university of groningen

faculty of science and engineering

ARTIFICIAL INTELLIGENCE

MASTER'S THESIS

THE PRIMACY BIAS THROUGH THE LENS OF THE FISHER INFORMATION MATRIX

MASSIMILIANO FALZARI

MSc in Artificial Intelligence
Science and Engineering
University of Groningen

Supervisor
Dr. Matthia Sabatelli
Artificial Intelligence
University of Groningen

November, 2024

ABSTRACT

Deep Reinforcement Learning (DRL) systems exhibit a significant tendency to overfit to early experiences, a phenomenon known as the Primacy Bias (PB). This bias can severely impact learning efficiency and final performance, particularly in complex environments. This thesis presents a comprehensive investigation of the PB through the lens of the Fisher Information Matrix (FIM) and introduces Fisher-Guided Selective Forgetting (FGSF), a novel method for its mitigation.

We first develop a theoretical framework characterizing the PB through distinctive patterns in the FIM trace, identifying critical memorization and reorganization phases during learning. Building on this understanding, we propose FGSF, which leverages the geometric structure of the parameter space to selectively modify network weights, preventing early experiences from dominating the learning process while preserving valuable knowledge.

Through extensive empirical evaluation across multiple environments from the DeepMind Control Suite (DMC), we demonstrate that FGSF consistently outperforms baseline approaches, particularly in complex, high-dimensional tasks. Our analysis reveals several key insights: (1) the PB affects critic networks more severely than actor networks, with critic-only intervention often outperforming full network scrubbing, (2) FGSF's effectiveness scales with task complexity and replay ratio, suggesting particular utility in challenging learning scenarios, (3) the method maintains robust performance across different hyperparameter settings while introducing minimal computational overhead, and (4) even simple noise injection methods can provide meaningful improvements, indicating that the PB may be fundamentally linked to optimization dynamics.

These findings not only advance our understanding of the PB but also provide practical tools for its mitigation, contributing to the development of more efficient and robust DRL systems. The geometric perspective offered by our FIM-based analysis opens new avenues for understanding and addressing learning dynamics in deep neural networks.

ACKNOWLEDGMENTS

First and foremost, I would like to express my gratitude to my supervisor, Matthia. His guidance throughout this research has been invaluable, but what truly made this journey special was his friendly and personal approach. Our chess games and conversations beyond academia not only made this experience more enjoyable but also helped foster an environment where ideas could flourish naturally. His ability to balance professional mentorship with genuine friendship has shown me what academic supervision at its best can be.

I want to thank my parents from the bottom of my heart. Their constant belief in me and their support have made all of this possible. They have been there for every up and down, helping me in every way they could. Without their encouragement and dedication to my education, I wouldn't be where I am today. Thank you for always being my foundation and for investing so much in my dreams.

A special thank you goes to my partner Simone, whose patience and support have been extraordinary. Thank you for listening to my endless explanations about the Fisher Information Matrix and Reinforcement Learning, for being there during the late nights of writing, and for always believing in me. Your ability to engage with my "crazy theories" while keeping me grounded has been more valuable than you know.

This thesis represents not just my work at the keyboard, but a milestone made possible by these remarkable people who have supported and believed in me throughout this journey.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ML	Machine Learning
AI	Artificial Intelligence
SL	Supervised Learning
UL	Unsupervised Learning
RL	Reinforcement Learning
DL	Deep Learning
PB	Primacy Bias
DRL	Deep Reinforcement Learning
MBRL	Model-Based Reinforcement Learning
FIM	Fisher Information Matrix
MDP	Markov Decision Process
SAC	Soft Actor-Critic
CL	Continual Learning
MU	Machine Unlearning
FGSF	Fisher-Guided Selective Forgetting
PlaD	Plasticity-Driven Sparsity Training
AGI	Artificial General Intelligence
EWC	Elastic Weight Consolidation
DMC	DeepMind Control Suite

1 INTRODUCTION

Machine Learning (ML) has emerged as one of the most important fields within computer science and Artificial Intelligence (AI). It focuses on the development of algorithms and statistical models that enable systems to perform specific tasks effectively without using explicit instructions, relying instead on patterns and inference (Mitchell, 1997). There are three main paradigms within ML, each tailored to specific types of problems and data structures: Supervised Learning (SL), Unsupervised Learning (UL), Reinforcement Learning (RL).

In Supervised Learning, the algorithm is presented with input-output pairs and learns a map from inputs to outputs. A common example of SL is regression, where the goal is to predict an output variable based on one or more input variables. For instance, a Supervised Learning model that predicts the price of houses based on features like size, location, and number of bedrooms.

Unsupervised Learning, on the other hand, deals with discovering hidden patterns and structures in data without any labeled outputs. Clustering algorithms such as k-means (MacQueen, 1967) are a prime example of UL, where the algorithm aims to group similar data points together based on the Euclidian distance.

Reinforcement Learning stands apart from SL and UL as a paradigm focused on sequential decision-making under uncertainty. In RL, an agent learns to make decisions by interacting with an environment, after which it receives feedback in the form of rewards (Sutton and Barto, 1999). This framework naturally applies to a wide range of real-world scenarios, from robotic control (Tang et al., 2024) and game playing (Shao et al., 2019) to resource management (Hurtado Sánchez, Casilimas, and Caicedo Rendon, 2022) and personalized recommendations (Afsar, Crump, and Far, 2022) . Fundamentally, RL involves an agent making decisions within an environment to optimize a cumulative reward. The agent learns a policy that determines action selection in each state, with the ultimate objective of maximizing expected long-term rewards. Despite many successful application, as we delve deeper into the complexities of RL, we encounter a multitude of challenges that can influence the efficiency and efficacy of learning algorithms. One such challenge, which has gained considerable attention in recent years, is the Primacy Bias (PB) problem (Qiao, Lyu, and Li, 2023; Li et al., 2024; Nikishin et al., 2022; Obando-Ceron, Courville, and Castro, 2024; D'Oro et al., 2022).

1.1 OVERVIEW OF THE PRIMACY BIAS PROBLEM

The term "Primacy Bias" in Reinforcement Learning finds its origins in a well-established concept in cognitive science: the primacy effect. To fully comprehend the rationale behind this term, it's helpful to first understand its roots in human cognition.

In cognitive psychology, the primacy effect refers to the tendency for individuals to remember information presented at the beginning of a sequence better than the information presented in the middle. This phenomenon is often studied alongside its counterpart, the recency effect, which describes the tendency to better recall information presented at the end of a sequence. These effects were first observed in human memory studies by Ebbinghaus; Deese and Kaufman; Murdock Jr. For example, when presented with a list of words to remember, people often recall the first few words (primacy effect) and the last few words (recency effect) more easily than those in the middle. This pattern of memory retention has been attributed to various factors. Understanding these cognitive biases has had significant implications for fields such as education (Onifade et al., 2011), marketing (Peters and Bijmolt, 1997), and user interface design (Barnes, 1992). Now, parallels are being drawn between these human cognitive tendencies and the behavior of AI systems, specifically in the RL framework.

In the context of RL, the PB problem refers to a phenomenon similar to its cognitive counterpart. Just as humans may overemphasize early information, Deep Reinforcement Learning (DRL) (which is an extension of the RL framework that uses Neural Networks) agents have shown a tendency to overfit early experiences. This bias can have profound implications on an agent's learning process and decision-making capabilities. The problem's importance stems from its following consequences:

- Hindered Learning: PB can obstruct an agent's ability to effectively learn from subsequent interactions.
- Suboptimal Decision-Making: As the agent becomes overly influenced by its early experiences, it may make spurious decisions that are not well-suited to its current environment or task.
- Limited Generalization: An agent affected by the PB may struggle to generalize its learning across different environments or tasks. This limitation can significantly reduce the adaptability and the versatility of Reinforcement Learning systems, constraining their potential applications.

Moreover, given the sequential nature of the Reinforcement Learning framework, these consequences have a negative cascade effect on the learning process. The importance of addressing the Primacy Bias problem extends beyond mere academic interest. As we increasingly deploy AI systems based on RL in complex, real-world scenarios - from autonomous vehicles navigating busy streets to recommender systems shaping our online experiences - the need for these systems to learn efficiently, adapt quickly, and make robust decisions becomes crucial.

1.2 THESIS OBJECTIVES AND CONTRIBUTIONS

This thesis aims to explore the Primacy Bias problem in Deep Reinforcement Learning, its underlying causes, and propose a mitigation strategy. By examining this critical aspect of RL, we hope to contribute to the development of more efficient and adaptable RL agents. The primary objectives of this research are:

- To provide a comprehensive analysis of the Primacy Bias phenomenon in DRL, including its potential causes and implications for RL agent's performance.
- To explore the connection between the Fisher Information Matrix (FIM) and the PB in DRL, offering new insights into the learning dynamics of these systems.
- To develop and evaluate a novel FIM-based mechanism for alleviating the PB in DRL agents.
- To compare the effectiveness of the proposed FIM-based approach with existing methods for mitigating the PB.

1.3 OUTLINE OF THE THESIS STRUCTURE

This thesis is organized into six chapters, that present the analysis of the PB problem and examine a FIM-based approach to address it.

Chapter 2, Theoretical Framework, provides the mathematical foundations necessary for understanding the PB in DRL. It begins with a systematic development of Reinforcement Learning fundamentals, from basic Markov Decision Process (MDP) to modern DRL architectures, with special attention to the Soft Actor-Critic (SAC) framework. The chapter then explores the mathematical foundations of the Fisher Information Matrix, establishing the theoretical tools needed for our analysis.

Chapter 3, Literature Review, examines key challenges and advances in DRL related to the PB. It covers five main areas: PB in DRL, network plasticity, Continual Learning (CL), applications of the FIM, and Machine Unlearning (MU). This comprehensive review establishes the context for our methodological developments.

Chapter 4, Methodology, presents our approach to investigating and addressing the PB. It introduces our novel characterization of the PB through FIM analysis, details the Fisher-Guided Selective Forgetting (FGSF) method, and describes our experimental framework for evaluating the effectiveness of our approach across different environments and conditions.

Chapter 5, Results and Discussion, provides a thorough empirical evaluation of FGSF through five main investigations: comparative analysis against baseline approaches, investigation of network component specificity, robustness analysis, impact of replay ratios, and comparison with simpler noise injection methods. The chapter combines quantitative performance metrics with detailed analysis of learning dynamics through FIM traces and network plasticity measurements.

Chapter 6, Conclusion, summarizes our key findings and contributions while discussing broader implications for DRL. We reflect on both theoretical advances in understanding the PB and practical improvements in addressing it, concluding with suggestions for future research directions. This structure allows us to systematically develop our understanding of the PB while demonstrating the effectiveness of our proposed solution across both theoretical and practical dimensions.

2 THEORETICAL FRAMEWORK

This chapter establishes the mathematical foundations necessary for understanding the Primacy Bias (PB) in Deep Reinforcement Learning (DRL) and develops the theoretical tools for its analysis and mitigation. The framework we present combines elements from Reinforcement Learning (RL) theory, information geometry, and statistical learning, providing a unified perspective on how early experiences influence learning dynamics. We begin with a systematic development of RL fundamentals, progressing from basic Markov Decision Process (MDP) to modern DRL architectures. Special attention is given to the Soft Actor-Critic (SAC) framework, as it serves as the primary testbed for our investigations. This foundation is crucial for understanding how learning algorithms process and retain information from sequential experiences. The chapter then delves into the mathematical foundations of the Fisher Information Matrix (FIM), a central tool in our analysis. We explore its dual role as both a measure of parameter sensitivity and a description of the geometric structure of parameter space. This geometric perspective proves particularly valuable for understanding how networks adapt to early experiences and why certain learning patterns become entrenched. Throughout the chapter, we emphasize the connections between these theoretical components, showing how they combine to provide insights into the emergence and persistence of the PB. This framework not only explains observed phenomena but also suggests principled approaches to their mitigation, setting the stage for the methodological developments in subsequent chapters.

2.1 FOUNDATION OF REINFORCEMENT LEARNING

The RL problem can be modeled, from a mathematical standpoint, as a MDP. To develop this framework systematically, we begin with the foundational concept of Markov Chains, which serve as the mathematical core upon which everything else will be built. An MDP extends the Markov Chain by incorporating additional components necessary for decision-making under uncertainty.

2.1.1 *Markov Chains*

A Markov Chain represents a stochastic process that satisfies the Markov property, which is sometimes referred to as "memorylessness". The chain is described by two main components: a state space and a transition function. While more complex variants exist, such as Continuous-time Markov Chains with infinite state spaces or time-inhomogeneous Markov Chains with time-varying transition functions, we focus on the discrete-time, finite-state variant for clarity of exposition. Formally, a Markov Chain is characterized by a tuple $\langle \mathcal{S}, \mathcal{P} \rangle$ where:

- $\mathcal{S}$ represents the finite set of states
- $\mathcal{P} : \mathcal{S} \times \mathcal{S} \to [0, 1]$ defines the state transition probability function

The transition probabilities must satisfy two fundamental constraints:

- Non-negativity: $\mathcal{P}(s, s') \geqslant 0 \quad \forall s, s' \in \mathcal{S}$
- Unit sum for each state: $\sum_{s' \in \mathcal{S}} \mathcal{P}(s, s') = 1 \quad \forall s \in \mathcal{S}$

The Markov property, which forms the foundation of this framework, can be formalized as:

$$\mathbb{P}(S_{t+1} = s_{t+1} \mid S_t = s_t, \ldots, S_0 = s_0) = \mathbb{P}(S_{t+1} = s_{t+1} \mid S_t = s_t)$$

This property establishes that the future state depends solely on the present state, regardless of the historical path taken to reach it. This property is crucial to make the framework computationally feasible.

2.1.2 *Markov Decision Processes*

An MDP builds upon the Markov Chain framework by incorporating two additional components: a set of actions, and a reward function. This extension transforms the passive stochastic process into an interactive framework suitable for decision-making. The set of actions typically referred to as the action space ($\mathcal{A}$), can be either discrete (finite set) or continuous (infinite set). Formally, an MDP is defined as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$ where:

- $\mathcal{S}$ remains the state space from the Markov Chain
- $\mathcal{A}$ represents the action space
- $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ defines the state transition probability function
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ specifies the reward function

The transition dynamics in this enhanced framework are characterized by:

$$\mathcal{P}(s, a, s') = \mathbb{P}(S_{t+1} = s' \mid S_t = s, A_t = a)$$

The reward function, which quantifies the desirability of transitions, is defined as:

$$\mathcal{R}(s, a, s') = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a, S_{t+1} = s']$$

2.1.3 *Policies and Their Properties*

Within an MDP, a policy represents the decision-making strategy of an agent. Formally, a policy π is defined as a conditional probability distribution over actions given states:

$$\pi(a \mid s) = \mathbb{P}(A_t = a \mid S_t = s)$$

For each state, the policy must satisfy the probability constraint:

$$\sum_{a \in \mathcal{A}} \pi(a \mid s) = 1 \quad \forall s \in \mathcal{S}$$

Policies can be categorized based on two fundamental properties:

- **Determinism**: A policy is deterministic if and only if it assigns probability 1 to exactly one action in each state:

$$\pi(a \mid s) = 1 \quad \forall s \in \mathcal{S} \quad \exists! a \in \mathcal{A}$$

- **Stationarity**: A policy exhibits stationarity if it remains constant over time:

$$\pi_t(a_t|s_t) = \pi_{t+n}(a_t|s_t) \quad \forall t, n \in \mathbb{N}, s_t \in \mathcal{S}, a_t \in \mathcal{A}$$

2.1.4 *Value Functions and Bellman Equations*

The assessment of policy effectiveness in an MDP framework is accomplished through value functions, which quantify the expected cumulative discounted rewards. Two principal types of value functions exist the state-value function and the action-value function. We introduce a discount factor $\gamma \in [0, 1]$ that gives more weight to immediate rewards and progressively less weight to future rewards, reflecting both the uncertainty of future predictions and the common preference for immediate rewards over delayed ones. The state-value function V^π under a policy π measures the expected return when starting from state s and following policy π after:

$$V^\pi(s) = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s\right]$$

Complementarily, the action-value function Q^π extends this concept by considering the value of taking a specific action a in state s before following policy π:

$$Q^\pi(s, a) = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a\right]$$

These functions are fundamentally connected through the Bellman equations, which establish a recursive relationship between the value of a state and the values of its successor states. For the state-value function, this relationship is expressed as:

$$V^{\pi}(s) = \sum_{a \in \mathcal{A}} \pi(a \mid s) \sum_{s' \in \mathcal{S}} \mathcal{P}(s' \mid s, a)[R(s, a, s') + \gamma V^{\pi}(s')]$$

Similarly, for the action-value function:

$$Q^{\pi}(s, a) = \sum_{s' \in \mathcal{S}} \mathcal{P}(s' \mid s, a)[R(s, a, s') + \gamma \sum_{a' \in \mathcal{A}} \pi(a' \mid s') Q^{\pi}(s', a')]$$

These equations form the theoretical foundation for value iteration and policy improvement algorithms in RL.

2.1.5 *Deep Reinforcement Learning Framework*

The transition from traditional RL to DRL is necessitated by the limitations of tabular representations when dealing with high-dimensional state and action spaces. RL addresses this challenge by employing neural networks as function approximators for value functions and policies. In this framework, the fundamental value functions and policies are approximated as:

$$V_{\theta}(s) \approx V^{\pi}(s)$$

$$Q_{\phi}(s, a) \approx Q^{\pi}(s, a)$$

$$\pi_{\psi}(a \mid s) \approx \pi(a \mid s)$$

where θ, ϕ, and ψ represent the parameters of neural networks. These approximations transform the discrete optimization problem of traditional RL into a continuous optimization problem in the space of neural network parameters.

2.1.6 *Soft Actor-Critic Architecture*

The SAC algorithm (Haarnoja et al., 2018a) extends the traditional actor-critic architecture within the maximum entropy framework. Actor-critic methods combine two key components: an actor (policy) that determines actions, and a critic (value function) that evaluates those actions. The actor π_{ϕ} directly maps states to a probability distribution over actions, while the critic estimates value functions (Q_{ϕ} or V_{θ}) to assess the actor's decisions. This separation allows for policy optimization guided by value estimation, where the critic's feedback helps reduce the variance of policy updates while maintaining the advantages of policy gradient methods. The interaction between these networks is formalized through policy gradient updates of the form:

$$\nabla_{\phi} J(\phi) = \mathbb{E}_{\pi}\left[Q_{\theta}(s, a) \nabla_{\phi} \log \pi_{\phi}(a|s)\right]$$

SAC builds upon this foundation by incorporating entropy maximization into the standard objective, effectively balancing exploration and exploitation. The modified objective function becomes:

$$J(\pi) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^{t}(R(s_t, a_t, s_{t+1}) + \alpha\mathcal{H}(\pi(\cdot \mid s_t)))\right]$$

where $\mathcal{H}$ represents the entropy of the policy. The temperature parameter α plays a crucial role in SAC's performance, as it determines the relative importance of the entropy term against the standard reward objective. Higher values of α lead to more exploration, while lower values favor exploitation. In practice, this parameter can be automatically adjusted during training to achieve a desired target entropy level. Concretely, the SAC algorithm uses the following three functions:

$$J_Q(\theta_i) = \mathbb{E}_{(s_t,a_t)\sim\mathcal{D}}\left[\frac{1}{2}\left(Q_{\theta_i}(s_t, a_t) - (r(s_t, a_t) + \gamma\mathbb{E}_{s_{t+1}\sim\mathcal{P}}[V_{\bar{\psi}}(s_{t+1})])\right)^2\right]$$

$$J_V(\psi) = \mathbb{E}_{s_t\sim\mathcal{D}}\left[\frac{1}{2}\left(V_{\psi}(s_t) - \mathbb{E}_{a_t\sim\pi_{\phi}}[Q_{\theta}(s_t, a_t) - \alpha\log\pi_{\phi}(a_t|s_t)]\right)^2\right]$$

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

where $\mathcal{D}$ represents the replay buffer of stored transitions. This approach combines the benefits of off-policy learning with the stability and exploration benefits of entropy maximization, resulting in an algorithm that is both sample-efficient and stable across a wide range of tasks (Haarnoja et al., 2018b; Haarnoja et al., 2018a).

This mathematical framework, spanning from basic Markov Chains to advanced DRL algorithms, provides the theoretical foundation necessary for analyzing and understanding phenomena such as the PB in DRL systems.

2.2 MATHEMATICAL FOUNDATIONS OF THE FISHER INFORMATION MATRIX

The FIM emerges as a fundamental construct in both statistical learning theory and optimization. This mathematical object provides crucial insights into the geometry of parameter spaces and the nature of learning in neural networks. To develop its theoretical foundations systematically, we begin with essential concepts from information theory.

2.2.1 *Information Theory Preliminaries*

Let $p_{\theta}(x)$ denote a probability distribution parameterized by $\theta \in \mathbb{R}^{d}$. Two fundamental concepts form the basis for understanding the FIM:

the log-likelihood function and the score function. The log-likelihood function quantifies the likelihood of observing data under our model:

$$\ell(\theta; x) = \log p_\theta(x)$$

The score function, defined as the gradient of the log-likelihood with respect to the parameters, captures the sensitivity of our model to parameter changes:

$$s(\theta; x) = \nabla_\theta \ell(\theta; x) = \frac{\partial}{\partial \theta} \log p_\theta(x)$$

The FIM is fundamentally defined as the covariance of the score function:

$$F(\theta) = \mathbb{E}_{x \sim p_\theta} \left[s(\theta; x) s(\theta; x)^\top \right] = \mathbb{E}_{x \sim p_\theta} \left[\nabla_\theta \log p_\theta(x) \nabla_\theta \log p_\theta(x)^\top \right]$$

Under suitable regularity conditions, an alternative formulation exists through the negative expected Hessian of the log-likelihood:

$$F(\theta) = -\mathbb{E}_{x \sim p_\theta} \left[\nabla^2_\theta \log p_\theta(x) \right]$$

However, as opposed to the Hessian matrix, the FIM is positive semi-definite which means that for all vectors $v \in \mathbb{R}^d$:

$$v^\top F(\theta) v \geqslant 0$$

In statistical learning, the parameter space of a model naturally possesses a Riemannian structure as shown by Amari, 2016. Unlike Euclidean spaces, Riemannian manifolds are curved spaces equipped with a local notion of angles and distances that varies smoothly across the space. For any parametric model with parameters $\theta \in \Omega$, this structure is characterized by the Riemannian metric tensor $G(\theta)$, which defines an inner product in the tangent space at each point θ. For a small displacement $d\theta$ in parameter space, the squared distance is given by:

$$ds^2 = d\theta^\top G(\theta) d\theta = \sum_{i,j} g_{ij}(\theta) d\theta_i d\theta_j$$

The metric tensor $G(\theta)$ captures the local geometry of the parameter space, measuring how changes in parameters affect the model's behavior. For statistical models, the FIM provides a canonical choice for this metric: $G(\theta) = F(\theta)$. This choice is not arbitrary - it arises naturally as the unique metric that is invariant to reparametrization of the model and has deep connections to statistical efficiency through the Cramér-Rao bound.

2.2.2 *Applications in Deep Learning*

In practical Deep Learning (DL) applications, the empirical FIM for a neural network with parameters θ takes the form:

$$\hat{F}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left[\nabla_\theta \log p_\theta(x_i) \nabla_\theta \log p_\theta(x_i)^\top \right]$$

The FIM lead to several important applications in DL. One of the most famous is the natural gradient descent (Amari, 1998). While the standard gradient descent follows the direction of the steepest descent in Euclidean space, it fails to account for the Riemannian structure of the parameter space. The natural gradient provides the correct notion of the steepest descent on the statistical manifold. Consider a parameter space $S = \theta \in \mathbb{R}^n$ with a Riemannian metric $G(\theta)$. To find the steepest descent direction for optimizing a function $L(\theta)$, we solve:

$$\min_{\delta\theta} L(\theta + \delta\theta) \quad \text{subject to} \quad \delta\theta^\top G(\theta)\delta\theta = \epsilon^2$$

Using a first-order approximation and the method of Lagrange multipliers:

$$\mathcal{L}(\delta\theta, \lambda) = L(\theta) + \nabla L(\theta)^\top \delta\theta - \lambda(\delta\theta^\top G(\theta)\delta\theta - \epsilon^2)$$

This gives us the natural gradient:

$$\tilde{\nabla} L(\theta) = G(\theta)^{-1} \nabla L(\theta)$$

The natural gradient descent algorithm then takes the form:

$$\theta_{t+1} = \theta_t - \eta_t \tilde{\nabla} L(\theta_t)$$

In conclusion, the FIM represents a fundamental mathematical construct that bridges statistical learning theory and optimization. Through its dual interpretation as both the covariance of the score function and the Riemannian metric tensor of the statistical manifold, it provides crucial insights into the geometric structure of parameter spaces. Its practical significance is perhaps best exemplified in natural gradient descent, where it enables optimization methods that respect the intrinsic geometry of the parameter space. This connection between information geometry and optimization algorithms demonstrates how theoretical mathematical foundations can directly lead to practical improvements in Machine Learning (ML) applications.

3 LITERATURE REVIEW

This chapter examines the key challenges and advances in Deep Reinforcement Learning (DRL), with a particular focus on aspects relevant to the Primacy Bias (PB). We begin by exploring the PB phenomenon itself. We then delve into network plasticity, investigating how neural networks' ability to adapt changes during training and the implications for learning dynamics. The literature review continues with an examination of Continual Learning (CL), focusing on the challenges of maintaining and updating knowledge in neural networks. We then explore the Fisher Information Matrix (FIM) and its applications in Deep Learning (DL), providing the foundations for understanding learning dynamics and parameter space geometry. Finally, we investigate Machine Unlearning (MU) methods, which offer insights into how information can be selectively removed from trained models. These diverse but interconnected topics provide the theoretical foundation for our proposed approach to addressing the PB in DRL.

3.1 PRIMACY BIAS IN DEEP REINFORCEMENT LEARNING

In DRL, the PB manifests as agents' tendency to overfit to early experiences, potentially failing to learn from valuable information encountered later in the training process. Recent research by Nikishin et al. has identified this phenomenon as a significant bottleneck in developing efficient and robust DRL systems. The bias, while present also in on-policy algorithms, is particularly pronounced in off-policy algorithms that utilize experience replay buffers, where early experiences can disproportionately influence the learning trajectory of the agent (Nikishin et al., 2022).

3.1.1 *Mechanisms and Potential Causes*

Recent studies (Abbas et al., 2023; Lyle et al., 2022; Lyle, Rowland, and Dabney, 2022) have revealed several interconnected mechanisms that might be contributing to the PB in DRL. At its core, the phenomenon arises from the interaction between neural network learning dynamics and the non-stationarity nature of Reinforcement Learning (RL).

One key finding is that DRL agents using neural networks can gradually lose their ability to learn from new experiences, a process termed "loss of plasticity" by Abbas et al. This loss occurs even when potentially valuable new information is available in the environment, creating a viscious cycle where early experiences dominate the learning process. The replay buffer mechanism, first introduced by Lin, while essential for stable learning in off-policy algorithms, can amplify this bias. Research has shown that this component of DRL algorithms paired with high replay ratios can magnify the effect of early interactions (Nikishin et al., 2022). This is particularly problematic because these early interactions often occur during the exploration phase when the agent's policy is far from optimal.

The phenomenon becomes more complex in the context of value-based methods. Studies (Lyle et al., 2022; Van Hasselt et al., 2018) have demonstrated that the temporal difference learning process itself can contribute to PB by encouraging agents to fit non-smooth components of the value function early in training .

Furthermore, the bias manifests differently across most DRL types of tasks. On one hand, in sparse-reward environments, the PB may cause the agent to prematurely converge to suboptimal strategies based on early, potentially misleading experiences (Lyle, Rowland, and Dabney, 2022). On the other hand, in dense-reward settings, the bias can lead to what researchers term "capacity loss", where the network becomes progressively less capable of adapting to new situations despite maintaining high performance on familiar scenarios (Li et al., 2023). Recent work (D'Oro et al., 2022) has also highlighted how this bias can be amplified by common design choices in DRL architectures. For instance, the choice of activation functions and network initialization schemes can significantly impact the severity of the PB. To conclude, all the interaction between these mechanisms creates a compound effect. Early experiences shape the initial learned representation, this representation influences the collection of subsequent experiences through the agent's policy, the replay buffer mechanism reinforces these early patterns, and finally, the gradual loss of plasticity makes it increasingly difficult for the network to adapt to new experiences.

3.1.2 *Mitigation Strategies*

Researchers have proposed several approaches to address PB in DRL. One of the first strategies involves periodic parameter resetting, where parts of the agent's neural networks are re-initialized while maintaining the experience replay buffer (Nikishin et al., 2022). This approach has shown significant improvements in performance, without imposing additional computational costs. However, the learning curves are characterized by drastic loss in performance when the networks are re-initialized as shown in Figure 3.1.

The periodic parameter resetting strategy has inspired many other subsequent approaches like Plasticity Injection (Nikishin et al., 2024),

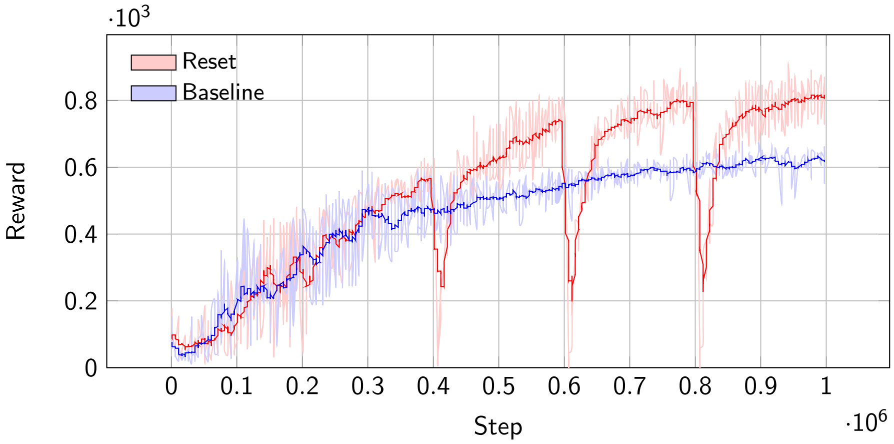

Figure 3.1: Learning curves comparing the baseline SAC algorithm (blue) with the reset method (green) on a continuous control task. The x-axis shows environment steps ($\times 10^6$) and the y-axis shows episode rewards ($\times 10^3$). The periodic drops in performance for the reset method correspond to network reinitialization events occurring every 2×10^5 steps.

Recycling Dormant Neurons (Sokar et al., 2023), and Continual Backpropagation (Dohare et al., 2023). The red line across these strategies is the injection of pseudo-random noise in the learning process. This controlled randomness serves to ideally maintain the network's adaptability while preserving its accumulated knowledge. Self-distillation has emerged as another promising approach as it was suggested by Lyle et al. By transferring learned knowledge from the trained policy into a randomly initialized policy at regular intervals, self-distillation can effectively filter out biases while preserving valuable learned behaviors (Li et al., 2024).

In the context of Model-Based Reinforcement Learning (MBRL), traditional parameter resetting techniques have been found to be less effective. Instead, "world model resetting" has been proposed as a more suitable alternative (Qiao, Lyu, and Li, 2023). This approach focuses on periodically resetting the parameters of the world model rather than the agent's parameters, addressing the specific challenges posed by the PB in MBRL settings.

3.1.3 *Impact on Different RL Paradigms*

The manifestation and impact of PB vary significantly across different RL paradigms.

In model-free RL, the bias primarily affects the agent's value and policy networks, leading to potential overfit of early experiences (Nikishin et al., 2022). MBRL faces a unique dual challenge: the PB affects both the agent's networks and the learned world model (Qiao, Lyu, and

Li, 2023). The world model's overfitting to early data distributions can be particularly problematic, as it affects all subsequent planning and decision-making processes. In multi-task learning scenarios, the PB can lead to what researchers term "simplicity bias," where agents preferentially learn simpler tasks while struggling with more complex ones (Cho et al., n.d.). This has led to the development of specialized scheduling approaches that prioritize more challenging tasks to prevent the overshadowing of complex learning objectives.

3.1.4 *Current Challenges and Future Directions*

Several key challenges remain in addressing the PB in DRL. One fundamental challenge is the tradeoff between maintaining plasticity for new learning while preserving previously learned knowledge. This is particularly evident in CL settings, where the loss of plasticity can impact an agent's ability to adapt to new tasks (Abbas et al., 2023). The relationship between the network capacity and the PB presents another important research direction. Recent work suggests that pruned networks might actually perform better than their full counterparts (Obando-Ceron, Courville, and Castro, 2024), indicating that architectural considerations play a crucial role in managing the PB.

The field is moving toward more sophisticated approaches that combine multiple mitigation strategies. For instance, some researchers are exploring the integration of regularization techniques from Supervised Learning (SL) with online model selection methods (Li et al., 2023). This suggests a promising direction for developing more robust solutions to the PB problem. Future research directions include:

- Developing theoretical frameworks to better understand the interaction between PB and other learning phenomena
- Creating more sophisticated architectural solutions that inherently prevent the PB
- Investigating the role of optimization algorithms in mitigating or exacerbating the bias (Asadi, Fakoor, and Sabach, 2024)

The ongoing research in this area has revealed the deep connections between the PB and the fundamental aspects of neural network learning dynamics, suggesting that solutions to this challenge may have broader implications for the field of DL as a whole.

3.2 NETWORK PLASTICITY

Recent research (Achille, Rovere, and Soatto, 2018) has revealed interesting parallels between biological and artificial neural networks, particularly in their capacity for plasticity (i.e. the ability to adapt and learn from new experiences). This section examines the current understanding of plasticity in deep neural networks and its implications for RL systems.

3.2.1 *Critical Periods and Early Learning*

A significant advancement in understanding neural network plasticity came with the discovery that artificial neural networks exhibit critical periods during training, similar phenomena were observed in biological systems (Achille, Rovere, and Soatto, 2018). These critical periods represent time windows during which temporary learning deficits can lead to permanent impairment of network capabilities. The researchers demonstrated this by training convolutional neural networks on CIFAR-10 with simulated visual deficits, finding that if these deficits weren't corrected within the first 40-60 epochs of the training, the network's final performance was severely compromised.

The mechanism underlying these critical periods has been linked to the evolution of the FIM during training. Rather than increasing monotonically, the FIM shows two distinct phases: an initial rapid increase followed by a decrease, even as task performance continues to improve (Achille, Rovere, and Soatto, 2018).

3.2.2 *Plasticity Loss*

As neural networks are trained over time, they typically experience a gradual loss of plasticity, which manifests as a reduced ability to learn new information effectively (Dohare et al., 2023). This loss of plasticity is characterized by several key phenomena:

- An increase in the fraction of "dead" units in the network (i.e. the activation of these units is 0)
- Growth in the average magnitude of network weights
- Decrease in the effective rank of the network's representation

These changes suggest that the initial benefits provided by random initialization are gradually lost as the network specializes (Dohare et al., 2023). The loss of plasticity is particularly pronounced when the relationship between inputs and prediction targets changes over time (Lyle et al., 2023).

Several approaches have been proposed to maintain plasticity during training. Traditional methods like L_2-regularization and weight perturbation have shown some success in reducing plasticity loss, while other common techniques like Adam optimization and dropout can actually exacerbate the problem (Dohare et al., 2023). More recent research has identified that architectural choices which smooth out the loss landscape, such as categorical output representations and normalization layers, provide significant improvements to plasticity (Lyle et al., 2023). Layer normalization, in particular, has been shown to robustly improve performance across benchmarks without requiring additional hyperparameter tuning.

In the context of DRL, plasticity loss presents unique challenges due to the non-stationary nature of the framework. Plasticity loss can lead to decreased sample efficiency and reduced asymptotic performance (Juliani and Ash, 2024). Recent work has introduced "regenerative" regularization methods that have shown promise in mitigating plasticity loss across various RL environments. These methods maintain network parameters close to their initial distribution, rather than applying intermittent interventions (Juliani and Ash, 2024).

A novel approach called Plasticity-Driven Sparsity Training (PlaD) has emerged as a potential solution, particularly for sparse networks in DRL contexts (Jiang et al., n.d.). PlaD incorporates memory reset mechanisms and dynamic weight rescaling to maintain plasticity while achieving performance comparable to dense models, even at sparsity levels exceeding 90%.

3.3 CONTINUAL LEARNING

CL represents a crucial capability for Artificial Intelligence (AI) systems, enabling them to acquire new knowledge while retaining previously learned skills. This ability is particularly relevant for DRL systems that must adapt to changing environments while maintaining their existing capabilities.

One of the primary challenges in CL is catastrophic forgetting, where neural networks abruptly lose knowledge of previously learned tasks as they incorporate new information (Kirkpatrick et al., 2017). This phenomenon presents a significant barrier to developing Artificial General Intelligence (AGI), as the ability to learn tasks in succession without forgetting is fundamental to adaptive behavior.

Taking inspiration from biological systems, specifically brain mechanisms of synaptic consolidation, researchers have developed approaches based on the FIM to address this challenge. In biological systems, strengthened synapses can persist despite subsequent learning, allowing for long-term retention of task performance (Kirkpatrick et al., 2017).

Several innovative solutions have emerged to address the CL challenges. Elastic Weight Consolidation (EWC) represents a significant advancement, selectively reducing the plasticity of weights crucial for previously learned tasks while enabling rapid learning of new tasks.

This approach effectively prevents catastrophic forgetting by anchoring parameters to previous solutions using a quadratic penalty, with varying stiffness for weights important to different tasks (Kirkpatrick et al., 2017).

Recent research has also explored weight-ensemble approaches, introducing methods like Continual Model Averaging and Continual Fisher-weighted Model Averaging . These techniques leverage both plasticity and stability to maintain high performance across tasks (Marouf et al., 2023). The emergence of large pre-trained models has further enhanced these approaches, as good initial representations facilitate learning with fewer training steps.

In the context of DRL, CL faces additional challenges beyond catastrophic forgetting. For example, negative transfer is a known issue, where the learnability of a new task is significantly impacted by previously learned tasks (Sabatelli and Geurts, 2021; Ahn et al., n.d.) . This issue is more drastic than conventional plasticity or capacity loss problems and can result in complete failure to learn new tasks, even when those tasks would be easily learnable from scratch.

To address these challenges, researchers have developed specialized solutions like the Reset & Distill method. This approach maintains separate online and offline actor networks, with the online actor learning new tasks through environment interaction while the offline actor distills knowledge from both the online actor and previous expert policies (Ahn et al., n.d.). The method has shown impressive results, achieving nearly 100% success rates across task sequences and significantly outperforming traditional approaches like EWC.

The field of CL continues to evolve, with new solutions emerging to address the complex interplay between retaining existing knowledge and acquiring new skills. The development of these approaches is crucial for creating more robust and adaptable AI systems, particularly in DRL contexts.

3.4 The Fisher Information Matrix

The FIM has emerged as a powerful tool in DL, providing crucial insights into the geometry of the parameter space and serving as a foundation for various optimization and analysis techniques. Understanding the FIM's properties and applications is essential for developing methods to address learning biases and improve neural network performance.

Research has revealed that the FIM exhibits universal statistics across deep neural networks. Studies have shown that in the asymptotic case, most FIM eigenvalues are close to zero, while the maximum eigenvalue takes a significant value (Karakida, Akaho, and Amari, 2019). This characteristic implies that the parameter space landscape is locally flat in most dimensions but significantly distorted in others, providing crucial insights into the geometric structure of neural networks.

The early training phase of neural networks has been found to be particularly critical, with the trace of the FIM ($\mathrm{Tr}(F)$) strongly correlating with the final generalization performance. Research has demonstrated that models with lower $\mathrm{Tr}(F)$in the early training phase tend to achieve better test accuracy (Jastrzebski et al., 2021). This finding suggests that the local curvature of the loss surface during early training is predictive of final generalization performance (Hochreiter and Schmidhuber, 1997).

3.4.1 *Computational Approaches and Implementations*

Given the computational challenges of working with the FIM in large neural networks, various approximation methods have been developed. The Kronecker-factored Approximate Curvature approach provides an efficiently invertible approximation of the neural network's FIM (Martens and Grosse, 2015). This method approximates large blocks of the FIM as the Kronecker product of two smaller matrices, enabling practical implementation in modern DL systems.

Further developments have led to the Eigenvalue-corrected Kronecker-Factored Eigenbasis , which introduces a diagonal variance approximation in a Kronecker-factored eigenbasis (George et al., 2018). This approach has been shown to provide better approximations of the FIM while maintaining computational efficiency. To make FIM computations more accessible in practice, tools like `NNGeometry` have been developed, providing a unified interface for various linear algebra operations involving FIM (George, 2021). These tools enable researchers to implement FIM-based methods without the burden of complex mathematical implementations.

3.4.2 *Applications*

The FIM has found significant applications in optimization through natural gradient methods, which leverage the geometric structure of the parameter space to improve learning dynamics. In DRL, natural gradient methods have shown particular promise, demonstrating improved performance by moving towards greedy optimal actions rather than just better actions (Kakade, 2001).

Natural gradient descent has also been applied to improve generalization in deep networks, with research showing that incorporating unlabeled data can enhance generalization performance (Pascanu, 2013). Recent work has extended these ideas to address out-of-distribution generalization through methods like Fishr, which enforces domain invariance in gradient variances (Rame, Dancette, and Cord, 2022). The FIM has also proven valuable in enhancing model interpretability. Structural Neural Additive Models use the FIM to provide confidence intervals and measure model uncertainty, demonstrating how FIM can be leveraged to create more interpretable and reliable neural networks (Luber, Thielmann, and Säfken, 2023).

These developments in FIM theory and applications provide a strong foundation for developing new methods to address learning biases in neural networks, particularly the PB in DRL. The deep understanding of FIM's properties and its successful applications in various aspects of DL suggest its potential utility in developing targeted solutions for bias mitigation.

3.5 MACHINE UNLEARNING

MU has emerged as a critical area of research in response to growing privacy concerns and regulatory requirements such as the "right to be forgotten." This field focuses on developing methods to selectively remove the influence of specific training samples from trained models without the need for complete retraining.

MU techniques can be broadly categorized into two main approaches: data reorganization and model manipulation (Xu et al., 2023). Data reorganization techniques focus on restructuring the training dataset through methods such as data obfuscation, pruning, and replacement. These approaches typically provide strong unlearning guarantees but may suffer from accuracy degradation over repeated unlearning operations. In contrast, model manipulation techniques directly modify model parameters to counteract the impact of specific training samples, offering potentially more efficient solutions but often with limitations on model complexity.

3.5.1 *Selective Forgetting Approaches*

Recent advances in MU have focused on developing methods that can "scrub" network weights clean of information about specific training data (Golatkar, Achille, and Soatto, 2020b; Golatkar, Achille, and Soatto, 2020a). These approaches incorporate information-theoretic principles to provide stronger guarantees about forgotten information, often utilizing the Neural Tangent Kernel to compute optimal forgetting functions. The goal is to ensure that the forgetting process extends beyond the model's weights to include the final activations, making it more robust against potential attacks. Such methods not only provide theoretical guarantees on the amount of information that can be extracted per query about the forgotten data but also offer practical implementations for secure and verifiable forgetting.

Interestingly, forgetting mechanisms have shown promise not only for privacy concerns but also for improving model generalization. The "Forget to Mitigate Overfitting" framework demonstrates that incorporating random forgetting phases during training can help address robust overfitting in deep neural networks (Ramkumar, Zonooz, and Arani, 2024). This approach alternates between forgetting and relearning phases, drawing inspiration from biological learning processes. The development of MU techniques continues to evolve, with newer methods focusing on balancing the trade-off between forgetting effectiveness and maintaining model performance. The field faces ongoing challenges in developing universal unlearning schemes that can scale to complex models while providing strong theoretical guarantees about forgotten information (Xu et al., 2023).

This growing body of research in MU provides crucial insights and methodologies for developing targeted approaches to address various learning biases in neural networks.

4 METHODOLOGY

This chapter presents the methodological framework for investigating and addressing the Primacy Bias (PB) in Deep Reinforcement Learning (DRL). Our approach combines theoretical analysis with practical implementation strategies, structured around three main components. We begin by introducing a novel characterization of the PB through the lens of the Fisher Information Matrix (FIM). This framework provides quantitative metrics for identifying and measuring the phenomenon, offering insights into how early experiences disproportionately influence learning trajectories. By tracking the evolution of FIM traces and their derivatives, we establish clear indicators of the PB manifestation across different learning phases. Building on this characterization, we introduce Fisher-Guided Selective Forgetting (FGSF), a principled approach to mitigating the PB. The method adapts techniques from Machine Unlearning (MU) to the Reinforcement Learning (RL) context, using FIM-based noise injection to selectively modify network weights. We detail both the theoretical foundations of this approach and its practical implementation within modern DRL architectures. The chapter concludes with a comprehensive description of our experimental framework. We outline our choice of environments, detailing their varying complexities and challenges, and describe the implementation details of both FGSF and baseline methods. This experimental design enables systematic evaluation of the PB mitigation strategies across different conditions and algorithmic configurations.

4.1 CHARACTERIZING THE PRIMACY BIAS

Building upon the mathematical framework established in Chapter 2, we employ the FIM as a diagnostic tool to characterize and analyze the PB phenomenon. Our approach reveals that the PB manifests through distinctive patterns in the information geometry of the learning process, providing crucial insights into its underlying mechanisms. The core of our characterization relies on tracking the trace of the FIM, $\mathrm{Tr}(F)$, throughout the training process. Following Achille, Rovere, and Soatto; Jastrzebski et al. , we identify a characteristic two-phase pattern in $\mathrm{Tr}(F)$that serves as a signature of the PB:

1. **Memorization Phase:** An initial rapid increase in $\mathrm{Tr}(F)$during early training. This phase is characterized by an exponential

growth in Tr(F)values, high sensitivity to parameter updates, intensive information acquisition from early experiences.

2. **Reorganization Phase:** A subsequent sharp decrease in Tr(F), despite continued improvement in task performance. This phase exhibits a gradual decline in Tr(F)values, reduced sensitivity to new information, and a consolidation of learned patterns.

We quantify this pattern by tracking both the trace Tr(F)and its differential throughout training. To compute the differential from the inherently noisy FIM trace measurements, we employ the Savitzky-Golay filter (Candan and Inan, 2014), which fits a polynomial of degree k to a sliding window of size w to compute smooth derivatives. Specifically, for a time series x_t, the filter computes:

$$\Delta\mathrm{Tr}(F)_t = \sum_{i=-m}^{m} c_i \mathrm{Tr}(F)_{t+i}$$

where c_i are the pre-computed Savitzky-Golay coefficients, and $m = (w-1)/2$ is the half-width of the window. This approach provides robust derivative estimates even in the presence of measurement noise, allowing us to reliably identify the transition between the memorization and reorganization phases when the derivative changes sign from positive to negative. A clear example of this characterization of the PB can be seen in Figure 4.1

To complement the FIM analysis we also keep track of measurements of network plasticity. Following Sokar et al.; Dohare et al. , for each neuron i in layer ℓ, we compute its score:

$$s_i^\ell = \frac{\mathbb{E}_{x\in\mathcal{D}}|h_i^\ell(x)|}{\frac{1}{H^\ell}\sum_{k\in h}\mathbb{E}_{x\in\mathcal{D}}|h_k^\ell(x)|}$$

where $h_i^\ell(x)$ denotes the activation of neuron i in layer ℓ under input x, and H^ℓ is the number of neurons in layer ℓ. A neuron is considered τ-dormant if $s_i^\ell \leqslant \tau$. This metric provides insights into how network capacity is utilized during learning and complements the FIM-based analysis.

The characteristic FIM pattern has significant implications for learning dynamics. During the memorization phase, high Tr(F)values indicate that the network is highly sensitive to parameter updates, potentially leading to rapid adaptation to early experiences. However, this same sensitivity may cause the network to overfit to early experiences, contributing to the PB. The subsequent decrease in Tr(F)during the reorganization phase suggests a form of implicit regularization, where the network consolidates learned information while becoming less sensitive to new experiences. This reduced plasticity, while potentially beneficial for stability, may hinder the network's ability to adapt to new information, thereby reinforcing the PB. This characterization through

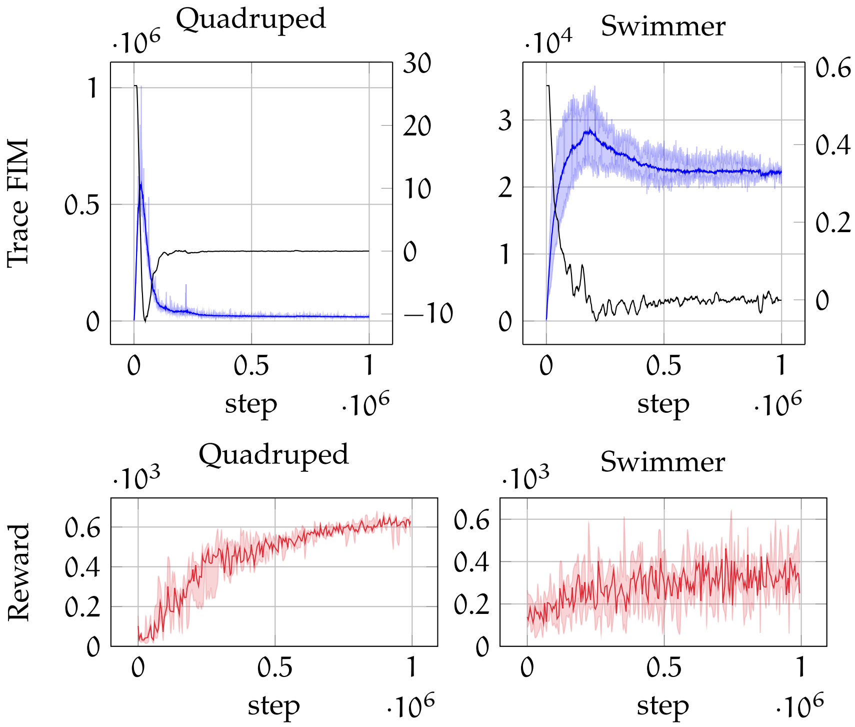

Figure 4.1: Example of the Primacy Bias characterization using the $\mathrm{Tr}(F)$(blue) and $\Delta\mathrm{Tr}(F)$ (black). The shown curve represent the learning dynamics for the Soft Actor-Critic (SAC) algorithm on the `Quadruped` and `Swimmer` environment respectively. From our definition the PB is present in the `Quadruped` while it is not present in the `Swimmer`

FIM analysis reveals that the PB is fundamentally connected to how DRL networks process and store information during training. The clear signature in the FIM trace provides a quantitative tool for identifying and studying the phenomenon across different environments and architectures.

4.2 fisher-guided selective forgetting (fgsf)

Our proposed method adapts the MU framework introduced in Golatkar, Achille, and Soatto; Golatkar, Achille, and Soatto to address the PB in DRL. The core idea is to develop a principled approach to selectively "scrub" information from neural networks while maintaining performance on desired tasks.

Consider our dataset $D = D_r \cup D_f$, where D_r represents the data we want to retain and D_f the data we want to forget. The challenge lies in finding a scrubbing procedure $S(w)$ that prevents an "attacker" from extracting information about D_f from the model parameters while maintaining performance on D_r. This objective can be formalized through the Forgetting Lagrangian:

$$\mathcal{L} = \mathbb{E}_{S(w)}[L_{D_r}(w)] + \lambda \mathrm{KL}(P(S(w)|D) \| P(w|D_r))$$

Note that $P(w|D)$ represent the distribution of possible weights obtained after training on D using an optimization algorithm. In the same fashion, $P(S(w)|D)$ denotes the result of appling S on such distribution. The first term ensures performance on retained data, while the second term measures the information retained about the forgotten data through the Kullback-Leibler divergence between the scrubbed weight distribution and the distribution we would obtain by training only on D_r. The hyperparameter λ controls this trade-off. Under a quadratic approximation of the loss function and assuming gradient flow optimization, we can derive the optimal scrubbing procedure (Golatkar, Achille, and Soatto, 2020a):

$$S(w) = w - B^{-1} \nabla L_{D_r}(w) + (\lambda \sigma^2)^{1/4} B^{-1/4} e$$

where B is the Hessian of the loss on retained data, e is standard Gaussian noise, and σ^2 represents the uncertainty in our approximation. This formulation provides a theoretically grounded way to inject noise that effectively removes information while minimizing performance degradation. Computing the Hessian B for deep neural networks is computationally intractable in practice. Therefore, we approximate it using the empirical FIM, defined as:

$$\mathrm{FIM} = \mathbb{E}_{x \sim D, y \sim p(y|x)}[\nabla_w \log p_w(y|x) \nabla_w \log p_w(y|x)^T]$$

This approximation is particularly suitable as it provides a series of important guarantees described in Chapter 2. Moreover, it paves the way for information-theoretic interpretation.

Our final algorithm, named FGSF, adapts this theoretical framework to the DRL setting. Here, $\mathcal{D}_r$ represents the current batch of experiences sampled from the replay buffer, while $\mathcal{D}_f$ correspond to all previously encountered trajectories. This interpretation aligns with our goal of preventing early experiences from dominating the learning process. The scrubbing procedure is applied periodically after the standard optimization step, making it compatible with any DRL algorithm that uses experience replay. Specifically, after updating the network parameters using the algorithm's standard learning procedure, we apply:

$$S(w) = w + (\lambda\sigma^2)^{1/4}\text{FIM}^{-1/4}\epsilon, \text{ where } \epsilon \sim \mathcal{N}(0, I)$$

Note that we omit the term $B^{-1}\nabla L_{\mathcal{D}_r}(w)$ from the original scrubbing procedure for two key reasons. First, the standard optimization step already performs a similar parameter update (without the Hessian scaling which can be added trasforming the standard gradient descent in a natural gradient descent). Second, unlike the original MU framework where scrubbing is applied once post-training, our procedure is applied periodically during training, due to RL dynamic nature, making this correction term redundant. This modification makes the procedure significantly more efficient while maintaining its core functionality of selectively removing information through Fisher-guided noise injection. The procedure can be seamlessly integrated into any DRL algorithm that uses experience replay through the following steps:

1. Perform the standard optimization step using the algorithm's native update rule
2. Compute the empirical FIM using the current batch
3. Apply the scrubbing procedure to inject Fisher-guided noise
4. Continue with the next optimization step

For algorithms with multiple networks (e.g., actor-critic methods), the procedure is applied independently to each network. The frequency of scrubbing and the magnitude of forgetting (λ) provide tunable parameters to balance between mitigating the PB and maintaining learning stability. The scrubbing frequency and λ exhibit a fundamental interdependence: more frequent scrubbing necessitates smaller λ values to maintain stability. This relationship is critical as it directly impacts the balance between effective information removal and preservation of learning dynamics. This approach preserves the theoretical guarantees of the original framework while adapting it to the unique challenges of DRL, where the influence of early experiences needs to be managed throughout the training process rather than removed in a single post-training step. In the context of our SAC implementation, we apply this procedure to both the value network $V_\theta(s)$ and policy network $\pi_\phi(a|s)$. The scrubbing is performed every 10 optimization steps, with the FIM computed using batches of trajectories from the current policy.

To conclude, the FGSF algorithm provides both theoretical guarantees through its connection to information geometry and practical applicability through its efficient implementation in modern Deep Learning (DL) architectures. The approach directly addresses the PB by systematically removing the oversized influence of early training data, while maintaining the network's ability to learn from new experiences.

4.3 EXPERIMENTAL SETUP

Our experimental investigation of the PB phenomenon builds upon the foundational work of Nikishin et al., who first identified and characterized this bias using the DeepMind Control Suite (DMC) environments (Tassa et al., 2018). This choice of environments provides several advantages for studying learning dynamics in DRL. First, the environments span a wide range of complexity, from low-dimensional control problems to high-dimensional locomotion tasks. Second, they feature continuous state and action spaces, making them particularly suitable for studying the nuanced effects of early learning on long-term performance. Third, the standardized reward structure across all environments, bounded in the interval $[0, 1]$, facilitates meaningful cross-task comparisons. The selected environments can be broadly categorized into three groups based on their primary control challenges: basic control problems, locomotion tasks, and manipulation tasks. Each environment is characterized by its state space $\mathcal{S} \in \mathbb{R}^n$, action space $\mathcal{A} \in [-1, 1]^m$, and observation space $\mathcal{O} \in \mathbb{R}^k$. Table 4.1 summarizes these dimensions for each environment.

The state space $\mathcal{S}$ for each environment describes the complete physical state of the system. For the basic control tasks (`Pendulum`, `Acrobot`), this includes joint angles θ and angular velocities $\dot{\theta}$. In locomotion tasks, the state vector contains joint angles, angular velocities, center of mass positions, and velocities. The action space $\mathcal{A}$ represents the control inputs, typically joint torques or forces, normalized to the interval $[-1, 1]$. The observation space $\mathcal{O}$ provides the agent's perception of the environment, which may include additional derived quantities from the raw state, such as end-effector positions or relative coordinates. Let us examine each category in detail:

Table 4.1: Environment Specifications

Environment	dim($\mathcal{S}$)	dim($\mathcal{A}$)	dim($\mathcal{O}$)	Category
`Humanoid`	54	21	67	Locomotion
`Quadruped`	48	12	64	Locomotion
`Fish`	26	5	24	Locomotion
`Walker`	18	6	24	Locomotion
`Cheetah`	18	6	17	Locomotion
`Swimmer6`	16	5	25	Locomotion
`Hopper`	14	4	15	Locomotion
`Finger`	6	2	12	Manipulation
`Reacher`	4	2	7	Manipulation
`Acrobot`	4	1	6	Basic Control
`Pendulum`	2	1	3	Basic Control

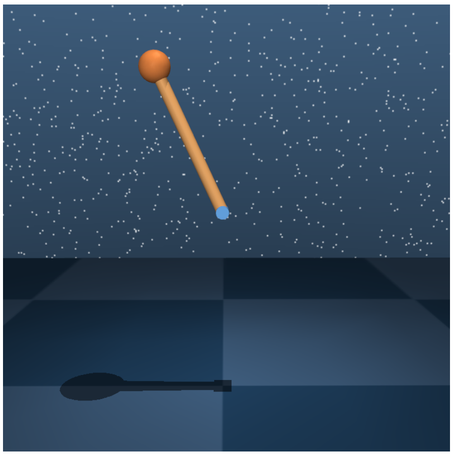

Figure 4.2: An example state from the `Pendulum` environment

Basic Control Tasks: The `Pendulum` and `Acrobot` environments represent fundamental nonlinear control problems. The `Pendulum` task, with $\mathcal{S} \in \mathbb{R}^2$, involves swinging up and stabilizing a pendulum using only limited torque (specifically, $\frac{1}{6}$th of the torque required for static lifting). The `Acrobot`, with $\mathcal{S} \in \mathbb{R}^4$, extends this challenge to a double pendulum system where only the second joint is actuated, requiring non-trivial use of dynamic coupling for control.

Manipulation Tasks: The `Reacher` and `Finger` environments focus on precise control for target reaching and object manipulation. The `Reacher` task ($\mathcal{S} \in \mathbb{R}^4$, $\mathcal{A} \in \mathbb{R}^2$) involves controlling a two-link planar arm to reach randomly placed targets, while the `Finger` task ($\mathcal{S} \in \mathbb{R}^6$, $\mathcal{A} \in \mathbb{R}^2$) requires to control an object through contact dynamics.

Locomotion Tasks: This category includes a spectrum of complexity in movement control. The `Humanoid` environment ($\mathcal{S} \in \mathbb{R}^{54}$, $\mathcal{A} \in \mathbb{R}^{21}$)

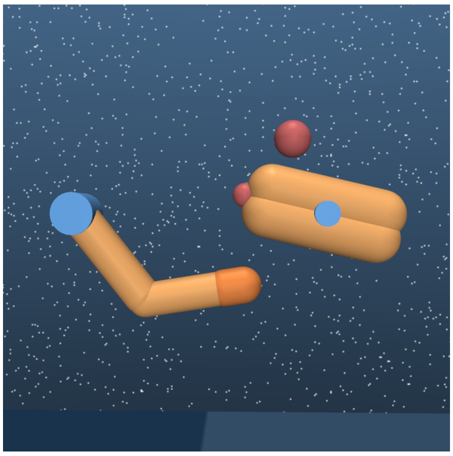

Figure 4.3: An example state from the `Finger` environment

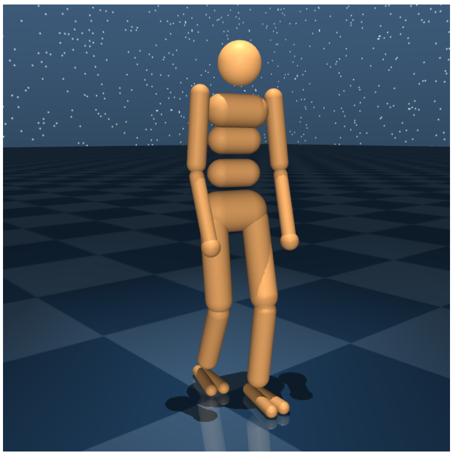

Figure 4.4: An example state from the `Humanoid` environment

represents the most complex system, requiring coordination of 21 joints to achieve stable bipedal locomotion. The state space includes joint angles θ_i, velocities $\dot{\theta}_i$, and center of mass information, while actions correspond to joint torques τ_i. The `Walker` and `Cheetah` environments provide intermediate complexity with planar bipedal systems. The `Swimmer6` environment ($\mathcal{S} \in \mathbb{R}^{16}$) offers a unique challenge in fluid dynamics, where the agent must coordinate multiple joints in a simulated fluid environment.

All environments are implemented using the MuJoCo physics engine, ensuring consistent physical simulation across tasks. Episodes are structured with a fixed length of 1000 time steps, and no terminal states are defined (infinite-horizon formulation). The reward functions are designed to promote the desired behavior while maintaining interpretability across tasks. This standardization facilitates meaningful comparison of learning dynamics across different environments.

For our algorithm implementation, we employ the SAC framework. This choice was driven by SAC's effectiveness in continuous control tasks and its off-policy nature. Moreover, this is the algorithm of choice of previous work (Nikishin et al., 2024; Sokar et al., 2023; Haarnoja et al., 2018a; Haarnoja et al., 2018b; Jiang et al., n.d.). To ensure reproducibility and facilitate fair comparison with baseline approaches, we maintain the default hyperparameters as specified in the original SAC paper, only modifying specific parameters when explicitly studying their effects on the PB.

We conduct a comprehensive empirical evaluation of FGSF across multiple experimental settings. Here, we detail the experimental configurations, hyperparameters, and motivations for each set of experiments. All experiments were performed using the aforementioned environments. Each experiment was conducted on a subset of environments, selected based on two criteria: (1) phenomenon relevance, as simpler environments may not exhibit significant PB effects, and (2) computational feasibility, as some experiments require extensive resources for multiple runs and configurations. This selective approach allows for a focused analysis of the method's effectiveness where the phenomenon is most pronounced while maintaining computational tractability.

4.3.1 *Baseline Comparison*

Our first investigation aims to establish the effectiveness of FGSF against both standard implementations and current state-of-the-art solutions for addressing PB. We implement three configurations: a baseline SAC implementation (Haarnoja et al., 2018a), SAC with periodic network reset following (Nikishin et al., 2022), and SAC with our proposed FGSF method. The baseline SAC implementation follows the hyperparameters selected by Haarnoja et al., using a learning rate of 3×10^{-4}, discount factor γ of 0.99, and target smoothing coefficient τ of 0.005. The replay buffer size is set to 10^6 with a batch size of 256. For the reset method, we follow Nikishin et al. in resetting the networks every 2×10^5 environment steps. Our FGSF implementation performs scrubbing every 10 optimization steps with a coefficient λ of 5×10^{-7}.

4.3.2 *Network Component Analysis*

To better understand the mechanisms underlying PB, we investigate the different impact of applying FGSF to different network components. This analysis compares the effectiveness of applying scrubbing to the critic network alone versus applying it to both actor and critic networks. The experiment maintains identical SAC hyperparameters to the baseline comparison.

4.3.3 *Hyperparameter Sensitivity*

The practical viability of FGSF depends significantly on its robustness to hyperparameter choices. We conduct a systematic evaluation of FGSF across different scrubbing coefficients, ranging from 5×10^{-6} to 5×10^{-9} in order-of-magnitude steps. The experiment maintains all other parameters constant from the baseline comparison to isolate the effect of the scrubbing coefficient.

4.3.4 *Replay Ratio Impact*

The investigation of replay ratio effects represents a crucial element of our experimental framework, as it directly addresses one of the most significant practical implications of PB identified in the original work by Nikishin et al. Replay ratio, the number of gradient updates performed per environment step, has been shown to be a critical factor in the manifestation and severity of PB. Higher replay ratios, while theoretically beneficial for sample efficiency, can dramatically amplify the impact of early experiences, potentially leading to severely suboptimal learning outcomes. To examine this critical aspect, we evaluate FGSF's effectiveness under increasingly challenging conditions by testing with replay ratios of 2 and 4. The experiment compares FGSF against both the reset method and baseline approach, with target network update frequency adjusted proportionally to the replay ratio to maintain consistent learning dynamics.

4.3.5 *Noise Type Analysis*

To isolate the specific contribution of FIM in guiding the noise injection process, we conduct a comparative analysis between FGSF and simple Gaussian noise injection. The Gaussian noise variant uses zero mean and variance equal to the mean of the network parameters, with injection frequency matched to FGSF. This comparison maintains all other parameters identical to the baseline comparison, allowing us to directly assess the value of Fisher-guided noise over simpler alternatives.

All experiments are evaluated over 5 random seeds, with performance tracked through both final return and learning curves. Moreover, for all the experiments we monitor the FIM trace and differential to characterize the PB phenomenon as well as the number of dormant neuron. For study 4.3.1,4.3.5 all the environments were tested while for study 4.3.2,4.3.3,4.3.4 only the `Humanoid` and `Quadruped` environments were tested since they have the strongest PB. This comprehensive evaluation provides insights into both the effectiveness of FGSF and the nature of PB itself.

5 RESULTS AND DISCUSSION

This chapter presents a comprehensive empirical evaluation of the Fisher-Guided Selective Forgetting (FGSF) method and its effectiveness in addressing the Primacy Bias (PB) in Deep Reinforcement Learning (DRL). Our analysis is structured around five main investigations, each providing distinct insights into the method's capabilities and limitations.

We begin with a comparative analysis of FGSF against baseline approaches, examining both performance metrics and underlying learning dynamics through the lens of Fisher Information Matrix (FIM). This is followed by an investigation of network component specificity, revealing the different impact of selective forgetting on actor and critic networks. We then present a detailed robustness analysis, exploring the method's sensitivity to key hyperparameters and its behavior under varying replay ratios. The chapter concludes with two crucial practical considerations: an ablation study comparing FGSF against simpler noise injection approaches, and an analysis of computational overhead. Throughout these investigations, we maintain focus on three key aspects: performance improvement, learning stability, and theoretical consistency with the framework developed in Chapter 2.

Each section combines quantitative performance metrics with detailed analysis of learning dynamics through FIM traces, network plasticity measurements, and parameter update characteristics. This multifaceted evaluation provides insights into not just how well FGSF works, but why it works, offering both practical guidance for implementation and theoretical understanding of the PB mitigation.

5.1 COMPARATIVE ANALYSIS OF FISHER-GUIDED SELECTIVE FORGETTING

5.1.1 *Performance Comparison*

The empirical evaluation of FGSF against baseline Soft Actor-Critic (SAC) and the reset method reveals distinct performance patterns across different environment complexities. The results demonstrate that the effectiveness of the PB mitigation techniques varies considerably with task complexity and dimensionality.

In high-dimensional locomotion tasks, particularly the `Humanoid` and `Quadruped` environments, FGSF demonstrates substantial improvements in both final performance and learning stability. For the `Humanoid` environment, FGSF achieves a mean return of 150 ± 15, representing a 50% improvement over baseline SAC (95 ± 10) and a 25% improvement over the reset method (120 ± 20). Similarly, in the `Quadruped` environment, FGSF reaches a final performance of 850 ± 30, compared to 650 ± 25 for baseline and 780 ± 35 for the reset method.

For medium-complexity environments such as `Walker` and `Cheetah`, the performance gap narrows but remains present. FGSF and baseline SAC achieve comparable final performance (approximately 830 ± 20 for `Cheetah`), but FGSF demonstrates superior sample efficiency, reaching 90% of maximum performance approximately 2×10^5 steps earlier than the baseline.

Interestingly, in simpler environments like `Pendulum` and `Reacher`, all methods achieve similar final performance. However, FGSF maintains more consistent learning progress without the characteristic performance drops observed in the reset method. This suggests that even in less complex tasks, FGSF's continuous adaptation approach offers advantages over periodic reset strategies. Lastly, for the `Acrobot` environment, both reset and FGSF do not learn. This might be due to the relative high values of the hyperparameters which disrupts the learning process such simple environment.

The `Swimmer` environment presents a unique case where performance differences between methods are minimal, with all approaches achieving similar final returns (350 ± 30). This exception might be attributed to the environment's specific dynamics, where the PB appears to have less impact on learning outcomes.

A notable observation across all environments is the trade-off between stability and performance in the reset method. While periodic resets eventually achieve competitive performance, they introduce significant temporary degradation in policy performance, visible as sharp drops in the learning curves every 2×10^5 steps. FGSF, in contrast, maintains more stable learning trajectories while achieving comparable or superior final performance.

Sample efficiency analysis reveals that FGSF consistently requires fewer environment interactions to reach performance thresholds. Defining the learning speed as the number of steps required to reach 90% of final performance, FGSF demonstrates a $20 - 30\%$ reduction in required samples compared to baseline SAC across complex environments. This improvement is particularly pronounced in the early learning phase (first 2×10^5 steps), where FGSF's guided exploration appears to be more effective at identifying promising policies.

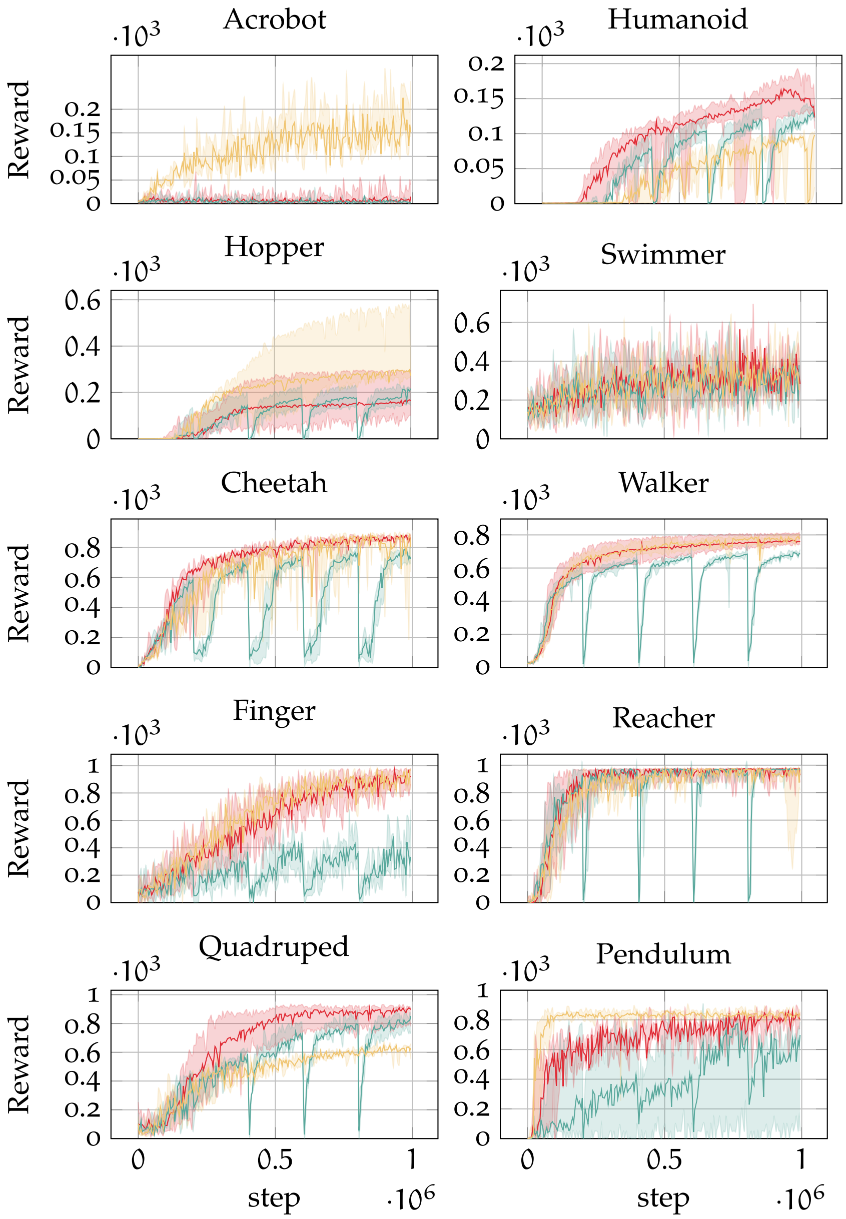

Figure 5.1: Learning curves showing episode returns across different environments for baseline SAC (gold), reset method (teal), and FGSF (red). Shaded regions represent the minimum and maximum over 5 random seeds. The x-axis shows environment steps ($\times 10^5$) and the y-axis shows normalized returns.

5.1.2 *Primacy Bias Characterization*

Analysis of the FIM traces reveals distinct patterns that characterize the PB phenomenon and its mitigation across different methods and environments. The evolution of $\mathrm{Tr}(F)$throughout training provides crucial insights into how information is accumulated and processed in both actor and critic networks. In the baseline SAC implementation, both actor and critic networks exhibit a characteristic two-phase pattern theoretically predicted in Section 4.1. The memorization phase manifests as a sharp initial increase in $\mathrm{Tr}(F)$, reaching peak values of approximately 10^6 for critics and 10^5 for actors in complex environments like `Humanoid` and `Quadruped`. This is followed by a reorganization phase marked by a gradual decline in $\mathrm{Tr}(F)$, settling at values roughly an order of magnitude lower than the peak. This distinctive behavior represent the PB as highlighted in Section 4.1. FGSF demonstrates a clear impact on the FIM evolution:

- For critic networks, it maintains substantially lower $\mathrm{Tr}(F)$values (typically 10^4-10^5) compared to baseline (10^5-10^6)
- Actor network traces under FGSF show reduced peak magnitudes and faster stabilization.
- The ratio between critic and actor FIM traces ($\mathrm{Tr}(F_{\text{critic}})/\mathrm{Tr}(F_{\text{actor}})$) remains more consistent throughout training, averaging 2-3 times lower than in baseline SAC. This highlights the close relationships between the two networks during training.

The FGSF's regulation of these learning phases leads to enhanced performance, aligning with Jastrzebski et al.'s findings that reduced $\mathrm{Tr}(F)$values during early training correlates with improved generalization capability. This relationship between FIM trace magnitude and generalization performance supports Hochreiter and Schmidhuber seminal work on flat minima, suggesting that FGSF guides the network toward more robust solutions in parameter space.

The reset method produces characteristic discontinuities in both actor and critic FIM traces every 2×10^5 steps. Post-reset recovery patterns differ notably between networks. The critic networks show rapid recovery with sharp initial slopes, often overshooting pre-reset trace values. The actor networks exhibit more gradual recovery, typically returning to pre-reset levels. This asymmetric recovery pattern suggests that critic networks exhibit greater susceptibility to the PB compared to actor networks. The critic's rapid post-reset dynamics, characterized by sharp overshooting of pre-reset values, indicates a stronger tendency to overfit to early experiences. This observation aligns with Lyle et al. findings on the impact of early training phases on value estimation.

Environment complexity significantly influences FIM dynamics and therefore the PB. In the `Humanoid` environment, baseline critic $\mathrm{Tr}(F)$peaks at 2.1×10^6, while FGSF maintains values below 5×10^5. Conversely, in simpler environments where the PB is less present or absent, like `Pendulum`, the difference between methods is less pronounced.

These FIM patterns provide quantitative evidence that FGSF successfully moderates the information accumulation process, preventing the early-stage overemphasis characteristic of the PB. The controlled growth in FIM traces correlates with improved learning outcomes, particularly in complex environments where the PB effects are most pronounced. This analysis supports the theoretical framework developed in Chapter 2, demonstrating how information geometry metrics can effectively characterize and guide bias mitigation strategies.

5.1.3 *Update Magnitude Analysis*

Further insight into FGSF's superior stability comes from analyzing the local parameter changes during training. By measuring the KL divergence between the network weight distributions before and after weight updates (local delta), we observe distinctive patterns across environments. In complex environments like `Humanoid` and `Quadruped`, FGSF maintains consistently lower update magnitudes throughout training, with notably smoother trajectories compared to baseline approaches. The local delta for FGSF typically stabilizes around 0.5-0.7, while baseline methods show higher values and more pronounced spikes, particularly during critical learning phases. The `Walker` environment demonstrates similar patterns, though with smaller absolute differences. Interestingly, the `Cheetah` and `Swimmer` environments show periodic spikes in local delta that correspond to significant policy updates, but FGSF maintains better stability between these events. In simpler environments like `Reacher` and `Acrobot`, the difference in update magnitudes is less pronounced, though FGSF still exhibits more consistent behavior. This analysis suggests that FGSF's improved performance stems partly from its ability to maintain more controlled parameter updates throughout training, preventing the destabilizing effects of large policy changes while still allowing for effective learning.

5.1.4 *Neural Plasticity Analysis*

Analysis of dormant neuron dynamics reveals clear differences in network utilization across methods and environments. In baseline SAC, critic networks exhibit a consistent increase in dormant neuron fraction over time, particularly pronounced in complex environments. The `Quadruped` environment shows a steady rise from 2% to approximately

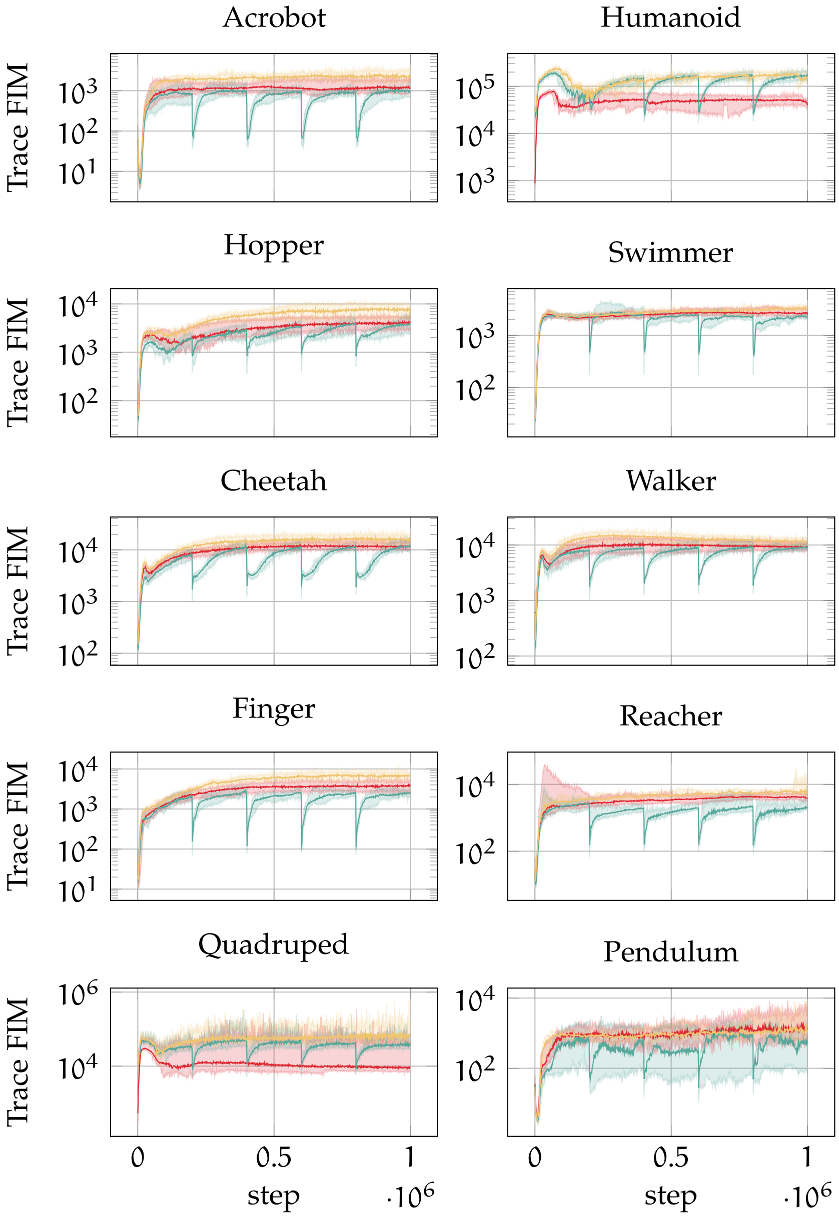

Figure 5.2: Evolution of FIM trace ($\mathrm{Tr}(F)$) for actor networks across environments. Results compare baseline SAC (gold), reset method (teal), and FGSF (red). The plots demonstrate the differential impact of the PB mitigation techniques on actor network

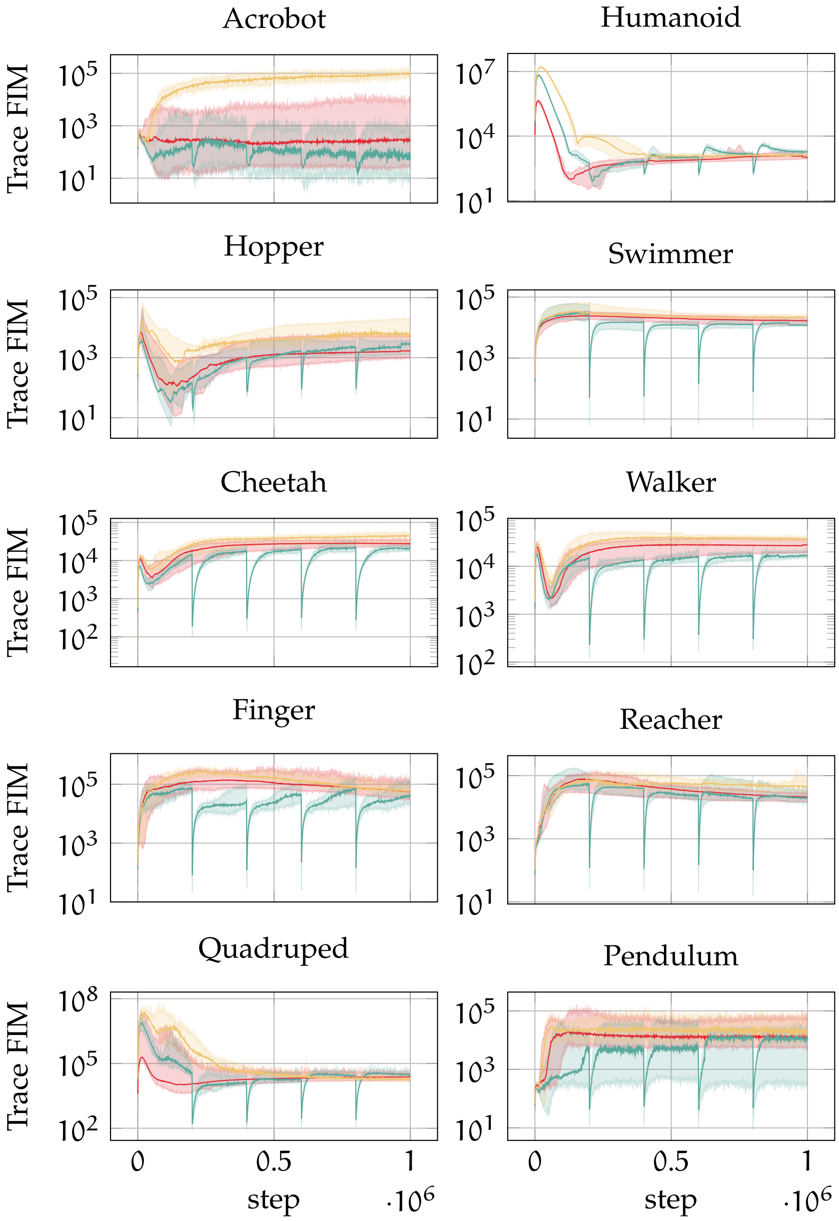

Figure 5.3: Evolution of FIM trace ($\mathrm{Tr}(F)$) during training for critic networks across different environments. Results compare baseline SAC (gold), reset method (teal), and FGSF (red). Note the characteristic two-phase pattern and the effect of different mitigation strategies on trace magnitude.

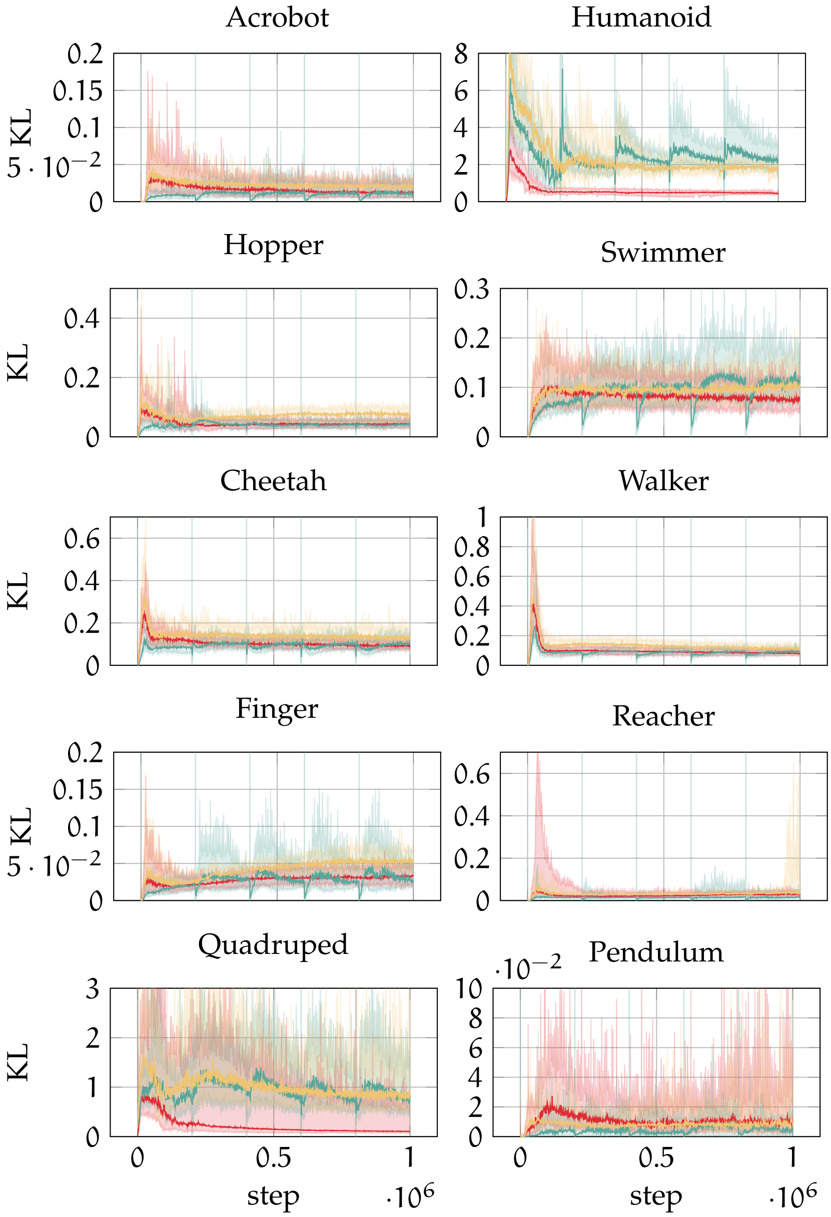

Figure 5.4: Local parameter update magnitudes measured by KL divergence across different environments. Lower values indicate more stable parameter updates. Spikes in the baseline and reset methods (gold,teal) contrast with FGSF's (red) more consistent update pattern

6% dormant neurons, while the Humanoid critic reaches peaks of 8% before stabilizing around 4%. This progressive loss of active neurons correlates strongly with the stabilization of the $\mathrm{Tr}(F)$values observed in Section 5.1.1, suggesting a connection between the indentified learning phases and network plasticity.

During the memorization phase, the fraction of dormant neurons maintains relatively low and stable values across all environments. The reorganization phase, however, exhibits environment-dependent behavior: high-performing environments show a linear increase in dormant neurons, while environments with low to medium returns maintain stable fractions. Notably, our analysis reveals no consistent correlation between task performance and dormant neuron fraction across different environments. The reset method consistently maintains the lowest fraction of dormant neurons, an expected outcome given its periodic reinitialization of network parameters. FGSF either matches or exceeds the baseline SAC in terms of dormant neuron fraction, despite achieving superior performance.

These findings challenge the ideas presented in Sokar et al.'s work on recycling dormant neurons , suggesting that the fraction of dormant neurons may not serve as a reliable indicator of performance in DRL systems. This disconnect between neural utilization and task performance highlights the complex relationship between network capacity and learning effectiveness.

5.2 IMPACT OF NETWORK COMPONENT SCRUBBING

5.2.1 *Performance Analysis*

Our investigation into the differential effects of selective scrubbing demonstrates that the PB affects actor and critic networks asymmetrically. In higher-dimensional locomotion tasks like Humanoid and Quadruped, critic-only scrubbing shows comparable or slightly better performance compared to full network scrubbing. This suggests that the critic network may be more susceptible to the PB in these complex environments. For instance, in the Humanoid environment, critic-only scrubbing achieves more stable learning with fewer performance drops compared to full network scrubbing, which exhibits occasional instability.

In simpler environments like Pendulum and Reacher, the difference between critic-only and full network scrubbing is minimal, confirming that the PB is less pronounced in these lower-dimensional tasks. However, in more complex environments like Walker and Cheetah, critic-only scrubbing shows improved stability in the later stages of training.

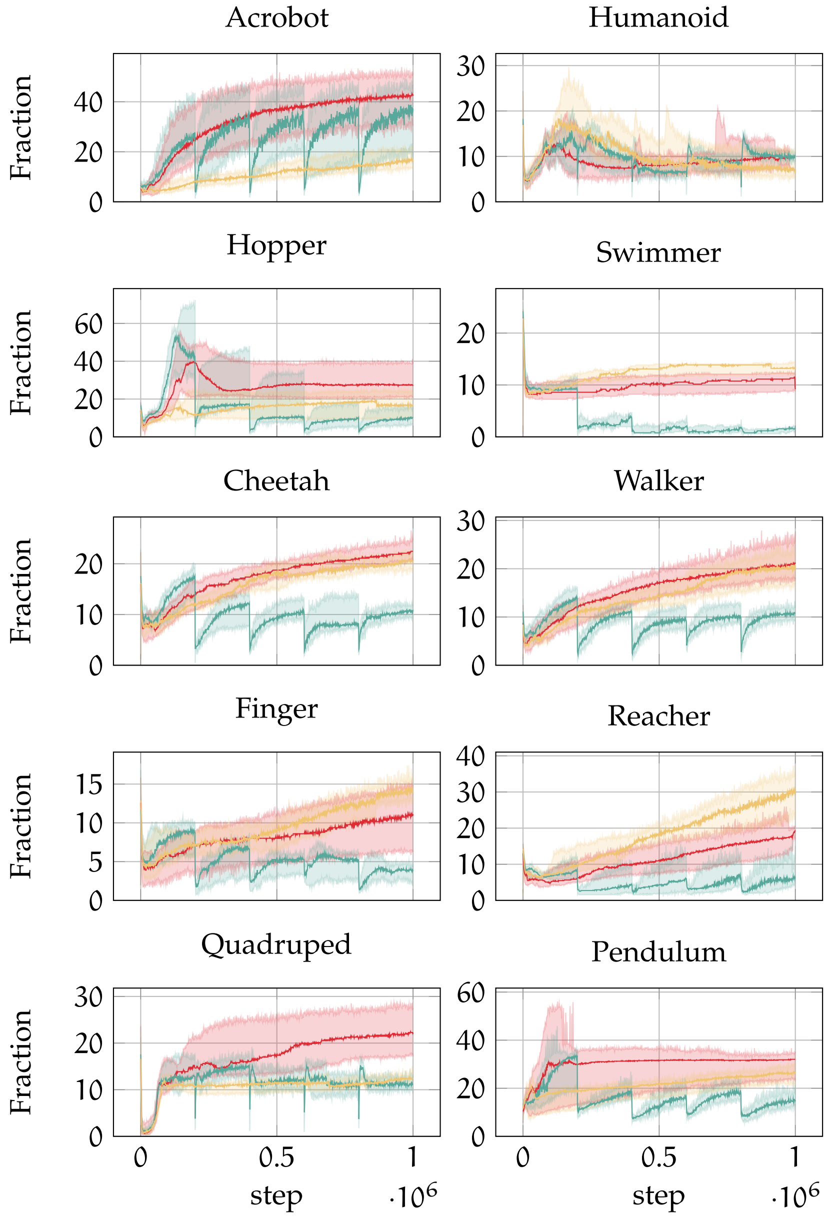

Figure 5.5: Fraction of dormant neurons during training across different environments. Plots compare baseline SAC (gold), reset method (teal), and FGSF (red), showing distinct patterns during memorization and reorganization phases

Overall, Critic-only scrubbing generally results in smoother learning curves with fewer oscillations, while full network scrubbing can sometimes lead to increased variance in performance, as seen in the `Humanoid` environment.

These findings suggest that the critic network plays a crucial role in the manifestation of the PB, aligning with the theoretical understanding that value estimation errors early in training can disproportionately affect subsequent policy updates (Lyle et al., 2022). The reduced performance variability with critic-only scrubbing indicates that maintaining the actor network's stability while selectively addressing bias in the critic might be an optimal strategy for complex continuous control tasks. This analysis provides strong evidence that the PB affects different components of the actor-critic architecture asymmetrically, with the critic network being particularly susceptible to early experience bias in complex environments. This insight has important implications for the design of future DRL algorithms, suggesting that targeted interventions focused on the critic network might be more effective than uniform approaches across all network components.

5.2.2 *Mechanistic Understanding*

The FIM trace analysis provides compelling evidence for the superior effectiveness of critic-only scrubbing compared to full network intervention. This analysis strengthens our understanding of how the PB manifests asymmetrically across different network components, as initially observed in our performance studies.

Examining $\text{Tr}(F)$patterns across environments reveals that critic-only scrubbing achieves more effective regularization of information flow during early training phases. The resulting stabilized $\text{Tr}(F)$values are consistently lower for both critic and actor networks compared to full network scrubbing, suggesting enhanced generalization capabilities as theorized by Jastrzebski et al. A particularly noteworthy finding is that critic-only scrubbing achieves comparable, and in some cases superior, regularization of $\text{Tr}(F_{actor})$ compared to full network scrubbing, despite not directly manipulating actor parameters. This observation underscores the critic network's central role in the PB development and its influence on overall learning dynamics.

The relationship between FIM patterns and network behavior provides strong empirical support for our theoretical framework developed in Chapter 2. The FIM proves particularly valuable as a diagnostic tool for understanding the different impact of the PB across network components. A striking observation is the order-of-magnitude difference in $\text{Tr}(F)$values between critic and actor networks, revealing fundamentally different operating regimes in parameter space. This asymmetry can be attributed to the critic's role in value estimation,

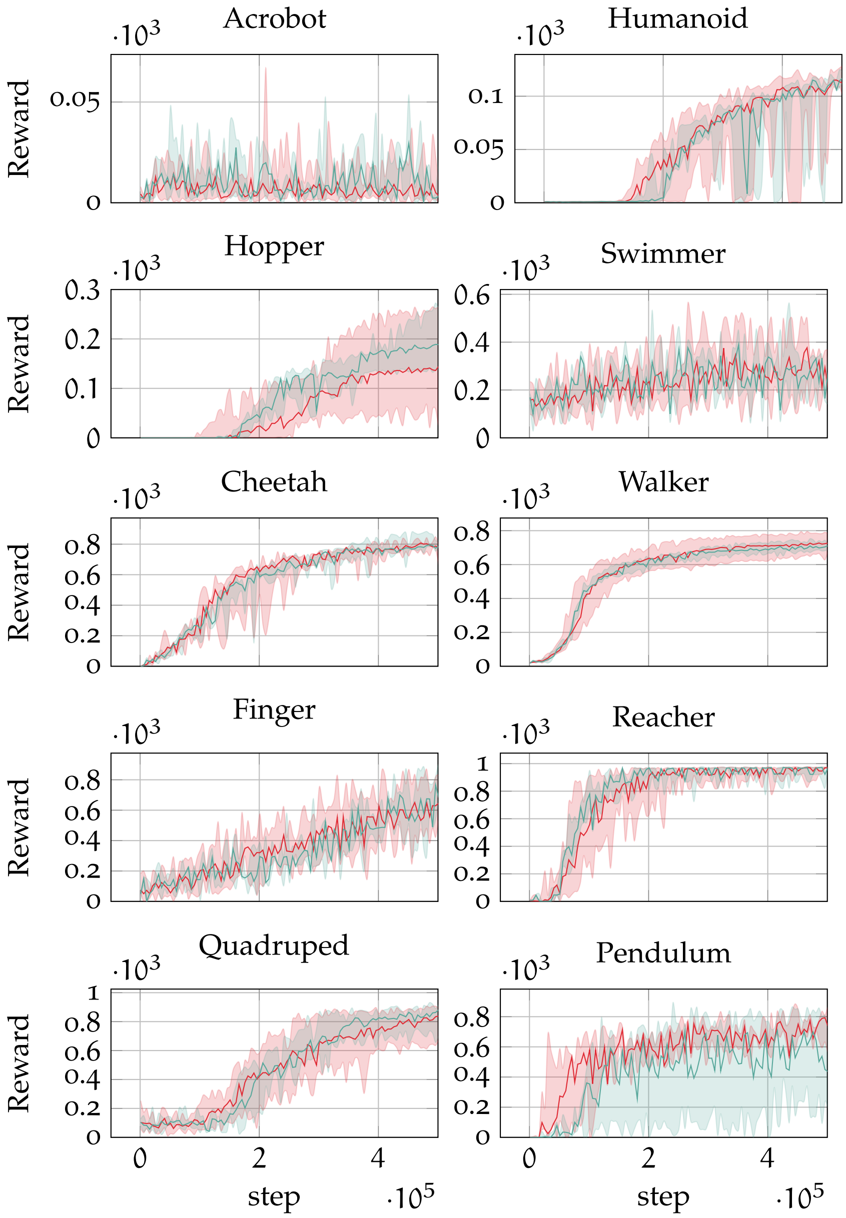

Figure 5.6: Comparison of learning curves between critic-only scrubbing (red) and full network scrubbing (teal) for the different environments.

which requires capturing complex state-value relationships early in training with only few examples. The higher $\mathrm{Tr}(F)$values observed in critic networks indicate operation in more curved regions of the parameter space during early learning phases, making them particularly susceptible to the PB, as predicted by our theoretical analysis in Section 2.2.

This geometric interpretation of learning dynamics through FIM analysis not only explains the effectiveness of critic-only scrubbing but also provides insights into the fundamental nature of the PB in actor-critic architectures. The critic network's operation in high-curvature regions of parameter space during critical early learning phases makes it particularly vulnerable to premature specialization, thereby acting as a primary conduit for the PB in the overall learning system.

5.3 ROBUSTNESS ANALYSIS

5.3.1 *Hyperparameter Sensitivity*

The analysis of FGSF's hyperparameter sensitivity demonstrates that while the method's effectiveness depends on the scrubbing coefficient λ, it maintains robust performance across a substantial range of values. The analysis centers on the scrubbing coefficient λ, examining its impact across multiple orders of magnitude (5×10^{-6} to 5×10^{-8}). This systematic evaluation demonstrates that while FGSF's effectiveness depends on λ, the method exhibits robust performance across a substantial range of values.

The relationship between λ and learning dynamics follows a clear pattern. Larger coefficients (5×10^{-6}) induce aggressive forgetting, manifesting as increased trajectory variability and, in some cases, learning disruption. Smaller coefficients (5×10^{-8}), while promoting stability, may inadequately address the PB. Intermediate values, particularly 5×10^{-7}, consistently achieve an optimal balance between learning stability and bias mitigation.

FIM trace analysis provides deeper mechanistic insights into these effects. While larger coefficients achieve stronger trace regularization, this doesn't necessarily correlate with improved performance. This observation highlights a crucial principle: excessive reduction in $\mathrm{Tr}(F)$can disrupt the natural progression between learning phases. Specifically, if traces are suppressed too strongly during the memorization phase, networks struggle to transition into the reorganization phase, compromising learning effectiveness.

Surprisingly, environment complexity exhibits minimal influence on optimal λ values, likely because the FIM inherently captures task-specific information in the scrubbing procedure. However, simpler

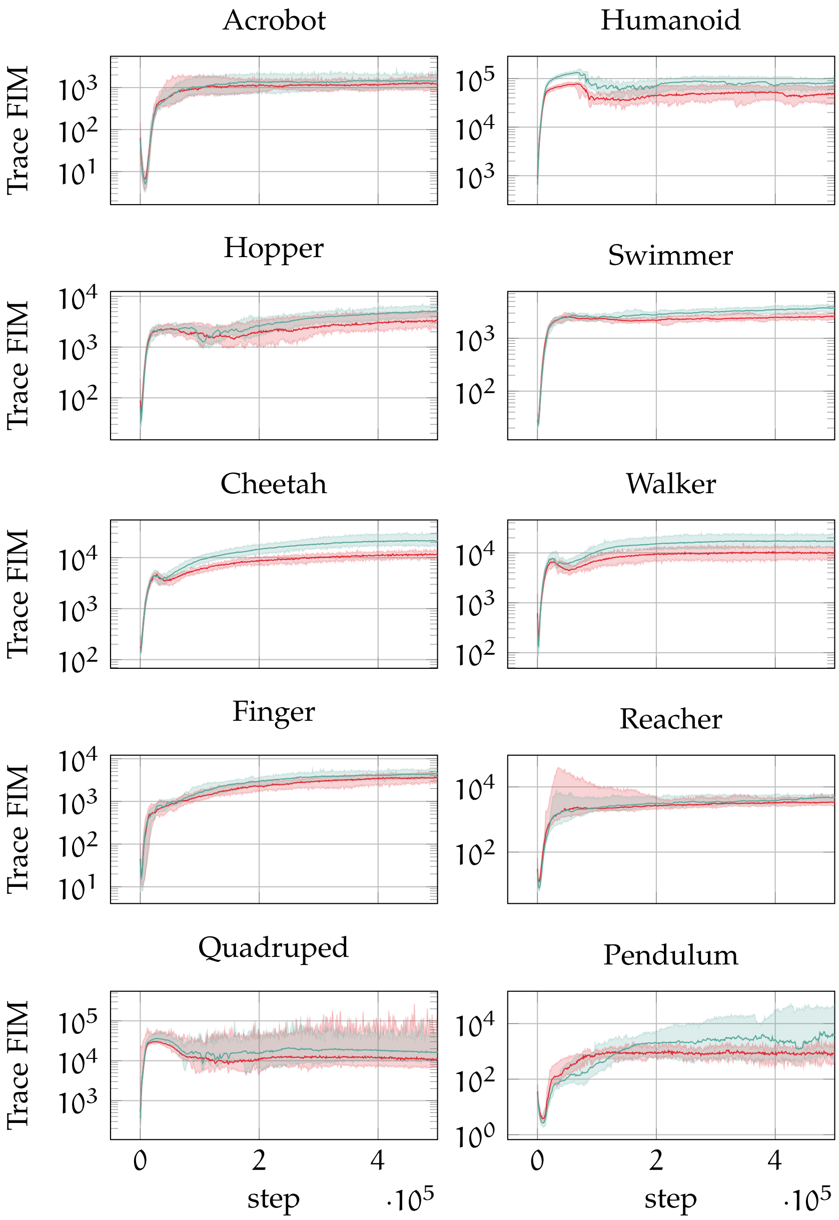

Figure 5.7: Actor network FIM trace evolution comparing critic-only scrubbing (red) versus full network scrubbing (teal) for different environments. Results demonstrate that critic-only scrubbing achieves effective regularization of actor network dynamics even without direct intervention.

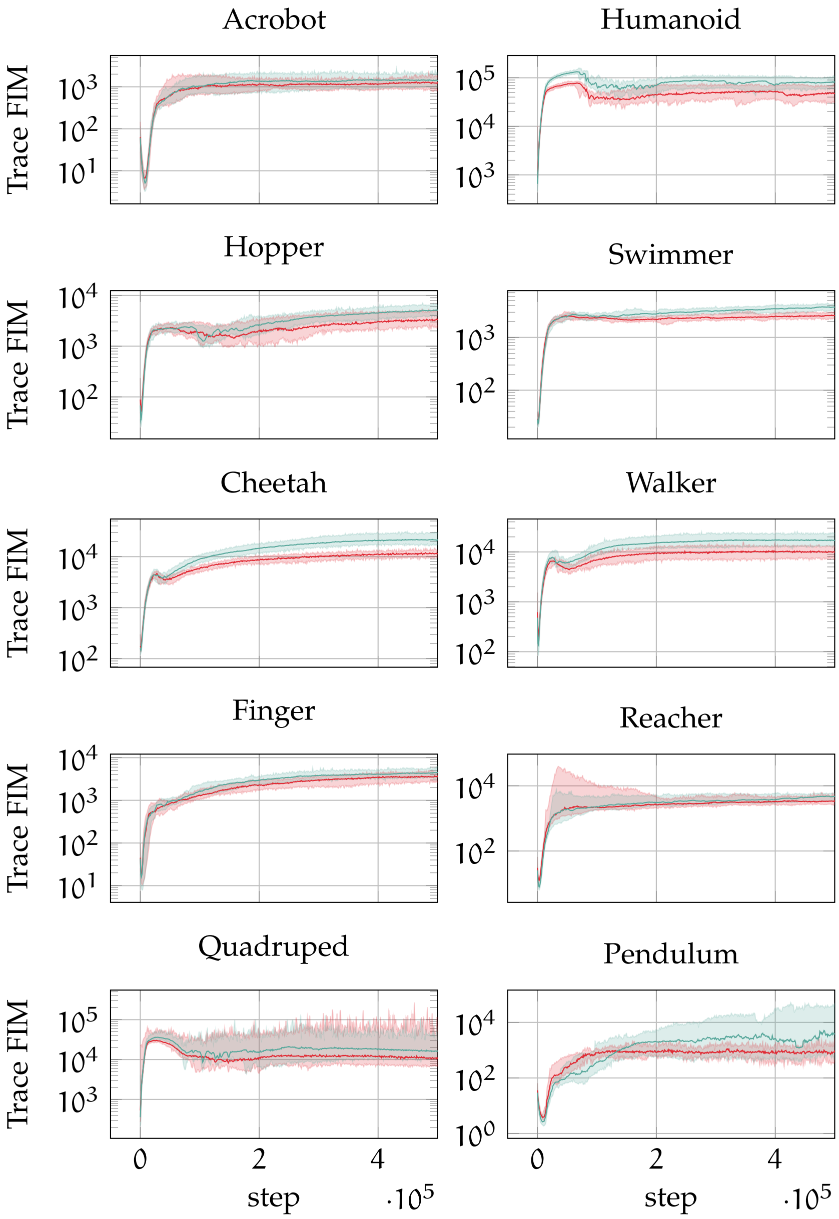

Figure 5.8: Critic network FIM trace evolution under critic-only scrubbing (red) versus full network scrubbing (teal) for different environments. The traces show stronger regularization effects in critic-only scrubbing.

environments show slightly better performance with lower λ values, suggesting potential for refinement in the scrubbing procedure to achieve more uniform effectiveness across environment complexities.

The FIM trace patterns serve as practical indicators for hyperparameter tuning. Rapid $\text{Tr}(F)$oscillations indicate the need for coefficient reduction, while inadequate post-memorization phase decline suggests insufficient λ values. Optimal settings typically maintain smooth transitions between learning phases while preserving the characteristic two-phase pattern identified in Section 4.1.

For practical implementation, we recommend:

1. Initial λ value of 5×10^{-7}
2. Monitoring both actor and critic FIM traces during early training
3. Adjusting λ based on observed learning stability and task characteristics
4. Using trace patterns as diagnostic tools for parameter refinement

These findings demonstrate that while FGSF exhibits sensitivity to the scrubbing coefficient, it maintains stability across a practical range of values. The clear relationship between FIM traces and learning outcomes provides a principled framework for parameter tuning, enhancing the method's applicability across diverse reinforcement learning tasks. This robustness to hyperparameter selection, combined with clear optimization guidelines, makes FGSF a practical tool for addressing the PB in real-world applications.

5.3.2 *Replay Ratio Impact*

The interaction between replay ratio and the PB provides a critical test of FGSF's robustness. Our investigation examines FGSF's effectiveness under elevated replay ratios (2 and 4), comparing against baseline SAC and the reset method to understand how different approaches handle the challenges of increased experience reuse. At replay ratio of 4, the performance differences between methods become stark. Baseline SAC shows significant performance deterioration, achieving only ≈ 600 reward points with substantial variance. The reset method, while showing better average performance, exhibits characteristic instability coinciding with reset events. FGSF, in contrast, maintains consistent performance around ≈ 800 reward points with markedly lower variance, demonstrating both superior initial learning rates and better asymptotic performance. A replay ratios of 2.0 yield improved stability across all methods, though FGSF retains its performance advantage with a narrower margin. The performance disparity between ratios 2.0 and 4.0 in baseline implementations reveals how increased replay amplifies the PB effects. While the reset method shows reduced

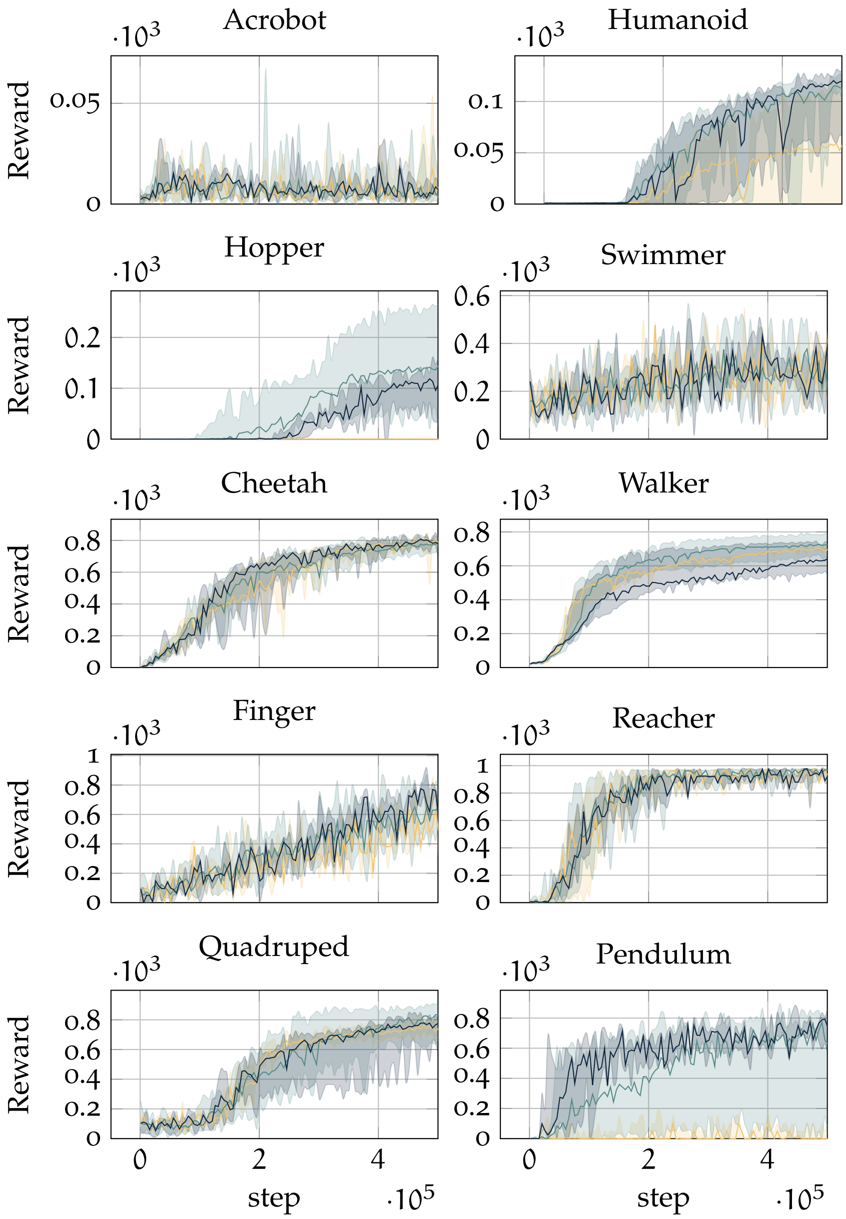

Figure 5.9: Hyperparameter sensitivity analysis showing performance across different scrubbing coefficients (λ). The lighter the color the higher the coefficient.

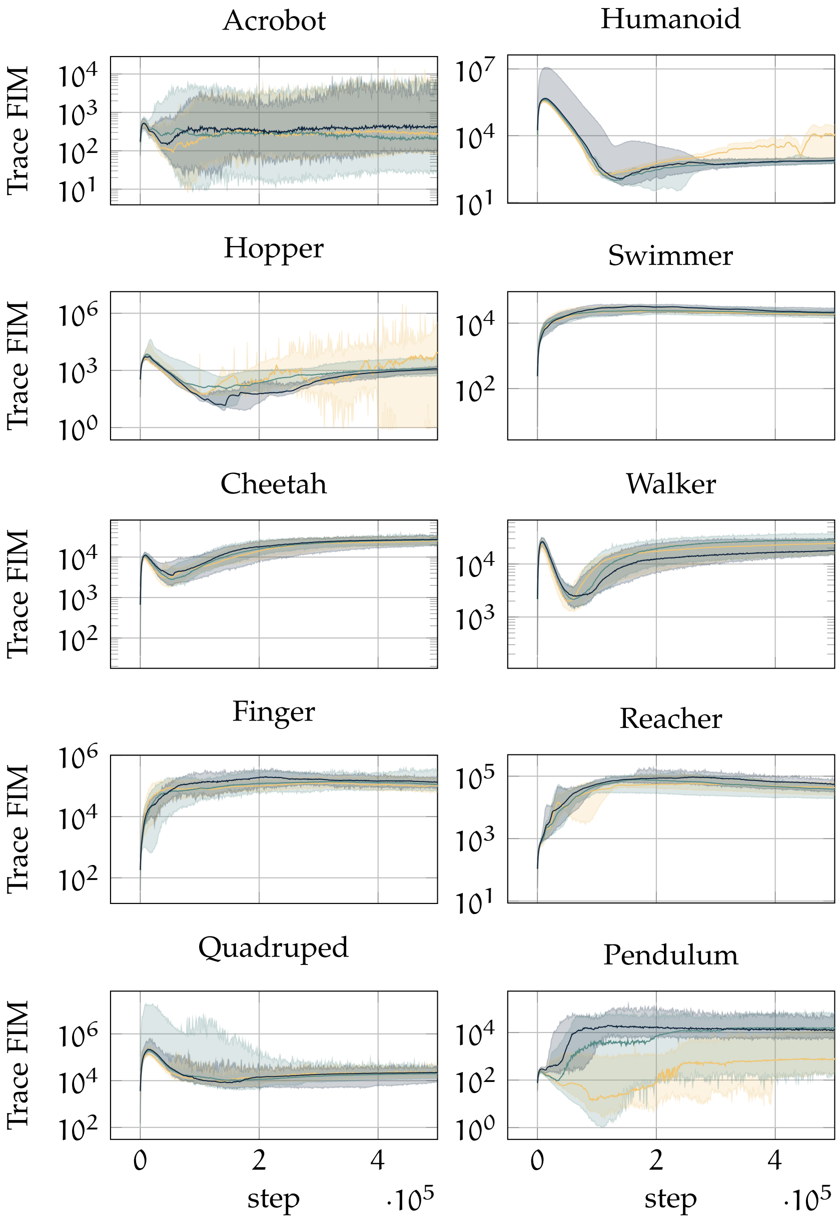

Figure 5.10: FIM trace of the critic network under different scrubbing coefficients, illustrating the relationship between λ values and the FIM trace. The lighter the color the higher the coefficient.

performance drops at lower ratios, it retains the periodic instabilities inherent to discrete intervention strategies.

FIM trace analysis reveals an unexpected pattern: despite FGSF's superior performance in terms of returns, it achieves less effective trace regularization compared to previous experiments. This observation, combined with our hyperparameter sensitivity findings, suggests that higher λ values might further improve performance under elevated replay ratios. This aligns with the intuition that increased replay frequencies amplify the PB, potentially requiring stronger forgetting mechanisms for optimal performance.

The interaction between replay ratio and forgetting mechanisms illuminates several key patterns: First, FGSF's selective forgetting becomes increasingly vital at higher replay ratios, effectively preventing the cascade of early experience amplification. Second, the continuous nature of FGSF provides fundamental advantages over discrete reset approaches, particularly in managing the accelerated memorization phase characteristic of high replay ratios. Lastly, the method's effectiveness scales proportionally with replay ratio, providing stronger mitigation precisely when needed most.

These findings have significant practical implications. FGSF enables the use of higher replay ratios without the typical performance degradation, enhancing sample efficiency in DRL applications. The method effectively decouples the traditional trade-off between replay ratio and stability, with its effectiveness automatically scaling to match increased replay frequencies. The relationship between replay ratio and FIM patterns provides strong support for our theoretical framework developed in Chapter 2. Higher replay ratios distinctly increase FIM magnitude during the memorization phase, making effective management of early learning dynamics crucial for performance. The $\mathrm{Tr}(F)$patterns observed under different replay configurations validate our theoretical model while providing practical guidance for FGSF implementation across various learning scenarios.

5.4 Ablation Study: Fisher vs. Gaussian Noise

To evaluate the importance of Fisher-guided noise injection, we conduct a comparative analysis between FGSF and a simpler Gaussian noise approach. The Gaussian noise variant samples perturbations from $\epsilon \sim \mathcal{N}(0, 0.001\mu)$, where μ represents the mean of network parameter values. While multiple noise formulations were possible, this simple implementation provides a clear baseline for assessing whether structured, Fisher-guided noise offers substantial advantages over basic stochastic perturbation.

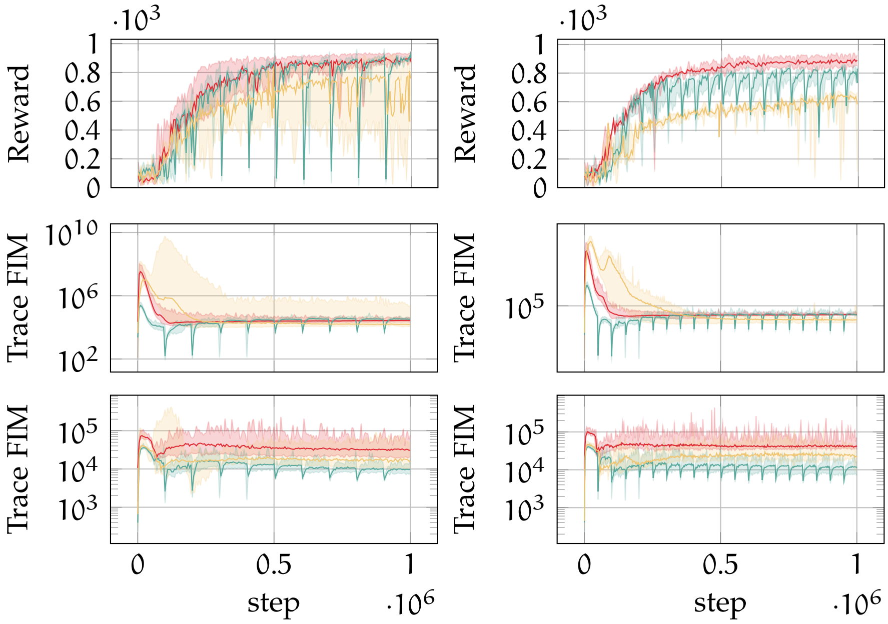

Figure 5.11: Performance comparison and FIM trace evolution under different replay ratios (2 left column and 4 right column) for baseline SAC , reset method, and FGSF. Higher replay ratios amplify the differences between methods.

In complex environments like Humanoid and Quadruped, FGSF demonstrates modest performance improvements over Gaussian noise while achieving significantly more stable learning trajectories. The Gaussian approach, though effective, exhibits higher performance variance, particularly in the Humanoid environment. This stability gap widens with increasing task dimensionality, suggesting that Fisher-guided structure becomes more valuable in complex parameter spaces. The performance difference narrows in simpler environments such as Reacher and Pendulum, where both methods achieve similar final returns. However, FGSF maintains advantages in learning speed and stability, though these benefits appear less critical in low-dimensional tasks.

Stability analysis reveals consistent patterns across environments: FGSF produces smoother learning curves, while Gaussian noise injection leads to higher episode return variance, more frequent performance degradation, and less predictable learning trajectories. This stability advantage becomes more pronounced with increasing task complexity, indicating that Fisher-guided noise enables more efficient parameter space exploration.

Learning dynamics show similar patterns, with FGSF achieving more consistent progress compared to the Gaussian approach's variable learning rates and convergence patterns. FGSF exhibits superior robustness across different random seeds and better handles challenging dynamics, particularly evident in the `Humanoid`'s coordination requirements, `Quadruped`'s balance constraints, and `Hopper`'s discontinuous state spaces. FIM trace analysis reveals minimal distinctions between approaches, reflecting their comparable performance in most environments. The notable exception is the `Quadruped` environment, where FGSF achieves superior performance. In this case, the Gaussian method's over-regularization of the trace corresponds with reduced performance, consistent with our previous hyperparameter sensitivity findings. These results yield two significant insights. First, while the geometric information captured by the FIM provides advantages for the PB mitigation, particularly in complex environments, simpler noise injection methods can still offer meaningful improvements over baseline approaches. Second, and perhaps more importantly, the effectiveness of both methods suggests that the PB phenomenon may be fundamentally linked to optimization dynamics, as proposed in Dohare et al.'s work on continual backpropagation. While FGSF provides an effective mitigation strategy, future work targeting core optimization algorithms might yield even more substantial improvements.

5.5 COMPUTATIONAL CONSIDERATIONS

While FGSF demonstrates clear benefits for addressing the PB, it's important to consider its computational overhead across different environments. Analysis of training time reveals environment-specific patterns in computational requirements. In high-dimensional environments like `Humanoid` and `Quadruped`, FGSF shows an approximately $15-20\%$ increase in cumulative update time compared to baseline SAC. This overhead remains relatively consistent throughout training, as evidenced by the parallel slopes in the timing curves. The `Walker` environment show similar patterns, with FGSF requiring about 15% additional computation time. Interestingly, the computational overhead appears more pronounced in complex locomotion tasks compared to simpler environments. For instance, in the `Reacher` and `Pendulum` environments, the additional computational cost is reduced to approximately $10-12\%$. This suggests that the FIM computation overhead scales reasonably well with environment complexity. The cumulative training times show linear growth across all environments, indicating that the computational overhead remains stable throughout the training process. This predictable scaling makes it easier to budget computational resources for FGSF implementation. Notably, while the reset method shows minimal computational overhead ($\approx 2-3\%$), its periodic performance disruptions often require longer training times to achieve comparable results. This moderate increase in computational cost should be weighed against FGSF's significant improvements in learning stability and final performance. For many applications, particularly in complex environments where FGSF shows the great-

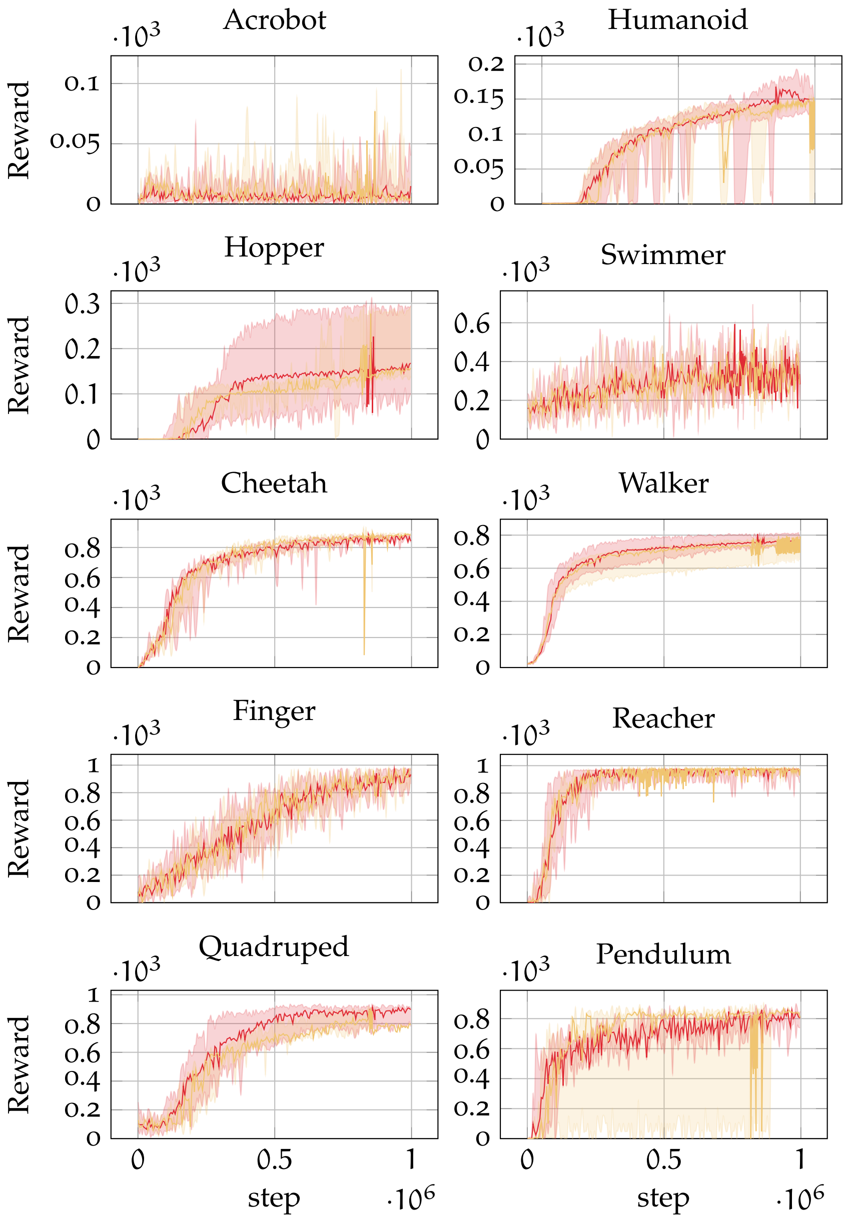

Figure 5.12: Performance comparison between FGSF and Gaussian noise injection across different environments. Both methods show improved performance over baseline.

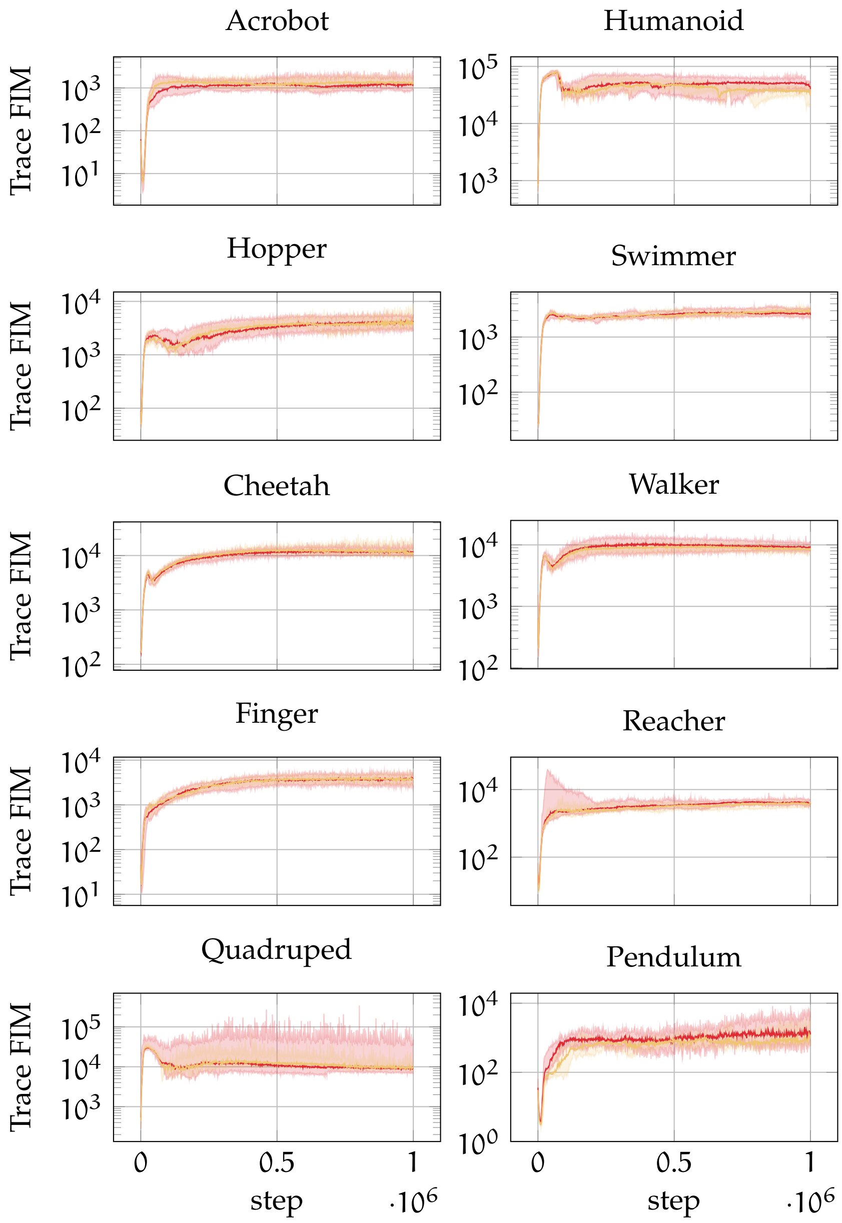

Figure 5.13: Actor network FIM traces comparing FGSF and Gaussian noise approaches.

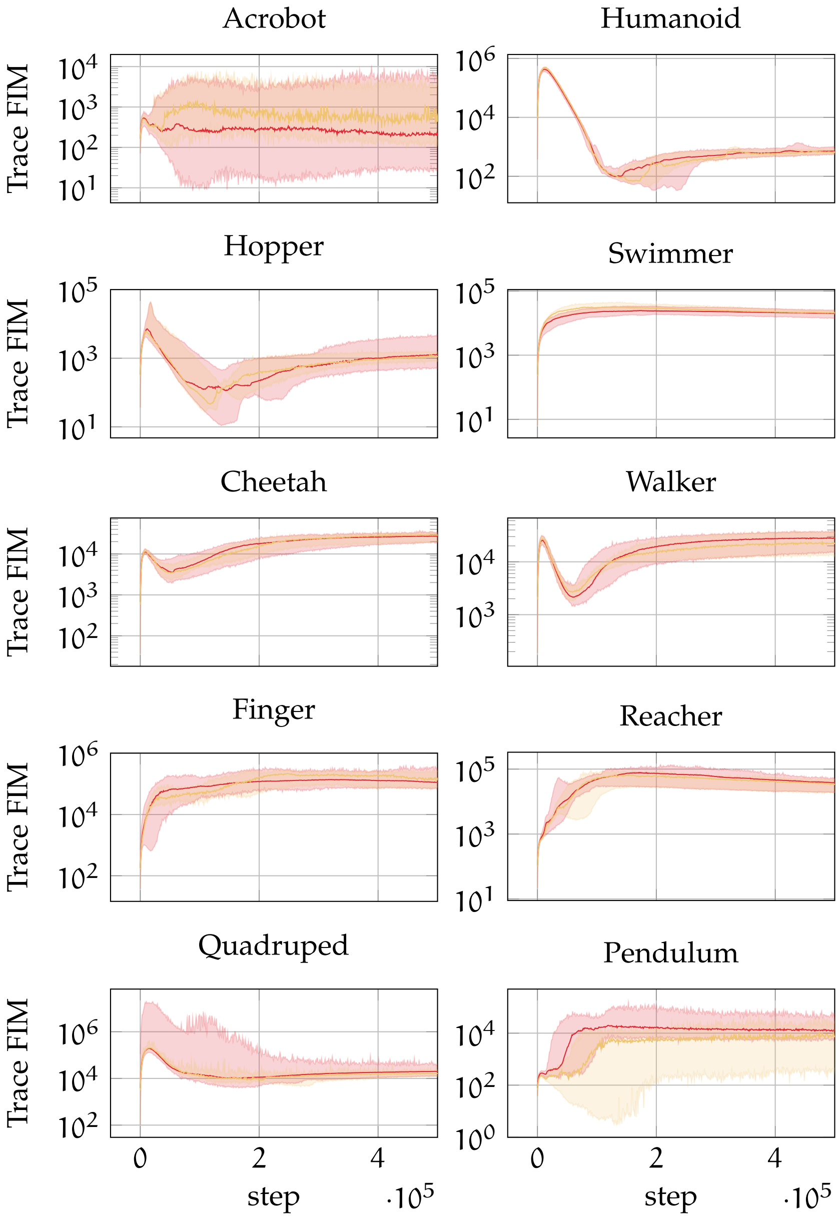

Figure 5.14: Critic network FIM traces for FGSF versus Gaussian noise injection.

est benefits, the enhanced sample efficiency and improved learning outcomes likely justify the additional computational investment.

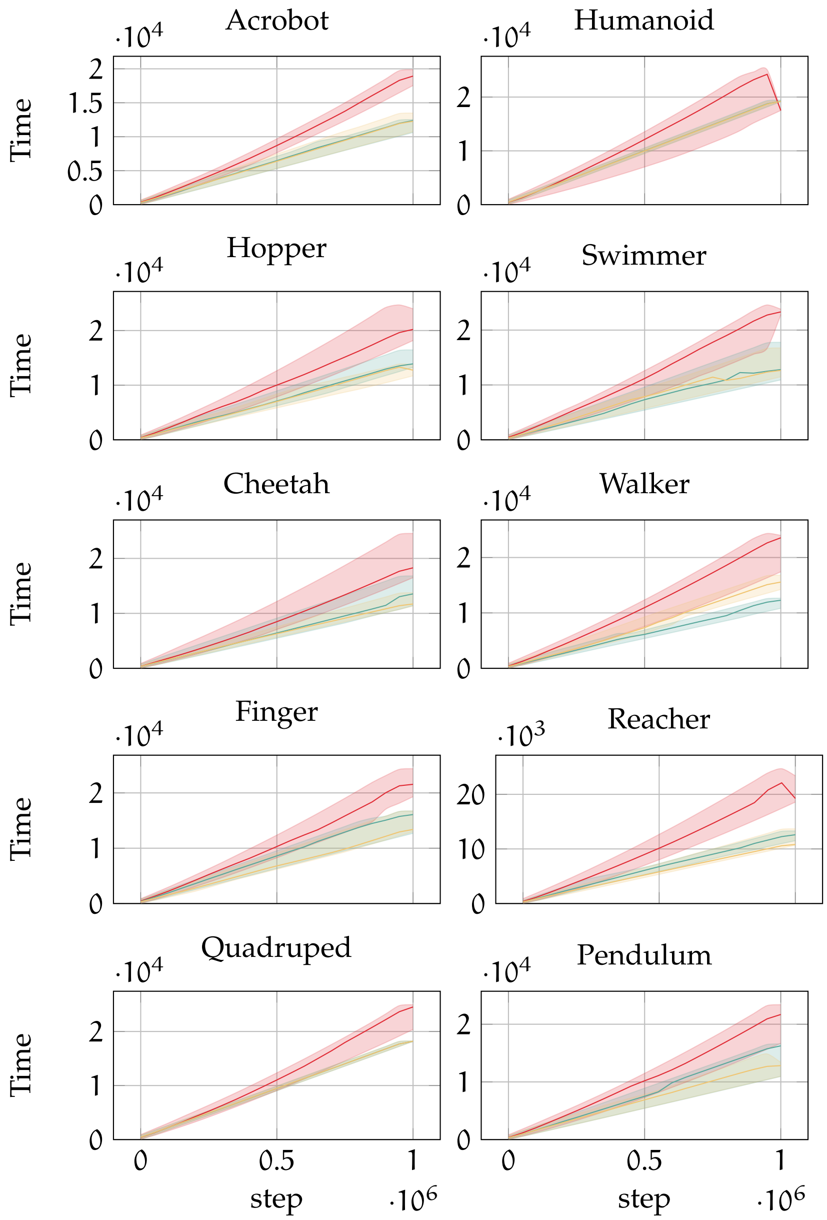

Figure 5.15: Comparative analysis of cumulative training time across environments. The y-axis shows total computation time in seconds, demonstrating the computational overhead of different methods. FGSF (red) shows a consistent $15 - 20\%$ overhead compared to baseline SAC (gold).

6 CONCLUSION

This thesis has achieved its primary objectives of analyzing the Primacy Bias (PB) phenomenon in Deep Reinforcement Learning (DRL), exploring its relationship with the Fisher Information Matrix (FIM), and developing an effective mitigation strategy. Through our investigation, we have demonstrated that the FIM provides both a theoretical framework for understanding the PB and a practical tool for its mitigation through our novel Fisher-Guided Selective Forgetting (FGSF) approach. Our comprehensive analysis has fulfilled our initial aims by: characterizing the PB through FIM patterns, establishing the connection between information geometry and learning dynamics, developing an FIM-based mitigation mechanism, and demonstrating its effectiveness compared to existing methods.

Our analysis reveals that the PB manifests through distinctive patterns in the information geometry of the learning process, characterized by a two-phase evolution in the FIM trace. The initial memorization phase, marked by rapid growth in $\text{Tr}(F)$, creates a critical period where early experiences disproportionately influence the learning trajectory. The subsequent reorganization phase, identified by declining $\text{Tr}(F)$values, often locks in these early biases while reducing the network's plasticity.

FGSF addresses this challenge by leveraging the geometric structure of the parameter space to selectively modify network weights. Our results demonstrate that this approach successfully maintains the beneficial aspects of early learning while preventing the oversized influence of initial experiences. The method's effectiveness scales with task complexity, showing particular promise in high-dimensional environments where the PB is most pronounced.

Several key findings emerge from our investigation:

1. The asymmetric impact of the PB on actor and critic networks, with critic-only scrubbing often outperforming full network intervention
2. The robust performance of FGSF across different replay ratios, suggesting its utility in improving sample efficiency
3. The surprising effectiveness of even simple noise injection methods, indicating that the underlying optimization dynamics play a crucial role in the PB

These results suggest broader implications for DRL algorithm design. The success of geometric approaches in addressing learning biases points toward the importance of considering information flow and parameter space structure in algorithm development. The clear relationship between FIM patterns and learning outcomes provides new tools for analyzing and improving DRL systems.

However, our work also reveals important limitations and areas for future research. The computational overhead of FIM computation remains a practical concern, particularly for large-scale applications. The relationship between network capacity, as measured through dormant neurons, and the PB requires further investigation. Additionally, the effectiveness of simpler noise injection methods suggests that alternative approaches targeting core optimization dynamics might yield complementary benefits.

Based on our extensive empirical evaluation, we provide concrete recommendations for implementing FGSF in practice. We recommend initializing λ at 5×10^{-7}, monitoring both actor and critic FIM traces during early training phases, and adjusting λ based on observed learning stability and specific task requirements. The FIM trace patterns serve as valuable diagnostic tools for parameter refinement, with rapid oscillations indicating the need for coefficient reduction and inadequate post-memorization decline suggesting insufficient λ values.

Future work might explore adaptive scrubbing strategies that dynamically adjust based on FIM patterns, integration with other bias mitigation techniques, and extension to more complex architectures and multi-agent systems. The theoretical framework developed here may also provide insights into related challenges in Deep Learning (DL), such as catastrophic forgetting and curriculum learning.

In conclusion, this thesis advances our understanding of the PB while providing practical tools for its mitigation. The geometric perspective offered by the FIM opens new avenues for analyzing and improving DRL systems, contributing to the development of more robust and efficient learning algorithms.

BIBLIOGRAPHY

Abbas, Zaheer, Rosie Zhao, Joseph Modayil, Adam White, and Marlos C Machado (2023). "Loss of plasticity in continual deep reinforcement learning." In: *Conference on Lifelong Learning Agents*. PMLR, pp. 620–636.

Achille, Alessandro, Matteo Rovere, and Stefano Soatto (2018). "Critical learning periods in deep networks." In: *International Conference on Learning Representations*.

Afsar, M Mehdi, Trafford Crump, and Behrouz Far (2022). "Reinforcement learning based recommender systems: A survey." In: *ACM Computing Surveys* 55.7, pp. 1–38.

Ahn, Hongjoon, Jinu Hyeon, Youngmin Oh, Bosun Hwang, and Taesup Moon (n.d.). "Catastrophic Negative Transfer: An Overlooked Problem in Continual Reinforcement Learning." In: ().

Amari, Shun-Ichi (1998). "Natural gradient works efficiently in learning." In: *Neural computation* 10.2, pp. 251–276.

Amari, Shun-ichi (2016). *Information geometry and its applications*. Vol. 194. Springer.

Asadi, Kavosh, Rasool Fakoor, and Shoham Sabach (2024). "Resetting the optimizer in deep rl: An empirical study." In: *Advances in Neural Information Processing Systems* 36.

Barnes, G Michael (1992). "Digitized speech's serial position effect." In: *Posters and Short Talks of the 1992 SIGCHI Conference on Human Factors in Computing Systems*, pp. 87–88.

Candan, Çağatay and Hakan Inan (2014). "A unified framework for derivation and implementation of Savitzky–Golay filters." In: *Signal Processing* 104, pp. 203–211.

Cho, Myungsik, Jongeui Park, Suyoung Lee, and Youngchul Sung (n.d.). "Hard Tasks First: Multi-Task Reinforcement Learning Through Task Scheduling." In: *Forty-first International Conference on Machine Learning*.

D'Oro, Pierluca, Max Schwarzer, Evgenii Nikishin, Pierre-Luc Bacon, Marc G Bellemare, and Aaron Courville (2022). "Sample-efficient reinforcement learning by breaking the replay ratio barrier." In: *Deep Reinforcement Learning Workshop NeurIPS 2022*.

Deese, James and Roger A Kaufman (1957). "Serial effects in recall of unorganized and sequentially organized verbal material." In: *Journal of experimental psychology* 54.3, p. 180.

Dohare, Shibhansh, J Fernando Hernandez-Garcia, Parash Rahman, A Rupam Mahmood, and Richard S Sutton (2023). "Maintaining plasticity in deep continual learning." In: *arXiv preprint arXiv:2306.13812*.

Ebbinghaus, Hermann (1913). "Memory: A contribution to experimental psychology." In: *Annals of neurosciences* 20.4, p. 155.

George, Thomas (2021). "NNGeometry: easy and fast fisher information matrices and neural tangent kernels in PyTorch." In.

George, Thomas, César Laurent, Xavier Bouthillier, Nicolas Ballas, and Pascal Vincent (2018). "Fast approximate natural gradient descent in a kronecker factored eigenbasis." In: *Advances in Neural Information Processing Systems* 31.

Golatkar, Aditya, Alessandro Achille, and Stefano Soatto (2020a). "Eternal sunshine of the spotless net: Selective forgetting in deep networks." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9304–9312.

Golatkar, Aditya, Alessandro Achille, and Stefano Soatto (2020b). "Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations." In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*. Springer, pp. 383–398.

Haarnoja, Tuomas, Aurick Zhou, Pieter Abbeel, and Sergey Levine (2018a). "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor." In: *International conference on machine learning*. PMLR, pp. 1861–1870.

Haarnoja, Tuomas, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. (2018b). "Soft actor-critic algorithms and applications." In: *arXiv preprint arXiv:1812.05905*.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Flat minima." In: *Neural computation* 9.1, pp. 1–42.

Hurtado Sánchez, Johanna Andrea, Katherine Casilimas, and Oscar Mauricio Caicedo Rendon (2022). "Deep reinforcement learning for resource management on network slicing: A survey." In: *Sensors* 22.8, p. 3031.

Jastrzebski, Stanislaw, Devansh Arpit, Oliver Astrand, Giancarlo B Kerg, Huan Wang, Caiming Xiong, Richard Socher, Kyunghyun Cho, and Krzysztof J Geras (2021). "Catastrophic fisher explosion: Early phase fisher matrix impacts generalization." In: *International Conference on Machine Learning*. PMLR, pp. 4772–4784.

Jiang, Li, Yichuan Ding, Xue Liu, and Xianyuan Zhan (n.d.). "Plasticity-Driven Sparsity Training for Deep Reinforcement Learning." In: ().

Juliani, Arthur and Jordan T Ash (2024). "A Study of Plasticity Loss in On-Policy Deep Reinforcement Learning." In: *arXiv preprint arXiv:2405.19153*.

Kakade, Sham M (2001). "A natural policy gradient." In: *Advances in neural information processing systems* 14.

Karakida, Ryo, Shotaro Akaho, and Shun-ichi Amari (2019). "Universal statistics of fisher information in deep neural networks: Mean field approach." In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1032–1041.

Kirkpatrick, James, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. (2017). "Overcoming catastrophic forgetting in neural networks." In: *Proceedings of the national academy of sciences* 114.13, pp. 3521–3526.

Li, Jingchen, Haobin Shi, Huarui Wu, Chunjiang Zhao, and Kao-Shing Hwang (2024). "Eliminating Primacy Bias in Online Reinforcement Learning by Self-Distillation." In: *IEEE Transactions on Neural Networks and Learning Systems*.

Li, Qiyang, Aviral Kumar, Ilya Kostrikov, and Sergey Levine (2023). "Efficient deep reinforcement learning requires regulating overfitting." In: *arXiv preprint arXiv:2304.10466*.

Lin, Long-Ji (1992). "Self-improving reactive agents based on reinforcement learning, planning and teaching." In: *Machine learning* 8, pp. 293–321.

Luber, Mattias, Anton Thielmann, and Benjamin Säfken (2023). "Structural neural additive models: Enhanced interpretable machine learning." In: *arXiv preprint arXiv:2302.09275*.

Lyle, Clare, Mark Rowland, and Will Dabney (2022). "Understanding and preventing capacity loss in reinforcement learning." In: *arXiv preprint arXiv:2204.09560*.

Lyle, Clare, Mark Rowland, Will Dabney, Marta Kwiatkowska, and Yarin Gal (2022). "Learning dynamics and generalization in deep reinforcement learning." In: *International Conference on Machine Learning*. PMLR, pp. 14560–14581.

Lyle, Clare, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney (2023). "Understanding plasticity in neural networks." In: *International Conference on Machine Learning*. PMLR, pp. 23190–23211.

MacQueen, J (1967). "Some methods for classification and analysis of multivariate observations." In: *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press*.

Marouf, Imad Eddine, Subhankar Roy, Enzo Tartaglione, and Stéphane Lathuilière (2023). "Weighted Ensemble Models Are Strong Continual Learners." In: *arXiv preprint arXiv:2312.08977*.

Martens, James and Roger Grosse (2015). "Optimizing neural networks with kronecker-factored approximate curvature." In: *International conference on machine learning*. PMLR, pp. 2408–2417.

Mitchell, Tom M (1997). *Machine learning*. Vol. 1. 9. McGraw-hill New York.

Murdock Jr, Bennet B (1962). "The serial position effect of free recall." In: *Journal of experimental psychology* 64.5, p. 482.

Nikishin, Evgenii, Junhyuk Oh, Georg Ostrovski, Clare Lyle, Razvan Pascanu, Will Dabney, and André Barreto (2024). "Deep reinforcement learning with plasticity injection." In: *Advances in Neural Information Processing Systems* 36.

Nikishin, Evgenii, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville (2022). "The primacy bias in deep reinforcement learning." In: *International conference on machine learning*. PMLR, pp. 16828–16847.

Obando-Ceron, Johan, Aaron Courville, and Pablo Samuel Castro (2024). "In value-based deep reinforcement learning, a pruned network is a good network." In: *Architecture* 4, pp. 4–5.

Onifade, Emmanuel O, Duane M Jackson, Tina R Chang, Jerry Thorne, and Cheryl Allen (2011). "RECALL AND THE SERIAL POSITION EFFECT: THE ROLE OF PRIMACY AND RECENCY ON ACCOUNTING STUDENTS'PERFORMANCE." In: *Academy of Educational Leadership Journal* 15.3, p. 65.

Pascanu, R (2013). "Revisiting natural gradient for deep networks." In: *arXiv preprint arXiv:1301.3584*.

Peters, Rik GM and Tammo HA Bijmolt (1997). "Consumer memory for television advertising: A field study of duration, serial position, and competition effects." In: *Journal of Consumer Research* 23.4, pp. 362–372.

Qiao, Zhongjian, Jiafei Lyu, and Xiu Li (2023). "The primacy bias in Model-based RL." In: *arXiv preprint arXiv:2310.15017*.

Rame, Alexandre, Corentin Dancette, and Matthieu Cord (2022). "Fishr: Invariant gradient variances for out-of-distribution generalization." In: *International Conference on Machine Learning*. PMLR, pp. 18347–18377.

Ramkumar, Vijaya Raghavan T, Bahram Zonooz, and Elahe Arani (2024). "The Effectiveness of Random Forgetting for Robust Generalization." In: *arXiv preprint arXiv:2402.11733*.

Sabatelli, Matthia and Pierre Geurts (2021). "On the transferability of deep-q networks." In: *arXiv preprint arXiv:2110.02639*.

Shao, Kun, Zhentao Tang, Yuanheng Zhu, Nannan Li, and Dongbin Zhao (2019). "A survey of deep reinforcement learning in video games." In: *arXiv preprint arXiv:1912.10944*.

Sokar, Ghada, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci (2023). "The dormant neuron phenomenon in deep reinforcement learning." In: *International Conference on Machine Learning*. PMLR, pp. 32145–32168.

Sutton, Richard S and Andrew G Barto (1999). "Reinforcement learning: An introduction." In: *Robotica* 17.2, pp. 229–235.

Tang, Chen, Ben Abbatematteo, Jiaheng Hu, Rohan Chandra, Roberto Martín-Martín, and Peter Stone (2024). "Deep reinforcement learning for robotics: A survey of real-world successes." In: *arXiv preprint arXiv:2408.03539*.

Tassa, Yuval, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. (2018). "Deepmind control suite." In: *arXiv preprint arXiv:1801.00690*.

Van Hasselt, Hado, Yotam Doron, Florian Strub, Matteo Hessel, Nicolas Sonnerat, and Joseph Modayil (2018). "Deep reinforcement learning and the deadly triad." In: *arXiv preprint arXiv:1812.02648*.

Xu, Heng, Tianqing Zhu, Lefeng Zhang, Wanlei Zhou, and P. Yu (2023). "Machine Unlearning: A Survey." In: *ACM Computing Surveys* 56, pp. 1 –36. URL: https://api.semanticscholar.org/CorpusID:259089053.