditional wavefront sensors, incorporating imitation learning could improve the learning process [239, 240].

- Leveraging offline learning to enhance online learning is also a valuable research direction. While online RL methods have proven effective, their dependence on continuous exploration can reduce efficiency in complex, rapidly changing environments. Incorporating offline learning, mainly through Shack-Hartmann wavefront sensor data, could enable the development of pre-trained models that improve online learning performance [241].

- Additionally, hierarchical learning could facilitate a unified AO system capable of managing quasi-static and dynamic turbulence conditions [242]. A hierarchical approach would allow an RL-based AO controller to adjust its strategies based on varying turbulence drift velocities and severities.

- Moreover, while the proposed RL environment leverages the HCIPy package to model key physical processes in AO with reasonable realism, future work could improve sim-to-real transferability by integrating more accurate and realistic atmospheric turbulence models, actuator dynamics, and sensor noise. This could be achieved by adopting more advanced simulation packages [243]. The RL environment could also be extended to include time-varying Fried parameters and to simulate a moving LEO satellite as it traverses the sky.

- Finally, a critical next step is to test the proposed RL controller in an optical setup and compare its performance against a traditional wavefront sensor-based AO system.

Appendix A

Pseudocode of Reinforcement Learning Algorithms

A.1 Soft Actor-Critic (SAC)

Algorithm 2 SAC algorithm

Input: ϕ, θ_1, θ_2 ▷ Initial policy and Q-function parameters
$\bar{\theta}_1 \leftarrow \theta_1$, $\bar{\theta}_2 \leftarrow \theta_2$ ▷ Initialize Q-function target network weights
$D \leftarrow \varnothing$ ▷ Initialize empty replay buffer
for each iteration **do**
    for each environment step **do**
        Observe state s
        $a \sim \pi_\phi(\cdot|s)$ ▷ Select action from the policy
        Execute action (a) in the env.
        $s' \sim P(s'|s,a)$ ▷ Sample transition from the env.
        $D \leftarrow D \cup \{(s,a,r(s,a),s')\}$ ▷ Store the transition in the replay buffer
    end for
    for each gradient step **do**
        $B = \{(s,a,r,s')\} \sim D$ ▷ Randomly sample a batch of transitions
        $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$ for $i \in \{1,2\}$ ▷ Update the Q-function parameters
        $\phi \leftarrow \phi - \lambda_\phi \hat{\nabla}_\phi J_\pi(\phi)$ ▷ Update the policy parameters
        $\alpha \leftarrow \alpha - \lambda \hat{\nabla}_\alpha J(\alpha)$ ▷ Temperature Adjustment
        Update target networks for $i \in \{1,2\}$

$$\bar{\theta}_i \leftarrow \rho\, \bar{\theta}_i + (1-\rho)\theta_i$$

    end for
end for
Output: ϕ, θ_1, θ_2 ▷ Optimized parameters

A.2 Deep Deterministic Policy Gradient (DDPG)

Algorithm 3 DDPG algorithm

Input: ϕ, θ ▷ Initial policy and Q-function parameters
$\bar{\theta} \leftarrow \theta$, $\bar{\phi} \leftarrow \phi$ ▷ Initialize Q-function target network weights
$D \leftarrow \varnothing$ ▷ Initialize empty replay buffer
for each iteration **do**
  for each environment step **do**
    Observe state s
    $a \sim \mu_\phi(s) + \epsilon$ ▷ Select action where $\epsilon \in \mathcal{N}$
    Execute action (a) in the env.
    $s' \sim P(s'|s,a)$ ▷ Sample transition from the env.
    $D \leftarrow D \cup \{(s,a,r(s,a),s')\}$ ▷ Store the transition in the replay buffer
  end for
  for each gradient step **do**
    $B = \{(s,a,r,s')\} \sim D$ ▷ Randomly sample a batch of transitions
    Compute Q-function target

$$y(r,s') = r + \gamma\, Q_{\bar{\theta}}(s', \mu_{\bar{\phi}}(s'))$$

    Update the Q-function parameters

$$\nabla_\theta \frac{1}{|B|} \sum_{(\boldsymbol{s},\boldsymbol{a},r,\boldsymbol{s}')\in B} \big(Q_\theta(\boldsymbol{s},\boldsymbol{a}) - y(r,\boldsymbol{s}')\big)^2$$

    Update the policy weights

$$\nabla_\phi \frac{1}{|B|} \sum_{s\in B} \big(Q_\theta(s, \mu_\phi(s))\big)$$

    Update target networks

$$\bar{\theta} \leftarrow \rho\bar{\theta} + (1-\rho)\theta$$
$$\bar{\phi} \leftarrow \rho\bar{\phi} + (1-\rho)\phi$$

  end for
end for
Output: ϕ, θ ▷ Optimized parameters

A.3 Proximal Policy Optimization (PPO)

Algorithm 4 PPO algorithm

Input: ϕ, θ ▷ Initial policy and value function parameters
for each iteration **do**
  for T time steps **do**
    Observe state s
    $a \sim \pi_{\phi_{old}}(.|s)$ ▷ Select action from the policy
    Execute action (a) in the env.
    $s' \sim P(s'|s,a)$ ▷ Sample transition from the env.
    $D \leftarrow D \cup \{\tau\} = \{(s,a,r,s')\}$ ▷ Collect the set of trajectories
    $\hat{\mathbb{A}} \sim Q_\theta(a,s) - V_\theta(s)$ ▷ Compute the advantage estimate
  end for
  for $k = 1, 2, .., N$ **do**
    Update the policy by maximizing the PPO-Clip objective:

$$\phi_{k+1} = \arg\max_{\phi} \frac{1}{|D_k|T} \sum_{\tau \in D_k} \sum_{t=0}^{T} \min\left(r_t(\phi)\hat{\mathbb{A}}_t^{\pi_{\phi_k}}, \text{clip}(r_t(\phi), 1-\epsilon, 1+\epsilon)\hat{\mathbb{A}}_t^{\pi_{\phi_k}}\right)$$

    Fit the value function

$$\theta_{k+1} = \arg\min_{\theta} \frac{1}{|D_k|T} \sum_{\tau \in D_k} \sum_{t=0}^{T} \left(V_\theta(s_t) - \hat{R}_t\right)^2$$

  end for
end for
Output: ϕ, θ ▷ Optimized parameters

Appendix B

Analyzing Policy Smoothness in MuJoCo Environments

This appendix presents the hyperparameters and plot results comparing the policy smoothness measure and average return of Vanilla PPO, PPO+SAPPS, and PPO+CAPS in MuJoCo environments. The corresponding performance plots are directly related to the discussion presented in Section 4.1. Table B.1 summarizes the common PPO hyperparameter settings used across all MuJoCo environments examined.

Parameter	Value
Optimizer	Adam
Activation function	Tanh
Clip range ϵ	0.2
Max gradient norm	0.5
Number of epochs	300
Steps per epoch	30,000
Steps per iteration	2,048
Network updates per iteration	10
Discount factor γ	0.99
GAE lambda λ	0.95
Value function loss coefficient	0.25
Replay buffer capacity	4,096
Advantage normalization	False
Reward normalization	True
Training / Testing environments	8 / 10

Table B.1: Common PPO hyperparameter settings used across MuJoCo environments

B.1 Walker2D-v4 Environment

Hyperparameters:

Method	Hyperparameter	Value
PPO	Learning rate lr	1×10^{-4}
	Batch size	64
	Actor/Critic hidden layer size	$[96, 96]$
CAPS	Regularization coefficient λ_T	1.0
	Regularization coefficient λ_S	0.25
SAPPS	Regularization coefficient λ_T	0.4
	Homogeneous ratio c_{hm}	25

Table B.2: Hyperparameter settings for PPO, CAPS, and SAPPS in Walker2D environment

Comparison of Average Returns:

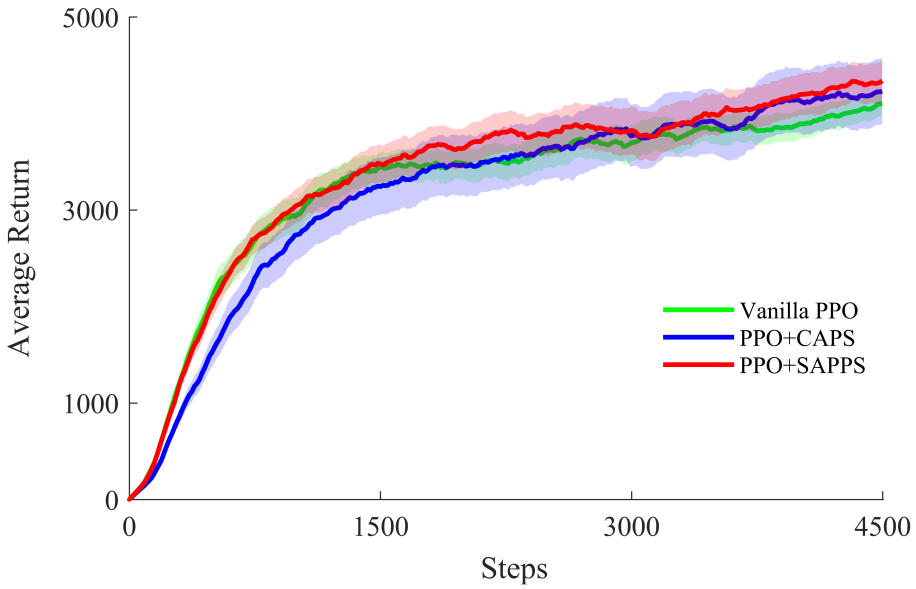

Figure B.1: Comparison of the average returns of policy smoothness methods and Vanilla PPO, evaluated using 15 different seeds per method in the Walker2D environment

Comparison of Smoothness Measure (Sm):

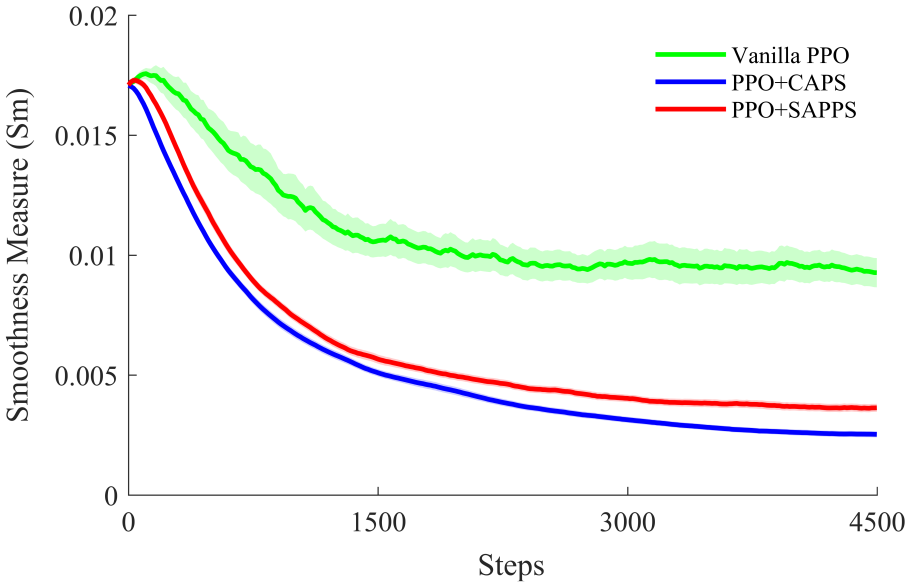

Figure B.2: Comparison of the smoothness measure (Sm) of policy smoothness methods and Vanilla PPO, evaluated using 15 different seeds per method in the Walker2D environment

B.2 Reacher-v4 Environment

Hyperparameters:

Method	Hyperparameter	Value
PPO	Learning rate lr	3×10^{-4}
	Batch size	64
	Actor/Critic hidden layer size	$[80, 80]$
CAPS	Regularization coefficient λ_T	1×10^{-1}
	Regularization coefficient λ_S	5×10^{-2}
SAPPS	Regularization coefficient λ_T	1×10^{-3}
	Homogeneous ratio c_{hm}	1×10^{2}

Table B.3: Hyperparameter settings for PPO, CAPS, and SAPPS in the Reacher environment

Comparison of Average Returns:

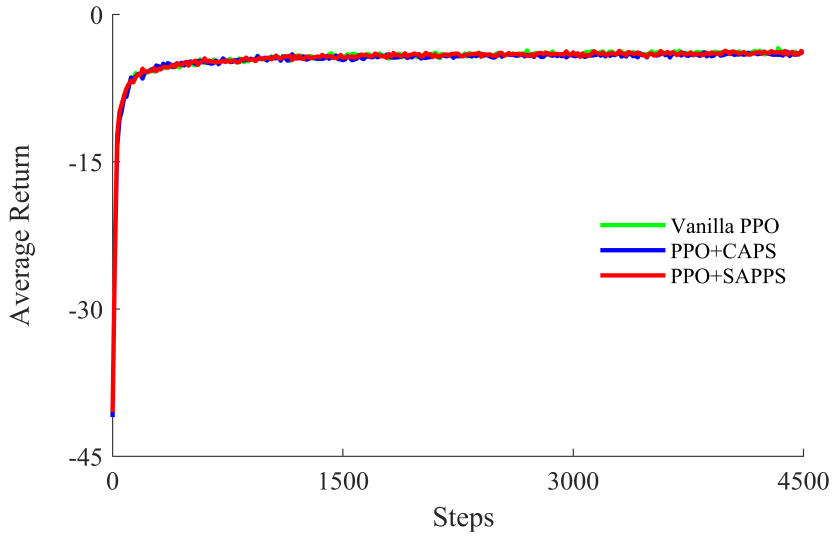

Figure B.3: Comparison of the average returns of policy smoothness methods and Vanilla PPO, evaluated using 20 different seeds per method in the Reacher environment

Comparison of Smoothness Measure (*Sm*):

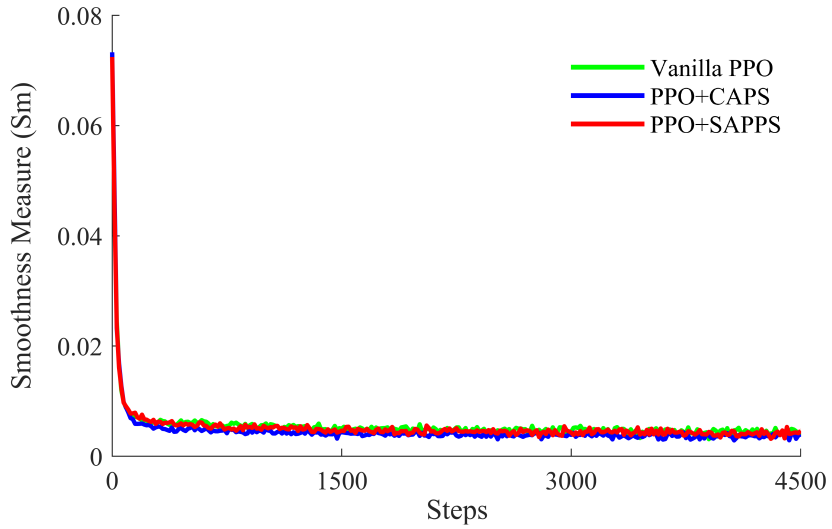

Figure B.4: Comparison of the smoothness measure (*Sm*) of policy smoothness methods and Vanilla PPO, evaluated using 20 different seeds per method in the Reacher environment

B.3 HalfCheetah-v4 Environment

Hyperparameters:

Method	Hyperparameter	Value
PPO	Learning rate lr	5×10^{-5}
	Batch size	32
	Actor/Critic hidden layer size	$[96, 96]$
CAPS	Regularization coefficient λ_T	1×10^{-3}
	Regularization coefficient λ_S	5×10^{-1}
SAPPS	Regularization coefficient λ_T	1×10^{-1}
	Homogeneous ratio c_{hm}	1×10^{3}

Table B.4: Hyperparameter settings for PPO, CAPS, and SAPPS in the Half Cheetah environment

Comparison of Average Returns:

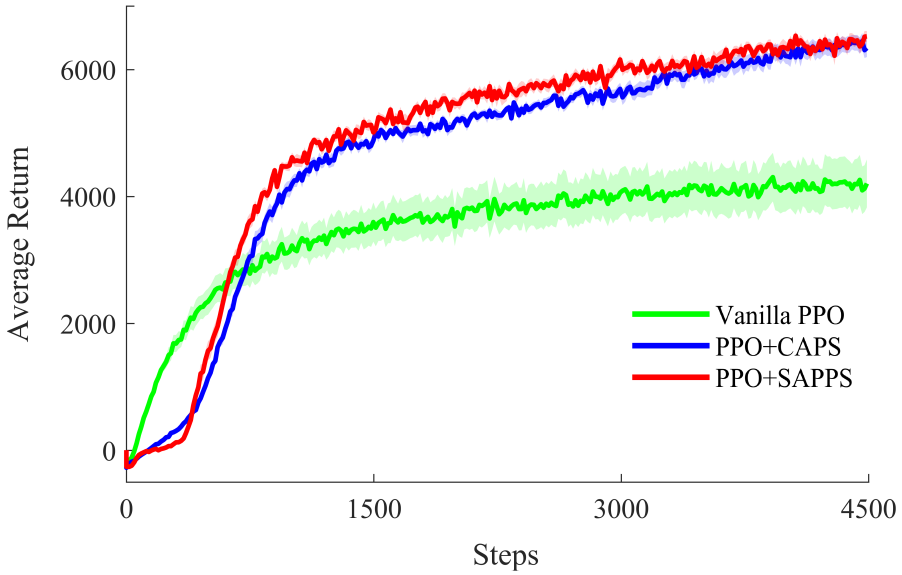

Figure B.5: Comparison of the average returns of policy smoothness methods and Vanilla PPO, evaluated using 20 different seeds per method in the Half Cheetah environment

Comparison of Smoothness Measure (Sm):

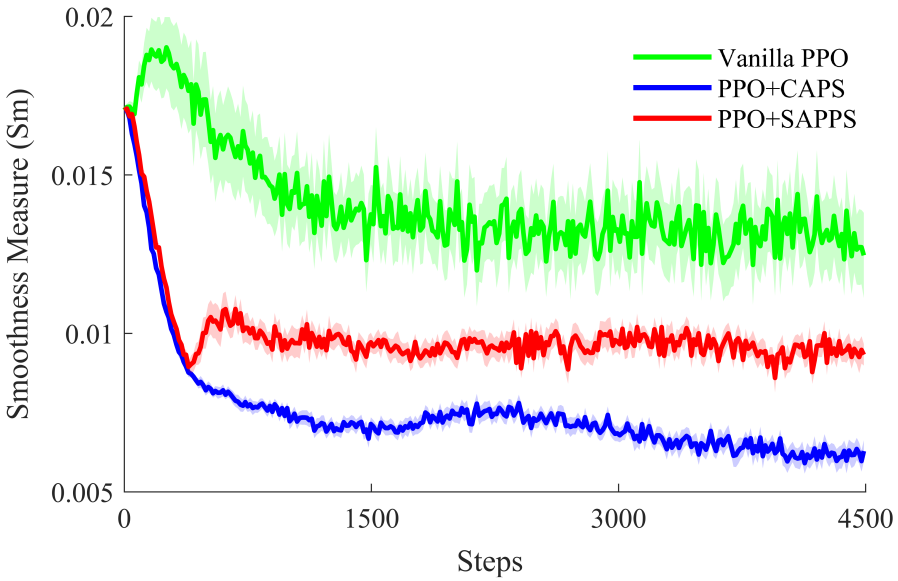

Figure B.6: Comparison of the smoothness measure (Sm) of policy smoothness methods and Vanilla PPO, evaluated using 20 different seeds per method in the Half Cheetah environment

B.4 Swimmer-v4 Environment

Hyperparameters:

Method	Hyperparameter	Value
PPO	Learning rate lr	5×10^{-4}
	Batch size	64
	Actor/Critic hidden layer size	$[80, 80]$
CAPS	Regularization coefficient λ_T	1×10^{-3}
	Regularization coefficient λ_S	5×10^{-3}
SAPPS	Regularization coefficient λ_T	1×10^{-4}
	Homogeneous ratio c_{hm}	5×10^{1}

Table B.5: Hyperparameter settings for PPO, CAPS, and SAPPS in the Swimmer environment

Comparison of Average Returns:

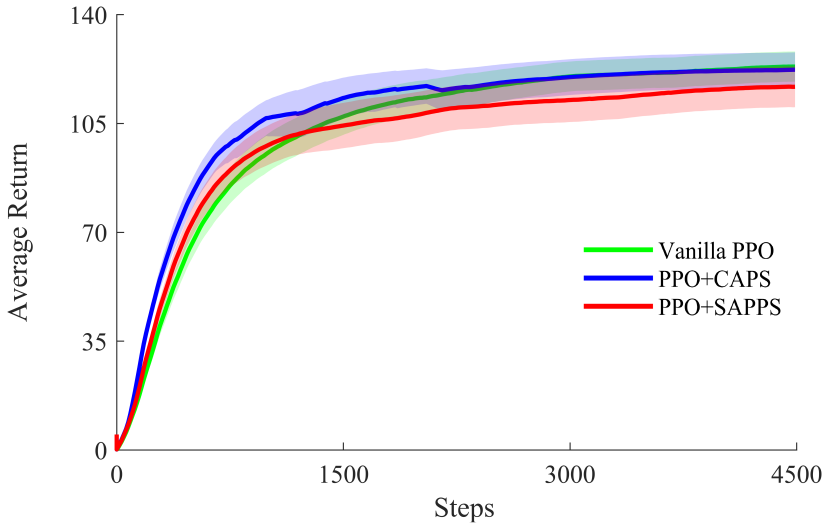

Figure B.7: Comparison of the average returns of policy smoothness methods and Vanilla PPO, evaluated using 20 different seeds per method in the Swimmer environment

Comparison of Smoothness Measure (*Sm*):

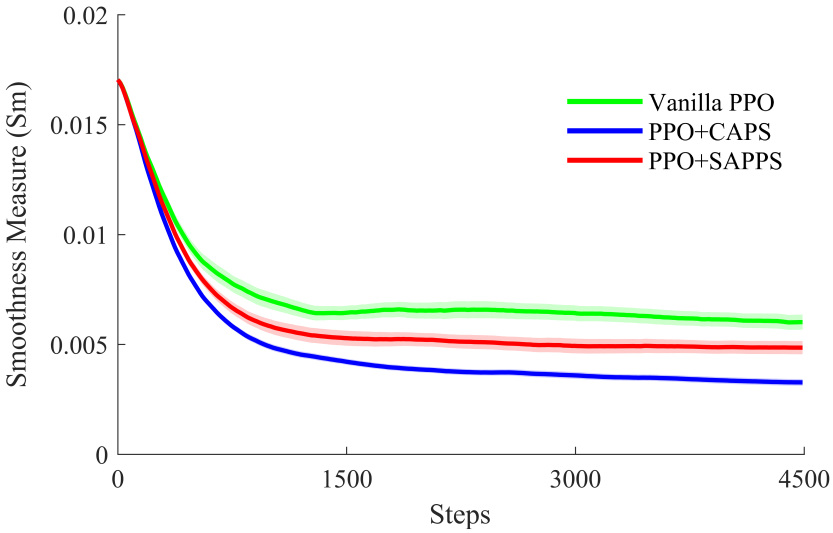

Figure B.8: Comparison of the smoothness measure (*Sm*) of policy smoothness methods and Vanilla PPO, evaluated using 20 different seeds per method in the Swimmer environment

B.5 Ant-v4 Environment

Hyperparameters:

Method	Hyperparameter	Value
PPO	Learning rate lr	5×10^{-5}
	Batch size	64
	Actor/Critic hidden layer size	$[112, 112]$
CAPS	Regularization coefficient λ_T	1×10^{-1}
	Regularization coefficient λ_S	5×10^{-1}
SAPPS	Regularization coefficient λ_T	5×10^{-1}
	Homogeneous ratio c_{hm}	5×10^{1}

Table B.6: Hyperparameter settings for PPO, CAPS, and SAPPS in the Ant environment

Comparison of Average Returns:

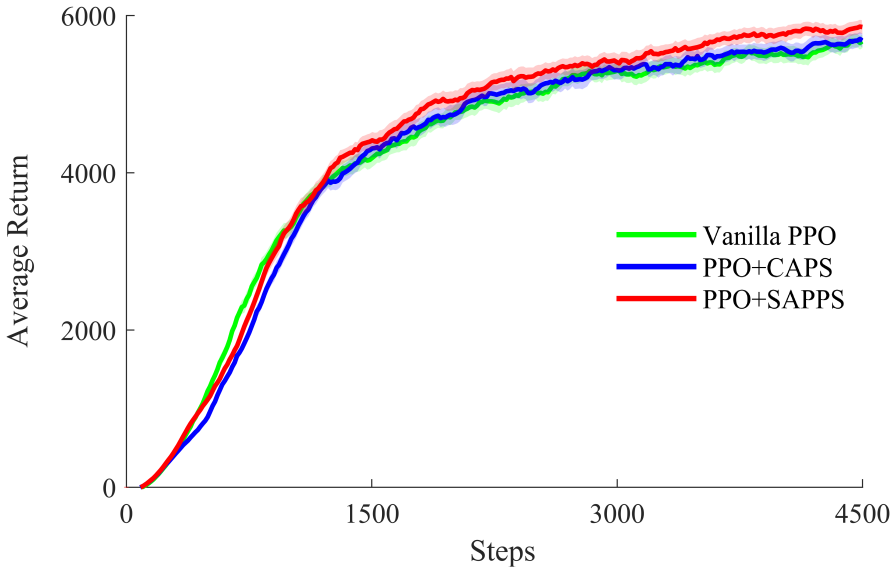

Figure B.9: Comparison of the average returns of policy smoothness methods and Vanilla PPO, evaluated using 20 different seeds per method in the Ant environment

Comparison of Smoothness Measure (*Sm*):

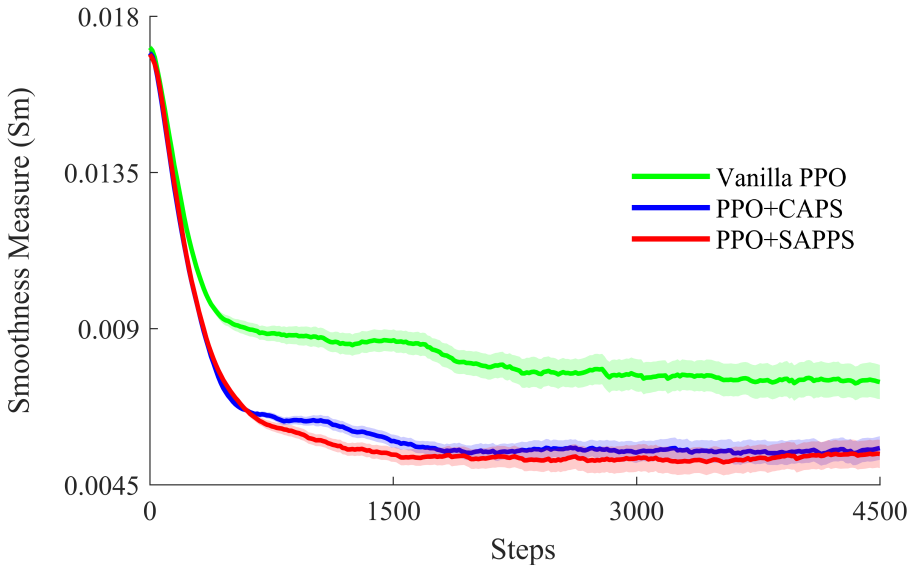

Figure B.10: Comparison of the smoothness measure (*Sm*) of policy smoothness methods and Vanilla PPO, evaluated using 20 different seeds per method in the Ant environment

Appendix C

Preliminary Tuning of RL Controllers for Wavefront Sensorless AO Systems

In the preliminary experiments on the quasi-static environment in a wavefront sensorless AO system, the hyperparameters were tuned for each RL algorithm employed. This tuning process corresponds to the preliminary experiments and results discussed in Section D.

For the quasi-static environment, the configuration consisted of a 2×2 pixel observation space, a 64-dimensional action space based on the Disk Harmonic basis function, and a Strehl ratio reward function r_{SR} (Eq. 5.1). A parametric study was conducted on the episode length, varying it from 10 to 100 time steps. The findings indicate that adopting an episode length of $20 - 30$ time steps is sufficient to achieve optimal actions. During these preliminary experiments, a single atmospheric condition was used for each policy.

C.1 Soft Actor-Critic (SAC)

The SAC algorithm is described in detail in Section 2.4.1. While SAC has shown promising results in various domains, a major drawback is its sensitivity to the choice of the temperature parameter α_t and the intuitively selected target entropy parameter. These parameters are critical to the algorithm's performance, and their selection can significantly impact the results.

In the preliminary assessment, SAC was evaluated under three different temperature (α_t) settings: *fixed*, *learned*, and *semi-learned*. For the *fixed temperature* setting, the best performance was achieved at a constant value of $\alpha_t = 0.4$, within a

possible range of 0 to 1. In the *learned temperature* setting, α_t was optimized using a learning rate of α_t-lr $= 10^{-1}$, selected from a range spanning 1×10^{-6} to 5×10^{-1}. Similarly, in the *semi-learned temperature* setting, optimization was performed with the same learning rate but with a minimum α_t value of 0.4.

The hyperparameters selected for the SAC algorithm, which is tuned for a quasi-static environment, are summarized in Table C.1.

Hyperparameter	Value
Batch size	128
Actor-lr	5×10^{-4}
Critic-lr	1×10^{-2}
Actor-Hidden dim.	150
Critic-Hidden dim.	80
Temp. α_t-lr	1×10^{-1}
Temp. α_t-min limit	0.4
No episodes per iteration	1
No updates per iteration	20
Polyak (ρ)	0.99
Discount (γ)	0.95
Reward scaling	No
learned α_t	*Semi*

Table C.1: SAC hyperparameters and corresponding values in quasi-static environment

Figure C.1 shows that the fixed and semi-learned temperature settings respond more quickly than the purely learned setting. Additionally, the semi-learned setting outperforms the others. Therefore, the semi-learned temperature setting was selected for all subsequent preliminary experiments in the SAC approach.

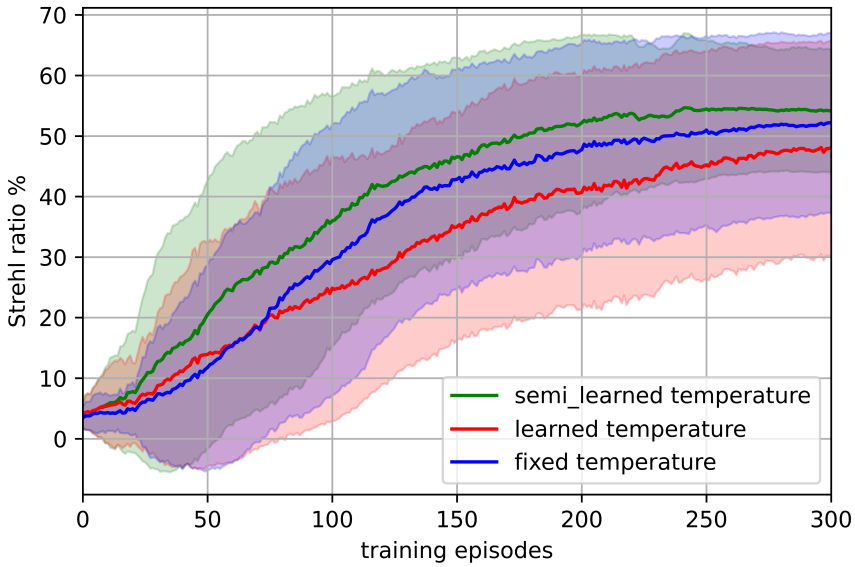

Figure C.1: Comparison of the selection of the temperature (α_t) in SAC applied on 20 randomly selected quasi-static atmospheric turbulences of $D/r_0 = 5$. Note that the shaded regions extend to negative values of the Strehl ratio because the standard deviation may be larger than the mean, represented by the solid curves

C.2 Deep Deterministic Policy Gradient (DDPG)

The DDPG algorithm is described in detail in Section 2.4.2. While the DDPG algorithm has the advantage of ease of implementation, it can be sensitive to the choice of hyperparameters and can be prone to instability due to the choice of the reward function [244].

In the preliminary assessment, the effects of reward normalization were evaluated by comparing three approaches: mean–standard deviation normalization (μ-σ norm), min-max normalization, and no normalization. Reward normalization scales reward values across episodes, which has been shown to help the model more effectively identify actions that lead to higher rewards. This process accelerates the algorithm's convergence. In addition, without normalization, the variance in rewards tends to decrease as the model approaches convergence, making it harder for the model to adjust itself. By normalizing rewards, the model can better recognize these reward differences and continue making adjustments.

The hyperparameters selected for the DDPG algorithm, which is tuned for a quasi-static environment, are summarized in Table C.2.

Figure C.2 shows the impact of reward normalization on training performance. The results show that avoiding reward normalization leads to slower convergence

Hyperparameter	Value
Batch size	256
Actor-lr	5×10^{-5}
Critic-lr	1×10^{-2}
Actor-Hidden dim.	250
Critic-Hidden dim.	65
No episodes per iteration	2
No updates per iteration	20
Polyak (ρ)	0.99
Discount (γ)	0.95
Reward scaling	μ-σ norm

Table C.2: DDPG hyperparameters and corresponding values in quasi-static environment

and lower overall performance. While both min-max normalization and mean–standard deviation normalization yield similar trends, the mean–standard deviation achieves a higher final Strehl ratio. Therefore, mean–standard deviation normalization is chosen for subsequent preliminary experiments in the DDPG approach.

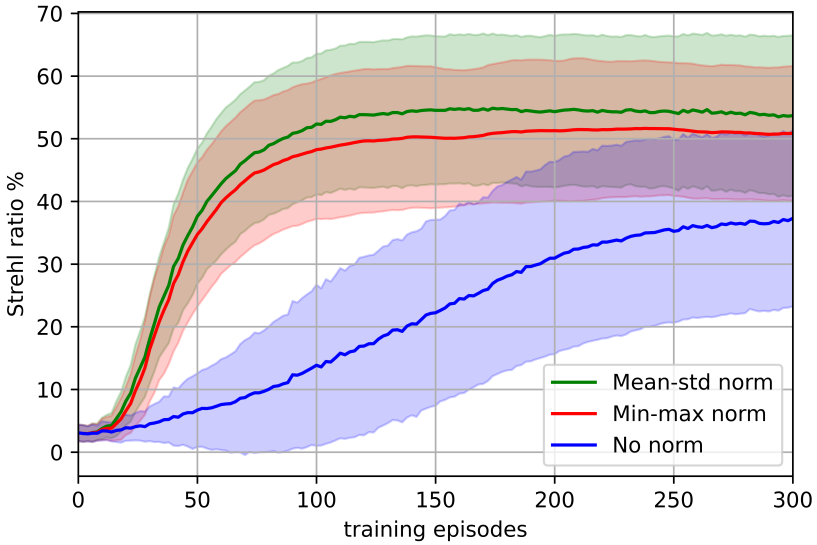

Figure C.2: Comparison of the selection of the reward normalization in DDPG applied on 20 randomly selected quasi-static atmospheric turbulences of $D/r_0 = 5$.

C.3 Proximal Policy Optimization (PPO)

The PPO algorithm is described in detail in Section 2.4.3. In PPO, the clipped surrogate objective function limits drastic changes in the policy during updates. Its effectiveness depends on the hyperparameter ϵ, which controls the size of policy updates to ensure stable and reliable training and prevent model collapse.

The hyperparameters selected for the PPO algorithm, which is tuned for a quasi-static environment, are summarized in Table C.3.

Hyperparameter	Value
Actor-lr	1×10^{-2}
Critic-lr	5×10^{-6}
Actor-Hidden dim.	150
Critic-Hidden dim.	50
Clipping ϵ	0.35
No episodes per iteration	2
No updates per iteration	20
Discount (γ)	0.95
Reward scaling	No

Table C.3: PPO hyperparameters and corresponding values in quasi-static environment

As shown in Figure C.3, the preliminary analysis demonstrates that PPO is robust to $\epsilon \in [0.05, 0.4]$ in this environment. Settings within this range show subtle differences in variance, convergence rate, and final performance. Generally, smaller ϵ values result in slightly slower initial convergence, while larger values converge more quickly but exhibit marginally lower performance. Based on this analysis, $\epsilon = 0.35$ was used for subsequent preliminary experiments to balance stability and convergence speed in the PPO approach.

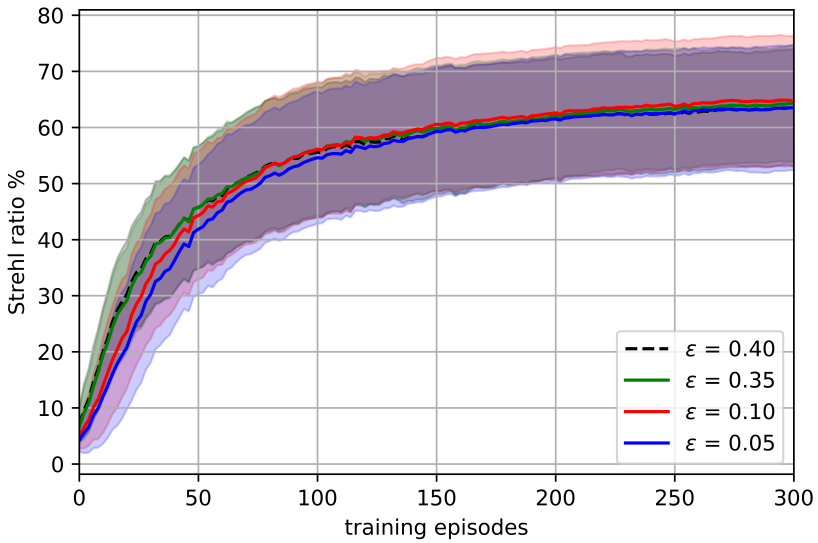

Figure C.3: Comparison of the selection of the ϵ in PPO applied on 20 randomly selected quasi-static atmospheric turbulences of $D/r_0 = 5$.

Appendix D

Evaluation of RL Controllers for Wavefront Sensorless AO Systems

To investigate the potential challenges and advantages of RL controllers in a wavefront sensorless AO system, a set of deep RL algorithms was evaluated in quasi-static and semi-dynamic environments with varying levels of turbulence severity.

D.1 Experimental Setup

The RL environment developed for the simulated wavefront sensorless AO system is described in detail in Chapter 5. It is implemented in accordance with the OpenAI Gymnasium framework standards (version 0.29.1) [213] and utilizes the HCIPy package (version 0.6.0) [223] and the PyTorch framework (version 2.0.1). The experiments were conducted on a computing system equipped with an Intel Core i7 processor, 16 GB of RAM, a 64-bit Windows operating system, and Python 3.9.6.

The RL algorithms—SAC, DDPG, and PPO—used in this setup were manually implemented from scratch, without relying on any RL libraries. The source code is available at the following GitHub Repository link: https://github.com/payamparvizi/adaptive_optics_gym.

The simulated AO system aims to optimize light coupling into a single-mode fiber at a wavelength of 1550 nm under varying levels of atmospheric turbulence severity. The experimental configurations are summarized as follows:

- **Action Space**: A 64-dimensional action space based on Disk Harmonic basis functions, and a 6-dimensional action space based on the first modes of

Zernike polynomials.

- **Observation Space**: Configurations of 2×2 pixels and 5×5 pixel photodetector array.
- **Reward Function**: The Strehl ratio and the initial version of the fiber-coupling reward function (Eq. 5.2): $(r_N = \omega_1 r_{SMF} + \omega_2 r_{SSIM})$
- **Training Setup**: Each policy was trained and tested within a single environment.

The performance of each RL algorithm was quantified by calculating the mean and standard deviation of the reward function over 20 independent trials. Hyperparameters for each algorithm were tuned, and the best-performing configurations were compared with a Shack–Hartmann wavefront sensor-based AO system. Details of the hyperparameter tuning process can be found in Appendix C.

D.2 Results

D.2.1 Quasi-Static Environment

The experiments were conducted under quasi-static turbulence conditions, where the turbulence profile remains static throughout training. This setting corresponds to scenarios where the convergence of the RL model is significantly faster than the atmospheric coherence time. The experimental setup for the quasi-static environment featured a configuration with a 2×2 pixel observation space (illustrated in Figure 5.5), a 64-dimensional action space based on the Disk Harmonic basis function (described in Section 5.3), and a Strehl ratio reward function, r_{SR} (Eq. 5.1).

Comparison of RL Algorithms

The Strehl ratio performance measured by a Shack-Hartmann sensor (described in Section 2.1.5), featuring 12 lenslets across the aperture diameter for a total of 112 lenslets, was used as a benchmark for comparison against the standard RL algorithms outlined in Section 2.4. This comparison was conducted under quasi-static turbulence conditions with $D/r_0 = 5$, where the turbulence profile remains static throughout training. This setting corresponds to scenarios where the convergence of the RL model is significantly faster than the atmospheric coherence

time—the timescale over which atmospheric turbulence can be considered approximately constant. This comparison enabled a comprehensive evaluation of the effectiveness of the proposed RL algorithms in improving wavefront correction in the presence of quasi-static atmospheric turbulence.

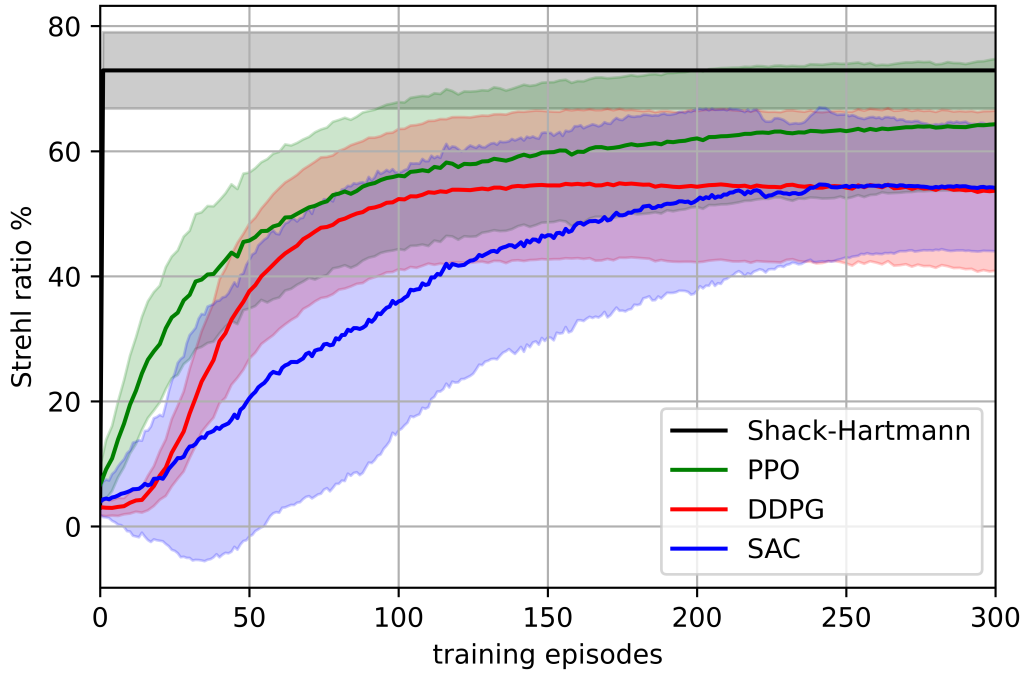

Figure D.1: Comparison of algorithms applied on 20 randomly selected quasi-static atmospheric turbulences of $D/r_0 = 5$. Note that the shaded regions extend to negative values of the Strehl ratio because the standard deviation may be larger than the mean, represented by the solid curves.

The results in Figure D.1 demonstrate that the PPO algorithm outperforms both the SAC and DDPG algorithms. Notably, PPO achieved a maximum reward of 67%, which, while lower than the 73% attained by the Shack–Hartmann sensor, remains the highest among the evaluated RL algorithms. In contrast, SAC and DDPG reached a peak reward of 53% in the early training episodes but did not improve further.

The improved performance of PPO over SAC and DDPG can be attributed to its policy optimization strategy, which constrains policy updates through a clipping mechanism. This approach promotes training stability and helps PPO avoid premature convergence to suboptimal local minima. In contrast, SAC and DDPG rely heavily on value function approximations to update the policy. Although SAC incorporates entropy regularization and soft target updates, and DDPG employs exploration noise strategies to mitigate convergence issues, both methods remain

vulnerable when value estimates are inaccurate. Such inaccuracies often arise from off-policy sampling, where the distribution of training data may deviate from the current policy's behavior, leading to poor value approximations and ineffective policy updates.

Given that PPO consistently outperformed the other RL algorithms, all subsequent experiments in this section are conducted using only the PPO algorithm.

PPO Comparison Under Varying Turbulence Severities

In this section, the performance of the PPO algorithm is evaluated under varying levels of quasi-static turbulent conditions. The results are also compared with those obtained using a Shack–Hartmann wavefront sensor, as presented in Figure D.2.

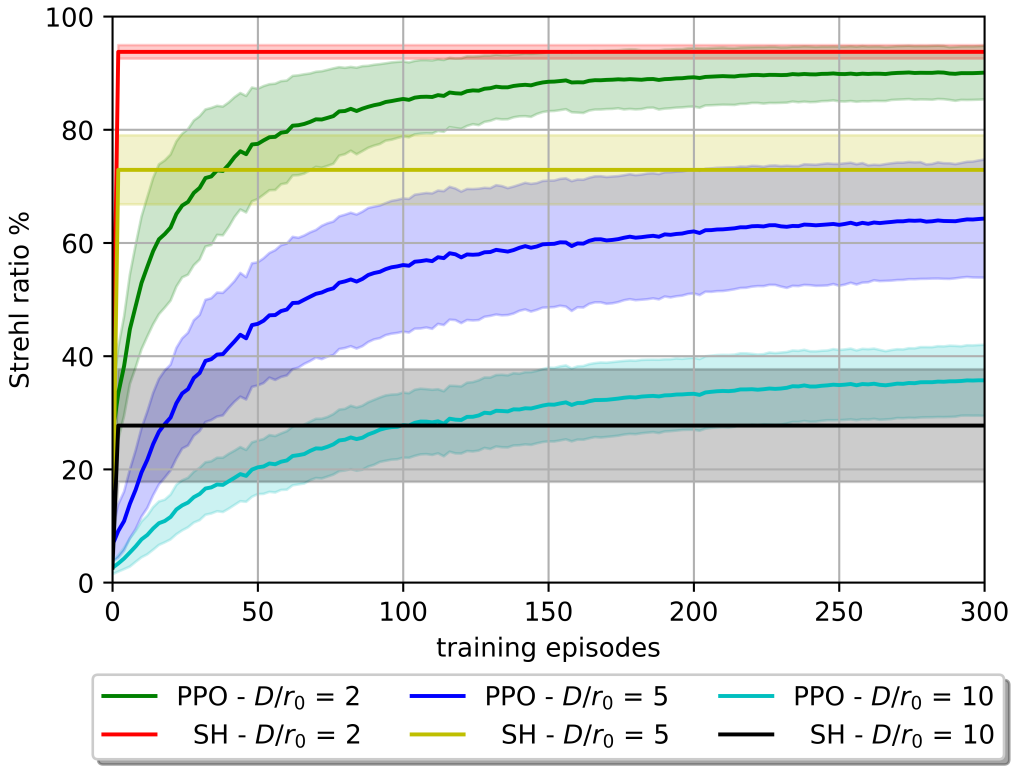

Figure D.2: Comparison of PPO algorithm and Shack-Hartmann wavefront sensor applied on 20 randomly selected quasi-static atmospheres of various D/r_0 ratios

As expected, Figure D.2 shows that as the Fried parameter r_0 (described in Section 5.1) decreases, the PPO model's ability to achieve a higher Strehl ratio decreases. Specifically, for lower D/r_0 values (e.g., $D/r_0 = 2$), the PPO algorithm achieves a Strehl ratio of approximately 90% and approaches the performance

of the Shack-Hartmann sensor. However, as D/r_0 increases (e.g., $D/r_0 = 10$), PPO struggles to achieve substantial correction, though it outperforms the Shack-Hartmann sensor. If the agent's performance does not improve significantly beyond the uncorrected Strehl ratio of 2% to 10% (depending on the D/r_0 value), it can be considered impractical.

Power Distribution Comparison

The impact of applying a PPO algorithm within an RL environment alongside a Shack-Hartmann wavefront sensor on the power distribution of a wavefront under quasi-static turbulence conditions with $D/r_0 = 5$ has been analyzed and is illustrated in Figure D.3.

In this figure, the columns correspond to different methods and episode numbers. The first column (left) shows the random actions performed by the PPO algorithm during the first episode. The second column (middle) shows the actions performed by the Shack–Hartmann wavefront sensor during the first episode. The third column (right) shows the actions performed by the PPO algorithm after training over multiple episodes (100^{th} episode). Also, the rows correspond to specific timesteps within each episode: timesteps 1 and 2 represent the initial stages, timesteps 10 and 11 represent the middle stages, and timesteps 19 and 20 represent the final stages. Timestep 1 shows the initial power distribution at the focal plane where the wavefront is reflected off a flat mirror without any AO correction.

As shown in Figure D.3, the Shack–Hartmann wavefront sensor (middle column) achieves a significant concentration of power at the center of the focal plane, with a Strehl ratio of approximately 70%. Similarly, the PPO algorithm (right column) also achieves a substantial concentration of power at the center of the focal plane, with a slightly lower Strehl ratio of approximately 60%.

It can also be seen that the learned PPO controller (right column) continues to generate varying actions even when the beam is centered on the fiber core. This instability may be due to the policy's sensitivity to small perturbations in the observations, resulting in action fluctuations even when the beam is already well centered.

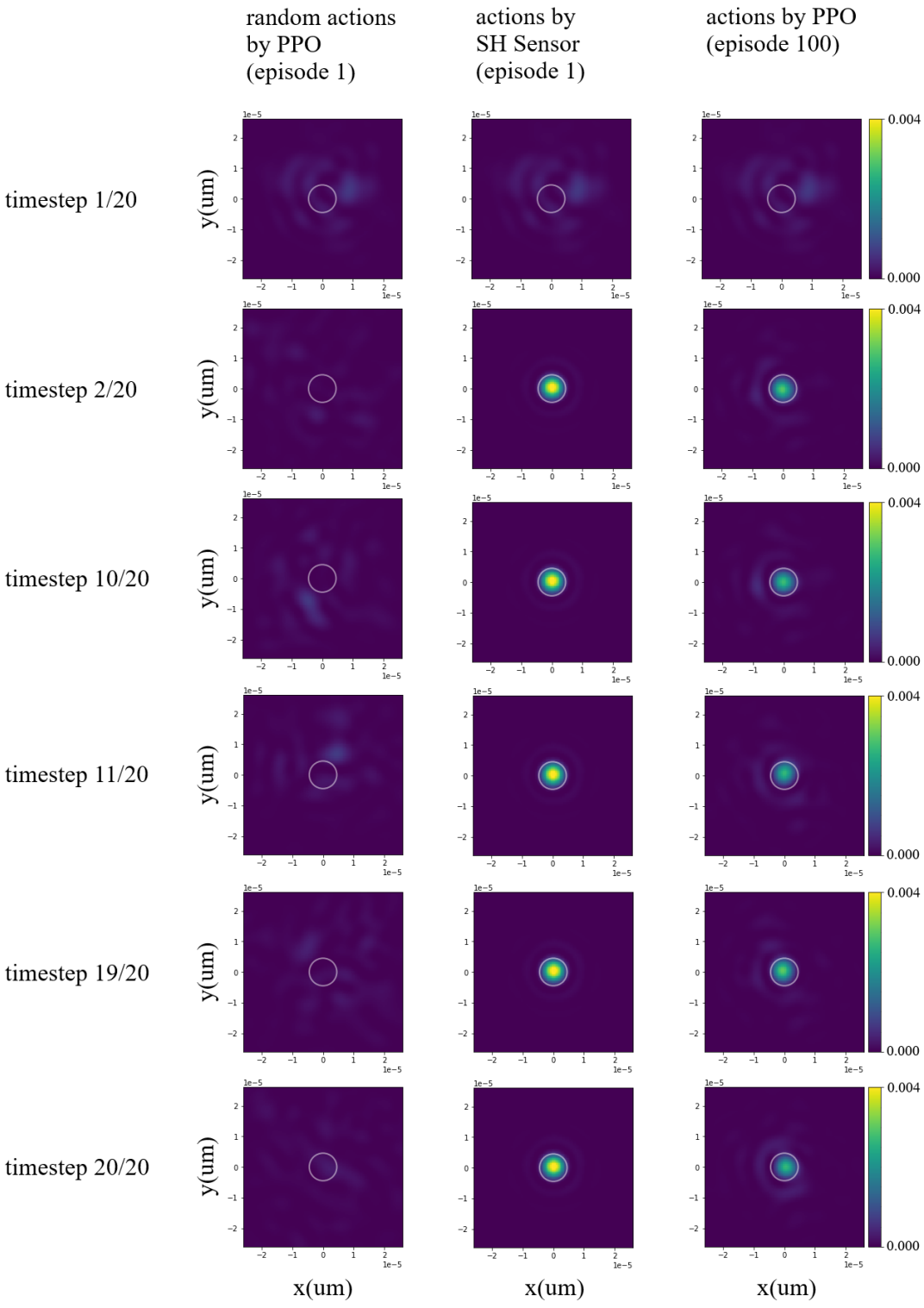

Figure D.3: Power distribution on the focal plane: columns: Shack–Hartmann wavefront sensor and PPO comparison across episodes; rows: specific timesteps within each episode.

D.2.2 Semi-Dynamic Environment

To extend and validate the algorithmic framework in dynamic environments that better approximate real-world scenarios, such as Earth's turbulent atmosphere, the RL environment must be refined to address the specific challenges posed by non-static conditions.

To accomplish this, a *semi-dynamic environment* was employed, in which the atmospheric configuration of the quasi-static environment changes between episodes. In the experiments in the semi-dynamic environment, the Strehl ratio reward function (Eq. 5.1) is replaced with the initial version of the fiber-coupling reward function (Eq. 5.2) ($r_N = \omega_1 r_{SMF} + \omega_2 r_{SSIM}$). The reason for changing the reward function is discussed in detail in Section 5.5.

In the conducted experiment, each iteration consisted of 100 episodes of a randomly selected quasi-static environment, with each episode comprising 20 time steps. The result of the experiment in a semi-dynamic environment, using a quasi-static configuration with the PPO algorithm, is presented in Figure D.4 (red). The results indicate that the configuration used for a quasi-static environment is insufficient for learning a policy in a semi-dynamic environment, achieving a mean coupling efficiency of 20%.

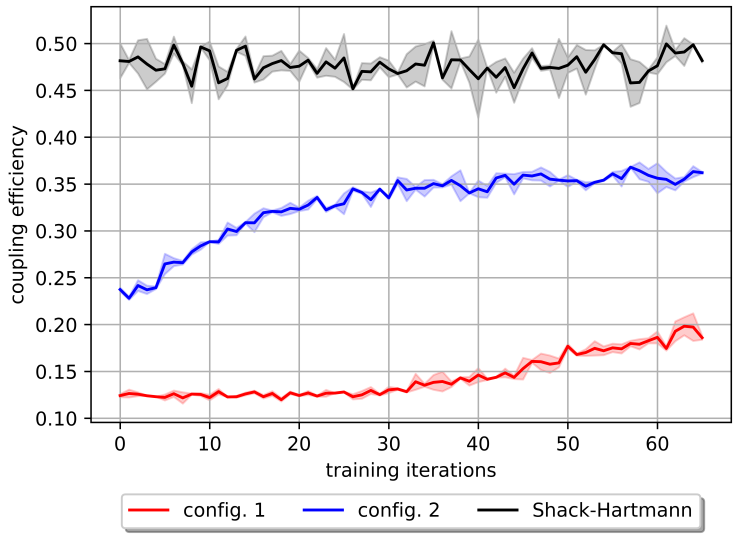

Figure D.4: Comparison between two configurations in atmospheric turbulence of $D/r_0 = 3.33$ in semi-dynamic environment. **Config. 1**: 2×2 pixels observation space, 64-D action space based on the Disk Harmonic basis function. **Config. 2**: 5×5 pixels observation space, 6-D action space based on first modes of Zernike polynomials.

Based on this poor performance, it is hypothesized that partial observability and the curse of dimensionality are significant contributing factors. In the 2×2 observation space, the agent receives low-resolution feedback, which results in a one-to-many mapping from observations to underlying environment states. In other words, different turbulence states can produce indistinguishable observations, leading to partial observability. Furthermore, using a high-dimensional action space can cause an exponential rise in computational complexity and lead to the curse of dimensionality [245].

Although these challenges were also present in the quasi-static setting, their impact was less severe. In that case, the policy was trained on a single, static turbulence profile, allowing the agent to consistently interact with the same environment and gradually adapt to its specific characteristics. This enabled successful convergence despite partial observability and a high-dimensional action space. However, in the semi-dynamic environment, the turbulence profile changes randomly between episodes. This variability prevents the policy from fully adapting to any single profile and instead requires the policy to generalize across a broader distribution of turbulence conditions. As a result, the impact of partial observability and high-dimensional actions is more pronounced in this setting, especially when the agent has limited information (low observations) and a large action space.

To address these challenges within the existing setup, the dimensionality of the action space was reduced by using the first six modes (second radial orders) of Zernike polynomials. Also, the observation space dimension was increased to a 5×5 pixels photodetector. To account for temporal dependencies, the frame stacking technique [211] was applied, using a sequence of three consecutive frames (Config. 2). The result of this approach is shown in Figure D.4 (blue). Hyperparameter tuning has been performed for this experiment.

By increasing the observation space's dimension and reducing the action space's dimension, a mean coupling efficiency of 36% was achieved, compared to the Shack- Hartmann wavefront sensor, which achieved a mean coupling efficiency of 48% (Figure D.4 (black)). This discrepancy may be attributed to the Shack–Hartmann sensor utilizing a 128×128 pixels camera for adjustments, whereas the proposed approach relies on a 5×5 pixels photodetector to provide the observation of the environment.

D.3 Discussion

This study addresses a challenging scenario involving a high 64-dimensional continuous action space and a low-dimensional 2×2 observation space in quasi-static atmospheric turbulence. This situation could potentially hinder the effectiveness of RL algorithms. However, the successful performance of the RL algorithms in this quasi-static environment demonstrated that their effectiveness is not solely dependent on the limited dimensions of the photodetector and the deformable mirror. Instead, it results from a combination of factors, including the robustness of the reward function, the quality of the photodetector employed, and the correlation within the action space. In particular, the Strehl ratio reward function used in the quasi-static environment offers a significant advantage due to its ability to provide high-quality rewards. These rewards are calculated prior to the wavefront propagating through the photodetector, which has a higher pixel count (128×128 as opposed to 2×2) and resolution. Consequently, by employing a proficient function approximator and a robust reward function, the agent can effectively generalize across related actions. Given the large number of degrees of freedom in the mirror, there are numerous possible paths to focus the light, and the agent only needs to find one of these paths.

While the preliminary experiments in this section demonstrate the applicability of RL algorithms to wavefront sensorless AO systems, the experimental setups do not fully reflect the complexity of real-world scenarios. In practice, atmospheric turbulence evolves continuously over time, and assumptions such as quasi-static or semi-dynamic turbulence may limit the practical relevance of the results. Moreover, in the quasi-static experiments, each policy is trained for a specific and static turbulence profile, which restricts its ability to generalize to other atmospheric conditions. This lack of generalization poses a significant limitation for real-world scenarios. Therefore, it is necessary to develop RL policies that can operate under a broader range of dynamic turbulence conditions to enhance their practical applicability.

D.4 Summary

The preliminary experiments demonstrated the feasibility of deep RL algorithms for wavefront sensorless AO in optical communication downlinks. Among the evaluated algorithms, PPO consistently outperformed SAC and DDPG in quasi-

static environments. This was primarily due to PPO's stable policy optimization strategy and its ability to avoid premature convergence. As a result, PPO was selected for subsequent experiments.

To extend and validate the RL controllers in environments that better approximate real-world conditions, the RL environment was refined to address the challenges posed by non-static turbulence, referred to as the semi-dynamic environment. Experiments in this semi-dynamic setting revealed significant limitations caused by partial observability and the curse of dimensionality. In the quasi-static scenario, each policy was trained on a single static turbulence profile, enabling the agent to gradually adapt to a specific environment. In contrast, the semi-dynamic environment introduced variations in turbulence between episodes, requiring the policy to generalize across a broader distribution of atmospheric conditions. This difference revealed the limitations associated with low observations and high-dimensional action spaces.

For this reason, the dimension of observation space was increased to address partial observability, while the action space was reduced using Zernike polynomials to address the curse of dimensionality. The choice of reward function also proved critical: while the Strehl ratio was suitable for static conditions, transitioning to the proposed fiber-coupling reward function was necessary for improved adaptation in dynamic scenarios.

Overall, these findings highlight the challenges and potential of using RL controllers for wavefront sensorless AO systems. They inform the development of more generalizable RL policies capable of handling fully dynamic atmospheric turbulence.

Appendix E

Evaluation of PPO Controller for WSL-AO Systems in a Quasi-Static Environment

Further evaluation of the RL-based controller for the simulated wavefront sensorless AO system was conducted using the Vanilla PPO algorithm in a quasi-static atmosphere. The quasi-static atmosphere refers to an environment in which the velocity field describing the atmosphere dynamics is zero, resulting in a static turbulence profile.

While the primary focus is on the dynamic environment, the quasi-static environment plays a critical role in the development process. It provides a controlled setting for thoroughly evaluating RL algorithms, which enables a more precise assessment of the efficiency of different configurations. This makes it a necessary step before transitioning to the dynamic environment. A configuration that fails to perform adequately in the quasi-static setting is unlikely to succeed under the more complex conditions of a dynamic environment.

The evaluation begins by investigating the effects of action space normalization, environment configuration, and reward function design. Next, the performance of the Vanilla PPO algorithm is compared with that of a Shack–Hartmann wavefront sensor-based AO system and a flat mirror scenario under varying levels of turbulence severity.

E.1 Experimental Setup

The RL environment developed for the simulated wavefront sensorless AO system is described in detail in Chapter 5. It is implemented following the OpenAI Gymnasium framework standards (version 0.29.1) [213] and utilizes the HCIPy package (version 0.6.0) [223], the PyTorch framework (version 2.0.1), and the Tianshou RL library (version 0.5.1) [238]. The Tianshou library was selected because it helps to minimize implementation errors and reduce development time.

The experiments were performed on a high-performance computing (HPC) platform provided by Compute Canada equipped with 5 GB of RAM per CPU, an NVIDIA P100 GPU, and Python 3.9.6.

The simulated AO system aims to optimize light coupling into a single-mode fiber at a wavelength of 1550 nm under varying levels of atmospheric turbulence severity in a quasi-static environment. Given the requirements of the physical setup, the experimental configurations are summarized as follows:

- **Action Space**: 10-dimensional action space based on the first modes of Zernike polynomials.
- **Observation Space**: 5×5 pixel photodetector array.
- **Reward Function**: The fiber-coupling reward function, defined in (5.2).
- **Training Setup**: Each policy was trained and tested on a broad range of atmospheric conditions.

E.2 Results and Discussion

This section presents the results of the action space normalization, environment configuration, and reward function analysis for the Vanilla PPO algorithm in a quasi-static environment. It also compares the performance of the Vanilla PPO algorithm with that of a Shack–Hartmann wavefront sensor-based AO system and a flat mirror scenario in varying turbulence severities.

In the initial stages of the evaluation, the policy was trained and tested across 10 different atmospheric conditions. This approach enabled faster iteration and reduced computational costs for action space normalization, environment configuration, and reward function design. After identifying a suitable configuration, the number of atmospheric conditions was increased to 50 in subsequent experiments

across varying turbulence severities to improve the generality and robustness of the learned policy.

E.2.1 Action Space Normalization

An action space consisting of the first 10 modes of the Zernike polynomial, as described in Section 2.1.2, is considered. However, the arrangement of these modes within the action space plays a crucial role in performance. To explore this, different normalizations are evaluated using the Vanilla PPO algorithm. These normalizations are as follows:

$$A_1 = \{a_1, a_2, \cdots, a_{10}\} \tag{E.1a}$$

$$A_2 = \left\{\frac{a_1}{p}, \frac{a_2}{2p}, \cdots, \frac{a_{10}}{10p}\right\} \tag{E.1b}$$

$$A_3 = \{c, a_2, a_3, \cdots, a_{10}\} \tag{E.1c}$$

$$A_4 = \left\{c, \frac{a_2}{p}, \frac{a_3}{2p}, \cdots, \frac{a_{10}}{9p}\right\} \tag{E.1d}$$

$$A_5 = \left\{c, \frac{a_2}{p}, \frac{a_3}{p}, \frac{a_4}{2p}, \frac{a_5}{2p}, \frac{a_6}{2p}, \frac{a_7}{3p}, \frac{a_8}{3p}, \frac{a_9}{3p}, \frac{a_{10}}{3p}\right\} \tag{E.1e}$$

where a_i for $i \in \{1, \ldots, 10\}$ represents the coefficient of the i^{th} Zernike mode. Lower values of i correspond to lower-order modes, while higher values correspond to higher-order modes. The parameter p is a positive constant used to adjust the influence of each Zernike mode and is tuned to achieve the desired balance between lower- and higher-order modes.

In A_1, the arrangement consists of the order with no weighting or modifications applied to the modes. The A_2 arrangement assigns progressively smaller weights to the modes, so the lower-order modes have more effect on the performance, while the higher-order modes contribute less. In A_3, the arrangement is similar to A_1, except that the first mode, known as the Piston, is set to a constant value, c. Since the Piston mode does not impact performance, it can remain fixed. The A_4 arrangement is similar to A_2, but the Piston mode is constant, and the subsequent modes are weighted in a decreasing manner. Finally, A_5 adjusts the effect of each mode based on the radial degree of the Zernike polynomials. Modes with lower radial degrees have more effect on the performance, so they have larger weights. A comparison of the different normalizations of the action space is presented in Figure E.1.

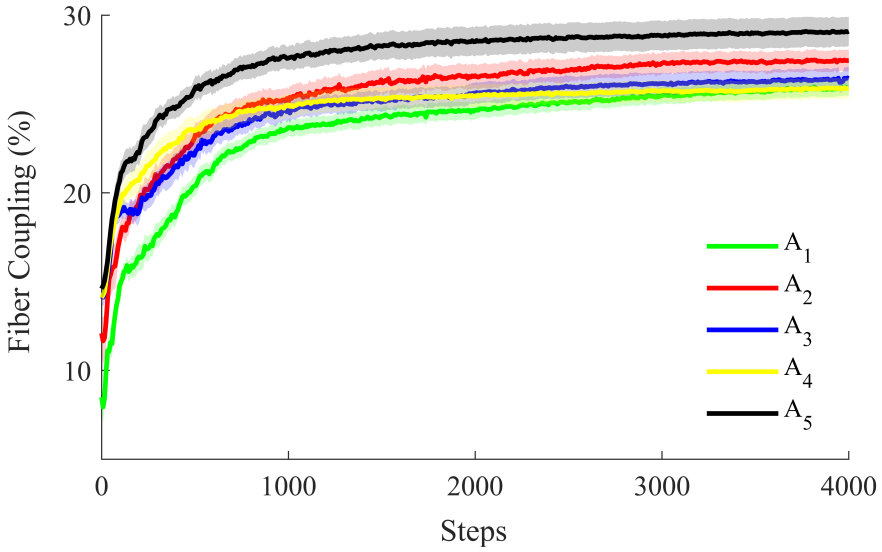

Figure E.1: Average fiber coupling (%) across various action space normalizations in Vanilla PPO, evaluated over 10 seeds from 10 distinct testing environments with $D/r_0 = 5$

As shown in Figure E.1, the action space arrangement A_5 (Eq. E.1e, represented by the black line) has achieved the highest fiber coupling efficiency and outperforms the other arrangements. Consequently, this arrangement is selected for further experiments.

E.2.2 Environment Configuration

As described in Section 5.2, the simulation operates in a continuous environment, where atmospheric motion is determined by a velocity field. In a quasi-static atmosphere, this velocity is zero. Although the atmospheric environment evolves continuously over time, the simulation discretizes this process into episodes to accommodate the digital nature of observations and policy updates. Each episode consists of 20 timesteps, with experience collected and policy updates performed every 20 episodes. The choice of 20 timesteps per episode was motivated by a preliminary study conducted across episode lengths ranging from 10 to 100 timesteps, which indicated that 20 timesteps were sufficient to achieve optimal action performance. In this setup, the initial timestep of episode $i + 1$ aligns with the final timestep of episode i.

At the start of each episode, there is an option to set the deformable mirror to a flat condition. Although the environment is quasi-static, the impact of this ini-

tialization is evaluated. Without resetting, the mirror shape at the start of each episode can vary depending on the preceding episode, introducing variability into the learning process that can lead to learning difficulties. By comparing episodes initialized with a flat mirror versus leaving the mirror as is, as shown in Figure E.2, it is observed that starting with a flat mirror is associated with slightly improved performance and lower standard error. This approach can contribute to more stable policy learning by providing a consistent starting point for each episode and reducing the likelihood of the policy getting stuck in suboptimal solutions.

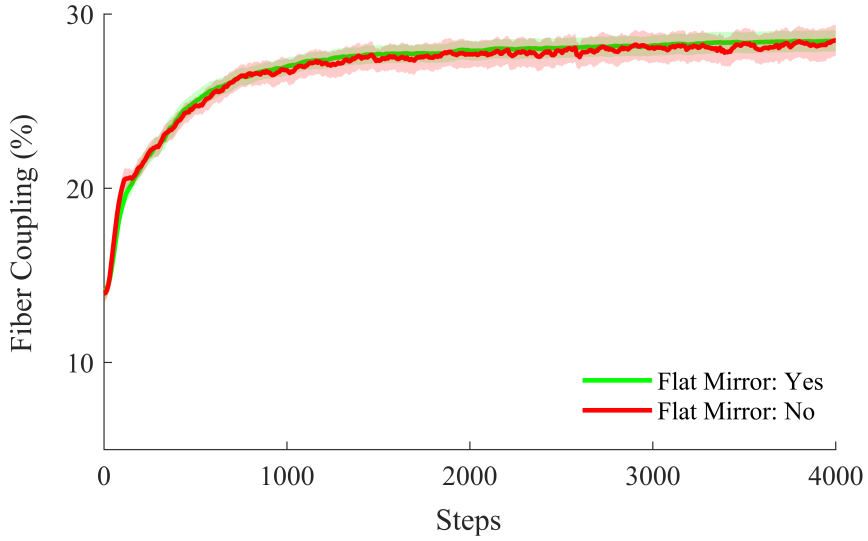

Figure E.2: Average fiber coupling (%) comparing the use of the flat mirror condition versus not using it at the first timestep of each episode in Vanilla PPO, evaluated over 10 seeds from 10 distinct testing environments with $D/r_0 = 5$

E.2.3 Reward Function Analysis

In the previous experiments, the reward function described in Equation 5.2 was used. Since many changes have been made to the parameters, it is necessary to re-analyze the reward function. As stated in Section 5.5, the reward function comprises three sub-rewards: Single-Mode Fiber total power (r_{SMF}; Eq. 5.3), Structural Similarity Index Metric (SSIM) (r_{SSIM}; Eq. 5.4), and tip/tilt error (r_{TE}; Eq. 5.7). Each of these sub-rewards, as well as their combinations, are analyzed in detail.

The rewards to be analyzed are defined as follows:

$$r_1 = r_{TE} \tag{E.2a}$$
$$r_2 = r_{SSIM} \tag{E.2b}$$
$$r_3 = r_{SMF} \tag{E.2c}$$
$$r_4 = \omega_1 r_{SMF} - \omega_3 r_{TE} \tag{E.2d}$$
$$r_5 = \omega_1 r_{SMF} + \omega_2 r_{SSIM} \tag{E.2e}$$
$$r_6 = \omega_1 r_{SMF} + \omega_2 r_{SSIM} - \omega_3 r_{TE} \tag{E.2f}$$

where ω_1, ω_2, and ω_3 are the weights assigned to the rewards to balance their contributions in the overall reward function. A comparison between the reward functions implemented in the proposed RL environment (described in Chapter 5) and the best configuration identified in Sections E.2.1 and E.2.2 is used. To ensure fair comparison among reward functions, the same environment and optical configuration specified in Table 6.1 were used, together with the same PPO hyperparameters reported in Table 6.2 and Table 6.4 ($D/r_0 = 5$ column). For each reward function, the experiment was conducted on 10 randomized training and testing environments generated using 10 identical random seeds. Only the reward definition was changed, while all other settings were kept identical across cases. The comparison results are presented in Figure E.3.

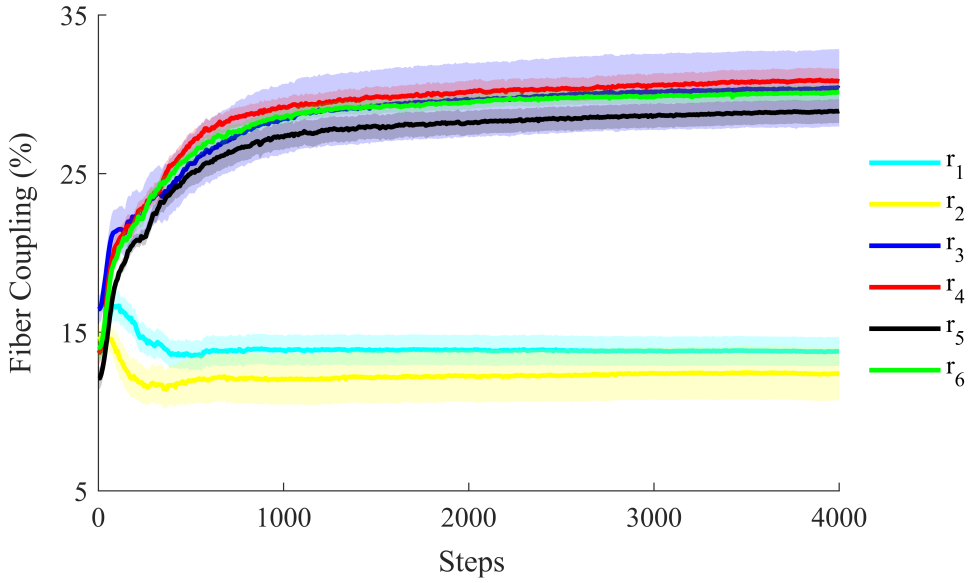

Figure E.3: Average fiber coupling (%) across various rewards in Vanilla PPO, evaluated over 10 seeds from 10 distinct testing environments with $D/r_0 = 5$

As shown in Figure E.3, neither the r_1 nor r_2 reward functions improve performance and result in a policy that behaves randomly. In contrast, the r_3 reward function improves performance but is associated with a high standard error. To address this, the effect of combining r_1 and r_2 with r_3 was investigated by tuning the balancing weights ω_2 and ω_3 within the range $[0, 1]$. Since r_1 demonstrated significantly better performance than the other reward components, its weighting factor ω_1 was fixed at 1. As all reward components are normalized between 0 and 1, this range ensures meaningful and comparable weighting among the terms.

The results indicate that the combination of r_1 and r_3 (r_4), with weights $\omega_1 = 1$ and $\omega_3 = 0.05$, leads to the best performance while reducing standard error, as illustrated by the red line. Additionally, the results for r_5 and r_6 indicate that the r_{SSIM} reward function fails to improve performance and degrades the effectiveness of r_{SMF}. This outcome may be attributed to SSIM behaving similarly to the Strehl ratio, where improvements in SSIM do not necessarily correspond to improvements in fiber coupling, as discussed in Section 5.5. Consequently, the r_4 reward function was selected for further experiments.

E.2.4 Performance with Varying Turbulence Severity

The hyperparameters of the Vanilla PPO algorithm in a quasi-static environment are retuned based on the action space arrangement defined by Equation (E.1e), using a flat mirror at the first timestep of each episode and the reward function described in Equation (E.2d). Additionally, to evaluate the policy's generality across various atmospheric conditions in a quasi-static atmosphere, 50 training and testing environments are employed. These hyperparameters are summarized in Table 6.2 and Table 6.4.

The Vanilla PPO algorithm is evaluated under low ($D/r_0 = 2$) and high ($D/r_0 = 5$) quasi-static turbulence conditions. The performance of the Vanilla PPO algorithm is also compared to a Shack-Hartmann wavefront sensor-based AO system (with 40 lenslets across the aperture diameter) and the flat mirror scenario. The results of these comparisons are presented in Figs E.4 and E.5.

By examining these figures, it is observed that as D/r_0 increases, fiber coupling efficiency decreases. This occurs because higher D/r_0 values correspond to stronger turbulence, leading to more significant wavefront distortion and reduced coupling performance. Figure E.4 demonstrates that under high turbulence conditions ($D/r_0 = 5$), Vanilla PPO achieves a fiber coupling efficiency of 21%,

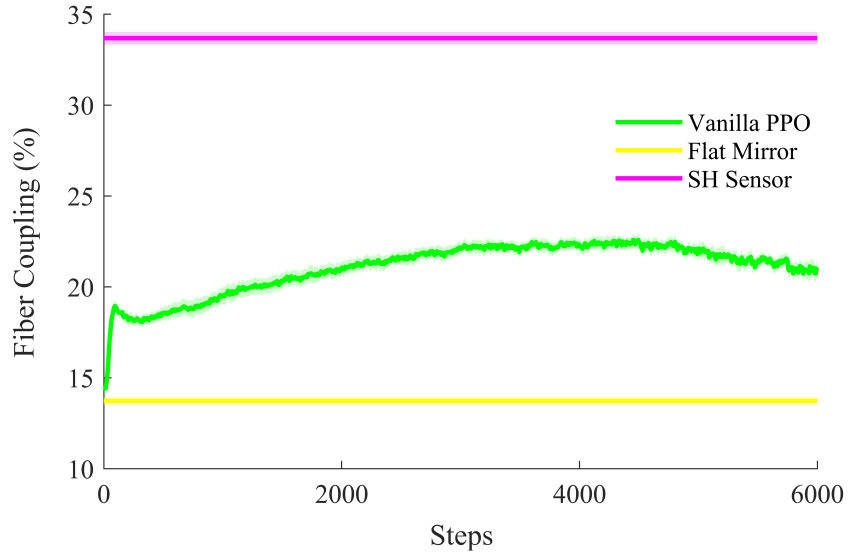

Figure E.4: Average fiber coupling (%) using the Vanilla PPO algorithm, Shack-Hartmann wavefront sensor, and flat mirror scenario evaluated over 10 seeds from 50 distinct testing environments with $D/r_0 = 5$

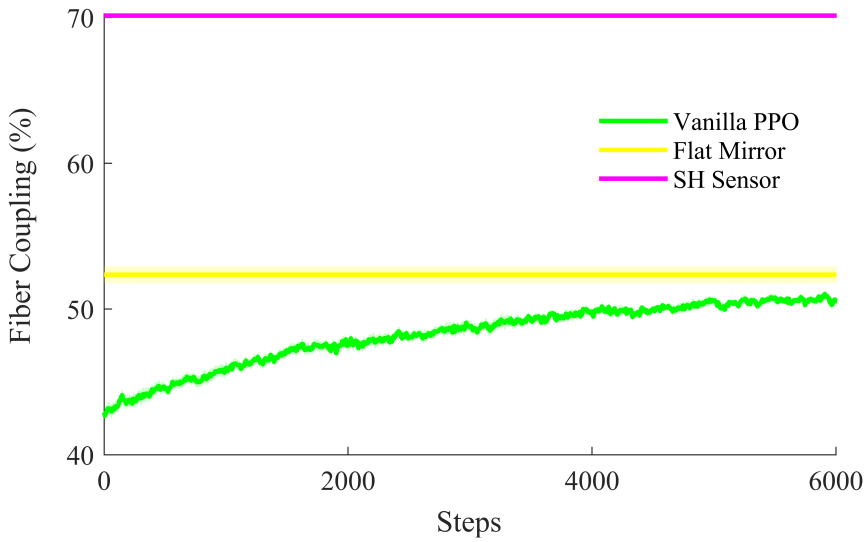

Figure E.5: Average fiber coupling (%) using the Vanilla PPO algorithm, Shack-Hartmann wavefront sensor, and flat mirror scenario evaluated over 10 seeds from 50 distinct testing environments with $D/r_0 = 2$

representing 62% of the performance achieved by the Shack-Hartmann wavefront sensor and exceeding the performance of the flat mirror scenario by 52%.

When comparing Figure E.3 and Figure E.4, both experiments use the Vanilla PPO algorithm in a quasi-static atmosphere with $D/r_0 = 5$. However, the training setups differ in the number of distinct training and testing environments used. In the experiments shown in Figure E.3, the policy was trained and tested on 10 distinct environments, achieving a fiber coupling efficiency of 31%. In contrast, the experiments shown in Figure E.4 used 50 distinct environments to improve the generality of the policy, resulting in a lower fiber coupling efficiency of 21%. The increased variability across environments makes the learning task more challenging, which explains the lower fiber coupling efficiency observed in Figure E.4.

Figure E.5 demonstrates that under low turbulence conditions ($D/r_0 = 2$), Vanilla PPO achieves a fiber coupling efficiency of 51%, which corresponds to 72% of the performance achieved using the Shack-Hartmann wavefront sensor. However, it underperforms the flat mirror scenario by 3%. This discrepancy arises because, in low-turbulence conditions, wavefront distortions are minimal, allowing a flat mirror to already achieve high fiber coupling efficiency. For this reason, during training, the RL controller tends to converge to a suboptimal flat mirror configuration.

Figures E.4 and E.5 demonstrate that the Vanilla PPO algorithm significantly underperforms the Shack-Hartmann wavefront sensor under both low and high turbulence severity conditions and also underperforms a flat mirror scenario under low turbulence severity in a quasi-static environment. This highlights the need for improvements to the Vanilla PPO algorithm.

One potential reason for performance degradation in the Vanilla PPO algorithm can be the high complexity of deep neural networks and their sensitivity to small perturbations in input observations. In such settings, the control outputs may exhibit significant variability even within acceptable actuation limits. This leads to high-frequency oscillations in the action signals. This behavior is evident in Figure D.3, where the PPO controller continues to generate fluctuating actions even when the beam is already centered on the fiber core. These fluctuations suggest that the policy struggles to maintain a stable control signal, likely due to its sensitivity to perturbations in the observations. This instability motivated the exploration of policy smoothness techniques to mitigate high-frequency oscillations and improve fiber coupling stability and overall system performance in the wavefront sensorless AO system. To address these challenges and enforce smoother control

while maintaining high fiber coupling efficiency, Vanilla PPO is integrated with the proposed SAPPS method introduced in Section 3.1.2.

E.3 Summary

This chapter presented the evaluation of the Vanilla PPO algorithm for a simulated wavefront sensorless AO system in a quasi-static environment. The initial evaluations focused on tuning the Vanilla PPO algorithm, including action space normalization, environment configuration, and reward function design. After identifying a suitable configuration, the performance of the Vanilla PPO algorithm was compared with that of a Shack–Hartmann wavefront sensor-based AO system and a flat mirror scenario under varying turbulence severities.

The results demonstrate that the Vanilla PPO algorithm's performance in the wavefront sensorless AO system is unsatisfactory, falling significantly short compared to the Shack–Hartmann wavefront sensor-based AO system. It is hypothesized that this poor performance is caused by the PPO policy's inability to maintain stable coupling within the single-mode fiber core due to oscillations in its actions.

To mitigate these oscillations and achieve smoother control while maintaining fiber coupling, the Vanilla PPO algorithm was integrated with the proposed State-Adaptive Proportional Policy Smoothing (SAPPS) method, as formulated in Chapter 3.

Bibliography

[1] Rayan A Alsemmeari, Sheikh Tahir Bakhsh, and Hani Alsemmeari. "Free space optics vs radio frequency wireless communication". In: *Int. J. Inf. Technol. Comput. Sci* 8.9 (2016), pp. 1–8.

[2] Hemani Kaushal and Georges Kaddoum. "Optical communication in space: Challenges and mitigation techniques". In: *IEEE communications surveys & tutorials* 19.1 (2016), pp. 57–96.

[3] Vincent WS Chan. "Free-space optical communications". In: *Journal of Lightwave technology* 24.12 (2006), pp. 4750–4762.

[4] Robert K Tyson and Benjamin West Frazier. *Principles of adaptive optics*. CRC press, 2022.

[5] Max Born and Emil Wolf. "Principles of Optics, 7th (expanded) edition". In: *United Kingdom: Press Syndicate of the University of Cambridge* 461 (1999), p. 93.

[6] Hemani Kaushal and Georges Kaddoum. "Optical Communication in Space: Challenges and Mitigation Techniques". In: *IEEE Communications Surveys & Tutorials* 19.1 (2017), pp. 57–96. DOI: 10.1109/COMST.2016.2603518.

[7] Jing Ma et al. "Performance analysis of satellite-to-ground downlink coherent optical communications with spatial diversity over Gamma-Gamma atmospheric turbulence". In: *Appl. Opt.* 54.25 (Sept. 2015), pp. 7575–7585. DOI: 10.1364/AO.54.007575. URL: https://opg.optica.org/ao/abstract.cfm?URI=ao-54-25-7575.

[8] Jiang Wenhan. "Overview of adaptive optics development". In: *Opto-Electronic Engineering* 45.3 (2018), pp. 170489-1-170489–15. ISSN: 1003-501X. DOI: 10.12086/oee.2018.170489.

[9] François Roddier. *Adaptive optics in astronomy*. Cambridge university press, 1999.

[10] Parham Taghina. "Wavefront sensorless adaptive optics for astronomical applications." In: (2023).

[11] Richard Davies and Markus Kasper. "Adaptive optics for astronomy". In: *arXiv preprint arXiv:1201.5741* (2012).

[12] Kai Sum Chan and HF Chau. "Reducing the impact of adaptive optics lag on optical and quantum communications rates from rapidly moving sources". In: *AIP Advances* 13.5 (2023).

[13] Hongxi Ren, Bing Dong, and Yan Li. "Alignment of the active secondary mirror of a space telescope using model-based wavefront sensorless adaptive optics". In: *Applied Optics* 60.8 (2021), pp. 2228–2234.

[14] Qinghua Tian et al. "DNN-based aberration correction in a wavefront sensorless adaptive optics system". In: *Optics express* 27.8 (2019), pp. 10765–10776.

[15] Jalo Nousiainen. "Model-based reinforcement learning and inverse problems in extreme adaptive optics control". PhD thesis. Lappeenranta-Lahti University of Technology LUT, 2023.

[16] Eduard Durech et al. "Wavefront sensor-less adaptive optics using deep reinforcement learning". In: *Biomedical optics express* 12.9 (2021), pp. 5423–5438.

[17] B Pou et al. "Adaptive optics control with multi-agent model-free reinforcement learning". In: *Optics express* 30.2 (2022), pp. 2991–3015.

[18] Jalo Nousiainen et al. "Adaptive optics control using model-based reinforcement learning". In: *Optics Express* 29.10 (2021), pp. 15327–15344.

[19] Jalo Nousiainen et al. "Towards on-sky adaptive optics control using reinforcement learning". In: *arXiv preprint arXiv:2205.07554* (2022).

[20] B Pou et al. "Model-free reinforcement learning with a non-linear reconstructor for closed-loop adaptive optics control with a pyramid wavefront sensor". In: *Adaptive Optics Systems VIII*. Vol. 12185. SPIE. 2022, pp. 945–958.

[21] Hu Ke et al. "Self-Learning Control for Wavefront Sensorless Adaptive Optics System through Deep Reinforcement Learning". In: *Optik* 178 (2019), pp. 785–793.

[22] K Hu et al. "Build the structure of wfsless ao system through deep reinforcement learning". In: *IEEE Photonics Technology Letters* 30.23 (2018), pp. 2033–2036.

[23] Tomi Krokberg. "Reinforcement learning in multi-mirror adaptive optics". PhD thesis. LUT University, 2022. URL: %5Curl%7Bhttps://urn.fi/URN:NBN:fi-fe2022081555333%7D.

[24] Ming Li et al. "Imaging performance evaluation and phasing error correction of sparse aperture telescope based on small satellites formation". In: *2019 International Conference on Optical Instruments and Technology: Optical Systems and Modern Optoelectronic Instruments*. Vol. 11434. SPIE. 2020, pp. 195–202.

[25] Siddharth Mysore et al. "Regularizing action policies for smooth control with reinforcement learning". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 1810–1816.

[26] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. "Adversarial examples in the physical world". In: *Artificial intelligence safety and security*. Chapman and Hall/CRC, 2018, pp. 99–112.

[27] A Rupam Mahmood et al. "Benchmarking reinforcement learning algorithms on real-world robots". In: *Conference on robot learning*. PMLR. 2018, pp. 561–591.

[28] I Lee et al. "Gradient-based Regularization for Action Smoothness in Robotic Control with Reinforcement Learning". In: *arXiv preprint arXiv:2407.04315* (2024).

[29] Henry Gouk et al. "Regularisation of neural networks by enforcing lipschitz continuity". In: *Machine Learning* 110 (2021), pp. 393–416.

[30] Timothy P Lillicrap et al. "Continuous control with deep reinforcement learning". In: *arXiv preprint arXiv:1509.02971* (2015).

[31] Scott Fujimoto, Herke Hoof, and David Meger. "Addressing function approximation error in actor-critic methods". In: *International conference on machine learning*. PMLR. 2018, pp. 1587–1596.

[32] Tuomas Haarnoja et al. "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor". In: *International conference on machine learning*. PMLR. 2018, pp. 1861–1870.

[33] John Schulman et al. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[34] James W Mock and Suresh S Muknahallipatna. "Sim-to-real: A performance comparison of ppo, td3, and sac reinforcement learning algorithms for quadruped walking gait generation". In: *Journal of Intelligent Learning Systems and Applications* 16.2 (2024), pp. 23–43.

[35] Bharathan Balaji et al. "Deepracer: Educational autonomous racing platform for experimentation with sim2real reinforcement learning". In: *arXiv preprint arXiv:1911.01562* (2019).

[36] Karol Arndt et al. "Meta reinforcement learning for sim-to-real domain adaptation". In: *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2020, pp. 2725–2731.

[37] Bangyu Qin, Yue Gao, and Yi Bai. "Sim-to-real: Six-legged robot control with deep reinforcement learning and curriculum learning". In: *2019 4th International Conference on Robotics and Automation Engineering (ICRAE)*. IEEE. 2019, pp. 1–5.

[38] Payam Parvizi et al. "Reinforcement Learning Environment for Wavefront Sensorless Adaptive Optics in Single-Mode Fiber Coupled Optical Satellite Communications Downlinks". In: *Photonics*. Vol. 10. 12. MDPI. 2023, p. 1371.

[39] Runnan Zou et al. "Wavefront Sensorless Adaptive Optics for Free-space Satellite-to-Ground Communication using Reinforcement Learning". In: *COAT2023*. 2023.

[40] Payam Parvizi et al. "Action-Regularized Reinforcement Learning for Adaptive Optics in Optical Satellite Communication". In: (Sept. 2025). DOI: 10.1364/opticaopen.30043543.v2.

[41] Walter Henry Angel Fincham and Michael Harold Freeman. *Optics*. Elsevier, 2013.

[42] Vincent Billault et al. "Evaluation of a multimode receiver with a photonic integrated combiner for satellite to ground optical communications". In: *arXiv preprint arXiv:2208.08869* (2022).

[43] Jacopo Antonello et al. "Optimization-based wavefront sensorless adaptive optics for multiphoton microscopy". In: *Journal of the Optical Society of America A* 31.6 (June 2014), pp. 1337–1347. DOI: 10.1364/JOSAA.31.001337.

[44] Martin J Booth. "Adaptive optics in microscopy". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 365.1861 (2007), pp. 2829–2843.

[45] Soo Sim Daniel Neo and Bert Lundy. *Free space optics communication for mobile military platforms*. Monterey, California. Naval Postgraduate School, 2012.

[46] N Pourré et al. "Low-wind-effect impact on Shack-Hartmann-based adaptive optics-Partial control solution in the context of SPHERE and GRAVITY+". In: *Astronomy & Astrophysics* 665 (2022), A158.

[47] John C Wyngaard. "Atmospheric turbulence". In: *Annual Review of Fluid Mechanics* 24.1 (1992), pp. 205–234.

[48] Oliver Graham Sutton. *Atmospheric turbulence*. Routledge, 2020.

[49] Hemani Kaushal et al. "Experimental study on beam wander under varying atmospheric turbulence conditions". In: *IEEE Photonics Technology Letters* 23.22 (2011), pp. 1691–1693.

[50] Joseph C Marron et al. "Atmospheric turbulence correction using digital holographic detection: experimental results". In: *Optics express* 17.14 (2009), pp. 11638–11651.

[51] Francois Rigaut and Eric Gendron. "Laser guide star in adaptive optics-The tilt determination problem". In: *Astronomy and Astrophysics* 261 (1992), pp. 677–684.

[52] Brent L Ellerbroek and François Rigaut. "Methods for correcting tilt anisoplanatism in laser-guide-star-based multiconjugate adaptive optics". In: *JOSA A* 18.10 (2001), pp. 2539–2547.

[53] Enrico Fedrigo, Riccardo Muradore, and Davide Zilio. "High performance adaptive optics system with fine tip/tilt control". In: *Control Engineering Practice* 17.1 (2009), pp. 122–135.

[54] Stephen A Burns et al. "Adaptive optics imaging of the human retina". In: *Progress in retinal and eye research* 68 (2019), pp. 1–30.

[55] Guang-ming Dai. *Wavefront optics for vision correction*. Vol. 179. SPIE press, 2008.

[56] Austin Roorda et al. "Adaptive optics scanning laser ophthalmoscopy". In: *Optics express* 10.9 (2002), pp. 405–412.

[57] Pablo Artal et al. "Compensation of corneal aberrations by the internal optics in the human eye". In: *Journal of vision* 1.1 (2001), pp. 1–1.

[58] Dariusz Stramski, Sorin Constantin, and Rick A Reynolds. "Adaptive optical algorithms with differentiation of water bodies based on varying composition of suspended particulate matter: A case study for estimating the particulate organic carbon concentration in the western Arctic seas". In: *Remote Sensing of Environment* (2022), p. 113360.

[59] ML Holohan and JC Dainty. "Low-order adaptive optics: a possible use in underwater imaging?" In: *Optics & Laser Technology* 29.1 (1997), pp. 51–55.

[60] Frits Zernike. "Diffraction theory of the knife-edge test and its improved form, the phase-contrast method". In: *Monthly Notices of the Royal Astronomical Society* 94 (1934), pp. 377–384.

[61] Robert J Noll. "Zernike polynomials and atmospheric turbulence". In: *JOsA* 66.3 (1976), pp. 207–211.

[62] Esdras Anzuola Valencia. "Atmospheric compensation experiments on free-space optical coherent communication systems". PhD thesis. Universitat Politècnica de Catalunya, 2015.

[63] Virendra N Mahajan and Guang-Ming Dai. "Orthonormal polynomials for hexagonal pupils". In: *Optics letters* 31.16 (2006), pp. 2462–2464.

[64] Horace W Babcock. "The possibility of compensating astronomical seeing". In: *Publications of the Astronomical Society of the Pacific* 65.386 (1953), pp. 229–236.

[65] Antoine Labeyrie. "Attainment of diffraction limited resolution in large telescopes by Fourier analysing speckle patterns in star images". In: *Astronomy and astrophysics* 6 (1970), p. 85.

[66] A Buffington et al. "First observatory results with an image-sharpening telescope". In: *JOSA* 67.3 (1977), pp. 304–305.

[67] John W Hardy, J E Lefebvre, and CL Koliopoulos. "Real-time atmospheric compensation". In: *JOSA* 67.3 (1977), pp. 360–369.

[68] John W Hardy. "Active optics: a new technology for the control of light". In: *Proceedings of the IEEE* 66.6 (1978), pp. 651–697.

[69] Horace W Babcock. "Adaptive optics revisited". In: *Science* 249.4966 (1990), pp. 253–257.

[70] John W Hardy. "Adaptive optics: a progress review". In: *Active and Adaptive Optical Systems* 1542 (1991), pp. 2–17.

[71] Darryl P Greenwood and Charles A Primmerman. "Adaptive optics research". In: *Lincoln Laboratory Journal* 5.1 (1992).

[72] Stewart Wills. *Infrared Imaging, adaptive optics spurred Nobel-worthy discovery*. Oct. 2020. URL: https://www.optica-opn.org/home/newsroom/2020/october/infrared_imaging_adaptive_optics_spurred_nobel-wor/.

[73] RH Freeman and James E Pearson. "Deformable mirrors for all seasons and reasons". In: *Applied Optics* 21.4 (1982), pp. 580–588.

[74] Thomas Bifano. "MEMS deformable mirrors". In: *Nature photonics* 5.1 (2011), pp. 21–23.

[75] Gleb Vdovin and PM Sarro. "Flexible mirror micromachined in silicon". In: *Applied optics* 34.16 (1995), pp. 2968–2972.

[76] Roberto Ragazzoni, Emiliano Diolaiti, and Elise Vernet. "A pyramid wavefront sensor with no dynamic modulation". In: *Optics communications* 208.1-3 (2002), pp. 51–60.

[77] Mark AA Neil, Martin J Booth, and Tony Wilson. "Closed-loop aberration correction by use of a modal Zernike wave-front sensor". In: *Optics letters* 25.15 (2000), pp. 1083–1085.

[78] Alexander D Corbett et al. "Designing a holographic modal wavefront sensor for the detection of static ocular aberrations". In: *JOSA A* 24.5 (2007), pp. 1266–1275.

[79] Daniel R Neal, James Copland, and David A Neal. "Shack-Hartmann wavefront sensor precision and accuracy". In: *Advanced Characterization Techniques for Optical, Semiconductor, and Data Storage Components*. Vol. 4779. SPIE. 2002, pp. 148–160.

[80] Richard W Wilson. "SLODAR: measuring optical turbulence altitude with a Shack–Hartmann wavefront sensor". In: *Monthly Notices of the Royal Astronomical Society* 337.1 (2002), pp. 103–108.

[81] Maham Aftab et al. "Adaptive Shack-Hartmann wavefront sensor accommodating large wavefront variations". In: *Optics Express* 26.26 (2018), pp. 34428–34441.

[82] Charlotte Z Bond et al. "Adaptive optics with an infrared pyramid wavefront sensor". In: *Adaptive Optics Systems VI*. Vol. 10703. SPIE. 2018, pp. 642–652.

[83] Victoria Hutterer, Ronny Ramlau, and Iuliia Shatokhina. "Real-time adaptive optics with pyramid wavefront sensors: part I. A theoretical analysis of the pyramid sensor model". In: *Inverse Problems* 35.4 (2019), p. 045007.

[84] Andreas Zepp et al. "Optimization of the holographic wavefront sensor for open-loop adaptive optics under realistic turbulence. Part I: simulations". In: *Applied Optics* 60.22 (2021), F88–F98.

[85] *A Brief History Of Shack-Hartmann Wavefront Sensing*. https://www.axiomoptics.com/blog/a-comprehensive-history-of-shack-hartmann-wavefront-sensing/. 2024.

[86] Brent L Ellerbroek and Curtis R Vogel. "Inverse problems in astronomical adaptive optics". In: *Inverse Problems* 25.6 (2009), p. 063001.

[87] J Primot, Gi Rousset, and JC Fontanella. "Deconvolution from wave-front sensing: a new technique for compensating turbulence-degraded images". In: *JOSA A* 7.9 (1990), pp. 1598–1608.

[88] Guang-ming Dai. "Modified Hartmann-Shack wavefront sensing and iterative wavefront reconstruction". In: *Adaptive Optics in Astronomy*. Vol. 2201. SPIE. 1994, pp. 562–573.

[89] D Russell Luke, James V Burke, and Richard G Lyon. "Optical wavefront reconstruction: Theory and numerical methods". In: *SIAM review* 44.2 (2002), pp. 169–224.

[90] Robert K Tyson. *Introduction to adaptive optics*. Vol. 41. SPIE press, 2000.

[91] John W Hardy. *Adaptive optics for astronomical telescopes*. Vol. 16. Oxford University Press on Demand, 1998.

[92] Jacques M Beckers. "Adaptive optics for astronomy-Principles, performance, and applications". In: *Annual review of astronomy and astrophysics* 31 (1993), pp. 13–62.

[93] Robin Swanson et al. "Wavefront reconstruction and prediction with convolutional neural networks". In: *Adaptive Optics Systems VI*. Vol. 10703. SPIE. 2018, pp. 481–490.

[94] Robert K Tyson. *Adaptive optics engineering handbook*. Marcel Dekker New York, 2000.

[95] Rachel E Morgan et al. "MEMS deformable mirrors for space-based high-contrast imaging". In: *Micromachines* 10.6 (2019), p. 366.

[96] Kerri L Cahoy et al. "Wavefront control in space with MEMS deformable mirrors for exoplanet direct imaging". In: *Journal of Micro/Nanolithography, MEMS, and MOEMS* 13.1 (2014), pp. 011105–011105.

[97] Isaku Kanno et al. "Development of deformable mirror composed of piezoelectric thin films for adaptive optics". In: *IEEE journal of selected topics in quantum electronics* 13.2 (2007), pp. 155–161.

[98] Chunlin Guan et al. "Piezoelectric deformable mirror technologies for astronomy at IOE, CAS". In: *Adaptive Optics Systems IV*. Vol. 9148. SPIE. 2014, pp. 141–151.

[99] Dustin C Johnston and Byron M Welsh. "Analysis of multiconjugate adaptive optics". In: *JOSA A* 11.1 (1994), pp. 394–408.

[100] Francois J Rigaut, Brent L Ellerbroek, and Ralf Flicker. "Principles, limitations, and performance of multiconjugate adaptive optics". In: *Adaptive Optical Systems Technology*. Vol. 4007. SPIE. 2000, pp. 1022–1031.

[101] Ben C Platt and Roland Shack. *History and principles of Shack-Hartmann wavefront sensing*. 2001.

[102] Theam Yong Chew, Richard M Clare, and Richard G Lane. "A comparison of the Shack–Hartmann and pyramid wavefront sensors". In: *Optics communications* 268.2 (2006), pp. 189–195.

[103] Wei Liu et al. "Performance analysis of coherent free space optical communications with sequential pyramid wavefront sensor". In: *Optics & Laser Technology* 100 (2018), pp. 332–341.

[104] Charlotte Z Bond et al. "Adaptive optics with an infrared pyramid wavefront sensor at Keck". In: *Journal of Astronomical Telescopes, Instruments, and Systems* 6.3 (2020), p. 039003.

[105] A Tozzi et al. "The double pyramid wavefront sensor for LBT". In: *Adaptive Optics Systems*. Vol. 7015. SPIE. 2008, pp. 1454–1462.

[106] V Deo et al. "A telescope-ready approach for modal compensation of pyramid wavefront sensor optical gain". In: *Astronomy & Astrophysics* 629 (2019), A107.

[107] Tariq Samad. "A survey on industry impact and challenges thereof [technical activities]". In: *IEEE Control Systems Magazine* 37.1 (2017), pp. 17–18.

[108] Zhizheng Wu, Azhar Iqbal, and Foued Ben Amara. "LMI-based multivariable PID controller design and its application to the control of the surface shape of magnetic fluid deformable mirrors". In: *IEEE Transactions on Control Systems Technology* 19.4 (2010), pp. 717–729.

[109] Xizheng Ke and Danyu Zhang. "Fuzzy control algorithm for adaptive optical systems". In: *Applied optics* 58.36 (2019), pp. 9967–9975.

[110] Youming Guo et al. "Adaptive optics based on machine learning: a review". In: *Opto-Electronic Advances* 5.7 (2022), pp. 200082–1.

[111] Jiaying Wang et al. "Experimental demonstration of LQG control with disturbance mitigation on multiple modes in adaptive optics system". In: *Optik* 202 (2020), p. 163594.

[112] Cyril Petit et al. "First laboratory validation of vibration filtering with LQG control law for adaptive optics". In: *Optics Express* 16.1 (2008), pp. 87–97.

[113] James Steven Gibson, Chi-Chao Chang, and Brent L Ellerbroek. "Adaptive optics: wave-front correction by use of adaptive filtering and control". In: *Applied optics* 39.16 (2000), pp. 2525–2538.

[114] Jonathan Tesch and Steve Gibson. "Optimal and adaptive control of aero-optical wavefronts for adaptive optics". In: *JOSA A* 29.8 (2012), pp. 1625–1638.

[115] Brice Le Roux et al. "Optimal control law for classical and multiconjugate adaptive optics". In: *JOSA A* 21.7 (2004), pp. 1261–1276.

[116] Jimmie J Perez, Gregory J Toussaint, and Jason D Schmidt. "Adaptive control of woofer-tweeter adaptive optics". In: *Advanced Wavefront Control: Methods, Devices, and Applications VII*. Vol. 7466. SPIE. 2009, pp. 108–119.

[117] Sebastiaan Y Haffert et al. "Data-driven subspace predictive control of adaptive optics for high-contrast imaging". In: *Journal of Astronomical Telescopes, Instruments, and Systems* 7.2 (2021), pp. 029001–029001.

[118] Maaike AM van Kooten et al. "Predictive wavefront control on Keck II adaptive optics bench: on-sky coronagraphic results". In: *Journal of Astronomical Telescopes, Instruments, and Systems* 8.2 (2022), pp. 029006–029006.

[119] Martin Glück, Jörg-Uwe Pott, and Oliver Sawodny. "Model predictive control of multi-mirror adaptive optics systems". In: *2018 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE. 2018, pp. 909–914.

[120] Olivier Guyon and Jared Males. "Adaptive optics predictive control with empirical orthogonal functions (EOFs)". In: *arXiv preprint arXiv:1707.00570* (2017).

[121] Luke C Johnson, Donald T Gavel, and Donald M Wiberg. "Bulk wind estimation and prediction for adaptive optics control systems". In: *JOSA A* 28.8 (2011), pp. 1566–1577.

[122] Krzysztof Patan. "Neural network-based model predictive control: Fault tolerance and stability". In: *IEEE Transactions on Control Systems Technology* 23.3 (2014), pp. 1147–1155.

[123] Alison P Wong et al. "Predictive control for adaptive optics using neural networks". In: *Journal of Astronomical Telescopes, Instruments, and Systems* 7.1 (2021), pp. 019001–019001.

[124] Olivier Guyon. "Limits of adaptive optics for high-contrast imaging". In: *The Astrophysical Journal* 629.1 (2005), p. 592.

[125] Cedric Taïssir Heritier et al. "A new calibration strategy for adaptive telescopes with pyramid WFS". In: *Monthly Notices of the Royal Astronomical Society* 481.2 (2018), pp. 2829–2840.

[126] Jalo Nousiainen et al. "Advances in model-based reinforcement learning for adaptive optics control". In: *Adaptive Optics Systems VIII*. Vol. 12185. SPIE. 2022, pp. 882–891.

[127] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*. Vol. 1. 1. MIT press Cambridge, 1998.

[128] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[129] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets". In: *Neural computation* 18.7 (2006), pp. 1527–1554.

[130] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. "Learning long-term dependencies with gradient descent is difficult". In: *IEEE transactions on neural networks* 5.2 (1994), pp. 157–166.

[131] Ilya Sutskever, James Martens, and Geoffrey E Hinton. "Generating text with recurrent neural networks". In: *Proceedings of the 28th international conference on machine learning (ICML-11)*. 2011, pp. 1017–1024.

[132] Tomas Mikolov et al. "Learning longer memory in recurrent neural networks". In: *arXiv preprint arXiv:1412.7753* (2014).

[133] Hojjat Salehinejad et al. "Recent advances in recurrent neural networks". In: *arXiv preprint arXiv:1801.01078* (2017).

[134] Alex Sherstinsky. "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network". In: *Physica D: Nonlinear Phenomena* 404 (2020), p. 132306.

[135] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[136] Yong Yu et al. "A review of recurrent neural networks: LSTM cells and network architectures". In: *Neural computation* 31.7 (2019), pp. 1235–1270.

[137] Ying Chen. "LSTM recurrent neural network prediction algorithm based on Zernike modal coefficients". In: *Optik* 203 (2020), p. 163796.

[138] Xuewen Liu et al. "Wavefront prediction using artificial neural networks for open-loop adaptive optics". In: *Monthly Notices of the Royal Astronomical Society* 496.1 (2020), pp. 456–464.

[139] Moez Krichen. "Convolutional neural networks: A survey". In: *Computers* 12.8 (2023), p. 151.

[140] Robin Swanson et al. "Closed loop predictive control of adaptive optics systems with convolutional neural networks". In: *Monthly Notices of the Royal Astronomical Society* 503.2 (2021), pp. 2944–2954.

[141] Hesam Hassanpour, Brandon Corbett, and Prashant Mhaskar. "Artificial neural network-based model predictive control using correlated data". In: *Industrial & Engineering Chemistry Research* 61.8 (2022), pp. 3075–3090.

[142] Zhe Wu et al. "Machine learning-based predictive control using noisy data: evaluating performance and robustness via a large-scale process simulator". In: *Chemical Engineering Research and Design* 168 (2021), pp. 275–287.

[143] Sebastien Gros and Mario Zanon. "Learning for MPC with stability & safety guarantees". In: *Automatica* 146 (2022), p. 110598.

[144] Troy R Ellis and Jason D Schmidt. "Wavefront sensor performance in strong turbulence with an extended beacon". In: *2010 IEEE Aerospace Conference*. IEEE. 2010, pp. 1–10.

[145] Jeffrey D Barchers et al. "Performance of wavefront sensors in strong scintillation". In: *Adaptive Optical System Technologies II*. Vol. 4839. SPIE. 2003, pp. 217–227.

[146] Carlos E Carrizo, Ramon Mata Calvo, and Aniceto Belmonte. "Intensity-based adaptive optics with sequential optimization for laser communications". In: *Optics express* 26.13 (2018), pp. 16044–16053.

[147] Gerard Theodore van Belle, Aden Baker Meinel, and Marjorie Pettit Meinel. "The scaling relationship between telescope cost and aperture size for very large telescopes". In: *Ground-based Telescopes*. Ed. by Jacobus M. Oschmann Jr. Vol. 5489. International Society for Optics and Photonics. SPIE, 2004, pp. 563–570. DOI: 10.1117/12.552181. URL: https://doi.org/10.1117/12.552181.

[148] *Astro Systeme Austria Products*. https://www.astrosysteme.com/products. Feb. 2022.

[149] *PlaneWave Instruments Observatory Systems*. https://planewave.com/observatory-systems. Feb. 2022.

[150] Xu He et al. "A rapid hybrid wave front correction algorithm for sensor-less adaptive optics in free space optical communication". In: *Optics Communications* 429 (2018), pp. 127–137.

[151] Jiaxun Li et al. "A Novel SPGD Algorithm for Wavefront Sensorless Adaptive Optics System". In: *IEEE Photonics Journal* 15.4 (2023), pp. 1–9.

[152] Huizhen Yang, Zhen Zhang, and Jian Wu. "Performance comparison of wavefront-sensorless adaptive optics systems by using of the focal plane". In: *International Journal of Optics* 2015 (2015).

[153] Hongxi Ren and Bing Dong. "Improved model-based wavefront sensorless adaptive optics for extended objects using N+ 2 images". In: *Optics Express* 28.10 (2020), pp. 14414–14427.

[154] Wen Lianghua et al. "A high speed model-based approach for wavefront sensorless adaptive optics systems". In: *Optics & Laser Technology* 99 (2018), pp. 124–132.

[155] Wen Lianghua et al. "Synchronous model-based approach for wavefront sensorless adaptive optics system". In: *Optics express* 25.17 (2017), pp. 20584–20597.

[156] Huizhen Yang, Oleg Soloviev, and Michel Verhaegen. "Model-based wavefront sensorless adaptive optics system for large aberrations and extended objects". In: *Optics express* 23.19 (2015), pp. 24587–24601.

[157] Huang Linhai and Changhui Rao. "Wavefront sensorless adaptive optics: a general model-based approach". In: *Optics express* 19.1 (2010), pp. 371–379.

[158] Ming Liu and Bing Dong. "Efficient wavefront sensorless adaptive optics based on large dynamic crosstalk-free holographic modal wavefront sensing". In: *Optics Express* 30.6 (2022), pp. 9088–9102.

[159] Dan Yue and Haitao Nie. "Wavefront sensorless adaptive optics system for extended objects based on linear phase diversity technique". In: *Optics Communications* 475 (2020), p. 126209.

[160] Dan Yue et al. "Fast correction approach for wavefront sensorless adaptive optics based on a linear phase diversity technique". In: *Applied Optics* 57.7 (2018), pp. 1650–1656.

[161] Hongxi Ren and Bing Dong. "Self-calibrated general model-based wavefront sensorless adaptive optics for both point-like and extended objects". In: *Optics Express* 30.6 (2022), pp. 9562–9577.

[162] Zhaoying Zheng et al. "Analysis and demonstration of PID algorithm based on arranging the transient process for adaptive optics". In: *Chinese Optics Letters* 11.11 (2013), p. 110101.

[163] Zhaokun Li et al. "Atmospheric compensation in free space optical communication with simulated annealing algorithm". In: *Optics Communications* 338 (2015), pp. 11–21.

[164] Liangliang Han, Yinkang Dai, and Yang Qiu. "Compensation for aberrant wavefront in UOWC based on adaptive optics technique employing genetic algorithm". In: *Optik* 281 (2023), p. 170832.

[165] Mikhail A Vorontsov and Viktor P Sivokon. "Stochastic parallel-gradient-descent technique for high-resolution wave-front phase-distortion correction". In: *JOSA A* 15.10 (1998), pp. 2745–2758.

[166] Mikhail A Vorontsov and Gary W Carhart. "Adaptive wavefront control with asynchronous stochastic parallel gradient descent clusters". In: *JOSA A* 23.10 (2006), pp. 2613–2622.

[167] Wang Xiong et al. "Numerical simulation of tilt-tip control in coherent beam combining using SPGD algorithm". In: *Optics & Laser Technology* 48 (2013), pp. 343–350.

[168] Kenan Wu et al. "Multi-perturbation stochastic parallel gradient descent method for wavefront correction". In: *Optics Express* 23.3 (2015), pp. 2933–2944.

[169] Jingtai Cao et al. "Stochastic parallel gradient descent laser beam control algorithm for atmospheric compensation in free space optical communication". In: *Optik* 125.20 (2014), pp. 6142–6147.

[170] Hui Zhao et al. "Nesterov-accelerated adaptive momentum estimation-based wavefront distortion correction algorithm". In: *Applied Optics* 60.24 (2021), pp. 7177–7185.

[171] Guoqing Yang et al. "Improved SPGD algorithm to avoid local extremum for incoherent beam combining". In: *Optics Communications* 382 (2017), pp. 547–555.

[172] Qintao Hu et al. "Adaptive stochastic parallel gradient descent approach for efficient fiber coupling". In: *Optics Express* 28.9 (2020), pp. 13141–13154.

[173] Martin J Booth. "Wave front sensor-less adaptive optics: a model-based approach using sphere packings". In: *Optics express* 14.4 (2006), pp. 1339–1352.

[174] Hu Ke et al. "Self-learning control for wavefront sensorless adaptive optics system through deep reinforcement learning". In: *Optik* 178 (2019), pp. 785–793.

[175] Rico Landman et al. "Self-optimizing adaptive optics control with reinforcement learning". In: *Adaptive Optics Systems VII*. Vol. 11448. SPIE. 2020, pp. 842–856.

[176] Rico Landman et al. "Self-optimizing adaptive optics control with reinforcement learning for high-contrast imaging". In: *Journal of Astronomical Telescopes, Instruments, and Systems* 7.3 (2021), p. 039002.

[177] Huimin Ma et al. "Numerical study of adaptive optics compensation based on convolutional neural networks". In: *Optics Communications* 433 (2019), pp. 283–289.

[178] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[179] Mingxing Tan and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.

[180] Yushuang Li, Dan Yue, and Yihao He. "Prediction of wavefront distortion for wavefront sensorless adaptive optics based on deep learning". In: *Applied Optics* 61.14 (2022), pp. 4168–4176.

[181] Yunlong Song et al. "Reaching the limit in autonomous racing: Optimal control versus reinforcement learning". In: *Science Robotics* 8.82 (2023), eadg1462.

[182] Yagang Zhang. *New advances in machine learning*. BoD–Books on Demand, 2010.

[183] Manuel Lopes et al. "Exploration in model-based reinforcement learning by empirically estimating learning progress". In: *Advances in neural information processing systems* 25 (2012).

[184] Frédérick Garcia and Emmanuel Rachelson. "Markov decision processes". In: *Markov Decision Processes in Artificial Intelligence* (2013), pp. 1–38.

[185] Frank L Lewis, Draguna Vrabie, and Vassilis L Syrmos. *Optimal control*. John Wiley & Sons, 2012.

[186] Tuomas Haarnoja et al. "Soft actor-critic algorithms and applications". In: *arXiv preprint arXiv:1812.05905* (2018).

[187] Petros Christodoulou. "Soft actor-critic for discrete action settings". In: *arXiv preprint arXiv:1910.07207* (2019).

[188] Solomon Kullback and Richard A Leibler. "On information and sufficiency". In: *The annals of mathematical statistics* 22.1 (1951), pp. 79–86.

[189] David Silver et al. "Deterministic policy gradient algorithms". In: *International conference on machine learning*. Pmlr. 2014, pp. 387–395.

[190] George E Uhlenbeck and Leonard S Ornstein. "On the theory of the Brownian motion". In: *Physical review* 36.5 (1930), p. 823.

[191] Guodong Zhang et al. "Three mechanisms of weight decay regularization". In: *arXiv preprint arXiv:1810.12281* (2018).

[192] Mahyar Fazlyab et al. "Efficient and accurate estimation of lipschitz constants for deep neural networks". In: *Advances in neural information processing systems* 32 (2019).

[193] Takeru Miyato et al. "Spectral normalization for generative adversarial networks". In: *arXiv preprint arXiv:1802.05957* (2018).

[194] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems* 27 (2014).

[195] Ryoichi Takase et al. "Stability-certified reinforcement learning control via spectral normalization". In: *Machine Learning with Applications* 10 (2022), p. 100409.

[196] Yi-Lun Wu et al. "Gradient normalization for generative adversarial networks". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 6373–6382.

[197] Xujie Song et al. "LipsNet: a smooth and robust neural network with adaptive Lipschitz constant for high accuracy optimal control". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 32253–32272.

[198] Wenxuan Wang et al. "Smooth Filtering Neural Network for Reinforcement Learning". In: *IEEE Transactions on Intelligent Vehicles* (2024).

[199] Shubham Pateria et al. "Hierarchical reinforcement learning: A comprehensive survey". In: *ACM Computing Surveys (CSUR)* 54.5 (2021), pp. 1–35.

[200] Chen Chen et al. "Addressing action oscillations through learning policy inertia". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 8. 2021, pp. 7020–7027.

[201] Pierre Thodoroff et al. "Temporal regularization for markov decision process". In: *Advances in Neural Information Processing Systems* 31 (2018).

[202] Taisuke Kobayashi. "L2c2: Locally lipschitz continuous constraint towards stable and smooth reinforcement learning". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2022, pp. 4032–4039.

[203] Andrew Ilyas et al. "A closer look at deep policy gradients". In: *arXiv preprint arXiv:1811.02553* (2018).

[204] Qianli Shen et al. "Deep reinforcement learning with robust and smooth policy". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 8707–8718.

[205] Hoang-Giang Cao et al. "Image-based regularization for action smoothness in autonomous miniature racing car with deep reinforcement learning". In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2023, pp. 5179–5186.

[206] Gerasimos Rigatos and Masoud Abbaszadeh. "Nonlinear optimal control for multi-DOF robotic manipulators with flexible joints". In: *Optimal Control Applications and Methods* 42.6 (2021), pp. 1708–1733.

[207] Janis Arents and Modris Greitans. "Smart industrial robot control trends, challenges and opportunities within manufacturing". In: *Applied Sciences* 12.2 (2022), p. 937.

[208] Shian Wang, Raphael Stern, and Michael W Levin. "Optimal control of autonomous vehicles for traffic smoothing". In: *IEEE Transactions on Intelligent Transportation Systems* 23.4 (2021), pp. 3842–3852.

[209] Szilárd Aradi. "Survey of deep reinforcement learning for motion planning of autonomous vehicles". In: *IEEE Transactions on Intelligent Transportation Systems* 23.2 (2020), pp. 740–759.

[210] Ji Woong Kim et al. "Towards autonomous eye surgery by combining deep imitation learning with optimal control". In: *Conference on Robot Learning*. PMLR. 2021, pp. 2347–2358.

[211] Volodymyr Mnih et al. "Human-level control through deep reinforcement learning". In: *nature* 518.7540 (2015), pp. 529–533.

[212] Yuhui Wang, Hao He, and Xiaoyang Tan. "Truly proximal policy optimization". In: *Uncertainty in artificial intelligence*. PMLR. 2020, pp. 113–122.

[213] Greg Brockman et al. "Openai gym". In: *arXiv preprint arXiv:1606.01540* (2016).

[214] Bitcraze AB. *Crazyflie 2.1*. https://store.bitcraze.io/products/crazyflie-2-1. Accessed: April 18, 2025. 2019.

[215] Matt Jordan and Alexandros G Dimakis. "Exactly computing the local lipschitz constant of relu networks". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 7344–7353.

[216] Bitcraze AB. *Flow Deck v2*. https://www.bitcraze.io/products/flow-deck-v2/. Accessed: April 18, 2025.

[217] Bosch Sensortec GmbH. *BMI088 – High-Performance Inertial Measurement Unit (IMU)*. BST-BMI088-DS001-13. Datasheet, Revision 1.3. Bosch Sensortec. 2021. URL: https://bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmi088-ds001.pdf.

[218] Bosch Sensortec GmbH. *BMP388 – Digital Pressure Sensor*. BST-BMP388-DS001-07. Datasheet, Revision 1.7. Bosch Sensortec. 2020. URL: https://bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmp388-ds001.pdf.

[219] STMicroelectronics. *VL53L1X – Time-of-Flight Ranging Sensor*. DS12385. Datasheet, Revision 8. STMicroelectronics. 2024. URL: https://www.st.com/resource/en/datasheet/vl53l1x.pdf.

[220] Bitcraze AB Forums. *Latency Measurement Between PC and Crazyflie*. Bitcraze AB. 2021. URL: https://forum.bitcraze.io/viewtopic.php?t=4691.

[221] Bitcraze AB. *Crazyradio PA – 2.4 GHz USB Radio Dongle*. Product documentation, Revision 3. Bitcraze AB. 2022. URL: https://www.bitcraze.io/products/crazyradio-pa/.

[222] Dániel Horváth et al. "Object detection using sim2real domain randomization for robotic applications". In: *IEEE Transactions on Robotics* 39.2 (2022), pp. 1225–1243.

[223] Emiel H Por et al. "High Contrast Imaging for Python (HCIPy): an open-source adaptive optics and coronagraph simulator". In: *Adaptive Optics Systems VI*. Vol. 10703. SPIE. 2018, pp. 1112–1125.

[224] David L Fried. "Optical resolution through a randomly inhomogeneous medium for very long and very short exposures". In: *Journal of the Optical Society of America* 56.10 (1966), pp. 1372–1379.

[225] Hanyu Zhan, Erandi Wijerathna, and David Voelz. "Is the formulation of the fried parameter accurate in the strong turbulent scattering regime?" In: *OSA Continuum* 3.9 (2020), pp. 2653–2659.

[226] Eakkachai Pengwang et al. "Scanning micromirror platform based on MEMS technology for medical application". In: *Micromachines (Basel)* 7.2 (Feb. 2016), p. 24.

[227] Lin Yang and Mengdi Wang. "Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 10746–10756.

[228] N Mark Milton and Michael Lloyd-Hart. "Disk harmonic functions for adaptive optics simulations". In: *Adaptive Optics: Methods, Analysis and Applications*. Optica Publishing Group. 2005, AWA3.

[229] Qinghua Liu et al. "When is partially observable reinforcement learning not scary?" In: *Conference on Learning Theory*. PMLR. 2022, pp. 5175–5220.

[230] Heidi Ottevaere and Hugo Thienpont. "Refractive optical microlenses: an introduction to nomenclature and characterization techniques". In: *Encyclopedia of Modern Optics*. Vol. 4. Elsevier, 2004, pp. 21–43.

[231] Daniel Sage et al. "DeconvolutionLab2: An open-source software for deconvolution microscopy". In: *Methods* 115 (2017), pp. 28–41.

[232] N Jovanovic et al. "Efficient injection from large telescopes into single-mode fibres: Enabling the era of ultra-precision astronomy". In: *Astronomy & Astrophysics* 604 (2017), A122.

[233] Virendra N Mahajan. "Strehl ratio for primary aberrations in terms of their aberration variance". In: *JOSA* 73.6 (1983), pp. 860–861.

[234] Cyril Ruilier and Frédéric Cassaing. "Coupling of large telescopes and single-mode waveguides: application to stellar interferometry". In: *JOSA A* 18.1 (2001), pp. 143–149.

[235] Zhou Wang et al. "Image quality assessment: from error visibility to structural similarity". In: *IEEE transactions on image processing* 13.4 (2004), pp. 600–612.

[236] Thorlabs, Inc. *Quadrant Photodetectors*. https://www.thorlabs.com/NewGroupPage9.cfm?ObjectGroup_ID=4400. Accessed: 2025-04-24.

[237] Ziqiang Li and Xinyang Li. "Centroid computation for Shack-Hartmann wavefront sensor in extreme situations based on artificial neural networks". In: *Optics Express* 26.24 (2018), pp. 31675–31692.

[238] Jiayi Weng et al. "Tianshou: A highly modularized deep reinforcement learning library". In: *Journal of Machine Learning Research* 23.267 (2022), pp. 1–6.

[239] Lili Chen et al. "Decision transformer: Reinforcement learning via sequence modeling". In: *Advances in neural information processing systems* 34 (2021), pp. 15084–15097.

[240] Faraz Torabi, Garrett Warnell, and Peter Stone. "Behavioral cloning from observation". In: *arXiv preprint arXiv:1805.01954* (2018).

[241] Philip J Ball et al. "Efficient online reinforcement learning with offline data". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 1577–1594.

[242] Gengchen Liu et al. "Hierarchical learning for cognitive end-to-end service provisioning in multi-domain autonomous optical networks". In: *Journal of Lightwave Technology* 37.1 (2019), pp. 218–225.

[243] Cedric Taissir Heritier, Christophe Verinaud, and Carlos M Correia. "Oopao: Object oriented python adaptive optics". In: *Adaptive Optics for Extremely Large Telescopes 7th Edition*. 2023.

[244] Guillaume Matheron, Nicolas Perrin, and Olivier Sigaud. "The problem with DDPG: understanding failures in deterministic environments with sparse rewards". In: *arXiv preprint arXiv:1911.11679* (2019).

[245] Luíza Caetano Garaffa et al. "Reinforcement learning for mobile robotics exploration: A survey". In: *IEEE Transactions on Neural Networks and Learning Systems* 34.8 (2021), pp. 3796–3810.