Democratic and Popular Republic of Algeria
Ministry of Higher Education and Scientific Research
University Mohamed Khider of Biskra

Faculty of Exact Sciences and Science of Nature and Life
Department of Computer Science

Ref: ……..

Thesis Presented to Obtain the Degree of
Doctorate in Computer Science

Entitled:

Evolutionary Developmental Robotics

Presented by:
Chourouk Guettas

Publicly defended on: 27/04/2026

Before the Jury Committee composed of:

Chaouki Babahenini	Prof.	President	University of Biskra
Cherif Foudil	Prof.	Supervisor	University of Biskra
Rachid Seghir	Prof.	Examiner	University of Batna 2
Fayçal Abbas	MCA	Examiner	University of Khenchla
Mohamed Kamel Benbraika	MCA	Examiner	University of El Oued

To the hearts that carried me further than I could go alone:

to my beloved mother, my dear father,
my amazing sisters and brothers,
and my lifelong friends.

Acknowledgments

Al-ḥamdu lillāh (all praise and thanks belong to Allah), whose mercy, guidance, and countless blessings sustained me throughout this doctoral journey. Without His grace, neither the strength nor the perseverance to complete this work would have been possible.

I express my sincere gratitude to **Professor Cherif Foudil, my supervisor**, for his invaluable guidance, unwavering patience, and steadfast support throughout this research journey. His mentorship has shaped not only this thesis but also my way of thinking as a researcher.

My deepest appreciation goes to the **esteemed jury members** for generously giving their time and expertise to evaluate this work. Your insights and critiques have been invaluable.

I am truly thankful to my **friends, Professors and colleagues**, whose encouragement, companionship, and thoughtful discussions lightened this demanding journey and enriched it with joy and support.

Most importantly, I extend my heartfelt gratitude to my **family—especially my loving mother, my dear father, my sisters, brothers, little Dani, and my extended family—for their constant love, sacrifices, and unwavering belief in me**. This accomplishment is as much theirs as it is mine.

Any shortcomings or errors that remain in this work are mine alone.

Chourouk Guettas

Abstract

Evolutionary Developmental Robotics

The field of robotics has long grappled with the challenge of creating adaptive learning systems that efficiently acquire new skills while maintaining previously learned capabilities. Traditional approaches rely on hand-engineered features and rigid architectures that struggle to generalize across diverse environments. While deep reinforcement learning has shown success, these methods suffer from sample inefficiency and require extensive computational resources that limit real-world deployment.

This research introduces two novel bio-inspired frameworks that translate natural regulatory mechanisms into computational advantages for robotic learning. The first contribution presents a Bio-Inspired Adaptive Rate Modulation (BIARM) framework that coordinates four computational modules corresponding to key neurotransmitter functions: dopamine for reward-based adaptation, serotonin for stability control, norepinephrine for attention gating, and acetylcholine for knowledge integration. Unlike existing methods that optimize single factors independently, BIARM implements coordinated multi-factor adaptation using organizational principles derived from biological neuromodulation.

The second contribution explores Gene Regulatory Networks (GRNs) for vision-based robotic control, representing robot states as gene expression levels and using evolutionary optimization to discover regulatory relationships that map sensory inputs to motor commands. This approach leverages fundamental biological properties including bounded dynamics, sparse connectivity patterns, and natural temporal reasoning without explicit memory modules.

Experimental validation demonstrates significant effectiveness. BIARM achieves performance improvements ranging from 1.06% to 155.94% over baseline Progressive Neural Networks across four OpenAI Gymnasium environments, while the GRN-based controller achieves 57.5% success rate in the KukaDiverseObjectEnv benchmark with up to 13.7× reduction in training time compared to established reinforcement learning baselines. These results demonstrate that biologically-inspired principles can offer algorithmic advantages for autonomous system design.

Keywords: Evolutionary developmental robotics, bio-inspired learning, neuromodulation, gene regulatory networks, adaptive systems, multi-task learning, robotic control, evolutionary computation.

الملخص

يُعَدُّ مجال الروبوتات من المجالات البحثية التي طالما واجهت تحدّي تطوير أنظمة تعلّم تكيفية قادرة على اكتساب مهارات جديدة بكفاءة عالية، مع الحفاظ في الوقت ذاته على القدرات المكتسبة سابقًا. وقد اعتمدت المقاربات التقليدية على خصائص مُصمَّمة يدويًا وبنى صارمة تعجز عن التعميم في البيئات المتنوعة. وعلى الرغم من النجاحات التي حققها التعلّم التعزيزي العميق، إلا أنّ هذه الأساليب ما تزال محدودة من حيث الكفاءة العيّنية وتتطلب موارد حسابية كبيرة، الأمر الذي يقيّد إمكانية تطبيقها العملي في الأنظمة المستقلة.

تهدف هذه الدراسة إلى اقتراح إطارين مبتكرين مستوحَيين من النظم البيولوجية، يسعيان إلى ترجمة الآليات التنظيمية الطبيعية إلى مزايا حسابية داعمة لعمليات التعلّم الروبوتي. يتمثل الإسهام الأول في إطار التنظيم الحيوي لمعدل التكيّف (BIARM)، الذي يقوم على تنسيق أربع وحدات حسابية تحاكي الوظائف الأساسية للناقلات العصبية، وهي: الدوبامين للتكيّف القائم على المكافأة، السيروتونين للتحكم في الاستقرار، النورأدرينالين لضبط الانتباه، والأسيتيل كولين لدمج المعرفة. وعلى خلاف النماذج السائدة التي تركز على تحسين العوامل بشكل مستقل، يعتمد هذا الإطار على التكيّف المتعدد العوامل بصورة منسّقة، مستندًا إلى المبادئ التنظيمية المستقاة من النمذجة العصبية البيولوجية.

أما الإسهام الثاني فيتمثل في استخدام الشبكات التنظيمية الجينية (GRNs) في التحكم الروبوتي المعتمد على الرؤية، حيث تُصاغ حالات الروبوت في صورة مستويات تعبير جيني، ويتم توظيف آليات التحسين التطوري لاكتشاف العلاقات التنظيمية التي تربط المدخلات الحسية بالمخرجات الحركية. ويستفيد هذا النهج من خصائص بيولوجية جوهرية مثل الديناميكيات المحدودة، وأنماط الترابط المتناثرة، والاستدلال الزمني الطبيعي دون الحاجة إلى وحدات ذاكرة صريحة.

وقد خلصت النتائج التجريبية إلى تحقيق مكاسب معتبرة، حيث أظهر إطار BIARM تحسّنًا في الأداء تراوح بين 1.06% و 155.94% مقارنة بخط الأساس المتمثل في الشبكات العصبية التقدمية عبر أربع بيئات في OpenAI Gymnasium. كما حقق المتحكّم المستند إلى GRN معدل نجاح بلغ 57.5% في معيار KukaDiverseObjectEnv، مع تقليص زمن التدريب بما يصل إلى 13.7 ضعفًا مقارنةً بالمقاربات المعتمدة في التعلّم التعزيزي. وتؤكد هذه النتائج أنّ مجال الروبوتات التطورية التنموية يُمثل إطارًا بحثيًا واعدًا، وتخلص الدراسة إلى أنّ المبادئ البيولوجية تتيح مزايا خوارزمية أصيلة من شأنها إحداث نقلة نوعية في تصميم الأنظمة المستقلة.

الكلمات المفتاحية: الروبوتات التطورية التنموية، التعلّم المستوحى من الأحياء، التنظيم العصبي، الشبكات التنظيمية الجينية، الأنظمة التكيفية، التعلّم متعدد المهام، التحكم الروبوتي، الحوسبة التطورية.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

5-HT 5-Hydroxytryptamine (Serotonin)

ACh Acetylcholine

AdaGrad Adaptive Gradient Algorithm

BIARM Bio-Inspired Adaptive Rate Modulation

CAP-cAMP Catabolite Activator Protein–cyclic Adenosine Monophosphate

CDF Cumulative Distribution Function

CMA-ES Covariance Matrix Adaptation Evolution Strategy

CNN Convolutional Neural Network

DA Dopamine

DEN Dynamically Expandable Network

DOF Degrees of Freedom

DQN Deep Q-Network

DRL Deep Reinforcement Learning

ER Evolutionary Robotics

ES Evolution Strategies

EWC Elastic Weight Consolidation

FC Fully Connected

GABA Gamma-Aminobutyric Acid

GEM Gradient Episodic Memory

GRN Gene Regulatory Network(s)

HER Hindsight Experience Replay

HyperNEAT Hypercube-based NEAT

iCaRL Incremental Classifier and Representation Learning

IPG Interpolated Policy Gradient

LSTM Long Short-Term Memory

MAML Model-Agnostic Meta-Learning

MAP-Elites Multi-dimensional Archive of Phenotypic Elites

NEAT NeuroEvolution of Augmenting Topologies

NE Norepinephrine

PackNet PackNet (Continual Learning Architecture)

PNN Progressive Neural Network(s)

PPO Proximal Policy Optimization

QT-Opt QT-Optimization

ReLU Rectified Linear Unit

Reptile Reptile (Meta-Learning Algorithm)

RGB Red Green Blue

RGB-D Red Green Blue–Depth

RL Reinforcement Learning

RMSprop Root Mean Square Propagation

SI Synaptic Intelligence

SN Substantia Nigra

SNN Spiking Neural Network

TD3 Twin Delayed Deep Deterministic Policy Gradient

VTA Ventral Tegmental Area

INTRODUCTION

The development of truly autonomous robotic systems remains one of the central challenges in robotics and artificial intelligence. Despite decades of progress, robots continue to struggle with the dynamic and unpredictable nature of real-world environments [1, 2]. This limitation stems from a fundamental mismatch between deterministic, model-dependent control paradigms and the inherent variability characterising real-world interactions [1, 3].

Classical behaviour-based architectures [4] provided important insights into reactive control systems but proved insufficient for complex manipulation tasks requiring sophisticated learning and adaptation. While these frameworks successfully demonstrated that intelligent behaviour could emerge from simple reactive mechanisms [5], they lacked the capabilities necessary for robots to acquire new skills autonomously or adapt based on experience.

The emergence of bio-inspired robotics offered a promising alternative by drawing inspiration from biological systems that have evolved sophisticated mechanisms for adaptation and robust control [6, 7]. However, many bio-inspired approaches have focused on superficial mimicry rather than incorporating the fundamental principles underlying biological intelligence. Evolutionary robotics, which applies evolutionary algorithms to automatically design robot controllers, has made significant contributions [8, 9], yet critical limitations remain.

Contemporary deep reinforcement learning methods, while achieving impressive results in specific domains [10, 11, 12], suffer from severe sample inefficiency, often requiring millions of interactions to learn relatively simple tasks [2, 13]. This inefficiency becomes particularly problematic in robotics, where each physical interaction carries temporal and economic costs that make extensive exploration prohibitively expensive [14].

Recent approaches have achieved remarkable success in game-playing and specific robotic tasks [15, 16, 17], yet fundamental challenges persist. Sample inefficiency, catastrophic forgetting in continual learning [18], and brittleness under distribution shift [19] constrain practical deployment. The gap between simulated and real-world performance, commonly known as the reality gap, continues to limit the transferability of learned behaviours [20].

The continual learning problem represents a particularly critical challenge. When learning new tasks, artificial systems typically suffer catastrophic forgetting, losing previously acquired knowledge [21, 18]. Architectural solutions like Progressive Neural Networks [22] and regularisation methods such as Elastic Weight Consolidation [18] address this problem partially, but these approaches fail to capture the sophisticated coordination mechanisms that enable biological systems to learn continuously throughout their lifetimes.

The efficiency gap between biological and artificial learning is striking [23, 24]. Biological organisms acquire adaptive capabilities through developmental programmes that coordinate growth, learning, and adaptation across multiple temporal and spatial scales [25, 26], achieving a degree of efficiency that artificial systems have yet to match. This efficiency stems not from unlimited computational resources but from the coordinated action of multiple biological mechanisms refined through evolutionary processes.

These observations motivate a fundamental reconsideration of how we approach learning and adaptation in robotic systems [27, 28]. Rather than pursuing incremental improvements to existing paradigms, this thesis explores whether insights from developmental biology, gene regulatory networks, and neuromodulatory systems can provide a more principled foundation for creating adaptive robotic systems [29, 30].

Research Questions

This investigation is guided by several key research questions addressing both theoretical and practical aspects of bio-inspired robotics.

First, how can sophisticated mechanisms governing biological development, from gene regulatory networks to neuromodulatory systems, be translated into computational frameworks that enhance learning efficiency and robustness in robotic systems? Biological development represents a proven solution to coordinating complex adaptive processes across multiple scales, from molecular interactions to organism-level behaviour [31].

Second, can neuromodulatory principles, which coordinate learning and adaptation in biological brains through multiple neurotransmitter systems, be effectively implemented in artificial systems to overcome the limitations of fixed-parameter learning algorithms? This question is motivated by extensive neuroscientific research demonstrating that biological systems employ sophisticated mechanisms for dynamically adjusting learning parameters based on context [28, 32, 29, 30].

Third, rather than viewing biological constraints as limitations, can these constraints serve as optimisation guides that focus the search process on robust solutions? This question challenges the conventional assumption that more flexibility necessarily leads to better learning performance [24, 33], aligning with recent work on inductive biases and architectural constraints in machine learning [23, 34, 35].

Fourth, what role can gene regulatory networks, the fundamental control systems governing cellular behaviour in living organisms, play in creating efficient and robust robot controllers? Gene regulatory networks have evolved to solve complex control problems in noisy, resource-constrained environments while maintaining remarkable robustness and adaptability [31, 36].

Fifth, how do bio-inspired constraints and mechanisms affect training efficiency, robustness, and generalisation capabilities compared to conventional approaches? This question is critical for establishing whether bio-inspired methods offer genuine practical advantages beyond interesting theoretical alternatives.

Finally, can the principles demonstrated in specific applications be generalised to create a coherent framework for developmental evolutionary robotics that guides the design of adaptive systems across multiple domains? This addresses the broader goal of establishing developmental evolutionary robotics as a principled approach rather than a collection of isolated techniques [27].

These research questions collectively frame an investigation into whether developmental evolutionary robotics can provide a more principled and effective approach to creating adaptive robotic systems. By grounding this investigation in specific biological mechanisms and computational implementations, this thesis aims to move beyond superficial biomimicry towards a deeper understanding of the computational principles that make biological systems remarkably efficient and robust.

Thesis Contributions

This thesis contributes to the development and advancement of developmental evolutionary robotics by demonstrating practical advantages through concrete implementations and comprehensive experimental validation. The contributions span theoretical insights, methodological innovations, and empirical findings.

Theoretical Contributions

The primary theoretical contribution lies in synthesising and extending the developmental evolutionary robotics approach through systematic investigation of specific biological mechanisms and their computational implementations. This work advances the field by establishing clear design principles for bio-inspired adaptive systems, particularly regarding the strategic use of biological constraints as optimisation guides rather than limitations. This represents a departure from traditional approaches that maximise flexibility, instead demonstrating empirically how carefully chosen constraints enhance learning efficiency and robustness.

The thesis provides a systematic analysis of how neuromodulatory principles and gene regulatory mechanisms can be effectively translated into computational frameworks. By formalising the coordination of multiple adaptive processes, inspired by biological neurotransmitter systems, this work contributes to understanding how multi-factor adaptation can address limitations of single, parameter optimisation approaches in artificial systems.

Methodological Contributions

Two methodological contibutions are presented:

Bio-Inspired Adaptive Rate Modulation (BIARM): A framework coordinating four computational modules corresponding to key neurotransmitter functions, dopamine for reward-based adaptation, serotonin for stability control, norepinephrine for attention gating, and acetylcholine for knowledge integration. Unlike existing methods optimising single factors independently, BIARM implements coordinated multi-factor adaptation using organisational principles derived from biological neuromodulation [37].

Gene Regulatory Networks (GRNs) for Robotic Control: An approach representing robot states as gene expression levels and using evolutionary optimisation to discover regulatory relationships mapping sensory inputs to motor commands. This leverages fundamental biological properties including bounded dynamics, sparse connectivity patterns, and natural temporal reasoning without explicit memory modules [38].

Empirical Contributions

Experimental validation demonstrates significant effectiveness. BIARM achieves performance improvements ranging from 1.06% to 155.94% over baseline Progressive Neural Networks across four OpenAI Gymnasium environments [39, 37]. The

GRN-based controller achieves a 57.5% success rate in the KukaDiverseObjectEnv benchmark with a 13.7× reduction in training time compared to established reinforcement learning baselines.

The robustness analysis demonstrates that bio-inspired controllers maintain high performance under noisy conditions and exhibit enhanced generalisation capabilities. GRN controllers retain over 91% of their clean performance when subjected to significant sensory noise, indicating that biological constraints contribute to robustness rather than limiting flexibility.

These contributions establish a foundation for future research in developmental evolutionary robotics whilst demonstrating immediate practical advantages for robotic applications. The work provides both theoretical insights and empirical evidence supporting the effectiveness of bio-inspired adaptive systems.

Thesis Structure

This thesis is organised into five chapters that progressively build from theoretical foundations through specific implementations to broader implications and future directions.

Chapter 1 presents the biological foundations of developmental evolutionary robotics, examining gene regulatory networks and neuromodulatory systems in biological organisms. The chapter establishes how these biological principles translate into computational advantages.

Chapter 2 provides a comprehensive literature review situating this work within the broader context of evolutionary robotics, continual learning, and bio-inspired computation. The review systematically examines historical development, key achievements, and persistent limitations motivating this work.

Chapter 3 presents the bio-inspired adaptive rate modulation framework for sequential multi-task learning, providing detailed mathematical formulations, implementation strategies, and experimental validation.

Chapter 4 examines gene regulatory networks for vision-based robot control, demonstrating how this biological paradigm adapts for robotic manipulation tasks whilst achieving remarkable efficiency improvements.

Chapter 5 provides comparative analysis integrating insights from both major contributions, examining commonalities and differences whilst exploring possibilities for their combination. This chapter addresses scalability considerations and discusses how demonstrated principles might extend to more complex robotic applications.

The Conclusion synthesises major findings and their implications for the field of robotics, emphasising both practical advantages demonstrated by bio-inspired approaches and theoretical insights emerging from this investigation.

Throughout the thesis, extensive experimental validation and ablation studies provide empirical support for theoretical claims whilst detailed mathematical formulations ensure reproducibility and extensibility.

List of Publications

Journal Papers:

1. **Gene Regulatory Networks for Enhanced Vision-Based Robot Control: A Bio-Inspired Approach**. MDPI Sensors 26(6), 1742, 2026. **Authors**: Chourouk Guettas, Foudil Cherif, Ammar Muthanna, Mohammad Hammoudeh, and Abdelkader Laouid.
2. **Bio-Inspired Adaptive Rate Modulation for Multi-task Learning in Progressive Neural Networks**. Informatica, 49(21):437–448, 2025. **Authors**: Chourouk Guettas, Foudil Cherif.

CHAPTER 1

BIOLOGICAL FOUNDATIONS OF DEVELOPMENTAL EVOLUTIONARY ROBOTICS

The remarkable capabilities of living organisms, from cellular homeostasis to complex adaptive behaviours, emerge from sophisticated control systems that have evolved over billions of years. Biological control systems differ fundamentally from engineered systems in their reliance on distributed decision-making, temporal dynamics, and robust performance under uncertainty. Understanding these control principles is essential for developing more capable and efficient artificial autonomous systems.

1.1 Gene Regulatory Networks in Biology

Biological control systems operate through molecular networks that work together to sense information, make decisions, and coordinate responses across different temporal and spatial scales. Unlike engineered control systems that typically employ centralised processing units with deterministic control laws, biological systems achieve complex behaviours as emergent properties of distributed molecular networks, stochastic processes, and stable regulatory mechanisms [31].

The evolution of regulatory complexity reflects fundamental computational advantages that arise from constrained, networked architectures [36]. Rather than maximising the capabilities of individual components, evolution has favoured systems that achieve reliable performance through coordinated interactions between relatively simple components operating within restricted parameter ranges. This biological design philosophy contrasts with traditional engineering approaches that

seek to enhance system capabilities through additional parameter flexibility and computational power.

Two classes of biological control systems are particularly relevant to autonomous robotics: gene regulatory networks that govern cellular development and behaviour, and neuromodulatory systems that coordinate neural activity and adaptive behaviour. Both exhibit common characteristics, sparse connectivity, bounded dynamics, multi-timescale operation, and inherent robustness, that could prove highly beneficial for artificial control systems when appropriately abstracted and implemented [40, 29].

1.1.1 Gene Regulation at the Molecular Level

The fundamental components of gene regulatory networks are regulatory DNA sequences, transcription factors (regulatory proteins), and genes. Transcription factors activate or repress the transcription of target genes by binding to specific DNA sequences located in gene promoter regions. This binding exhibits sigmoidal dose-response relationships with cooperative binding characteristics that are hallmarks of biological regulatory systems [41].

The mathematical description of transcriptional regulation follows Hill kinetics, where the gene expression rate depends on regulatory protein concentration:

$$\text{Expression Rate} = V_{\max} \cdot \frac{[\text{TF}]^n}{K_d^n + [\text{TF}]^n} \tag{1.1}$$

where [TF] represents transcription factor concentration, K_d is the dissociation constant, n is the Hill coefficient representing cooperativity, and $V_{\max}$ is the maximum expression rate [42].

This cooperative binding creates switch-like, threshold-like responses that enable gene regulatory networks to make discrete decisions from continuous input signals [43]. The steepness of the response is determined by the Hill coefficient, with higher values producing more switch-like behaviour. This mathematical relationship provides a natural mechanism for implementing decision circuits and logic gates without explicit programming.

1.1.2 Network Properties and Organisation

Gene regulatory networks exhibit specific organisational principles that distinguish them from random networks. They possess scale-free topology, meaning that a few

highly connected hub genes regulate numerous targets whilst most genes have limited connectivity [44]. This architecture provides both robustness to random gene disruptions and vulnerability to targeted removal of hubs, reflecting evolutionary optimisation for common perturbations rather than catastrophic failures.

Network motifs, recurring patterns of regulatory circuitry, serve as fundamental building blocks of gene regulatory networks [45, 46]. The most common motif, the feed-forward loop, filters transient signals and creates temporal delays in responses [47]. This motif enables cells to distinguish between sustained environmental signals and transient fluctuations, implementing temporal logic that makes decision-making more reliable.

Negative feedback loops provide homeostatic control and can generate oscillatory behaviours when combined with appropriate time delays [48]. The toggle switch motif, composed of two mutually repressing genes, implements bistability that enables discrete cell fate decisions [49]. These circuit motifs demonstrate how simple regulatory connections can produce complex, robust behaviours.

Figure 1.1 illustrates a simplified gene regulatory network exhibiting the key organizational principles discussed above. The sparse connectivity pattern (24% of possible connections) emerges from evolutionary and metabolic constraints, while the combination of activation and inhibition enables complex regulatory dynamics. The network exhibits several important motifs: a mutual regulatory relationship between G1 and G2 (positive activation from G1, negative inhibition from G2), a feed-forward loop through G2→G3→G5 with G2 also directly inhibiting G5, and self-regulation of G3 through auto-inhibition. These regulatory patterns allow the network to maintain homeostasis while responding to perturbations.

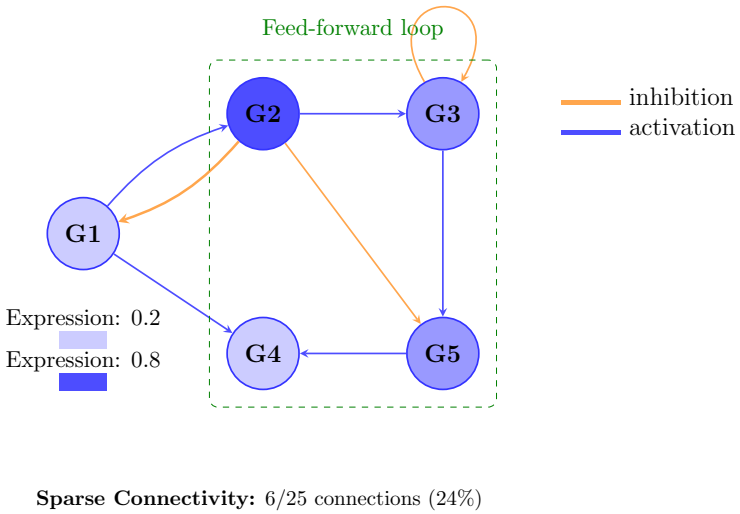


Figure 1.1: **Gene Regulatory Network Architecture.**

1.1.3 Examples from Developmental Biology

The circadian clock illustrates how gene regulatory networks generate autonomous rhythms with approximately 24-hour periods [50]. This system comprises transcriptional negative feedback loops where clock proteins repress their own transcription with appropriate delays. The robustness of circadian rhythms to molecular noise and environmental perturbations demonstrates the effectiveness of biological regulatory design principles.

Cell fate determination during development shows how gene regulatory networks make stochastic yet discrete decisions and maintain stable cellular states [51, 52]. Cellular differentiation involves regulatory networks with multiple stable attractor states, one for each cell type. These networks integrate developmental signals and progressively restrict cellular potential until cells commit to specific fates that are maintained throughout the organism's lifetime.

The lac operon in bacteria demonstrates how gene regulatory networks process environmental information to make metabolic decisions [53]. In the presence of lactose and absence of glucose, the lac operon switches to a high expression state enabling lactose utilisation. This switching involves both positive regulation (through CAP-cAMP) and negative regulation (through the lac repressor), creating a logical AND gate that integrates information about carbon source availability.

Morphogen gradient interpretation during embryogenesis shows how gene regulatory networks translate spatial information into sharp spatial patterns of gene expression [54]. Morphogen proteins like Bicoid in *Drosophila* form concentration gradients across developing tissues, where different threshold concentrations activate different target genes. This establishes spatial domains of gene expression that define body segments and organ boundaries with remarkable precision despite the stochastic nature of molecular processes.

1.2 Neuromodulatory Systems in the Brain

Neuromodulatory systems represent a distinct class of neural circuits that differ fundamentally from classical point-to-point synaptic transmission. Rather than conveying specific information about sensory stimuli or motor commands, neuromodulatory systems broadcast signalling molecules that modify the behaviour of large neural populations, thereby influencing learning, attention, arousal, and other global brain states [29].

1.2.1 The Four Major Neuromodulatory Systems

The brain employs four major neuromodulatory systems that project diffusely throughout cortical and subcortical structures (Figure 1.2). These systems originate from small nuclei but influence vast regions simultaneously, enabling coordinated regulation of learning, attention, and behavioral states [55, 56].

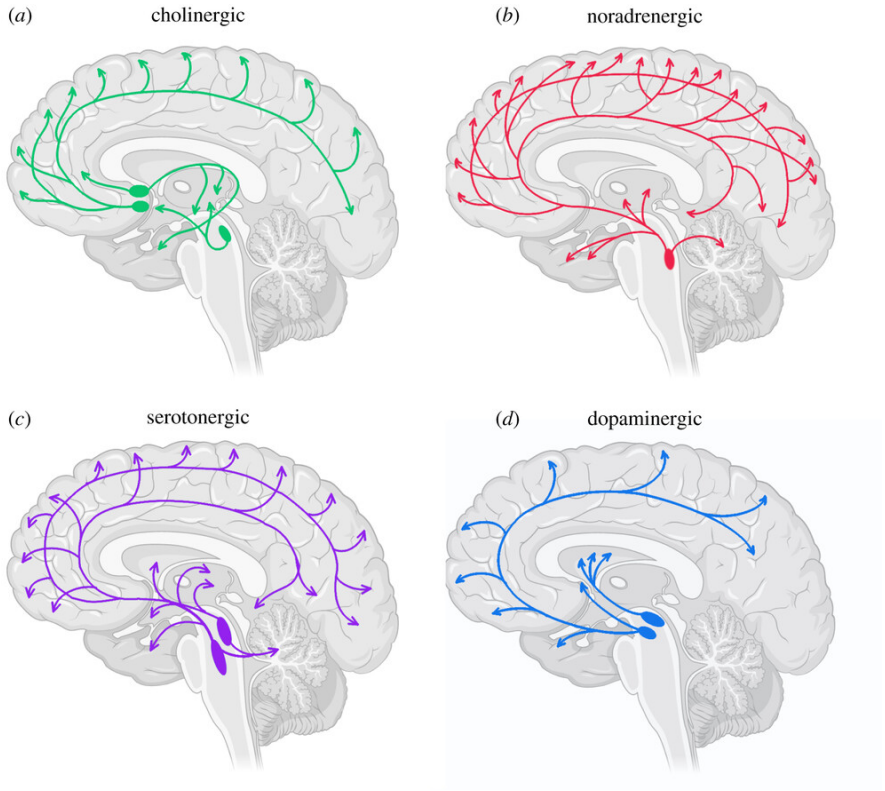


Figure 1.2: **Ascending Neuromodulatory Pathways in the Brain.** The four major neuromodulatory systems demonstrate diffuse projection patterns enabling system-wide coordination: (a) cholinergic system (attention/arousal); (b) noradrenergic system (vigilance/novelty); (c) serotonergic system (mood/inhibition); (d) dopaminergic system (reward/motivation). These widespread projections enable simultaneous modulation of multiple brain regions. Adapted from O'Callaghan et al. [56]

Dopaminergic System: The dopaminergic system, with projections primarily from the ventral tegmental area (VTA) and substantia nigra (SN), plays a crucial role in reward processing, motivation, and learning [57]. Dopamine neurons exhibit phasic responses to reward prediction errors, the difference between expected and received rewards [58]. When reward exceeds expectations, dopamine neuron firing increases, signalling a positive prediction error. When reward falls short of expectations, dopamine neuron activity decreases below baseline, signalling a

negative prediction error. This reward prediction error signal provides a teaching signal that updates value estimates and guides future behaviour.

Serotonergic System: The serotonergic system, originating primarily from the dorsal and median raphe nuclei, projects widely throughout the brain and influences mood, anxiety, and behavioural inhibition [32]. Serotonin operates on slower timescales than dopamine, providing tonic control that affects behavioural policies over extended periods. This system regulates the balance between immediate and delayed rewards, influences risk sensitivity, and modulates behavioural flexibility.

Noradrenergic System: The noradrenergic system, emanating from the locus coeruleus, regulates arousal, attentional engagement with salient or novel stimuli, and behavioural responses to unexpected events [59]. Noradrenergic neurons fire phasically to novelty regardless of valence, signalling general surprise or unexpectedness. The noradrenergic system primarily modulates cortical gain, adjusting how vigorously neurons respond to their inputs.

Cholinergic System: The cholinergic system, with major projections from the basal forebrain, enhances cortical plasticity and facilitates memory formation [60]. Acetylcholine regulates the balance between encoding new information and retrieving previously stored information, operating on intermediate timescales between tonic serotonin modulation and phasic dopamine responses.

1.2.2 Neuromodulatory Mechanisms

Neuromodulatory systems exert control over neural computation at multiple levels of organisation through various mechanisms. At the synaptic level, neuromodulators alter neurotransmitter release probability, postsynaptic receptor sensitivity, and the balance between excitation and inhibition [29]. These actions enable the same neural circuit to produce different output patterns depending on neuromodulatory state.

At the network level, neuromodulators adjust the gain of neural responses, effectively controlling how vigorously neurons respond to their inputs. This gain modulation enables adaptive control of signal-to-noise ratios and determines how much influence different information sources have on decision-making [61, 62].

At the systems level, neuromodulators coordinate activity patterns between brain regions, enabling different behavioural modes and states of consciousness [28]. The interaction between different neuromodulatory systems creates a high-dimensional space of brain states, with each combination of neuromodulator levels associated with specific computational properties and behavioural tendencies.

1.2.3 Functional Coordination

The four major neuromodulatory systems do not operate independently but interact to provide coordinated influences on learning and behaviour. Dopamine and serotonin exhibit opponent interactions, with dopamine promoting approach behaviour and serotonin promoting avoidance [32]. The balance between these systems affects risk-taking, persistence in the face of obstacles, and temporal discounting of future rewards.

Norepinephrine interacts with acetylcholine to regulate the encoding-retrieval balance in memory systems [60]. High norepinephrine and acetylcholine favour encoding new information, whilst low levels favour retrieval of previously stored information. This coordinated adjustment enables dynamic switching between learning and exploitation.

The temporal dynamics of neuromodulatory systems span multiple timescales, from phasic dopamine responses occurring within hundreds of milliseconds to tonic serotonin effects persisting over hours or days [32]. This multi-timescale organisation enables biological systems to simultaneously adapt to immediate circumstances whilst maintaining long-term behavioural policies.

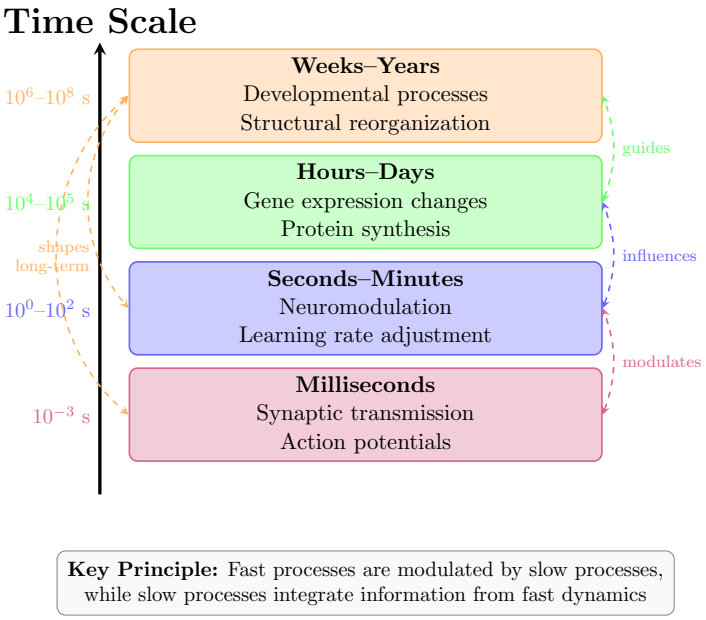


Figure 1.3: **Simplified Multi-timescale coordination in biological adaptive system.**

Figure 1.3 presents a simplified scheme of temporal hierarchy that enables a sophisticated form of meta-adaptation: fast processes provide immediate responses while slower processes adjust the parameters governing those responses based on accumulated experience. Critically, the interactions are bidirectional—rapid synaptic

activity drives slower changes through activity-dependent plasticity, while developmental processes establish constraints that guide learning. Current artificial learning systems rarely integrate more than two timescales (typically training time and inference time), representing a significant gap that developmental evolutionary robotics addresses. Our framework (Chapters 3 and 4) implements this principle by combining evolutionary optimization (slow timescale), developmental processes (intermediate), and real-time learning (fast timescale).

1.3 Computational Advantages of Biological Control Systems

The control principles observed in gene regulatory networks and neuromodulatory systems offer several computational advantages that could be translated to artificial autonomous systems.

1.3.1 Efficiency Through Constraints

Biological systems achieve high efficiency despite, or perhaps because of, the constraints under which they operate. Bounded gene expression levels, limited network connectivity, and restricted parameter ranges focus evolutionary search towards robust, generalisable solutions rather than exploration of the entire parameter space [31]. These constraints act as inductive biases that accelerate learning and improve generalisation.

Sparse biological networks reduce computational complexity whilst maintaining rich dynamics [40]. Biological gene regulatory networks exhibit extreme sparsity compared to artificial neural systems. Empirical analyzes and inference studies indicate that genes are typically regulated by only a few transcription factors (average $\sim$2–3 per gene) in biologically realistic networks [63], whereas artificial neural networks often assume dense or fully connected architectures with orders of magnitude more connections per node. This sparsity arises from metabolic constraints, maintaining regulatory connections costs energy, but provides computational advantages including improved evolvability, reduced parameter and optimisation complexity.

1.3.2 Robustness Mechanisms

Biological systems operate reliably despite pervasive molecular noise, environmental fluctuations, and component failures. This robustness emerges from network

architecture rather than precise parameter tuning [51]. Redundancy, feedback control, and distributed decision-making ensure that system-level behaviour remains stable even when individual components behave stochastically.

The nonlinear dynamics of gene regulatory networks suppress noise and sharpen signals, enabling reliable detection of meaningful environmental cues whilst ignoring stochastic fluctuations [46]. Cooperative binding produces sharp threshold functions that convert graded signals into binary decisions with reduced sensitivity to parameter variations within physiological ranges.

1.3.3 Multi-timescale Adaptation

The temporal hierarchy described in Section 1.2.3 suggests a key computational principle for artificial autonomous systems: simultaneous adaptation across multiple timescales, integrating rapid synaptic responses (milliseconds), intermediate neuromodulatory adjustments (seconds to minutes), and slow developmental changes (hours to years) [32, 29].

This separation of timescales enables systems to remain responsive to immediate conditions without destabilizing learned long-term policies. Fast processes provide immediate behavioral responses whilst slower processes adjust the parameters governing those responses based on accumulated experience. Critically, the interactions are bidirectional: rapid activity drives slower changes through activity-dependent plasticity, whilst developmental processes establish constraints that guide learning [28].

Implementing multi-timescale adaptation in artificial systems remains challenging. Single learning rates or monolithic update mechanisms struggle to achieve the balance between rapid responsiveness and long-term stability that biological systems maintain naturally through their layered temporal architecture.

1.4 Computational Foundations for Bio-Inspired Robotics

The biological principles described in preceding sections provide inspiration for adaptive robotic systems, but their translation to artificial systems requires understanding the computational frameworks within which they will be implemented. This section provides essential background on three foundational areas, reinforcement learning, evolutionary computation, and robotic control, that form the technical substrate for the bio-inspired approaches developed in subsequent chapters.

1.4.1 Reinforcement Learning Fundamentals

Reinforcement learning provides a mathematical framework for learning optimal behavior through interaction with an environment [64]. An agent observes states $s \in \mathcal{S}$, executes actions $a \in \mathcal{A}$, and receives rewards $r \in \mathbb{R}$, with the objective of maximizing cumulative reward. The Markov Decision Process (MDP) formalism captures this interaction through the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{P}$ represents state transition probabilities, $\mathcal{R}$ defines the reward function, and $\gamma \in [0, 1]$ is the discount factor balancing immediate and future rewards.

The fundamental challenge involves learning a policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes the expected return:

$$J(\pi) = \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right]. \tag{1.2}$$

Value-based methods estimate action-value functions $Q(s, a)$ to approximate expected returns, while policy-based and actor–critic methods directly optimize policy parameters through gradient-based updates. Modern deep reinforcement learning algorithms unify these paradigms using deep neural networks, enabling learning in high-dimensional and continuous control settings relevant to robotics [65, 66, 10].

The challenges specific to robotic applications include continuous action spaces, high-dimensional sensory inputs (particularly in vision-based manipulation [14]), and sample inefficiency that makes extensive real-world training prohibitively expensive [2]. Biological systems overcome these challenges and achieve robust learning under strict resource constraints, motivating the bio-inspired approaches developed in Chapters 3 and 4, which leverage biological principles to achieve more efficient learning whilst maintaining robust performance.

1.4.2 Evolutionary Computation Essentials

Evolutionary algorithms provide population-based optimization methods inspired by biological evolution [67, 68]. A population of candidate solutions (individuals) undergoes iterative cycles of evaluation, selection, and variation. Fitness evaluation assigns quality scores to individuals based on their performance on the target task. Selection mechanisms, such as tournament or ranking selection, preferentially choose high-fitness individuals for reproduction. Variation operators, including mutation and crossover, generate offspring with modified characteristics, introducing the diversity necessary for adaptation.

The general evolutionary cycle operates as follows:

1. Initialize population P_0 of N individuals
2. For generation $t = 1, 2, \ldots$:
   - Evaluate fitness $f(x_i)$ for each individual $x_i \in P_t$
   - Select parents based on fitness values
   - Apply variation operators to generate offspring
   - Form new population P_{t+1} through selection from parents and offspring
3. Terminate when convergence criteria are satisfied or maximum generations reached

Unlike gradient-based optimization, evolutionary algorithms do not require differentiable objectives and can explore multimodal search spaces effectively [27]. This property proves particularly valuable for optimizing neural network architectures and parameters where the fitness landscape exhibits multiple local optima and discontinuities.

However, evolutionary approaches typically require many function evaluations, often millions, which becomes problematic in robotics where fitness evaluation requires physical experiments or computationally expensive simulations. The gene regulatory network approach developed in Chapter 4 addresses this limitation through biological constraints that reduce the effective search space.

1.4.3 Robotic Control Context

Robotic manipulation presents unique challenges that distinguish it from purely computational learning problems [1, 3]. The sensorimotor loop connects perception (visual, tactile, proprioceptive sensors) to action (motor commands controlling actuators) through a controller that must operate in real-time, handle sensor noise and calibration drift, and respond to dynamic environmental changes including unexpected perturbations and novel objects.

Vision-based manipulation [14] adds substantial complexity through high dimensional image inputs ($\sim 10^4$–10^5 dimensions) that must be processed to extract task-relevant features whilst discarding irrelevant visual information. The challenge intensifies in unstructured environments where object appearance, position, and orientation vary unpredictably, requiring controllers that generalize beyond training data.

Key challenges that constrain robotic learning include:

- **Partial observability**: Sensors provide limited information about true environmental state, with occlusions, limited field-of-view, and sensor noise preventing complete observation of task-relevant variables
- **Continuous control**: Many robotic tasks require continuous action spaces (joint velocities, torques) rather than discrete choices, complicating exploration and credit assignment.
- **Sample efficiency**: Physical experiments and high-fidelity simulations limit the number of learning trials, making sample-inefficient methods like deep reinforcement learning impractical for many applications.
- **Sim-to-real transfer**: Controllers trained in simulation often fail when deployed on physical robots due to modeling approximations, unmodeled dynamics, and physical parameter uncertainties [20].
- **Safety constraints**: Physical robots must avoid collisions and unsafe configurations during exploration, restricting the learning process in ways that purely simulated agents do not face.

These constraints motivate the search for alternative control paradigms that can learn efficiently from limited data whilst maintaining robust performance under real-world uncertainty. The biological principles, evolved over billions of years to enable efficient, robust control in noisy, resource-constrained environments, provide a principled foundation for robotic learning that transcends the limitations of conventional optimization approaches. Subsequent chapters address these challenges through multiple developped mechanisms.

1.4.4 Multi-Task and Sequential Learning Fundamentals

The challenge of learning multiple tasks sequentially without forgetting previously acquired knowledge represents one of the most fundamental obstacles to creating truly autonomous systems. While humans and other biological organisms routinely acquire new skills throughout their lifetimes whilst maintaining existing capabilities, artificial neural networks typically suffer from catastrophic forgetting when trained on sequential tasks. This section establishes the fundamental nature of the forgetting problem and examines how biological systems provide insight into potential solutions, motivating the bio-inspired approaches developed in subsequent chapters.

The Catastrophic Forgetting Problem

Catastrophic forgetting, also termed catastrophic interference, occurs when a neural network trained on one task suffers severe performance degradation upon training on a subsequent task. This phenomenon, first systematically documented in connectionist systems [69, 70], reveals a fundamental limitation of gradient-based learning in distributed representations.

Consider a sequence of tasks $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n\}$, where each task $\mathcal{T}_i$ is characterized by a dataset $\mathcal{D}_i = \{(x_j^{(i)}, y_j^{(i)})\}_{j=1}^{N_i}$ drawn from a task-specific distribution $p_i(x, y)$. A neural network with parameters θ learns task $\mathcal{T}_i$ by minimizing the empirical risk:

$$\mathcal{L}_i(\theta) = \frac{1}{N_i} \sum_{j=1}^{N_i} \ell(f(x_j^{(i)}; \theta), y_j^{(i)}) \tag{1.3}$$

where $f(\cdot; \theta)$ represents the network function and $\ell(\cdot, \cdot)$ is the task-specific loss function.

When learning tasks sequentially, the network parameters evolve through a series of optimization phases. After training on task $\mathcal{T}_1$, the parameters reach $\theta_1^* = \arg\min_\theta \mathcal{L}_1(\theta)$. Subsequent training on task $\mathcal{T}_2$ then updates these parameters toward $\theta_2^* = \arg\min_\theta \mathcal{L}_2(\theta)$. However, this optimization typically causes $\mathcal{L}_1(\theta_2^*) \gg \mathcal{L}_1(\theta_1^*)$, indicating severe degradation of performance on the first task.

The severity of catastrophic forgetting can be quantified through the *forgetting measure* [71]:

$$\mathcal{F}_i^{(t)} = \max_{k \in \{1,\ldots,t-1\}} (\text{Acc}_{i,k} - \text{Acc}_{i,t}) \tag{1.4}$$

where $\text{Acc}_{i,k}$ represents the accuracy on task $\mathcal{T}_i$ after training on task $\mathcal{T}_k$. High values of $\mathcal{F}_i^{(t)}$ indicate substantial forgetting, with empirical studies showing that conventional neural networks can experience forgetting of 40%–90% when learning just two sequential tasks [18]. Figure 1.4 illustrates this phenomenon, showing how performance on Task 1 degrades rapidly when the network begins training on Task 2.

The fundamental cause of catastrophic forgetting lies in the distributed nature of neural network representations. Unlike localist representations where each unit encodes independent information, distributed representations require coordinated patterns of activation across many units to encode specific information [70]. When gradient descent modifies weights to accommodate new task information, it disrupts the precise patterns required for previously learned tasks. This interference becomes particularly severe when task domains overlap substantially, requiring shared representations to encode different information; when network capacity is limited, forcing competition between tasks for representational resources; and

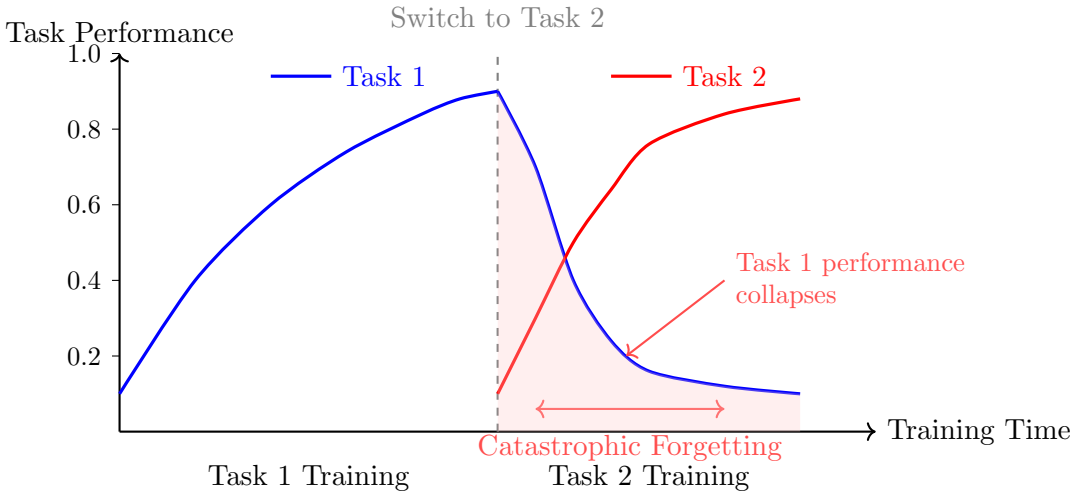


Figure 1.4: **Catastrophic forgetting in sequential task learning.** When a neural network trained on Task 1 begins training on Task 2, performance on Task 1 degrades rapidly (shaded region), often dropping to near-random levels. This catastrophic interference occurs because new learning overwrites the network parameters responsible for previously acquired skills.

when learning rates remain high, enabling rapid parameter modification that overwrites previous knowledge.

The severity and ubiquity of catastrophic forgetting in artificial neural networks stands in stark contrast to the lifelong learning capabilities observed in biological systems, suggesting that fundamental principles underlying biological learning differ from those implemented in standard gradient-based optimization.

Biological Perspectives and Motivation for Bio-Inspired Approaches

The catastrophic forgetting problem reveals a fundamental gap between artificial and biological learning systems. Humans and other animals routinely acquire new skills throughout their lifetimes whilst maintaining previously learned capabilities, often exhibiting positive transfer where new learning enhances rather than disrupts existing knowledge [70, 72]. This remarkable capability suggests that biological neural systems employ mechanisms that artificial networks lack.

Neuroscientific research has identified multiple complementary mechanisms operating at different timescales and organizational levels that enable biological systems to avoid catastrophic forgetting [73]. *Synaptic consolidation* stabilizes important connections through molecular modifications that resist subsequent plasticity.

Critical synapses undergo structural changes, including spine enlargement and increased receptor density, that make them resistant to disruption by new learning whilst allowing less important synapses to remain plastic [73]. This selective stabilization enables the brain to protect important knowledge whilst maintaining the capacity to learn new information.

Systems-level consolidation gradually transfers information from rapid-learning structures to slower-learning but higher-capacity structures. The hippocampus initially encodes new experiences rapidly, then gradually transfers this information to neocortical structures through repeated reactivation during sleep and rest periods [74]. This hierarchical organization enables fast learning of new information without immediately disrupting stable long-term memories stored in cortical networks.

Neuromodulatory regulation dynamically adjusts plasticity based on behavioral context and environmental demands. The neuromodulatory systems examined in Section 1.2 provide global signals that increase learning rates when encountering novel situations whilst reducing plasticity in well-learned domains. Dopamine signals reward prediction errors that indicate when learning is necessary, whilst acetylcholine modulates the encoding-retrieval balance, promoting memory formation during novel experiences and memory consolidation during familiar situations [60, 28].

Structural organization creates specialized circuits for different functions whilst maintaining connections that enable knowledge transfer. The modular organization of cortical and subcortical structures provides natural separation between different domains of knowledge, reducing interference whilst allowing coordinated function through long-range connections [75]. This architectural principle suggests that physical separation of task-specific processing, combined with controlled knowledge transfer mechanisms, may be essential for continual learning.

These biological observations motivate the central hypothesis of this thesis: that computational implementations of biological regulatory mechanisms can provide more effective solutions to the continual learning challenge than purely algorithmic approaches that ignore biological organizational principles. The biological systems that successfully avoid catastrophic forgetting employ not single mechanisms but coordinated systems operating across multiple scales, from molecular processes affecting individual synapses to system-level dynamics coordinating entire brain regions.

This multi-scale coordination suggests that effective artificial continual learning systems may require similar integration of mechanisms operating at different organizational levels. The neuromodulatory framework presented in Chapter 3 implements dynamic regulation of learning parameters based on task context and

performance feedback, translating the principles of biological neuromodulation into practical computational mechanisms.

1.5 From Biology to Robotics

1.5.1 Abstraction Principles

Translating biological regulatory principles into computational implementations requires careful abstraction to capture essential features whilst omitting unnecessary biological details [27]. The goal is to extract the computational principles that make biological systems effective, not to replicate biological systems in full detail.

Three key abstraction principles guide this translation. First, preserve the structural properties that enable biological functionality, sparsity, modularity, hierarchy, rather than replicating specific molecular mechanisms. Second, implement the essential nonlinearities, thresholds, saturation, cooperative interactions, that enable biological systems to filter noise and make decisions. Third, respect the temporal dynamics and multi-timescale structure that enable biological systems to adapt across different time horizons.

1.5.2 Previous Computational Applications

Gene regulatory networks have been applied in evolutionary robotics with promising results [76, 77]. These implementations demonstrated that evolving both the topology and parameters of networks can produce effective robot controllers with the interpretability and modularity observed in natural systems.

Neuromodulatory principles have inspired learning rate adaptation methods [28], attention mechanisms [78], and meta-learning strategies [79]. However, most existing implementations treat individual neuromodulatory functions in isolation rather than implementing the coordinated interactions observed in biological systems.

1.5.3 Novel Contributions of This Work

This thesis makes three significant advances over previous bio-inspired approaches. First, it implements coordinated multi-system neuromodulation rather than isolated neuromodulatory functions, capturing the interaction effects that characterise biological coordination. Second, it demonstrates that biological constraints

can provide optimisation advantages rather than limiting system capabilities. Third, it provides systematic validation through ablation studies showing that performance improvements result from specific biological principles rather than general implementation characteristics.

The systematic mapping between biological mechanisms and computational implementations establishes a principled framework for translating biological insights to artificial systems. Tables 1.1 and 1.2 summarise these mappings for neuromodulatory systems and gene regulatory networks, respectively, making explicit the connections between biological properties, computational advantages, and specific implementations.

Table 1.1: Biological-to-Computational Mapping: Neuromodulatory Systems

Mechanism	**Key Properties**	**Computational Advantage**	**Implementation**
Dopamine reward signaling	Phasic responses to prediction errors; modulates plasticity	Adaptive learning rates based on performance; efficient credit assignment	s_{DA} modulates $\eta_{\text{effective}}$ (Chapter 3)
Serotonin stability	Tonic regulation; long timescales (hours-days)	Prevents oscillations; maintains learned knowledge; temporal discounting	$\tau_{\text{5-HT}} = 0.5$ (slowest adaptation)
Norepinephrine attention	Phasic responses to novelty; arousal modulation	Dynamic attention allocation; exploration-exploitation balance	s_{NE} gates lateral connections $\alpha_{i,k}^{(l)}$
Acetylcholine consolidation	Enhances plasticity; memory formation	Knowledge integration; selective transfer; prevents forgetting	Coordinates transfer between PNN columns
Multi-system coordination	Neurotransmitters interact (e.g., DA-5HT opponent processes)	Coordinated multi-factor adaptation exceeds individual mechanisms	4-module interaction (98.0% retention vs. 93.2% baseline)
Multiple timescales	NE (seconds) < DA (minutes) < ACh (hours) < 5-HT (days)	Hierarchical temporal processing; rapid + slow adaptation	$\tau_{\text{NE}} = 0.05$, $\tau_{\text{5-HT}} = 0.5$

Table 1.2: Biological-to-Computational Mapping: Gene Regulatory Networks

Mechanism	**Key Properties**	**Computational Advantage**	**Implementation**
Bounded gene expression	Expression levels constrained to [0,1] by molecular concentrations	Prevents parameter explosion; natural regularization; inherent stability	Gene states $x_i \in [0.0, 1.0]$ with clipping
Sparse connectivity	Only essential regulatory connections maintained due to metabolic costs	Reduced computational complexity; improved evolvability; enhanced interpretability	$\sim 20\%$ sparse regulatory matrix $W^{(r)}$
Hill function cooperativity	Cooperative protein binding creates sigmoidal responses	Natural threshold-based switching; noise filtering; sharp decision boundaries	$H(u, \theta) = \frac{u^2}{\theta^2+u^2} \cdot \text{sign}(u)$
Molecular decay processes	Proteins naturally degrade over time	Temporal memory without explicit recurrence; multi-timescale dynamics	Decay rates $\delta_i \in [0.05, 1.0]$
Biological noise	Stochastic molecular processes ($\eta \sim 10 - 20\%$)	Natural exploration; prevents local minima; robust solutions	$\eta_i(t) \sim \mathcal{N}(0, 0.01^2)$
Modular network motifs	Regulatory circuits (feed-forward loops, oscillators)	Reusable control patterns; hierarchical organization	Emergent from evolution (Chapter 4)

1.6 Conclusion

The biological and computational foundations presented in this chapter form the scientific and technical basis for the systems developed in subsequent chapters. Gene regulatory networks and neuromodulatory systems exemplify control principles that have been refined through evolution to achieve high efficiency, robustness, and adaptability.

The biological understanding presented here underpins the mathematical models, experimental designs, and theoretical analyses that demonstrate the practical advantages of developmental evolutionary robotics approaches. By grounding computational innovations in well-established biological principles, we ensure that bio-inspired systems leverage genuine biological advantages rather than superficial analogies.

The following chapters demonstrate how these principles translate into measurable performance improvements in robot learning and control.

CHAPTER 2

THEORETICAL FOUNDATIONS AND LITERATURE REVIEW

2.1 Introduction

The development of adaptive robotic systems capable of continual learning and autonomous behavior remains one of the central challenges in artificial intelligence and robotics. This chapter establishes the theoretical foundations and reviews the research traditions that inform the approaches presented in this thesis. The review synthesizes insights from evolutionary computation, continual learning, bio-inspired neural architectures, and developmental biology into a unified framework that guides our methodological innovations.

The review progresses from historical foundations toward contemporary approaches and open challenges. We examine evolutionary robotics, tracing how biology-inspired optimization methods have evolved from simple controller evolution to sophisticated approaches incorporating morphological adaptation and novelty search. We explore continual and multi-task learning, focusing on catastrophic forgetting and strategies that enable sequential skill acquisition. The examination continues with adaptive learning rate methods, bio-inspired neural network design, and computational neuromodulation. Finally, we synthesize these research streams to identify critical gaps that motivate the contributions of this thesis.

2.2 Evolutionary Robotics: Historical Perspective

Evolutionary robotics emerged from the convergence of evolutionary computation and autonomous robotics in the early 1990s, representing a fundamental shift from traditional engineering approaches toward bio-inspired automatic design methods [8, 80]. Rather than hand-designing robot controllers through laborious engineering processes, evolutionary algorithms could automatically discover effective control strategies through processes analogous to natural selection.

The intellectual roots trace back to evolutionary computation, which had demonstrated remarkable success in optimization across diverse domains [67]. However, applying evolutionary principles to robotics presented unique challenges that required significant methodological innovations. Unlike abstract optimization problems, robotic applications demanded controllers that operated in real-time, handled noisy sensory inputs, and produced robust behaviors in dynamic environments [20].

Early work by Harvey et al. established foundational principles that continue to guide evolutionary robotics research [80]. Their experiments demonstrated that evolutionary algorithms could discover elegant solutions to navigation and obstacle avoidance tasks using minimal neural architectures. The key insight was that evolution could exploit morphological and environmental properties that human designers might overlook.

Sims' pioneering work on evolving virtual creatures showcased the potential for co-evolving morphology and control [81]. By allowing both body structures and neural controllers to evolve, Sims demonstrated the emergence of locomotion strategies remarkably similar to those observed in biological organisms. This work highlighted that separating morphology from control, as traditional robotics approaches do, may fundamentally limit evolutionary design processes.

The development of NeuroEvolution of Augmenting Topologies (NEAT) by Stanley and Miikkulainen represented a major methodological advancement [82]. NEAT addressed the competing conventions problem in evolving neural network topologies through historical markings, protected structural innovation through speciation, and grew networks incrementally from minimal structures. These innovations enabled more efficient evolution of network topologies compared to fixed-topology methods.

Novelty search, introduced by Lehman and Stanley, challenged the assumption that optimization toward explicit objectives necessarily produces optimal results [83]. By rewarding behavioral novelty rather than objective performance, novelty search

demonstrated that removing performance pressure can paradoxically lead to discovering solutions that objective-based search fails to find. This approach proved particularly effective in deceptive fitness landscapes where local optima trap traditional evolutionary algorithms.

Despite these advances, evolutionary robotics faces persistent challenges that limit broader adoption. The computational expense of evolutionary processes, particularly when applied to complex neural architectures, remains a significant barrier [27]. The reality gap, the difficulty of transferring evolved behaviors from simulation to physical robots, continues to challenge practitioners [20]. Contemporary research addresses fundamental questions regarding fitness function design, scalability, and integration with other machine learning techniques.

Recent developments in evolutionary robotics have expanded the field's scope and capabilities through several parallel tracks. Quality-diversity algorithms, particularly MAP-Elites and its variants [84, 85], have emerged as powerful alternatives to traditional optimization approaches by simultaneously discovering both high-performing and diverse solution sets. These methods have proven especially effective for modular robotics where they generate diverse morphological stepping stones that enable escape from local optima while building repertoires of adaptive behaviors [86]. The integration of deep learning with evolutionary methods has led to neural architecture search techniques that automatically discover optimal network topologies for robotic control tasks [87]. Gradient-assisted quality-diversity algorithms such as Policy Gradient MAP-Elites and CMA-MAE [88, 89] combine the exploration benefits of diversity search with the exploitation power of gradient-based optimization. The field has also witnessed advances in real-world robot evolution, moving beyond simulation-only approaches toward systems capable of autonomous reproduction and physical evolution [90], though ethical concerns around robot multiplication and autonomous adaptation require careful consideration. Simultaneously, evolutionary swarm robotics [91] has demonstrated the emergence of collective behaviors including task specialization and division of labor through selection on group performance alone. These contemporary developments suggest evolutionary robotics is transitioning from a niche research area toward practical applications while maintaining theoretical connections to biological evolution and open-ended evolutionary dynamics.

Table 2.1: Historical Development of Evolutionary Robotics Methods

Era	Key Work	Innovation	Limitations	Impact
Early 1990s	Harvey et al. [80]; Nolfi & Floreano [8]	Evolved neural controllers for simple behaviors	Small-scale; simulation only; limited transfer	Foundation
Late 1990s	Sims [81]; Lipson & Pollack [9]	Co-evolution of morphology and control	Reality gap; computational cost	High
2000s	NEAT [82]; HyperNEAT [92]	Topology evolution; indirect encoding	Scalability challenges	Moderate
2010s	Novelty Search [83]; MAP-Elites [84]	Quality-diversity; exploration-based	Fitness function design	High
2017-2020	ES for RL [93]; CMA-ES [94]	Deep RL integration; parallel evaluation	Sample efficiency	High
2020-2022	Surrogate-Assisted MAP-Elites [95]; AutoMoDe [96]	Data efficiency; automatic modular design	Domain specificity	Moderate
2022-2024	PGA-MAP-Elites [97]; CMA-MAE [98]; Evolutionary NAS [87]	Gradient-assisted QD; neural architecture search integration	Computational cost; theoretical understanding	High
2024-Present	Real-world evolution systems [99]; Multi-task MAP-Elites [100]; LLM-guided evolution [101]	Physical robot evolution; task generalization; language-model integration	Ethical concerns; validation complexity	Emerging
This Work	**GRN + Evolution**	**Biological constraints as optimization guides**	**Validation in complex domains**	**Novel**

2.3 Sequential Learning Architectures and Methods

Having established the fundamental challenge of catastrophic forgetting in Section 1.4.4, we now examine how researchers have attempted to address this problem through various architectural and algorithmic innovations. Table 2.2 presents a comprehensive taxonomy of continual learning strategies, organized into five

major categories: approaches based on regularization, replay, optimization, representation, and architecture. Each strategy represents distinct trade-offs between memory efficiency, forgetting prevention, and knowledge transfer capabilities.

This section focuses on architecture-based approaches, particularly Progressive Neural Networks (PNN) [22], which forms the foundation for the bio-inspired extensions developed in Chapter 3. We provide detailed technical analysis of PNN's mechanisms and examine representative methods from other categories, Elastic Weight Consolidation (EWC) as a regularization approach and PackNet as an efficient parameter allocation method, to establish the technical context and motivate our contributions.

2.3.1 Progressive Neural Networks: Architectural Foundation

Progressive Neural Networks [22] address catastrophic forgetting through architectural innovation, allocating dedicated capacity for each new task whilst maintaining lateral connections that enable knowledge transfer from previously learned tasks. This approach achieves zero forgetting by construction, representing a fundamental departure from regularization-based methods that attempt to protect important parameters whilst allowing modification.

Architectural Design and Mathematical Formulation

The PNN architecture consists of a sequence of neural network columns $\{C_1, C_2, \ldots, C_n\}$, where each column C_i is dedicated to task $\mathcal{T}_i$. When learning task $\mathcal{T}_i$, a new column is instantiated with randomly initialized parameters, whilst all parameters from previous columns $\{C_1, \ldots, C_{i-1}\}$ are frozen to preserve learned knowledge. This architectural isolation guarantees that no catastrophic forgetting can occur, as previous task parameters remain unchanged during new task training.

The forward pass through a PNN computes layer activations through a combination of within-column processing and lateral connections from previous columns:

$$h_i^{(l)} = f\left(W_i^{(l)} h_i^{(l-1)} + \sum_{k=1}^{i-1} U_{i,k}^{(l)} h_k^{(l)}\right) \tag{2.1}$$

where $h_i^{(l)} \in \mathbb{R}^{d_l}$ represents the activations at layer l of column i, $W_i^{(l)} \in \mathbb{R}^{d_l \times d_{l-1}}$ are the standard within-column weights that process information vertically through

the network, $U_{i,k}^{(l)} \in \mathbb{R}^{d_l \times d_l}$ are the lateral connection weights that transfer knowledge horizontally from column k to column i, and $f(\cdot)$ is the element-wise activation function (typically ReLU).

The lateral connections $U_{i,k}^{(l)}$ serve two critical functions. First, they enable positive transfer by allowing new tasks to exploit features learned from related previous tasks. The network can leverage low-level visual features, motor primitives, or abstract representations developed during earlier training phases. Second, they provide a mechanism for the network to discover which previous knowledge is relevant to the current task, with the weights $U_{i,k}^{(l)}$ learned through standard gradient descent during training on task $\mathcal{T}_i$. This learned selectivity enables the architecture to automatically identify and exploit beneficial knowledge transfer whilst ignoring irrelevant previous learning.

The complete parameter set for a PNN with n columns, each having L layers with d hidden units per layer, grows as:

$$|\Theta_{\text{PNN}}| = nLd^2 + \frac{n(n-1)}{2}Ld^2 = \frac{n(n+1)}{2}Ld^2 = O(n^2d^2) \tag{2.2}$$

This quadratic growth in the number of tasks represents the primary limitation of the basic PNN architecture, limiting practical deployment to scenarios with fewer than 10–15 tasks before memory requirements become prohibitive.

Training Procedure and Task-Specific Optimization

Training a PNN proceeds sequentially through tasks. For the first task $\mathcal{T}_1$, a single column C_1 is trained using standard supervised or reinforcement learning procedures, optimizing the within-column weights $W_1^{(l)}$ to minimize the task-specific loss $\mathcal{L}_1(\theta_1)$. Upon completion of training, all parameters of column C_1 are frozen.

For subsequent tasks $\mathcal{T}_i$ where $i > 1$, a new column C_i is instantiated with randomly initialized within-column weights $W_i^{(l)}$ and lateral connection weights $\{U_{i,k}^{(l)}\}_{k=1}^{i-1}$. The optimization objective becomes:

$$\theta_i^* = \arg\min_{\theta_i} \mathcal{L}_i(\theta_i) \quad \text{subject to} \quad \{\theta_k\}_{k=1}^{i-1} = \{\theta_k^*\}_{k=1}^{i-1} \tag{2.3}$$

where $\theta_i = \{W_i^{(l)}, U_{i,k}^{(l)}\}$ represents the trainable parameters for column i. This constrained optimization ensures that previous task performance cannot degrade, achieving zero forgetting by construction.

The lateral connections introduce additional expressiveness that can accelerate learning when beneficial knowledge transfer is possible. Empirical studies demonstrate that PNN consistently matches or exceeds the performance of networks

trained independently on each task, with performance improvements of 10%–50% observed when tasks share relevant structure [22].

Advantages and Limitations

Progressive Neural Networks provide several compelling advantages. They achieve *zero forgetting* as previous task parameters remain unchanged during new task training, providing theoretical guarantees that are difficult to achieve through regularization approaches. They enable *automatic feature transfer* through learned lateral connections, avoiding the need to manually specify which knowledge should be shared between tasks. The lateral connection weights provide some interpretability, revealing which previous tasks contribute most to current task performance. They maintain *task-specific output heads*, allowing specialization for different action spaces or objective functions across tasks, a critical advantage for control applications where tasks may have incompatible action mappings.

However, these advantages come at computational cost. The quadratic growth in parameters with task number ($O(n^2)$) limits scalability. The frozen columns represent wasted capacity if early tasks were learned with excessive parameters, as overprovisioning in initial columns permanently allocates resources that cannot be reclaimed. The architecture provides no mechanism for *backward transfer*, where knowledge from new tasks might improve performance on previous tasks. The approach requires careful tuning of column capacity, as undersized columns limit individual task performance whilst oversized columns waste memory.

2.3.2 Representative Methods from Other Strategies

Regularization-Based: Elastic Weight Consolidation

Elastic Weight Consolidation (EWC) [18] represents the regularization-based strategy, using Fisher Information to identify important parameters and constraining their modification during subsequent learning. When learning task $\mathcal{T}_B$ after task $\mathcal{T}_A$, EWC augments the standard loss function with a quadratic penalty:

$$\mathcal{L}_{\text{EWC}}(\theta) = \mathcal{L}_B(\theta) + \sum_i \frac{\lambda}{2} F_i \left(\theta_i - \theta^*_{A,i}\right)^2 \tag{2.4}$$

where F_i is the Fisher information quantifying parameter importance, $\theta^*_{A,i}$ are the optimal parameters from task A, and λ controls consolidation strength.

EWC achieves $O(np)$ memory complexity (storing Fisher information and optimal parameters for n tasks with p parameters), enabling application to longer

task sequences than PNN. The method provides no explicit transfer mechanism and suffers from high sensitivity to the hyperparameter λ, which must balance plasticity for new learning against stability of previous knowledge.

Parameter Allocation: PackNet

PackNet [102] addresses PNN's parameter efficiency through structured pruning, identifying and freezing task-specific subnetworks whilst freeing unused capacity for future learning. For each task, PackNet trains the network on available (unfrozen) parameters, applies magnitude-based pruning to identify essential weights, and freezes the pruned subnetwork.

PackNet achieves near-zero forgetting through parameter isolation whilst requiring less memory than PNN. However, maximum task capacity is determined by initial network size. The approach provides no mechanism for dynamic adaptation of pruning strategy based on task characteristics or learning progress.

2.3.3 Control-Specific Challenges and Architectural Necessity

Beyond the general limitations of consolidation and architectural approaches, sequential learning of control tasks presents distinctive challenges that are particularly relevant to robotic applications [21].

Control tasks differ fundamentally from vision classification in ways that exacerbate the stability-plasticity dilemma. State and action spaces vary dramatically across tasks. Control objectives can be diametrically opposed and reward structures span from dense immediate feedback to sparse delayed signals.

These characteristics create a *parameter conflict problem* for consolidation, based approaches. When tasks require opposite control strategies, damping versus amplifying, stabilization versus exploration, immediate response versus delayed planning, the network parameters optimized for one task directly conflict with those needed for another. Empirical evaluation across multiple control environments demonstrates that Low consolidation enables initial learning but permits severe catastrophic forgetting, whilst higher consolidation prevents both learning and forgetting by maintaining uniformly poor performance.

These control-specific challenges motivate architectural approaches that provide task-specific output layers whilst enabling selective knowledge transfer in hidden representations. The architectural isolation guarantees that action mappings for

one task cannot corrupt those of another, addressing the parameter conflict problem directly. For robotics applications involving sequential learning of diverse manipulation, navigation, or interaction skills where control tasks naturally exhibit the diversity that causes consolidation approaches to fail, the zero-forgetting guarantees and perfect retention demonstrated by architectural approaches with adaptive mechanisms become critical.

2.3.4 Comparative Analysis and Research Gaps

Table 2.2 reveals that different strategies occupy distinct regions of the trade-off space between memory efficiency, forgetting prevention, and knowledge transfer capabilities. Regularization-based methods (EWC, SI [103]) offer no memory overhead but limited capacity; replay-based methods (iCaRL [104], GEM [105]) provide strong baselines but raise privacy concerns; optimization-based approaches (GPM [106], MAML [107]) enable explicit interference control or fast adaptation at computational cost; representation-based methods (L2P [108], DualPrompt [109]) leverage pre-trained models but require large-scale pre-training; and architecture-based approaches (PNN, PackNet, HAT [110]) achieve zero or near-zero forgetting through parameter isolation but face scalability constraints.

Despite their different strategies, all reviewed approaches share fundamental limitations that motivate the bio-inspired solutions developed in subsequent chapters. None provide mechanisms for *dynamic adaptation* of learning parameters based on task context or performance feedback. PNN, EWC, and PackNet all employ fixed learning strategies: constant learning rates, static regularization strengths, or predetermined pruning schedules. This lack of adaptivity contrasts sharply with biological learning systems that dynamically adjust plasticity based on behavioral context and environmental demands.

All approaches treat forgetting prevention as a binary decision: either protect parameters completely (PNN, PackNet) or apply uniform regularization (EWC). This fails to capture the nuanced, context-dependent regulation observed in biological systems, where different brain regions and connections exhibit different degrees of plasticity depending on behavioral state, task demands, and learning phase. The neuromodulatory framework developed in Chapter 3 addresses this limitation by implementing coordinated multi-factor adaptation inspired by biological neuromodulation.

The reviewed approaches employ either architectural isolation (PNN, PackNet) or parameter regularization (EWC) but do not coordinate these mechanisms with other adaptive processes such as attention allocation, exploration-exploitation balance, or knowledge consolidation. Biological systems integrate multiple adaptive mechanisms operating at different organizational levels, from molecular processes

at individual synapses to system-level dynamics coordinating entire brain regions. This coordination enables sophisticated forms of adaptation that would be impossible with any single mechanism operating alone.

These fundamental limitations motivate the central hypothesis of this thesis: that computational implementations of biological regulatory mechanisms can provide more effective solutions to the continual learning challenge than approaches that ignore biological organizational principles.

Table 2.2: Taxonomy of Continual Learning Strategies

Strategy	**Representative Methods**	**Mechanism**	**Trade-offs**
Regularization-Based			
Parameter Protection	EWC [18], SI [103], RWalk [71], AFEC [111]	Constrain important parameters using Fisher information, path integral, or expansion-renormalization	**Pro:** No memory overhead **Con:** Limited capacity
Knowledge Distillation	LwF [112], PODNet [113], Co2L [114], DDE [115]	Distill knowledge from old model to new model via intermediate/final outputs	**Pro:** Preserves learned representations **Con:** Requires old model storage
Replay-Based			
Experience Replay	iCaRL [104], GEM [105], DER++ [116], X-DER [117]	Store/replay subset of old data with improved selection strategies	**Pro:** Effective, strong baseline **Con:** Memory/privacy concerns
Generative Replay	DGR [118], MeRGANs [119], LifelongGAN [120]	Generate pseudo-data from old tasks using GANs/VAEs	**Pro:** No storage required **Con:** Generator quality dependent
Feature Replay	GFR [121], REMIND [122], FeTrIL [123]	Replay generated features instead of raw data	**Pro:** More efficient than data replay **Con:** Representation shift issues
Optimization-Based			
Gradient Projection	OGD [124], GPM [106], AdamNSCL [125]	Project gradients orthogonal to important subspaces	**Pro:** Explicit interference control **Con:** Subspace estimation overhead

Continued on next page

Table 2.2 – continued from previous page

Strategy	Representative Methods	Mechanism	Trade-offs
Meta-Learning	MAML [107], OML [126], ANML [127], La-MAML [128]	Learn initialization or update rules for fast adaptation	**Pro:** Fast task adaptation **Con:** Meta-training complexity
Flat Minima Seeking	Stable-SGD [129], MC-SGD [130], SAM [131]	Find flatter loss landscape regions for better generalization	**Pro:** Better stability-plasticity **Con:** Additional computation
Representation-Based			
Self-Supervised Learning	LUMP [132], Co2L [114], DualNet [133]	Learn robust representations via contrastive or self-supervised objectives	**Pro:** Robust to forgetting **Con:** Requires unlabeled data
Pre-trained Models	L2P [108], DualPrompt [109], CODA-Prompt [134], SLCA [135]	Leverage pre-trained backbones with prompt tuning or adapters	**Pro:** Strong baseline performance **Con:** Requires large-scale pre-training
Architecture-Based			
Parameter Allocation	PackNet [102], HAT [110], SupSup [136]	Dedicate parameter subsets to each task via masking	**Pro:** Near-zero forgetting **Con:** Limited scalability
Network Expansion	Progressive NN [22], DEN [137], DyTox [138]	Dynamically expand network capacity for new tasks	**Pro:** Controlled forgetting **Con:** Linear growth in parameters
Modular Network	Expert Gate [139], Model Zoo [140], CoSCL [141]	Use mixture of experts or ensemble of models	**Pro:** Task specialization **Con:** Task identity inference needed
This Work (BIARM)	**Bio-inspired neuromodulation**	**Coordinate learning dynamics via four neuromodulatory modules (DA, 5-HT, NE, ACh)**	**Pro:** 98% retention, coordinated adaptation **Con:** Runtime overhead (+52-87%)

2.4 Adaptive Learning Rate Methods

The optimization of learning rates represents one of the most critical aspects of training artificial neural networks, with the choice of learning schedule often determining the difference between successful learning and failure to converge [142]. Traditional stochastic gradient descent uses a single, manually tuned learning rate, but the optimization landscape of deep networks suggests that different parameters may benefit from different learning rates at different times during training [143].

2.4.1 Mathematical Foundations of Adaptive Optimization

Consider the standard gradient descent update rule for minimizing a loss function $\mathcal{L}(\theta)$:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}(\theta_t) \tag{2.5}$$

where η is the learning rate and $\nabla_\theta \mathcal{L}$ is the gradient. The fundamental limitation of this approach lies in using a single scalar η for all parameters regardless of their individual optimization landscapes.

Adaptive methods address this by introducing parameter-specific learning rates. AdaGrad [144] maintains an accumulation of squared gradients for each parameter:

$$G_{t,i} = \sum_{\tau=1}^{t} g_{\tau,i}^2 \tag{2.6}$$

where $g_{t,i} = \frac{\partial \mathcal{L}}{\partial \theta_i}\big|_{\theta_t}$ is the gradient for parameter i at time t. The update rule becomes:

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,i} + \epsilon}} g_{t,i} \tag{2.7}$$

where ϵ is a small constant for numerical stability. This formulation provides larger updates for parameters with historically small gradients and smaller updates for parameters with large gradients. The theoretical justification stems from online convex optimization, where AdaGrad achieves regret bounds of $O(\sqrt{T})$ for convex functions [144].

However, the monotonic accumulation in G_t causes learning rates to decay indefinitely, potentially halting learning prematurely. RMSprop [145] addresses this through exponential moving averages:

$$v_{t,i} = \beta v_{t-1,i} + (1 - \beta) g_{t,i}^2 \tag{2.8}$$

with update rule:

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{v_{t,i} + \epsilon}} g_{t,i} \tag{2.9}$$

where $\beta \in (0, 1)$ controls the decay rate. This formulation allows learning rates to increase when recent gradients are small, preventing premature convergence.

Adam [146] extends this by incorporating momentum through first-moment estimation:

$$m_{t,i} = \beta_1 m_{t-1,i} + (1 - \beta_1) g_{t,i} \tag{2.10}$$

$$v_{t,i} = \beta_2 v_{t-1,i} + (1 - \beta_2) g_{t,i}^2 \tag{2.11}$$

with bias-corrected estimates:

$$\hat{\mathbf{m}}_{t,i} = \frac{m_{t,i}}{1 - \beta_1^t}, \quad \hat{\mathbf{v}}_{t,i} = \frac{v_{t,i}}{1 - \beta_2^t} \tag{2.12}$$

and update:

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{\hat{\mathbf{v}}_{t,i}} + \epsilon} \hat{\mathbf{m}}_{t,i} \tag{2.13}$$

The bias correction terms $(1 - \beta_1^t)^{-1}$ and $(1 - \beta_2^t)^{-1}$ compensate for initialization bias, ensuring that moment estimates are unbiased, especially during initial training steps.

2.4.2 Limitations of Single-Factor Adaptation

Despite their success, these adaptive methods share a fundamental limitation: they optimize each parameter independently based solely on gradient statistics [144, 146, 147]. Formally, the effective learning rate for parameter i at time t is:

$$\eta_{t,i}^{\text{eff}} = f(g_{1:t,i}) \tag{2.14}$$

where f is a function of the parameter's gradient history. This formulation ignores:

Cross-parameter dependencies: Parameters do not optimize independently. The loss landscape exhibits coupling between parameters, where the optimal learning rate for one parameter may depend on the state and recent updates of others.

Task-level context: The gradient statistics provide local information about the loss surface but no information about task difficulty, progress toward convergence, or the relationship between current and previous tasks in continual learning scenarios.

Multiple timescales: Biological learning operates across multiple timescales simultaneously, rapid synaptic changes, slower neuromodulatory adjustments, and long-term structural plasticity [28]. Single-exponential moving averages capture only one timescale.

2.4.3 Reinforcement Learning Optimization Challenges

The reinforcement learning framework described in Section 1.4.1 presents unique optimization challenges that compound the limitations of adaptive learning methods discussed above. Unlike supervised learning where the data distribution remains fixed, reinforcement learning faces non-stationary distributions as the policy evolves during training. Each policy update changes the distribution of states visited and actions taken, creating a moving target for optimization [148].

This non-stationarity interacts problematically with adaptive learning rate methods. AdaGrad's monotonic accumulation of gradient statistics becomes particularly unsuitable when the optimal learning rates should change as the policy evolves. RMSprop and Adam partially address this through exponential moving averages, but their fixed decay rates cannot adapt to the varying dynamics of policy improvement.

Additional challenges specific to reinforcement learning include:

- **High variance gradients**: Policy gradient estimates exhibit high variance, particularly in sparse reward environments where successful behaviors occur. rarely [149]
- **Exploration-exploitation trade-offs**: Learning rates affect not only convergence speed but also the exploration behavior, with overly aggressive learning potentially prematurely converging to suboptimal policies.
- **Credit assignment delays**: Temporal credit assignment challenges require learning rates that can adapt to different temporal horizons across state-action pairs.

These reinforcement learning-specific challenges motivate the bio-inspired neuromodulatory framework developed in Chapter 3, where different adaptive mechanisms operate at multiple timescales to provide context-aware learning rate modulation.

2.4.4 Toward Multi-Factor Coordination

Recent work has explored various forms of adaptive learning rate control, including joint optimization of meta-parameters [150], multiple learning rate candidates [151], and layer-specific adaptation [152, 153]. Building on these ideas, we propose a unified view where the effective learning rate is decomposed into multiple coordinated factors:

$$\eta_t^{\text{eff}} = \eta_0 \cdot \prod_{k=1}^{K} f_k(s_k(t)) \tag{2.15}$$

where f_k are modulation functions and $s_k(t)$ are state variables. This formulation makes explicit several limitations in current approaches:

- Factors are typically treated independently: $f_k(s_k(t))$ depends only on s_k
- Optimization occurs separately for each factor rather than jointly
- Coordination strategies are heuristic rather than principled

Biological neuromodulatory systems provide existence proofs that coordinated multi-factor adaptation is achievable and beneficial. Dopamine, serotonin, norepinephrine, and acetylcholine each modulate different aspects of learning and behavior while interacting to produce coordinated effects [28, 154, 59, 29]. Importantly, these systems do not operate independently, their effects are interdependent and coordinated [32].

The key difference from existing adaptive methods lies in this explicit coordination. Rather than treating adaptive factors independently as multiplicative components (Equation 2.15), biological systems suggest a jointly coordinated formulation.

2.5 Bio-Inspired Neural Network Design

Bio-inspired neural network architectures draw inspiration from various aspects of biological neural systems, ranging from individual neuron models to large-scale network organization [155]. These approaches differ in the level of biological detail they incorporate and the specific biological principles they emphasize.

Spiking neural networks (SNNs) model individual neurons more realistically than traditional artificial neurons by representing information through the precise timing of discrete action potentials [155, 156]. SNNs can theoretically compute functions that rate-coded networks cannot, and they offer potential advantages for

neuromorphic hardware implementation. However, training SNNs remains challenging due to the non-differentiable nature of spike generation [157].

Reservoir computing approaches, including echo state networks and liquid state machines, employ large pools of randomly connected recurrent neurons with fixed weights [158, 159]. Only the output connections are trained, reducing computational requirements while maintaining rich temporal dynamics. These architectures demonstrate that biological neural systems may leverage random connectivity and intrinsic dynamics rather than requiring all connections to be precisely tuned.

Attention mechanisms, inspired by selective attention in biological vision, enable neural networks to focus computational resources on relevant parts of input [78, 160]. The Transformer architecture's self-attention mechanism has proven remarkably effective across diverse domains, suggesting that biological attention principles provide computational advantages even when abstracted far from their biological implementation.

Modular and hierarchical architectures reflect the organization observed in biological neural systems, where specialized modules process different aspects of information and are organized hierarchically [75]. Capsule networks [161] attempt to incorporate explicit hierarchical relationships between visual features, while mixture-of-experts models [162] dynamically route information to specialized subnetworks.

2.6 Computational Neuromodulation

Computational neuromodulation represents a relatively recent but promising direction in bio-inspired artificial intelligence, drawing inspiration from the neuromodulatory systems that regulate learning and adaptation in biological brains [28, 32]. Unlike traditional neural networks where all adaptation occurs through synaptic weight modification, neuromodulatory systems provide global signals that modify how synapses adapt and how neurons respond to their inputs.

Doya proposed a unified framework linking specific neuromodulatory systems to computational functions in reinforcement learning [28]. Dopamine signals reward prediction errors, providing a teaching signal for value learning. Serotonin modulates the timescale of reward prediction, influencing temporal discounting. Norepinephrine signals unexpected uncertainty, modulating exploration rates. Acetylcholine distinguishes expected from unexpected uncertainty, controlling the balance between memory encoding and retrieval.

Early computational implementations of neuromodulation explored various isolated functions. Some approaches used neuromodulatory signals to adaptively adjust learning rates based on prediction errors or other performance metrics [163], while others employed neuromodulation to control attention, synaptic plasticity, or network dynamics [164, 165]. Meta-learning approaches shared conceptual similarities with neuromodulation by learning how to learn [166], optimizing not just task-specific parameters but also learning algorithms, learning rates, or architectural choices.

Recent work has begun to address the key limitation of these earlier approaches, their focus on isolated neuromodulatory functions rather than coordinated multi-system organization. Mei et al. [167] demonstrated that integrating dopamine, acetylcholine, serotonin, and noradrenaline dynamics in concert enables adaptive learning that addresses catastrophic forgetting and robustness in non-stationary environments. Wang et al. [168] developed a Neuromodulated Meta-Learning framework that jointly optimizes structural and synaptic parameters, extending traditional meta-learning to bio-inspired regimes that echo biological flexibility. Similarly, neuromodulatory extensions to meta-reinforcement learning have shown improved dynamic representations and task adaptation compared to standard meta-RL approaches [169], while bio-plausible spiking network models with reward-modulated plasticity further demonstrate the computational advantages of neuromodulation in biologically grounded learning systems [170].

These advances demonstrate that biological neuromodulatory systems do not operate independently; their interactions create emergent computational properties that exceed what any single system could achieve [29]. This observation motivates the coordinated neuromodulatory framework developed in this thesis, which builds upon these recent multi-system approaches to further explore the computational principles underlying coordinated neuromodulation.

2.7 Research Gaps and Opportunities

Despite significant advances across the fields reviewed above, several fundamental gaps limit the effectiveness of current approaches to adaptive robotic systems. The identification of these gaps (Figure 2.2) provides crucial motivation for the developmental evolutionary robotics approach presented in this thesis.

2.7.1 Limited Coordination Between Adaptive Mechanisms

The first major gap lies in the limited coordination between different adaptive mechanisms in current systems. While individual techniques such as adaptive

learning rates, attention mechanisms, and regularization methods demonstrate effectiveness in isolation, most approaches fail to coordinate these mechanisms in ways that provide synergistic benefits [72]. Biological systems, by contrast, exhibit sophisticated coordination between multiple adaptive processes operating at different timescales, suggesting that artificial systems might benefit from similar coordination strategies.

This lack of coordination manifests in several ways. Adaptive learning rate methods adjust parameters based solely on gradient statistics without considering task-level context. Attention mechanisms operate independently of learning rate adaptation. Continual learning approaches protect parameters from modification without coordinating this protection with other adaptive mechanisms. The potential for coordination between these different forms of adaptation remains largely unexplored.

The progression from fixed to coordinated adaptation (Figure 2.1) reflects an evolution toward more sophisticated learning control. While meta-learning introduced per-parameter adaptation, it lacks the coordinated modulation that characterizes biological systems. BIARM bridges this gap by implementing coordinated multi-factor adaptation inspired by neuromodulatory coordination.

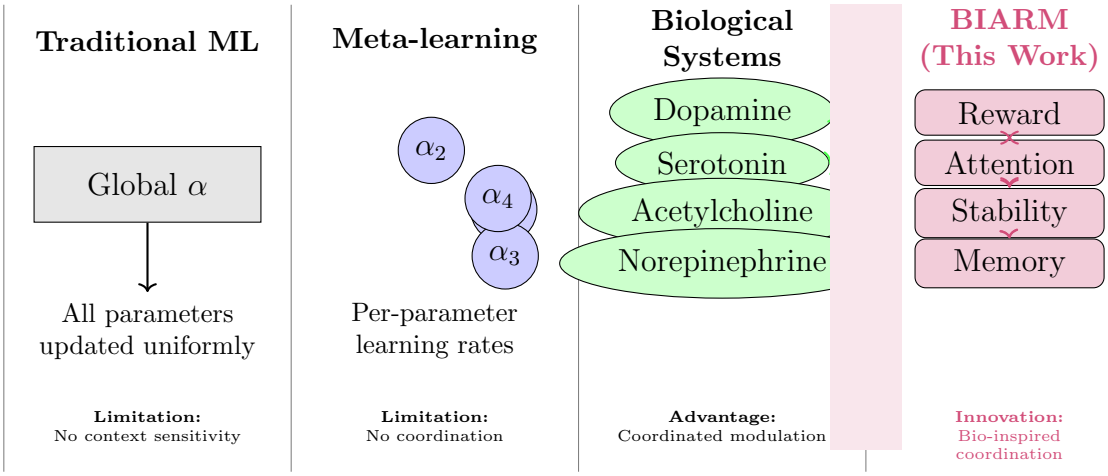


Figure 2.1: **Evolution of coordination mechanisms in adaptive learning systems.**

2.7.2 Sample Inefficiency and Computational Cost

Evolutionary robotics and deep reinforcement learning both suffer from severe sample inefficiency [93]. Evolutionary approaches require millions of fitness evaluations, while reinforcement learning methods demand extensive interaction with environments. This inefficiency becomes particularly problematic in robotics, where

each interaction carries temporal and economic costs that make extensive exploration prohibitively expensive.

Biological learning systems achieve remarkable sample efficiency through multiple mechanisms operating in concert. Developmental constraints focus exploration on viable solutions. Neuromodulatory systems adapt learning rates based on performance feedback. Transfer learning reuses knowledge across related tasks. Current artificial systems rarely integrate these mechanisms, instead relying on brute-force exploration that requires enormous computational resources.

2.7.3 Superficial Biological Inspiration

Many bio-inspired approaches incorporate superficial aspects of biological systems without capturing fundamental organizational principles [27]. For example, neural networks use biological terminology (neurons, synapses) but lack key properties of biological neural systems such as sparse connectivity, bounded dynamics, and multi-timescale adaptation. This superficial biomimicry limits the extent to which artificial systems can benefit from billions of years of evolutionary optimization.

The challenge lies in identifying which aspects of biological systems provide genuine computational advantages versus which are implementation details specific to biological substrates. Successful bio-inspired approaches must abstract essential computational principles while omitting irrelevant biological details. This requires a deeper understanding of why biological systems work as they do, not just how they work.

2.7.4 Integration of Multiple Timescales

Biological learning systems seamlessly integrate adaptation across multiple timescales, from rapid synaptic plasticity (milliseconds to seconds) to slower neuromodulatory changes (minutes to hours) to developmental processes (days to years) [25]. Current artificial systems typically operate at a single timescale, either training networks from scratch (slow timescale) or fine-tuning pre-trained networks (fast timescale), but rarely integrate multiple timescales of adaptation within a single system.

The integration of evolutionary optimization, developmental processes, and real-time learning represents a fundamental challenge receiving limited attention in current research. While biological systems seamlessly integrate these different types of adaptation, artificial systems typically treat them as separate processes.

Understanding how to coordinate evolutionary optimization, developmental processes, and real-time learning could enable more capable and robust adaptive systems.

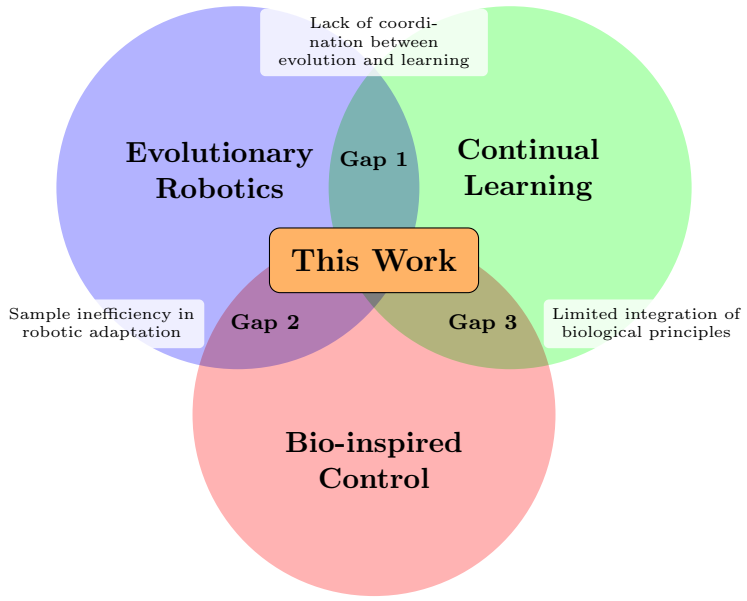


Figure 2.2: **Research landscape and contribution positioning.**

2.8 Conclusion

This chapter has surveyed the theoretical foundations underlying adaptive robotic systems, spanning evolutionary computation, continual learning, optimization methods, bio-inspired neural architectures, and computational neuromodulation. Several key themes emerge from this comprehensive review that directly inform the research direction of this thesis.

The historical trajectory of evolutionary robotics demonstrates both the promise and limitations of biology-inspired automatic design methods. While evolutionary approaches have successfully discovered elegant solutions to complex control problems, challenges of computational expense, scalability, and the reality gap continue to constrain practical applications. The persistent problem of catastrophic forgetting in continual learning systems reveals fundamental limitations in how artificial systems acquire sequential knowledge, motivating approaches that can better coordinate stability and plasticity.

The exploration of bio-inspired neural design and computational neuromodulation reveals that biological systems employ sophisticated coordination mechanisms operating across multiple temporal and spatial scales. These mechanisms enable adaptation, self-regulation, and robust behavior in ways that current artificial systems rarely achieve. Insights from developmental biology suggest that processes of growth, differentiation, and self-organization may be as important as final architectures in determining system capabilities.

Most critically, this review has identified a fundamental gap in current research: the limited coordination between different adaptive mechanisms. While individual techniques demonstrate effectiveness in isolation, few approaches achieve the sophisticated integration characteristic of biological systems, where evolutionary optimization, developmental processes, learning mechanisms, and real-time adaptation operate in concert across multiple timescales.

The theoretical framework established in this chapter provides the foundation for the specific implementations presented in subsequent chapters. The mathematical formalisms of gene regulatory networks and neuromodulatory dynamics offer principled approaches to translating biological insights into computational mechanisms. The design principles articulated here guide the development of bio-inspired systems that capture genuine biological advantages rather than superficial mimicry. The research opportunities identified suggest that advancing adaptive robotic systems requires moving beyond incremental improvements to existing methods toward more fundamental integration of biological principles with computational approaches.

CHAPTER 3

BIO-INSPIRED ADAPTIVE RATE MODULATION FOR MULTI-TASK LEARNING

3.1 Introduction and Motivation

This chapter discusses the research article: "Bio-Inspired Adaptive Rate Modulation for Multi-task Learning in Progressive Neural Networks", Informatica, December 2025 [37].

Sequential multi-task learning represents one of the most challenging problems in robotics, where systems must continuously acquire new skills while preserving previously learned capabilities. The fundamental limitation of existing approaches lies in their reliance on fixed learning strategies. Most current methods treat learning as a monolithic process, applying uniform learning rates, attention mechanisms, and memory consolidation strategies regardless of the specific demands of the current learning context. This one-size-fits-all approach ignores the dynamic nature of learning, where different phases of skill acquisition may benefit from different optimization strategies.

Consider how humans learn a sequence of related motor skills, such as playing different musical instruments. Initially, learning a piano piece requires focused attention on finger placement and timing, with rapid parameter updates as basic motor patterns are established. Later, when learning violin, the system can leverage some existing musical knowledge while adapting to new physical constraints. Throughout this process, different aspects of learning are coordinated: reward processing adapts based on performance feedback, attention shifts between

different aspects of the task, stability mechanisms prevent interference between instruments, and memory systems selectively consolidate knowledge that transfers across domains.

Recent advances in neuroscience have revealed that biological learning systems employ complex neuromodulatory mechanisms that coordinate multiple aspects of learning and adaptation [28, 29, 32, 30]. Unlike artificial systems that typically rely on fixed learning parameters, biological systems dynamically adjust learning rates, attention allocation, and memory consolidation based on contextual information provided by specialized neurotransmitter systems.

This chapter presents a bio-inspired adaptive rate modulation framework that translates the organizational principles of biological neuromodulatory systems into practical computational mechanisms for sequential multi-task learning. Rather than treating dopamine, serotonin, norepinephrine, and acetylcholine as isolated mechanisms, the key insight is that their coordinated interaction enables forms of adaptive learning impossible with any single system operating alone.

3.2 Framework Architecture and Mathematical Formulation

3.2.1 Overall System Design

The Bio-Inspired Adaptive Rate Modulation (BIARM) framework translates the organizational principles of biological neuromodulation established in Section 1.2 into a practical computational architecture that can be integrated with existing neural network approaches. The framework consists of four specialized modules that correspond to the major neuromodulatory systems, each responsible for monitoring different aspects of the learning environment and contributing to coordinated adaptation of learning parameters.

The system operates through a hierarchical coordination mechanism where each neuromodulatory module maintains its own internal state while contributing to system-wide learning parameter modulation. Unlike traditional approaches that treat learning parameters as fixed or schedule them according to predetermined patterns, our framework enables dynamic adaptation based on current learning context, environmental demands, and inter-module coordination. The integration of the framework with existing neural network architectures is achieved through a standardized interface that allows the neuromodulatory influences to be applied to various types of learning algorithms without requiring fundamental changes to the underlying learning procedures.

3.2.2 Neuromodulatory State Dynamics

Each neuromodulatory module $m \in \{\text{DA}, \text{5HT}, \text{NE}, \text{ACh}\}$ maintains an internal state $s_m(t) \in [0, 1]$ that evolves during training according to discrete exponential moving averages:

$$s_m(t+1) = s_m(t) + \tau_m \cdot (s_m^{\text{target}}(t) - s_m(t)) \tag{3.1}$$

where τ_m represents the discrete adaptation rate for module m, determining how quickly the module state responds to target changes within each training step. The τ_m values represent relative adaptation speeds rather than absolute biological timing, and $s_m^{\text{target}}(t)$ is computed by a module-specific neural network. The time constants are assigned in accordance with the functional role of each module: DA uses $\tau_{DA} = 0.1$ to support performance-based feedback, NE uses $\tau_{NE} = 0.05$ for attention gating, ACh uses $\tau_{ACh} = 0.3$ for knowledge integration, and 5-HT uses $\tau_{5HT} = 0.5$ for learning stabilisation. The precise numerical values were determined through systematic grid search over the parameter space $[0.01, 1.0]$ with 0.05 increments, evaluating 10,000 parameter combinations using a composite score weighting convergence speed (30%), training stability (40%), and final performance (30%) across multiple environments. The identified configuration was validated across 10 independent runs with different random seeds, confirming robustness to initialisation.

3.2.3 Target State Computation Networks

Each module computes its target state using a dedicated neural network that processes relevant contextual information. This design allows each module to specialize in detecting and responding to specific types of environmental and learning context while maintaining the flexibility to adapt to novel situations.

Dopamine Module The dopamine module monitors reward signals and task performance to adaptively modulate learning rates based on the reliability and magnitude of learning progress. Unlike fixed learning rate schedules, this module adjusts learning rates dynamically based on current performance trends, increasing rates when learning is progressing rapidly and decreasing rates when performance has stabilized.

The dopamine module processes current state and reward signal to adapt learning based on performance feedback:

$$s_{\text{DA}}^{\text{target}} = \sigma(f_{\text{DA}}([s_{\text{current}}, r_{\text{current}}])) \tag{3.2}$$

The dopamine module takes as input the concatenation of the current state $s_{current}$ and the current reward $r_{current}$, producing a scalar target state via sigmoid normalisation. This design allows the module to modulate learning intensity based on immediate reward feedback, increasing its activation when rewards are high and decreasing it otherwise, thereby coupling the effective learning rate to ongoing task performance.

Serotonin Module The serotonin module regulates learning stability and temporal discounting, helping to balance rapid adaptation with the preservation of previously learned knowledge. This module monitors learning variance and adjusts parameters to maintain stable learning dynamics while preventing excessive oscillations that could disrupt acquired knowledge:

$$s_{\text{5HT}}^{\text{target}} = \sigma(f_{\text{5HT}}([s_{\text{current}}, s_{\text{DA}}])) \tag{3.3}$$

The serotonin module takes as input the concatenation of the current state $s_{current}$ and the current dopamine state s_{DA}, producing a scalar target state via sigmoid normalisation. This inter-module dependency allows the serotonin state to be directly conditioned on the ongoing reward-processing dynamics of the dopamine module, implementing the biological principle that serotonergic stability regulation operates in response to dopaminergic activity.

Norepinephrine Module The norepinephrine module manages attention allocation and the exploration-exploitation trade-off, directing learning resources toward the most relevant aspects of the current task while maintaining the flexibility to explore novel strategies when current approaches are insufficient:

$$s_{\text{NE}}^{\text{target}} = \sigma(f_{\text{NE}}([s_{\text{current}}, n_{\text{task}}])) \tag{3.4}$$

The norepinephrine module takes as input the concatenation of the current state $s_{current}$ and the current task novelty measure n_{task}, producing a scalar target state via sigmoid normalisation. This design directly couples the module's activation to how novel the current task remains, allowing it to modulate attention allocation based on learning progress rather than fixed schedules. Task novelty n_{task} starts at 1.0 for each new task and decays exponentially:

$$n_{\text{task}}(t+1) = n_{\text{task}}(t) \times 0.995 \tag{3.5}$$

This simple but effective measure quantifies how new the current task remains relative to initial conditions, providing the norepinephrine module with a continuous signal that naturally diminishes as the agent accumulates experience within

a task. The implementation draws upon research on attention and arousal, incorporating mechanisms for detecting task novelty and environmental changes that require attention reallocation.

Acetylcholine Module The acetylcholine module facilitates memory consolidation and knowledge integration, helping to coordinate the transfer of knowledge between different tasks while protecting important memories from interference:

$$s_{\text{ACh}}^{\text{target}} = \sigma(f_{\text{ACh}}([s_{\text{current}}, s_{\text{NE}}])) \tag{3.6}$$

The acetylcholine module takes as input the concatenation of the current state $s_{current}$ and the current norepinephrine state s_{NE}, producing a scalar target state via sigmoid normalisation. This dependency allows the acetylcholine state to be conditioned on the current level of attentional engagement, implementing the biological principle that cholinergic memory integration operates in coordination with noradrenergic attention signals.

3.2.4 Coordinated Learning Parameter Modulation

The coordination between these four modules represents enables the framework to achieve performance improvements that exceed what any individual module could provide in isolation. The coordinated influence of all four neuromodulatory modules determines the effective learning parameters used during each training step, moving beyond isolated adaptive mechanisms toward integrated optimization strategies.

The effective learning rate for each training step is computed as a weighted combination of all neuromodulatory states:

$$\eta_{\text{effective}} = \eta_{\text{base}} \cdot (w_{\text{DA}} \cdot s_{\text{DA}} + w_{\text{5HT}} \cdot s_{\text{5HT}} + w_{\text{NE}} \cdot s_{\text{NE}} + w_{\text{ACh}} \cdot s_{\text{ACh}}) \tag{3.7}$$

where η_{base} is the base learning rate and the weighting coefficients reflect the relative contributions of each neuromodulatory system: $w_{\text{DA}} = 0.3$ provides primary influence from reward processing, $w_{\text{5HT}} = 0.2$ contributes stability control, $w_{\text{NE}} = 0.3$ implements attention-based modulation, and $w_{\text{ACh}} = 0.2$ manages memory integration influence. This weighted formulation ensures that the effective learning rate reflects the consensus of all four adaptive modules rather than being dominated by any single factor, with the fixed coefficients reflecting the relative contribution of each module's functional role to the overall learning dynamics.

The attention coefficients for lateral connections between Progressive Neural Network columns are computed through a three-component mechanism:

$$\alpha_{i,k}^{(l)} = \sigma(W_\alpha h_k^{(l)}) \cdot V(h_k^{(l)}) \cdot \sigma\left(\sum_{m \in \{\text{DA,5HT,NE,ACh}\}} w_m s_m\right) \tag{3.8}$$

This formulation enables selective knowledge transfer by combining content-based attention through $\sigma(W_\alpha h_k^{(l)})$ providing learned attention weights that focus on relevant features, value transformation through $V(h_k^{(l)})$ implementing learned projections that extract useful information, and context-dependent modulation through coordinated neuromodulator influence. The sigmoid activation on the modulation term ensures bounded influence while allowing each module to contribute according to its current state and learned importance weights.

3.3 Integration with Progressive Neural Networks

3.3.1 Architectural Enhancement

While Progressive Neural Networks (Section 2.3.1) provide guaranteed zero forgetting through architectural isolation [22], they use fixed learning parameters and static connection strengths that limit the system's ability to adapt to varying task demands and learning contexts in sequential learning of control tasks.

Building on PNN's foundation of task-specific columns with lateral connections, BIARM integrates adaptive rate modulation with careful consideration of how the neuromodulatory influences should be applied at different stages of the learning process. During the initial training of a new task column, the neuromodulatory system must balance rapid learning with stability to ensure that the new task is learned effectively without disrupting the knowledge transfer from previous tasks.

Our framework as shown in Figure 3.1 computes layer outputs as:

$$h_i^{(l)} = f\left(W_i^{(l)} h_i^{(l-1)} + \sum_{k=1}^{i-1} \alpha_{i,k}^{(l)} U_{i,k}^{(l)} h_k^{(l)}\right) \tag{3.9}$$

The implementation begins with the integration of neuromodulatory state variables into the Progressive Neural Network architecture. Each task column is augmented with neuromodulatory state vectors that track the current activation levels of the four neuromodulatory systems. These state vectors evolve over time based

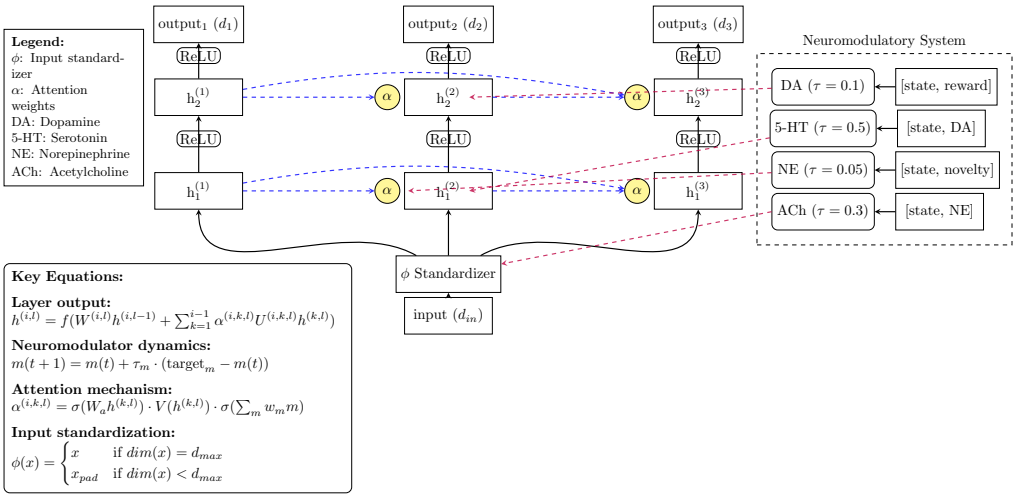


Figure 3.1: **BIARM Architecture.** Task-specific columns with attention-modulated lateral connections (α) enable progressive learning and knowledge transfer. Neuromodulatory system (bottom) regulates learning dynamics. Solid/dashed/dotted arrows indicate forward/lateral/neuromodulatory connections.

on the current learning context, task performance, and environmental conditions, providing dynamic context information that guides learning parameter adaptation.

3.3.2 Input Standardization and Task Management

To handle varying input dimensions across different tasks, we implement adaptive input standardization that ensures compatibility across different task dimensions:

$$\phi(x) = \begin{cases} x & \text{if } \dim(x) = d_{\max} \\ [x; 0_{d_{\max} - \dim(x)}] & \text{if } \dim(x) < d_{\max} \end{cases} \tag{3.10}$$

where $d_{\max}$ is the maximum input dimension across all tasks. This standardization layer enables the framework to handle heterogeneous task sequences without requiring manual specification of transformation rules for each new task.

3.3.3 Complete Training Algorithm

The complete training procedure integrates all components of the BIARM framework into a cohesive algorithm that coordinates neuromodulatory adaptation with Progressive Neural Network training. Algorithm 1 maintains neuromodulatory

state variables throughout training, updating them at each step based on current context while using their coordinated influence to modulate learning parameters.

Algorithm 1 Bio-Inspired Adaptive Rate Modulation Training

Require: Task sequence $\mathcal{T} = \{T_1, T_2, \ldots, T_n\}$, episodes E, batch size B
Ensure: Trained progressive network with adaptive modulation

```
Initialize modulation system with states s_m = 0.5 for all m
Initialize base learning rate η_base and network parameters
for each task T_i in 𝒯 do
    Initialize new column C_i
    Calculate initial task novelty n_task = 1.0
    for episode = 1 to E do
        Reset environment
        while episode not done do
            Observe state s_current, execute action, get reward r
            Update module target states (Eqs. 3.2–3.6)
            Update module states (Eq. 3.1)
            Compute effective learning rate (Eq. 3.7)
            Compute attention coefficients (Eq. 3.8)
            Store transition in replay buffer
            if buffer size ≥ B then
                Sample mini-batch from buffer
                Compute loss with modulated parameters
                Update network parameters with η_effective
            end if
            Update task novelty: n_task ← n_task × 0.995
        end while
    end for
end for
```

Key implementation details include the network architecture where each task column employs two hidden layers with 128 units each using ReLU activation, with input and output dimensions adapted to environment requirements and Xavier initialization for weights with zero initialization for biases. Each neuromodulatory module uses a two-layer network with 64 hidden units, employing ReLU activation in the hidden layer and sigmoid in the output layer. The training configuration uses a batch size of 64, base learning rate of 0.001 modulated by the neuromodulatory system, Adam optimizer with modulated learning rates, experience replay maintaining 10,000 transitions per task, and epsilon-greedy exploration starting at 1.0 and decaying to 0.01.

3.4 Experimental Validation and Results

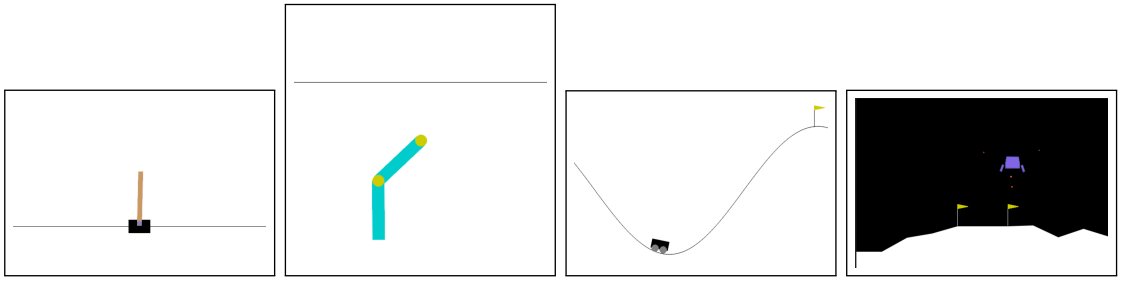
3.4.1 Experimental Setup and Environment Selection

The experimental validation of the bio-inspired adaptive rate modulation framework was conducted across four distinct OpenAI Gymnasium environments that provide diverse challenges for sequential multi-task learning. The environments were selected to test different aspects of adaptive learning, including control precision, exploration requirements, temporal reasoning, and multi-objective optimization. Each environment presents unique characteristics that allow for comprehensive evaluation of the neuromodulatory mechanisms under different learning conditions. The four selected environments are visualized in Figure 3.2 and described as follows:

CartPole-v1 serves as a foundational test of control precision and stability maintenance. This environment requires learning precise control policies that can balance competing objectives while maintaining system stability over extended periods. The task provides clear performance feedback that enables detailed analysis of how the neuromodulatory systems respond to different phases of the learning process, from initial skill acquisition through performance optimization and maintenance. With a 4-dimensional continuous state space consisting of cart position, velocity, pole angle, and angular velocity, and a 2-dimensional discrete action space for left or right force application, the environment awards +1 per timestep while the pole remains upright. Episode termination occurs when the pole angle exceeds $\pm 12^\circ$ or cart position exceeds ± 2.4 units, with a maximum episode length of 500 timesteps and training conducted over 400 episodes.

The Acrobot-v1 environment tests the framework's ability to handle complex dynamics and long-horizon planning requirements. The under-actuated nature of this system requires coordination between exploration and exploitation, making it an ideal testbed for evaluating the norepinephrine and dopamine modules' ability to manage the exploration-exploitation trade-off effectively. This two-link pendulum system features a 6-dimensional continuous state space comprising joint angles and velocities, with a 3-dimensional discrete action space for torque application to the second joint. The environment provides a reward of -1 per timestep until the goal is reached, which requires swinging the end-effector above a target height. Training proceeds for 500 episodes with a maximum episode length of 500 timesteps.

The MountainCar-v0 environment presents sparse reward challenges that test the framework's ability to maintain learning motivation in the absence of frequent positive feedback. This environment is particularly valuable for evaluating the dopamine module's reward prediction mechanisms and the serotonin module's



(a) CartPole Visualization (b) Acrobot Visualization (c) MountainCar Visualization (d) LunarLander Visualization

Figure 3.2: **Visualization of OpenAI Gymnasium environments** : (a) CartPole-v1; (b) Acrobot-v1; (c) MountainCar-v0; and (d) LunarLander-v3.

ability to maintain learning stability during extended periods of limited feedback. The 2-dimensional continuous state space captures position and velocity, while the 3-dimensional discrete action space offers left, no action, and right movements. The environment awards -1 per timestep, requiring the agent to reach a flag at the top of a hill within a maximum episode length of 200 timesteps across 800 training episodes.

The LunarLander-v3 environment provides the most complex learning challenge, requiring coordination of multiple control objectives including precise positioning, velocity control, and fuel efficiency optimization. This environment tests the framework's ability to coordinate all four neuromodulatory modules simultaneously while managing the complex interactions between different learning objectives. The 8-dimensional continuous state space includes position, velocity, orientation, and ground contact information, with a 4-dimensional discrete action space controlling no action, left engine, main engine, and right engine. The complex reward function evaluates landing success, fuel efficiency, and stability, with episode termination occurring upon landing, crash, or timeout within a maximum episode length of 1000 timesteps over 800 training episodes.

All experiments employed standardized training procedures across all environments to ensure fair comparison between the adaptive framework and baseline Progressive Neural Networks. Each experiment was repeated across multiple random seeds to provide statistical validation of the observed performance improvements, with comprehensive data collection enabling detailed analysis of the mechanisms underlying the performance gains. The training configuration maintained identical network architectures across all comparisons, with task-specific training episodes determined based on environment complexity as detailed above. Training hyperparameters are standardized across all experiments as outlined in Table 3.1:

Table 3.1: Training Configuration Parameters

Parameter	Value
Batch size	64
Base learning rate	0.001
Optimizer	Adam
Discount factor (γ)	0.99
Experience replay buffer	10,000 transitions per task
Epsilon-greedy exploration	ϵ starts at 1.0, decays to 0.01 with rate 0.995
Task-specific Training Episodes	
CartPole-v1	400 episodes
Acrobot-v1	500 episodes
MountainCar-v0	800 episodes
LunarLander-v3	800 episodes

3.4.2 Performance Results and Comprehensive Analysis

The results as presented in Table 3.2 and Figure 3.3 demonstrate consistent and substantial performance improvements across all tested environments. The experimental validation reveals that the magnitude of improvement correlates with task complexity and the benefits of coordinated adaptation, providing compelling evidence for the value of bio-inspired neuromodulatory mechanisms in sequential multi-task learning.

Table 3.2: Performance Comparison Between Baseline PNN and BIARM

Environment	**Baseline PNN**	**BIARM**	**Improvement**
CartPole-v1	491.00 ± 11.94	$\mathbf{500.00 \pm 0.00}$	+1.83%
Acrobot-v1	-94.50 ± 24.96	$\mathbf{-93.50 \pm 18.91}$	+1.06%
MountainCar-v0	-165.90 ± 46.31	$\mathbf{-119.80 \pm 28.02}$	+27.79%
LunarLander-v3	-417.17 ± 412.62	$\mathbf{233.37 \pm 34.38}$	+155.94%

Note: Improvement is computed as

$$\frac{\text{BIARM} - \text{Baseline PNN}}{|\text{Baseline PNN}|} \times 100$$

where the absolute value in the denominator accounts for environments in which the baseline performance is negative.

The most substantial improvement occurred in the LunarLander-v3 environment, where the adaptive framework achieved positive performance scores of 233.37 ± 34.38 while the baseline Progressive Neural Networks consistently failed to learn

effective policies with negative performance of -417.17 ± 412.62. This 155.94% improvement represents a qualitative difference in learning capability rather than merely incremental performance enhancement, demonstrating that the neuromodulatory framework enables successful task completion in scenarios where baseline methods fail entirely. The substantial variance reduction from 412.62 to 34.38 represents a 91.7% decrease, indicating that the framework not only improves average performance but also enhances the consistency and reliability of learned behaviors.

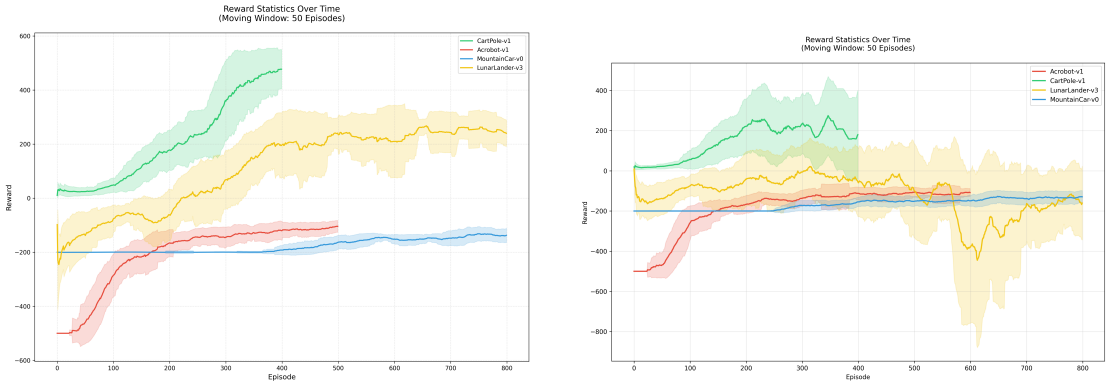
The CartPole-v1 results, while showing a more modest 1.83% improvement in mean performance, demonstrated a remarkable reduction in performance variance, achieving zero variance across test episodes compared to the baseline variance of 11.94. This result indicates that the neuromodulatory framework not only improves average performance but also enhances the consistency and reliability of learned behaviors, which is crucial for practical robotic applications. The achievement of perfect consistency demonstrates that the adaptive mechanisms can eliminate performance variability in well-structured control tasks.

The Acrobot-v1 and MountainCar-v0 environments showed intermediate performance improvements of 1.06% and 27.79% respectively, with substantial reductions in performance variance in both cases. For Acrobot-v1, the variance decreased by 24.3% from 24.96 to 18.91, while MountainCar-v0 achieved a 39.5% variance reduction from 46.31 to 28.02. These results demonstrate that the framework provides benefits across diverse learning scenarios, from those requiring minimal adaptation to those demanding substantial strategic adjustments. The MountainCar-v0 results are particularly noteworthy given the sparse reward structure, which typically challenges learning algorithms by providing minimal feedback for extended periods.

The correlation between environment complexity and performance improvement magnitude suggests that the neuromodulatory framework provides the greatest benefits in scenarios that require coordination between multiple learning objectives. The improvement in LunarLander-v3, which requires managing multiple competing objectives simultaneously, demonstrates that biological principles may be particularly valuable for complex control problems that exceed the capabilities of fixed-parameter approaches.

3.4.3 Computational Overhead Analysis

The BIARM framework introduces computational overhead (Table 3.3) through four primary sources: forward passes through neuromodulatory networks (4 modules $\times$ 64 hidden units each), computation of modulated learning rates and attention coefficients, state updates for neuromodulatory values, and attention-gated lateral connection computations.



(a) BIARM learning curves

(b) Baseline PNN learning curves

Figure 3.3: **Learning curves showing training progress across all four environments.** (a) BIARM framework (b) baseline Progressive Neural Networks. The shaded regions represent standard deviation across multiple runs, demonstrating superior stability and convergence with BIARM.

Table 3.3: Computational Overhead Analysis

Environment	**Baseline Time/Step**	**BIARM Time/Step**	**Overhead**
CartPole-v1	0.45 ms	0.68 ms	+52%
Acrobot-v1	0.52 ms	0.89 ms	+71%
MountainCar-v0	0.41 ms	0.71 ms	+73%
LunarLander-v3	0.63 ms	1.18 ms	+87%

The overhead ranges from +52% (CartPole) to +87% (LunarLander) per training step. This scales linearly with task count due to lateral connection computations. Detailed comparative analysis with other continual learning approaches, including memory complexity and scalability considerations, is provided in Section 5.3.2.

For the immediate application context, the overhead is justified by performance improvements: the 87% overhead in LunarLander enables the difference between complete failure (baseline) and successful task completion (BIARM), while the 52% overhead in CartPole achieves perfect consistency (zero variance) compared to baseline variance of 11.94.

3.5 Comprehensive Ablation Study

3.5.1 Methodology and Component Isolation

To validate the contribution of each adaptive module and understand the mechanisms underlying the observed performance improvements, we conducted systematic ablation studies. The comprehensive ablation studies provide crucial insights into the mechanisms underlying these performance improvements. The methodology involved removing individual neuromodulatory modules while maintaining the computational framework, allowing us to isolate the specific contribution of each component. The systematic removal of individual neuromodulatory modules reveals that each component contributes meaningfully to overall system performance, with the dopamine and acetylcholine modules showing the largest individual contributions in most environments.

When ablating modules with dependencies (5HT depends on DA, ACh depends on NE), we replaced missing dependency inputs with fixed neutral values of 0.5, representing moderate activation. This approach allows fair comparison across configurations while eliminating the specific module's adaptive contribution without disrupting the computational pipeline.

3.5.2 Ablation Results and Component Analysis

The ablation results, as shown in Table 3.4 and Figure 3.4, reveal distinct contributions from each module that validate the biological inspiration behind the framework design. The dopamine and acetylcholine modules show the most severe impact when removed, confirming their critical roles in reward-based learning adaptation and knowledge integration between task columns.

Table 3.4: Ablation Study Impact on Performance Across All Environments

Environment	Full BIARM	w/o DA	w/o 5-HT	w/o NE	w/o ACh
CartPole-v1	500.00 ± 0.00	195.77 ± 148.93	173.29 ± 198.46	233.33 ± 212.19	247.78 ± 210.25
Acrobot-v1	-93.50 ± 18.91	-109.82 ± 20.67	-110.78 ± 21.01	-110.27 ± 24.56	-106.48 ± 22.55
MountainCar-v0	-119.80 ± 28.02	-125.90 ± 32.98	-126.26 ± 26.96	-139.63 ± 31.10	-133.39 ± 29.71
LunarLander-v3	233.37 ± 34.38	-18.63 ± 190.72	218.15 ± 97.80	202.99 ± 135.61	-39.61 ± 152.67

Dopamine Module Ablation The dopamine module ablation results confirm its crucial role in reward-based learning adaptation, with its removal leading to substantial performance degradation across all environments. In LunarLander-v3, removing the dopamine module resulted in a 108.0% performance impact, dropping performance from 233.37 ± 34.38 to -18.63 ± 190.72. The magnitude of this

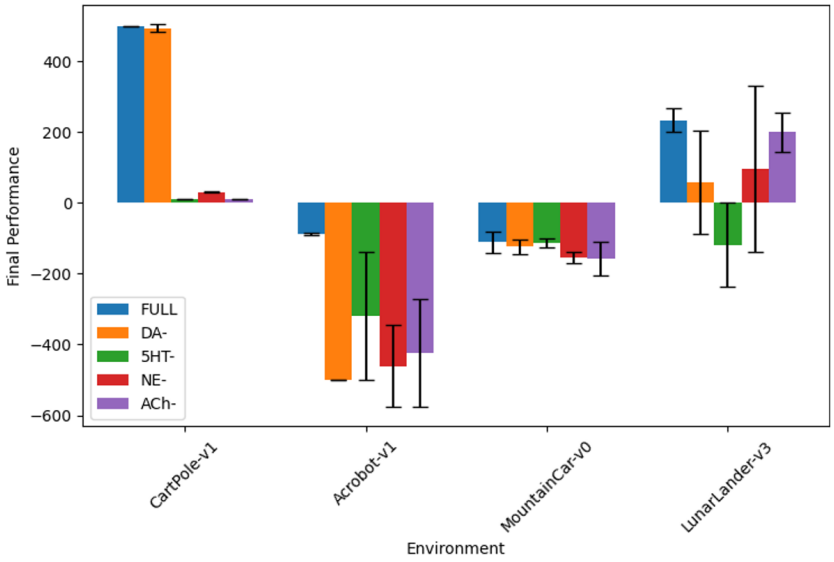


Figure 3.4: **Final Performance Evaluation of Catastrophic Forgetting for Ablation Study.** Bar chart showing performance degradation when removing individual neuromodulatory modules across all four environments, demonstrating the critical contribution of each component to the overall system performance.

degradation correlates with the reward structure complexity of each environment, with more complex reward landscapes showing greater dependence on adaptive reward processing mechanisms.

Acetylcholine Module Ablation The acetylcholine module ablation studies reveal its essential role in knowledge integration and transfer between task columns. The removal of this module showed the highest overall impact in Lunar-Lander-v3 with a 117.0% performance degradation, reducing performance to -39.61 ± 152.67. This particularly impacted performance in environments where knowledge transfer could provide substantial benefits, demonstrating the importance of adaptive knowledge integration mechanisms in continual learning scenarios.

Serotonin Module Ablation The serotonin module, while showing smaller individual contributions to mean performance with only a 6.5% impact in LunarLander-v3, demonstrated substantial impacts on learning stability and variance reduction. The ablation of this module led to increased performance variability across all environments, with variance increasing from 34.38 to 97.80 in LunarLander-v3, confirming its role in promoting stable and consistent learning dynamics.

Norepinephrine Module Ablation The norepinephrine module ablation studies revealed its importance for attention allocation and exploration management with a 13.0% performance impact in LunarLander-v3. The removal of this module particularly impacted performance in environments requiring complex exploration

strategies or attention allocation across multiple task aspects. While showing moderate direct performance reduction, the variance increased fourfold from 34.38 to 135.61, demonstrating its critical role in maintaining training stability also.

3.5.3 Temporal Dynamics and Coordination Patterns

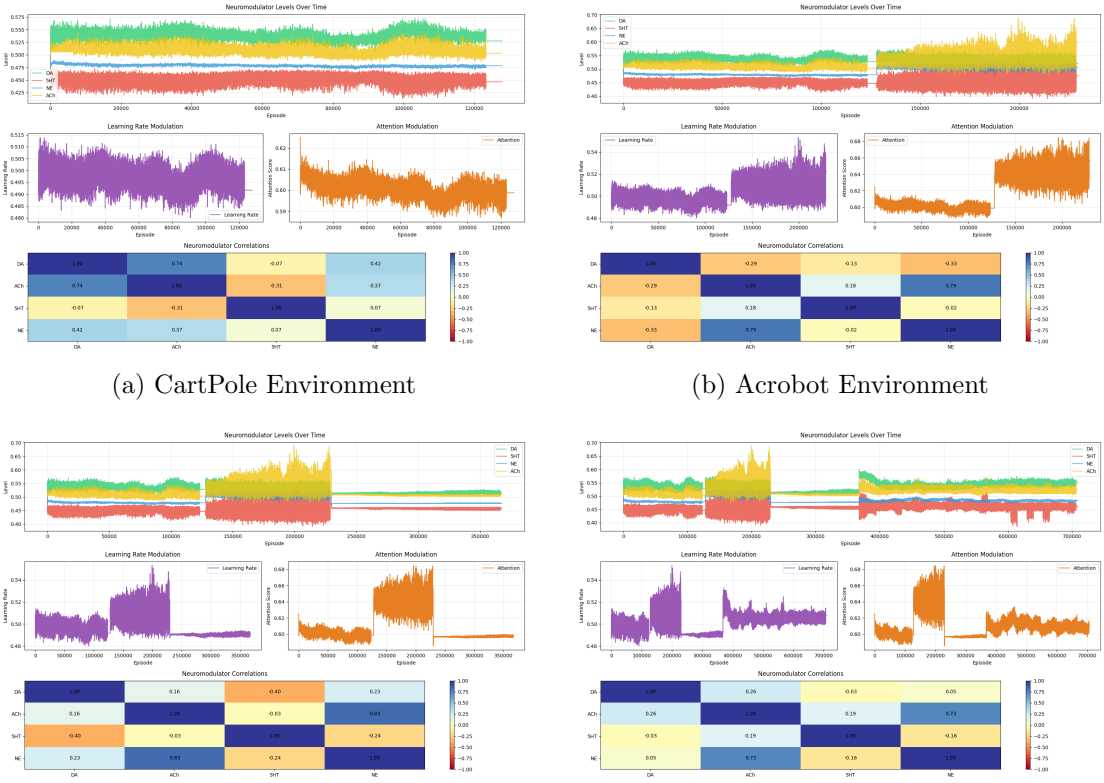
Analysis of neuromodulatory dynamics (Figure 3.5) during training reveals the coordination patterns that emerge from the interaction of the four modules. The temporal analysis of learning progress shows that the neuromodulatory framework provides benefits throughout the learning process, from initial skill acquisition through performance optimization and long-term maintenance. These patterns suggest that the framework develops environment-specific adaptation strategies that optimize learning for the particular characteristics of each task domain. The emergence of these coordinated patterns without explicit programming demonstrates the power of bio-inspired organizational principles for creating adaptive behaviors.

In simple tasks like CartPole-v1, neuromodulator levels remain relatively stable with minimal fluctuations once optimal control is achieved. The dopamine module shows slight increases during reward acquisition, while serotonin maintains consistent stability control. This reflects the predictable nature of the balancing task once basic control is mastered. Complex environments like LunarLander-v3 demonstrate pronounced fluctuations that mirror their challenging nature. The landing task requires continuous adaptation to varying spacecraft dynamics, leading to dynamic dopamine responses and norepinephrine adjustments for attention allocation between frozen columns. These fluctuations represent appropriate adaptive responses rather than instability, demonstrating the framework's ability to scale modulation intensity based on task demands.

3.6 Performance Retention Evaluation

One of the most critical aspects of sequential multi-task learning is the system's ability to maintain performance on previously learned tasks while acquiring new capabilities. Our framework's approach to this challenge combines the architectural benefits of Progressive Neural Networks with adaptive mechanisms that optimize knowledge transfer and minimize interference.

Table 3.5 presents the performance retention analysis comparing initial task performance to performance after learning all subsequent tasks. The analysis demonstrates that BIARM achieves superior knowledge preservation compared to baseline Progressive Neural Networks. The average retention rate of 98.0% across all



(a) CartPole Environment

(b) Acrobot Environment

(c) MountainCar Environment

(d) LunarLander Environment

Figure 3.5: **Neuromodulator Dynamics Analysis.** For each environment: (top) Temporal activation patterns of each neuromodulator, showing the evolution of DA, 5-HT, NE, and ACh levels over training; (middle-left) Learning rate modulation, demonstrating how the neuromodulators influence learning speed; (middle-right) Attention modulation, showing the dynamic focusing of learning; and (bottom) Correlation heatmap revealing interactions between neuromodulators.

tasks significantly exceeds the baseline rate of 93.2%, representing a 4.8 percentage point improvement in knowledge retention. Perfect retention was achieved in CartPole-v1 and LunarLander-v3, demonstrating that the adaptive mechanisms can completely eliminate catastrophic forgetting in both simple and complex tasks when appropriately coordinated.

Table 3.5: Initial vs Final Performance Comparison Between BIARM and Baseline Model

Environment	Model	Initial Performance	Final Performance	Retention
CartPole-v1	BIARM	500.00 ± 0.00	500.00 ± 0.00	**100%**
	Baseline	496.20 ± 5.31	491.00 ± 11.94	98.95%
Acrobot-v1	BIARM	-88.20 ± 3.84	-93.50 ± 18.91	**94.3%**
	Baseline	-86.80 ± 10.71	-94.50 ± 24.96	91.85%
MountainCar-v0	BIARM	-117.20 ± 29.84	-119.80 ± 28.02	**97.82%**
	Baseline	-150.60 ± 49.33	-165.90 ± 46.31	90.78%
LunarLander-v3	BIARM	233.37 ± 34.38	233.37 ± 34.38	**100%**
	Baseline	-380.88 ± 346.54	-417.17 ± 412.62	91.30%

3.7 Discussion and Implications

3.7.1 Biological Plausibility and Validation

The experimental results provide compelling evidence that bio-inspired adaptive rate modulation offers genuine advantages over fixed-parameter learning approaches in sequential multi-task scenarios. However, the analysis of these results reveals several important insights about when, why, and how these advantages emerge, providing guidance for future applications and improvements of the framework.

The success of our framework provides computational validation for several key principles from neuroscience. The hierarchical importance of different neuromodulatory systems revealed by our ablation studies aligns well with biological literature. Dopamine's critical role in learning rate modulation matches extensive research on its function in reward-based learning. Acetylcholine's importance for knowledge integration corresponds to its known roles in attention and memory consolidation. The structured dependency chains between modules, designed to reflect biological coordination principles, produce coherent adaptive behaviour across diverse environments, suggesting that even simplified implementations of neuromodulatory organisation can capture functionally meaningful aspects of biological learning regulation.

The substantial variance reduction, observed across all environments, is particularly important characteristic for practical robotic applications where consistent performance is often more valuable than occasionally exceptional performance accompanied by frequent failures. The achievement of zero variance in CartPole-v1 represents a particularly striking example of how adaptive mechanisms can enhance learning reliability.

3.7.2 Comparison with Alternative Approaches

Our approach shares conceptual similarities with meta-learning frameworks that seek to learn how to learn more effectively [107, 171, 79]. However, BIARM differs in its explicit biological grounding and its focus on coordinated adaptation rather than parameter optimization. While meta-learning approaches typically learn adaptation strategies through gradient-based optimization, our framework implements biologically-motivated coordination principles that emerge from the interaction of specialized modules.

This biological grounding provides several advantages. First, it offers principled guidance for system design grounded in established neuroscientific principles rather than purely empirical exploration.. Second, it naturally incorporates multi-timescale dynamics that are difficult to achieve through gradient-based approaches. Third, it provides interpretable mechanisms that can be analyzed and understood in terms of their biological counterparts, facilitating system debugging and improvement.

The comparison with other continual learning approaches reveals that the neuromodulatory framework provides complementary benefits that could enhance the effectiveness of alternative architectural and algorithmic solutions. BIARM can be viewed as complementary to rather than competitive with other continual learning methods. The framework's modulation mechanisms could potentially enhance the effectiveness of regularization-based methods like Elastic Weight Consolidation [18] by providing adaptive control over the strength and timing of regularization. Similarly, replay-based methods [104, 118] could benefit from neuromodulatory control over when and how stored experiences are utilized for learning.

3.7.3 Generalization and Future Extensions

The principles demonstrated in this work suggest several promising directions for future research. The coordination mechanisms could be extended to include additional neuromodulatory systems, such as histamine for arousal or GABA for inhibitory control. The temporal dynamics could be made more sophisticated by incorporating circadian rhythms or other biological timing mechanisms.

The framework could also be extended beyond sequential task learning to address other challenges in adaptive robotics. Multi-agent coordination, long-horizon planning, and dynamic environment adaptation all present scenarios where coordination between multiple adaptive mechanisms might provide significant benefits.

3.8 Conclusion

This chapter has presented a comprehensive bio-inspired adaptive rate modulation framework that successfully translates the organizational principles of biological neuromodulatory systems into practical computational mechanisms for sequential multi-task learning. The framework demonstrates that coordination between multiple adaptive mechanisms, inspired by dopamine, serotonin, norepinephrine, and acetylcholine systems, provides substantial performance improvements over fixed-parameter approaches across diverse learning scenarios.

Key Contributions

1. **Theoretical Framework**: A principled approach to coordinating multiple adaptive mechanisms based on biological neuromodulation principles.

2. **Mathematical Formulation**: Precise specifications for implementing coordinated neuromodulatory control in artificial systems, including:
   - State dynamics with multi-timescale adaptation
   - Target computation through specialized neural networks
   - Coordinated learning rate and attention modulation
   - Integration with Progressive Neural Networks

3. **Empirical Validation**: Comprehensive experimental results showing consistent improvements ranging from 1.06% to 155.94% across four challenging environments.

4. **Mechanistic Understanding**: Detailed ablation studies confirming that performance improvements stem from fundamental biological principles rather than superficial implementation details.

The experimental validation demonstrates that the framework provides the greatest benefits in scenarios requiring coordination between multiple learning objectives, enhanced learning reliability, effective knowledge transfer between tasks, and substantially reduced variance.

The success of this bio-inspired approach establishes a foundation for the next contribution of this thesis, which explores how biological principles from gene regulatory networks can provide complementary improvements in learning efficiency and robustness for robotic control applications. The coordination mechanisms developed here demonstrate the power of biological inspiration for creating adaptive systems, setting the stage for investigating how other biological control architectures might further advance the field of developmental evolutionary robotics.

CHAPTER 4

GENE REGULATORY NETWORKS FOR VISION-BASED ROBOT CONTROL

4.1 Introduction and Biological Motivation

This chapter discusses the research article: "Gene Regulatory Networks for Enhanced Vision-Based Robot Control: A Bio-Inspired Approach", MDPI Sensors 26(6), 1742, 2026 [38].

While Chapter 3 demonstrated how neuromodulatory principles enable adaptive multi-task learning, this chapter explores a fundamentally different biological control paradigm. Gene Regulatory Networks (GRNs), introduced in Section 1.1, can offer distinct advantages for vision-based robotic control.

Recent work in vision-based robot control has demonstrated that systems can learn to grasp diverse objects using only RGB images [14, 172], but the training requirements remain substantial. State-of-the-art approaches like QT-Opt achieved 96% grasp success rates but required distributed training on seven robots for 800 hours, highlighting the computational resources needed for competitive performance [172].

This chapter presents the first comprehensive application of Gene Regulatory Networks to vision-based robot control, achieving a 57.5% success rate on the challenging KukaDiverseObjectEnv benchmark while reducing training time by 13.7× compared to established baselines, demonstrating that biological regulatory principles can improve training efficiency while maintaining competitive performance..

4.2 Mathematical Formulation and System Architecture

We illustrate in this section how GRNs are applied to vision-based robot control, giving the mathematical formulation, system architecture, and optimization framework. We begin by describing the core dynamics that govern gene expression, including the mapping from sensory inputs to regulatory signals and the generation of motor commands from the evolved gene expression patterns.

4.2.1 Discrete-Time Gene Expression Dynamics

Our computational implementation of GRNs adapts the continuous-time dynamics of biological gene regulation to discrete-time updates suitable for robotic control applications. The core dynamics are governed by:

$$x_i(t+1) = x_i(t) + \Delta t \cdot [f_i(s,x) - \delta_i x_i(t) + \eta_i(t)] \tag{4.1}$$

where $x_i(t) \in [0.0, 1.0]$ represents the expression level of gene i at discrete time step t, and the gene expression vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ captures the current state of the regulatory network. The discrete timestep $\Delta t = 0.05$ provides temporal resolution sufficient for robotic control while ensuring numerical stability.

The sensory input vector $\mathbf{s} \in \mathbb{R}^8$ combines visual and proprioceptive information, where the first four dimensions represent processed visual features extracted from RGB camera input, and the remaining four dimensions encode proprioceptive feedback from robot joint states. This multimodal sensory integration enables the GRN to respond to both global environmental context and local robot state information.

The activation function $f_i(\mathbf{s}, \mathbf{x})$ determines how gene i responds to environmental stimuli and regulatory influences from other genes, implementing the core logic of the regulatory network:

$$f_i(\mathbf{s}, \mathbf{x}) = H\left(\sum_{j=1}^{8} w_{ji}^{(s)} s_j + \sum_{k=1}^{n} w_{ki}^{(r)} x_k \cdot \alpha_k, \theta_i\right) \tag{4.2}$$

The sensor-to-gene connection weights $w_{ji}^{(s)}$ determine how strongly each sensory input influences gene i, allowing the network to develop specialized responses to different types of environmental information. The regulatory interaction matrix $w_{ki}^{(r)}$ encodes gene-gene interactions, where positive values represent activation and

negative values represent repression, creating the complex regulatory relationships that drive network behavior.

4.2.2 Hill Function Activation and Regulatory Logic

The Hill function $H(u, \theta)$ implements the sigmoidal activation characteristic of biological gene regulation, capturing the cooperative binding behavior observed in transcriptional control:

$$H(u, \theta) = \frac{u^2}{\theta^2 + u^2} \cdot \text{sign}(u) \tag{4.3}$$

This modification of the standard Hill function enables bidirectional regulation while maintaining the characteristic sigmoid response profile. The squared terms create sigmoidal activation curves with adjustable sensitivity through the threshold parameter $\theta_i \in [0.1, 1.0]$, where lower values make genes more responsive to weak regulatory signals. The sign function preserves the activating or repressing nature of the regulatory input, allowing genes to both promote and inhibit the expression of other genes.

The interaction strength parameters $\alpha_k \in [0.1, 3.0]$ provide an additional layer of regulatory control, modulating the influence of gene k on other genes in the network. This parameter enables the evolution of regulatory hierarchies where some genes serve as master regulators with strong influence over many targets, while others function as specialized effectors with limited regulatory scope. Each gene has an associated decay rate $\delta_i \in [0.05, 1.0]$ that models the natural degradation of gene products in biological systems, providing temporal dynamics that enable the network to implement sequential behaviors and temporal reasoning without explicit memory modules.

4.2.3 Biological Noise and Robustness

Biological noise $\eta_i(t) \sim \mathcal{N}(0, 0.01^2)$ introduces stochasticity that serves multiple important functions in the system. In biological systems, molecular noise is not merely a nuisance to be filtered out, but plays essential roles in state-space exploration, preventing convergence to suboptimal solutions, and enabling robust control strategies that work across varying conditions [51, 173]. The noise level is calibrated to match the signal-to-noise ratios observed in biological gene expression systems, where fluctuations typically represent 10–20% of mean expression levels. The clipping operation bounds expression levels to the unit interval $[0.0, 1.0]$, preventing numerical instabilities while maintaining biological plausibility.

4.2.4 Motor Command Generation

Motor commands are generated through a linear combination of gene expression levels that maps the high-dimensional gene expression space to the robot's actuator space:

$$\mathbf{a}(t) = \tanh(W^{(m)}\mathbf{x}(t)) \tag{4.4}$$

The motor command vector $\mathbf{a}(t) \in [-1, 1]^7$ provides control signals for the seven degrees of freedom of the robotic manipulator. The motor output weight matrix $W^{(m)} \in \mathbb{R}^{7\times 50}$ maps the 50-dimensional gene expression space to the 7-dimensional motor command space. This linear mapping enables complex, coordinated motor behaviors to emerge from distributed gene expression patterns. Unlike traditional approaches that require explicit coordination mechanisms, the GRN naturally generates coordinated control signals through the regulatory relationships between genes. The hyperbolic tangent activation function ensures that motor commands remain bounded within the acceptable range for joint control, preventing damage to the robotic system while providing smooth, continuous control signals.

4.3 Four-Stage Architecture and Visual Processing

Figure 4.1 illustrates the four-stage GRN architecture. Stage 1 processes RGB images through a lightweight CNN, extracting 4-dimensional visual features. Stage 2 integrates visual features with 4-dimensional proprioceptive feedback to create an 8-dimensional sensor input vector. Stage 3 implements a core gene regulatory network with 50 genes that evolves during optimization. Stage 4 transforms the gene expression vector into 7-dimensional motor commands through the learned motor output weight matrix. The subsections below provide detailed explanations of each stage.

4.3.1 Stage 1: Lightweight Visual Feature Extraction

The first stage implements a lightweight convolutional neural network designed specifically for computational efficiency while maintaining sufficient representational power for manipulation tasks. Processing $48 \times 48 \times 3$ RGB images through two convolutional layers, the system extracts essential visual features without the computational overhead of deeper architectures.

The first convolutional layer transforms the 3-channel RGB input to 16 feature channels using 8×8 kernels with stride 4, capturing large-scale spatial patterns

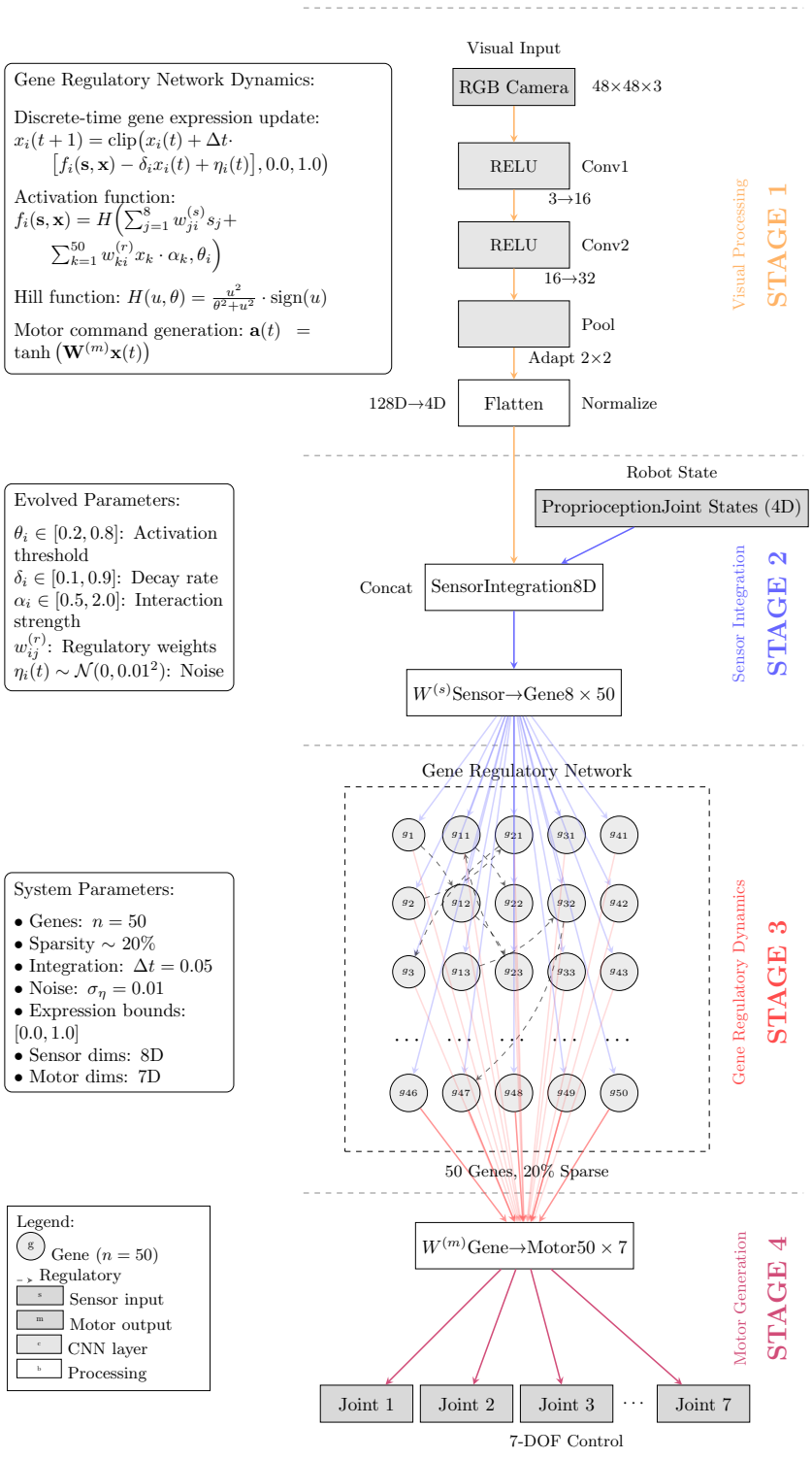


Figure 4.1: **Gene regulatory network architecture.** Four-stage vision-based robot control system: (1) visual processing, (2) sensor integration, (3) gene regulatory dynamics, and (4) motor command generation.

relevant for object detection and localization. The second layer expands to 32 channels using 4×4 kernels with stride 2, refining the feature representation to capture more detailed spatial relationships necessary for grasp planning.

Adaptive pooling reduces the spatial dimensions to a fixed 2×2 grid regardless of input variations, ensuring consistent feature dimensionality while preserving spatial relationships. The final flattening operation produces a 128-dimensional feature vector that is normalized and reduced to 4 dimensions through learned linear projection, providing compact visual representation suitable for GRN processing. This architecture balances computational efficiency with representational power, requiring only a fraction of the parameters and computation needed by typical deep vision systems while extracting sufficient information for successful manipulation [174].

4.3.2 Stage 2: Multimodal Sensor Integration

The second stage integrates visual features with proprioceptive feedback to create a comprehensive sensory representation that captures both environmental context and robot state. The 4-dimensional visual feature vector is concatenated with 4-dimensional proprioceptive information including joint positions, creating an 8-dimensional sensor input vector $\mathbf{s}$.

This multimodal integration is crucial for successful manipulation, as visual information alone is insufficient for precise motor control. The sensor integration stage also implements normalization and scaling to ensure that visual and proprioceptive signals contribute appropriately to gene regulation.

4.3.3 Stage 3: Gene Regulatory Network Dynamics

The third stage implements the core GRN computation, where 50 interconnected genes exhibit complex regulatory dynamics that process sensory information and generate appropriate control responses. The choice of 50 genes provides sufficient complexity for manipulation tasks while maintaining computational tractability for evolutionary optimization. The regulatory network exhibits $\sim 20\%$ sparse connectivity (to be evolved), meaning that approximately 525 of the possible 2,500 gene-gene connections are active.

The gene expression update process implements the discrete-time dynamics described in Equation 4.1, where each gene's expression level evolves based on sensory inputs, regulatory influences from other genes, decay processes, and biological noise. This parallel update mechanism enables the network to process multiple

sensory inputs simultaneously while generating coordinated responses through regulatory coupling.

4.3.4 Stage 4: Motor Command Generation

The fourth stage transforms the 50-dimensional gene expression vector into 7-dimensional motor commands through the learned motor output weight matrix (Equation 4.4). This transformation enables complex motor behaviors to emerge from the distributed pattern of gene expression, where different expression patterns correspond to different motor behaviors. The linear mapping allows for both direct gene-to-motor relationships and more complex combinatorial relationships where multiple genes contribute to each motor command, enabling the evolution of hierarchical control architectures.

Algorithm 2 details the motor command generation process implemented in our GRN controller. The algorithm processes visual inputs through the CNN encoder, integrates proprioceptive feedback, updates gene expression levels according to the regulatory dynamics, and generates motor commands.

Algorithm 2 GRN Controller Motor Command Generation

Require: Evolved genome $\mathcal{G} = \{W^{(s)}, W^{(r)}, W^{(m)}, \boldsymbol{\theta}, \boldsymbol{\delta}, \boldsymbol{\alpha}\}$
Require: Visual encoder CNN, initial gene expression $\mathbf{x}(0)$
Initialize $\mathbf{x} \leftarrow \mathbf{x}(0)$
while robot episode active **do**
  $I_{\text{RGB}} \leftarrow \text{CaptureRGBImage}()$
  $\mathbf{v} \leftarrow \text{CNNExtractFeatures}(I_{\text{RGB}})$
  $\mathbf{p} \leftarrow \text{GetProprioception}()$
  $\mathbf{s} \leftarrow [\mathbf{v}; \mathbf{p}]$ ▷ Concatenate visual and proprioceptive
  for $i = 1$ to n **do**
    $u_i \leftarrow \sum_{j=1}^{8} w_{ji}^{(s)} s_j + \sum_{k=1}^{n} w_{ki}^{(r)} x_k \alpha_k$
    $f_i \leftarrow H(u_i, \theta_i)$
    $\eta_i \leftarrow \mathcal{N}(0, 0.01^2)$
    $\Delta x_i \leftarrow f_i - \delta_i x_i + \eta_i$
    $x_i^{\text{new}} \leftarrow \text{clip}(x_i + \Delta t \cdot \Delta x_i, 0.0, 1.0)$
  end for
  $\mathbf{x} \leftarrow \mathbf{x}^{\text{new}}$
  $\mathbf{a}_{\text{raw}} \leftarrow W^{(m)}\mathbf{x}$
  $\mathbf{a} \leftarrow \tanh(\mathbf{a}_{\text{raw}})$
  $\text{ExecuteMotorCommands}(\mathbf{a})$
end while

4.4 Evolutionary Optimization Framework

4.4.1 Genome Representation and Encoding

The evolutionary optimization framework treats each GRN controller as a genome encoding the complete regulatory architecture required for vision-based manipulation. Each genome $\mathcal{G}$ contains six parameter matrices and vectors:

$$\mathcal{G} = \{W^{(s)}, W^{(r)}, W^{(m)}, \boldsymbol{\theta}, \boldsymbol{\delta}, \boldsymbol{\alpha}\} \tag{4.5}$$

The sensor-to-gene connection weight matrix $W^{(s)} \in \mathbb{R}^{8\times 50}$ determines how sensory inputs influence gene expression, enabling the network to develop specialized responses to different types of environmental information. The gene regulatory interaction matrix $W^{(r)} \in \mathbb{R}^{50\times 50}$ encodes the complex web of regulatory relationships between genes, implementing the core logic of the control system. The gene-to-motor connection weight matrix $W^{(m)} \in \mathbb{R}^{50\times 7}$ maps gene expression patterns to motor commands. The activation threshold vector $\boldsymbol{\theta} \in \mathbb{R}^{50}$ sets the sensitivity of each gene to regulatory inputs, the decay rate vector $\boldsymbol{\delta} \in \mathbb{R}^{50}$ controls temporal dynamics, and the interaction strength vector $\boldsymbol{\alpha} \in \mathbb{R}^{50}$ modulates regulatory influence.

4.4.2 Population Initialization and Diversity

The evolutionary algorithm maintains a population of $N_{\text{pop}} = 12$ genomes, balancing computational efficiency with sufficient diversity for effective search. This relatively small population size is enabled by the constrained search space imposed by biological bounds. Initial genomes are sampled from biologically motivated distributions that reflect the statistical properties observed in natural GRNs.

Connection weights are drawn from normal distributions: $W^{(s)} \sim \mathcal{N}(0, 0.3^2)$, $W^{(r)} \sim \mathcal{N}(0, 0.1^2)$ with $\sim 20\%$ evolved sparsity, and $W^{(m)} \sim \mathcal{N}(0, 0.2^2)$. Gene-specific parameters use uniform distributions: $\theta_i \sim \mathcal{U}(0.2, 0.8)$, $\delta_i \sim \mathcal{U}(0.1, 0.9)$, and $\alpha_i \sim \mathcal{U}(0.5, 2.0)$.

The sparsity constraint in the regulatory matrix is implemented by randomly setting $\sim 80\%$ of connections to zero during initialization, then allowing evolutionary processes to modify connection patterns through structural mutations. This approach ensures that the search begins with biologically plausible connectivity patterns while allowing evolution to discover optimal regulatory architectures.

4.4.3 Fitness Evaluation and Multi-Objective Optimization

The fitness function combines multiple objectives to encourage both task performance and behavioral robustness:

$$F(\mathcal{G}) = \bar{R} + 10 \cdot S_r + 2 \cdot C + 1 \cdot \min(S_r, 0.8) \tag{4.6}$$

The average episode reward $\bar{R}$ provides a baseline measure of task performance, capturing the system's ability to achieve manipulation objectives. The success rate S_r represents the fraction of episodes resulting in successful object grasping and receives high weighting to prioritize task completion. The consistency measure $C = 1/(1 + \sigma_R)$ is based on the standard deviation of episode rewards and encourages stable, reproducible performance. The final term $\min(S_r, 0.8)$ provides additional reward for success rates up to 80%, after which the marginal benefit decreases to prevent overfitting.

4.4.4 Robust Training Protocol

To improve training efficiency while maintaining robustness, we employ a dual evaluation strategy that balances computational speed with generalization capability. During fitness evaluation, 70% of episodes occur under normal conditions to assess basic task performance, while 30% involve Gaussian noise $\mathcal{N}(0, 0.03^2)$ added to visual features to test robustness. This approach prevents overfitting to clean sensory conditions, encourages the evolution of robust control strategies that work under uncertainty, and maintains computational efficiency by limiting the fraction of noisy evaluations.

4.4.5 Evolutionary Operators and Mutation Strategies

The evolutionary process employs three distinct types of mutations that collectively enable exploration of both network structure and parameters while maintaining biological plausibility:

Structural Mutations Structural mutations occur with 10% probability and modify the regulatory connectivity pattern by adding or removing connections between genes, maintaining the sparse connectivity characteristic of biological GRNs while allowing evolution to discover optimal regulatory architectures.

Weight Mutations Weight mutations apply Gaussian perturbations with 20% probability to connection weights in all three weight matrices. The perturbation magnitude (standard deviation 0.1) is calibrated to enable gradual exploration while avoiding disruptive changes that could destroy functional regulatory relationships.

Parameter Mutations Parameter mutations modify gene-specific properties with 20% probability, adjusting activation thresholds, decay rates, and interaction strengths through constrained Gaussian perturbations. The biological bounds are strictly enforced to maintain parameter values within ranges observed in natural systems.

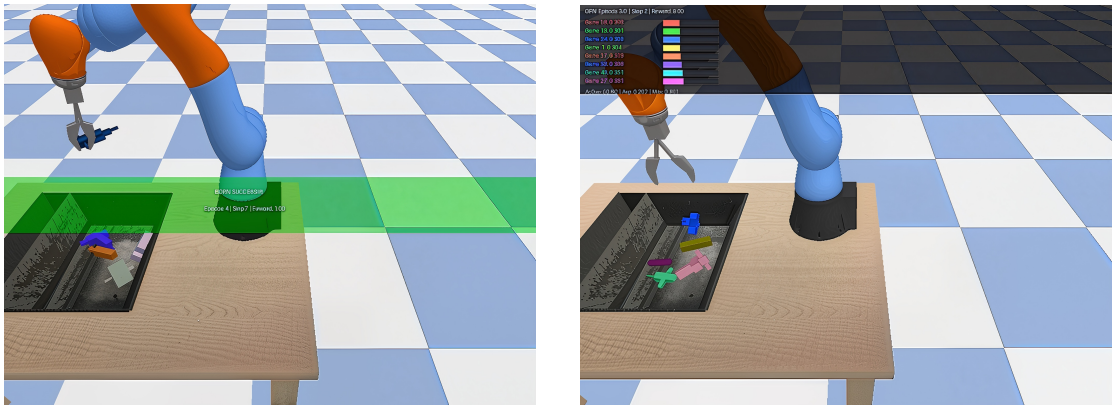
The evolutionary algorithm presented in Algorithm 3 employs elitism, preserving the top 25% of performers across generations, while the remaining positions are filled by mutated offspring generated from the elite population. Early stopping is triggered when a controller achieves 70% success rate during training and maintains at least 50% success rate during robust validation with 20 episodes, preventing unnecessary computation while ensuring controller quality.

Algorithm 3 GRN Evolution Algorithm

Require: Population size N_{pop}, max generations $G_{\max}$, elite fraction f_e
Initialize population $\mathcal{P}_0$ with N_{pop} random genomes
for $g = 1$ to $G_{\max}$ **do**
    for each genome $\mathcal{G}_i \in \mathcal{P}_g$ **do**
        $F_i \leftarrow \text{EvaluateGenome}(\mathcal{G}_i)$
    end for
    Sort population by fitness: $F_1 \geq F_2 \geq \cdots \geq F_{N_{\text{pop}}}$
    if $S_r(\mathcal{G}_1) \geq 0.7$ **then**
        Perform robustness test on $\mathcal{G}_1$
        if robust performance confirmed **then**
            break
        end if
    end if
    $N_e \leftarrow \lfloor f_e \cdot N_{\text{pop}} \rfloor$
    $\mathcal{P}_{g+1} \leftarrow \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_{N_e}\}$
    while $|\mathcal{P}_{g+1}| < N_{\text{pop}}$ **do**
        $\mathcal{G}_{\text{parent}} \leftarrow \text{SelectElite}(\mathcal{P}_{g+1})$
        $\mathcal{G}_{\text{offspring}} \leftarrow \text{Mutate}(\mathcal{G}_{\text{parent}})$
        $\mathcal{P}_{g+1} \leftarrow \mathcal{P}_{g+1} \cup \{\mathcal{G}_{\text{offspring}}\}$
    end while
end for
return Best genome $\mathcal{G}_1$

4.5 Experimental Setup and Baseline Comparisons

Experiments used Python 3.12 with PyTorch 2.8.0+cu126 and PyBullet 3.2.7 on Google Colab's free tier. The evolutionary optimization used tournament selection (size three), elitism (top 25%), and Gaussian mutation. Population size (12 individuals) and generation limit (15) ensured reasonable training times within computational constraints. The evaluation was conducted in the KukaDiverseObjectEnv from the Gymnasium RL environment suite (Gymnasium v1.2.0; Farama Foundation) [39], a standardized benchmark for vision-based robotic grasping (Figure 4.2).



(a) Successful grasp (b) Failed grasp

Figure 4.2: **KukaDiverseObjectEnv benchmark.** (a) Successful object lift above 0.2 m threshold using RGB visual input. (b) Failed manipulation attempt. The environment provides sparse rewards and diverse object configurations with over 1000 object types.

4.5.1 KukaDiverseObjectEnv Benchmark

KukaDiverseObjectEnv from PyBullet is a standardized benchmark for vision-based robotic grasping that provides challenging and realistic manipulation scenarios. The environment simulates a 7-DOF KUKA iiwa manipulator performing grasps on objects with varying shapes, sizes, and materials positioned at diverse locations within the workspace.

Visual observations consist of $48 \times 48 \times 3$ RGB images from a fixed camera viewpoint, providing limited resolution that requires efficient visual processing. The action space consists of continuous values in $[-1, 1]$ for each joint representing desired velocities.

Episodes are limited to 30 timesteps with a sparse reward structure that provides +1.0 reward only when objects are lifted above a 0.2 m threshold. No intermediate rewards or shaping signals are provided, creating a challenging exploration problem. The environment includes over 1000 diverse object types with varying grasping difficulties and colors, from simple geometric shapes to complex irregular objects, with random positioning within the robot's workspace requiring controllers to generalize across different spatial configurations.

4.5.2 Baseline Method Configurations

The proposed method was compared against four baseline approaches representing distinct control paradigms, with detailed configurations summarized in Table 4.1.

Table 4.1: Baseline method configurations.

Method	**Type**	**Key Parameters**	**Architecture**
PPO	On-policy RL	$\beta = 0.01$, $\gamma = 0.993$ $\epsilon = 0.07$, LR=2×10^{-4}	Conv: $3 \to 16 \to 32 \to 32$ FC: $128 \to 64$
NEAT	Evolutionary	Pop=12, Gen=15 Mutation: 80% weights	Evolving topology Input: 128D→Output: 7D
TD3	Off-policy RL	Buffer=50K, Batch=128 LR: 1e-4 (actor), 1e-3 (critic)	Conv: $3 \to 16 \to 32$ FC: $256 \to 256$
IPG+HER Standard RL	Hybrid RL	Success rates: $25 - 45\%$ (Kumar et al. [175])	Literature baseline Various architectures

Proximal Policy Optimization (PPO) employs an actor–critic architecture with a clipped surrogate objective and entropy regularization. NeuroEvolution of Augmenting Topologies (NEAT) evolves both network topology and weights through a population-based search. Twin Delayed Deep Deterministic Policy Gradient (TD3) employs twin critic networks with experience replay for off-policy learning. Literature baselines from Kumar et al. [175] established performance ranges for well-tuned reinforcement learning approaches on the same benchmark environment.

4.5.3 Evaluation Protocol

The evaluation protocol ensures fair comparison through standardized procedures and comprehensive statistical analysis. Each method receives identical computational budgets with 15 generations for evolutionary approaches and equivalent

training time for reinforcement learning methods. Following training, standardized testing utilizes fixed random seeds for reproducible evaluation across 40 test episodes for the GRN approach. Bootstrap confidence intervals provide statistical comparison with appropriate correction for multiple comparisons. Robustness assessment introduces Gaussian noise (standard deviation 0.03) to visual features during evaluation, testing controller performance under realistic sensory uncertainty.

4.6 Experimental Results and Performance Analysis

4.6.1 Comparative Performance Results

Table 4.2 presents performance metrics for all evaluated methods. Efficiency represents success rate divided by training time (percentage points per minute). PPO results used the repository implementation [176], while IPG+HER and Standard RL were literature baselines from Kumar et al. [175].

Table 4.2: Comprehensive Performance Comparison on KukaDiverseObjectEnv Benchmark

Method	Success Rate	Training Time	Efficiency	Test Episodes
GRN (Ours)	**57.5%**	**6.5 min**	**8.85**	40
PPO (Repository)	50.0%	89.0 min	0.56	10
NEAT	33.3%	47.4 min	0.70	30
TD3	10.0%	70.0 min	0.14	40
IPG+HER	∼45%	N/A	N/A	N/A
Standard RL	∼25%	N/A	N/A	N/A

The experimental results demonstrate that the GRN approach achieves superior performance while requiring less training time than conventional methods. The GRN approach achieved a 57.5% success rate while requiring only 6.5 minutes of training time, representing substantial improvements over all baseline methods. The 7.5 percentage point improvement over PPO (50.0%) is achieved with 13.7× faster training, demonstrating both superior final performance and improved efficiency.

The efficiency improvement is particularly striking, with the GRN method achieving 8.85 percentage points per minute compared to 0.56 for PPO, representing a 1,480% improvement in training efficiency. This improvement stems from the constrained search space imposed by biological constraints, which focuses evolutionary

optimization on promising regions of the parameter space. NEAT achieved 33.3% success rate with 47.4 minutes of training, while TD3 achieved only 10.0% success rate despite 70.0 minutes of training, highlighting the challenges that traditional methods face with sparse rewards and high-dimensional continuous control.

4.6.2 Training Dynamics and Convergence

Figure 4.3 presents a comprehensive analysis of the evolved GRN controller, revealing characteristics contributing to performance advantages: evolution progress showing fitness improvement over six generations with rapid convergence, success rate evolution demonstrating 80% peak performance during training with a final test result of 57.5%, and network complexity evolution showing growth in regulatory connections from 400 to 1600 connections.

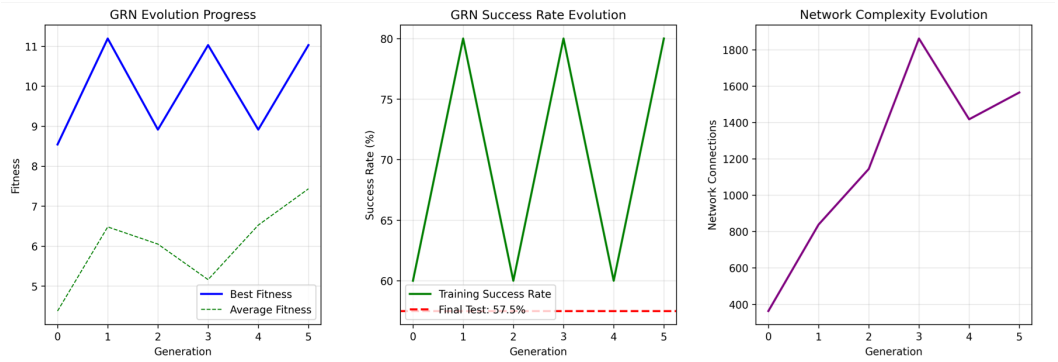


Figure 4.3: **Gene regulatory network analysis.** Evolution progress, success rates, and network complexity over training generations. Left panel shows fitness evolution across 6 generations with rapid initial improvement. Middle panel displays success rate progression peaking at 80% during training. Right panel illustrates regulatory network growth from sparse initialization to evolved complexity of ∼1600 active connections.

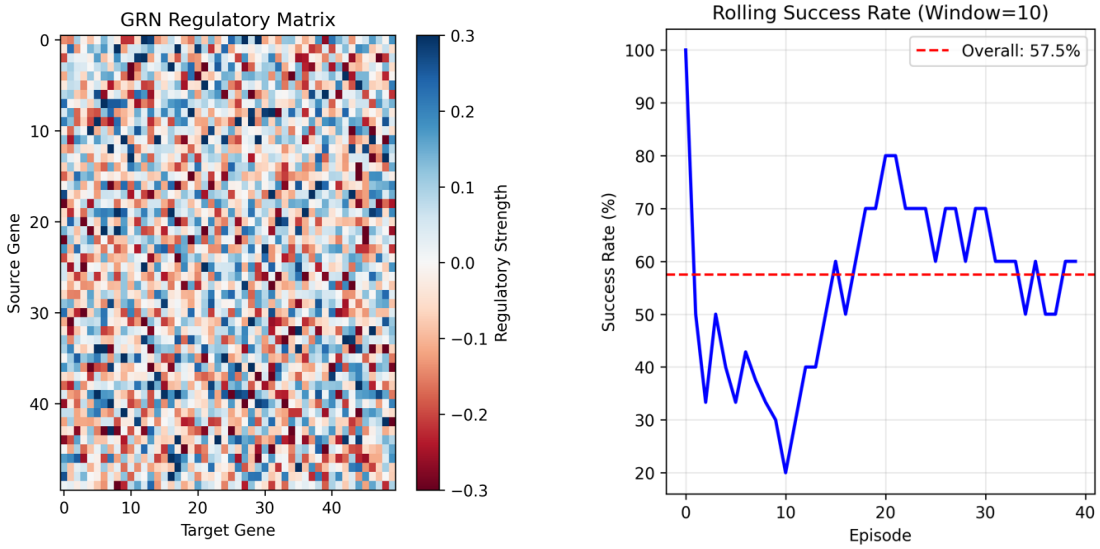
The rapid convergence reflects the effectiveness of biological constraints in guiding optimization. Unlike unconstrained neural evolution that must explore vast parameter spaces, the GRN approach benefits from bounded dynamics, sparse connectivity, and other biological principles that eliminate many suboptimal solutions from consideration. This focused search enables the algorithm to discover effective controllers with far fewer evaluations than traditional approaches require.

4.6.3 Robustness and Noise Tolerance

The robustness evaluation reveals that GRN controllers maintain 91.8% of their clean performance when Gaussian noise is added to visual features. This represents exceptionally strong resilience to sensory perturbations, with performance dropping only from 57.5% to 52.8% under noisy conditions. This robustness stems from several biological principles incorporated in the GRN architecture: The bounded gene expression levels prevent extreme responses to noisy inputs, while the biological noise inherent in the system dynamics provides natural filtering of external perturbations.

4.6.4 Regulatory Matrix and Evaluation Test

The evolved regulatory matrix (Figure 4.4a) demonstrates adaptive connectivity, starting from sparse initialization and evolving to ∼1600 active connections (64% of total) as optimization balances network complexity with performance demands (Figure 4.3). The heatmap reveals both positive (blue) and negative (red) regulatory interactions distributed throughout the network, demonstrating the evolution of complex regulatory relationships essential for coordinated motor control.



(a) Regulatory matrix heatmap

(b) Rolling success rate

Figure 4.4: **Regulatory matrix and rolling success rate results.** (a) Regulatory matrix heatmap of the final evolved gene network with both activating (blue) and repressing (red) interactions. (b) Rolling success rate analysis over 40 test episodes showing performance consistency with 57.5% overall success rate.

The rolling success rate analysis (Figure 4.4b) demonstrated consistency of the evolved controller across 40 test episodes, validating robustness through relatively stable performance throughout testing episodes.

4.7 Ablation Study

To validate the individual contributions of GRN components, nine architectural ablations were systematically applied to the best-evolved genome, each isolating a specific biological mechanism. Figure 4.5 presents the results, confirming that biological mechanisms provide substantial and distinct performance contributions.

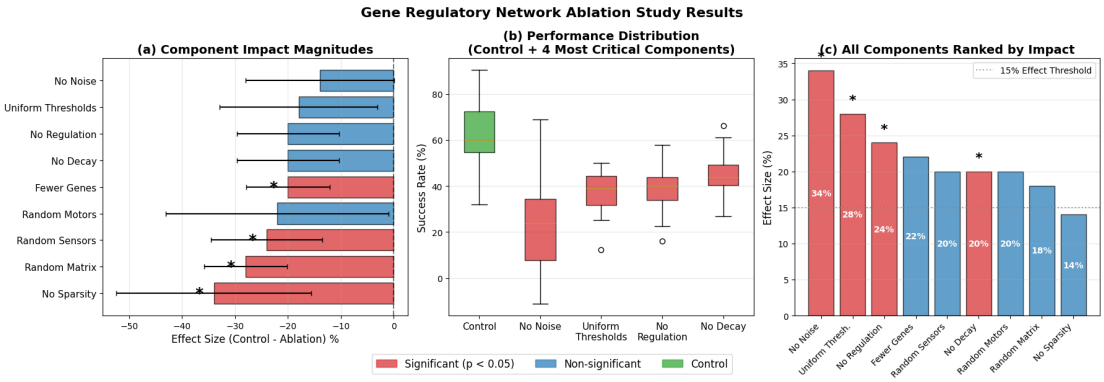


Figure 4.5: **GRN ablation study results.** (a) Effect sizes for all nine ablated components. (b) Performance distribution for control and the four most critical components. (c) Complete ranking of component importance. Asterisks indicate statistical significance.

No Noise (34% degradation): the stochastic term $\eta_i(t)$ was removed by setting `noise_level = 0.0`, eliminating all biological noise from gene dynamics. This produced the largest degradation, confirming that molecular stochasticity is not merely a nuisance but serves an essential functional role in state-space exploration and in preventing premature convergence to suboptimal regulatory patterns.

Uniform Thresholds (28% degradation): all evolved activation thresholds were replaced with a uniform value ($\theta_i = 0.5$ for all genes), removing the specialisation encoded in threshold diversity. The resulting degradation confirms that different genes rely on distinct sensitivity levels to respond selectively to different sensory conditions and regulatory states, and that this specialisation is critical to coordinated motor output.

No Regulation (24% degradation): all regulatory interaction weights were set to zero ($w_{ki}^{(r)} = 0$), effectively reducing the GRN to a set of independent genes

with no inter-gene communication. This confirmed that the evolved regulatory structure is responsible for the coordinated gene expression patterns necessary for complex sequential manipulation behaviours.

Fewer Genes (22% degradation): the network was reduced from 50 to 25 genes by truncating all parameter matrices accordingly, including the regulatory matrix, sensor weights, motor weights, activation thresholds, decay rates, and interaction strengths. The significant degradation indicates that a minimum level of gene diversity is required to sustain the sensory-motor mappings demanded by the manipulation task.

Random Sensors (20% degradation): the evolved sensor-to-gene weight matrix $W^{(s)}$ was replaced with random values drawn from $\mathcal{N}(0, 0.3^2)$, destroying the specialised sensory processing patterns developed during evolution. This confirms that the evolved sensor weights encode task-relevant perceptual structure that cannot be substituted by arbitrary connectivity.

No Decay (20% degradation): all gene decay rates were fixed to zero ($\delta_i = 0$), removing the dissipative dynamics that enable natural temporal sequencing in gene expression. Without decay, genes accumulate activation indefinitely, losing the multi-timescale temporal memory that underpins sequential control behaviour.

Random Motors (20% degradation): the evolved gene-to-motor weight matrix $W^{(m)}$ was replaced with random values drawn from $\mathcal{N}(0, 0.2^2)$, severing the learned mapping from gene expression patterns to motor commands. The resulting degradation confirms that the motor output layer encodes evolved coordination structure rather than functioning as a generic linear readout.

Random Matrix (18% degradation): the regulatory interaction matrix was randomised while preserving the original sparsity pattern of the evolved genome, replacing learned regulatory relationships with arbitrary ones. This isolates the contribution of the specific regulatory structure independently of connectivity density, confirming that the identity of regulatory interactions matters beyond sparsity alone.

No Sparsity (14% degradation): the regulatory matrix was replaced with a dense random initialisation, removing the evolved sparse connectivity pattern. The comparatively small degradation indicates that while sparse connectivity contributes to computational efficiency and robustness, a dense network can partially preserve performance at higher computational cost. Taken together, these results validate that the performance improvements of the GRN controller stem from fundamental biological principles rather than superficial implementation details, and establish a clear hierarchy of component importance for future architectural design decisions.

4.8 Discussion

4.8.1 Biological Constraints as Optimization Guides

The success of our GRN approach demonstrates a counterintuitive principle: carefully chosen constraints can enhance rather than limit optimization effectiveness. Traditional machine learning approaches typically seek to maximize flexibility and minimize assumptions, operating under the belief that fewer constraints lead to better solutions. Our results suggest that this assumption may be fundamentally flawed for certain classes of problems.

Biological systems have evolved under severe resource constraints and environmental uncertainties, yet they consistently demonstrate remarkable efficiency and robustness. Rather than viewing these constraints as limitations overcome through clever engineering, our work suggests that the constraints themselves serve as powerful optimization guides that focus search processes on solutions that are both effective and robust. The bounded dynamics of gene expression prevent the parameter explosion problems that can destabilize neural network training, while sparse connectivity reduces the dimensionality of the search space without sacrificing representational power [31].

This principle has profound implications for machine learning and optimization more broadly. Rather than seeking ever-more-flexible architectures, researchers might benefit from identifying and implementing constraints that have been validated through biological evolution. The challenge lies in distinguishing between constraints that genuinely enhance optimization and those that merely limit flexibility.

4.8.2 Temporal Dynamics Without Explicit Architecture

One of the most striking aspects of the GRN approach is its ability to implement sophisticated temporal reasoning and memory functions without explicit recurrent connections or memory modules. The temporal dynamics emerge naturally from the interplay between gene activation, regulatory interactions, and molecular decay, creating natural filters and integrators that enable complex sequential behaviors.

This natural temporal processing offers several advantages over explicit memory architectures. It avoids the stability problems often associated with recurrent neural networks, where gradient flow through time can lead to vanishing or exploding gradients. It provides multiple timescales naturally through different decay rates,

enabling the system to integrate information across different temporal horizons simultaneously.

The temporal dynamics are inherently stable due to the bounded nature of gene expression and the dissipative effects of decay processes. Unlike unbounded neural activations that can grow without limit, gene expression levels naturally return to baseline in the absence of driving inputs, providing inherent stability that prevents runaway dynamics. The implications extend beyond robotics to any domain requiring temporal reasoning and sequential decision-making, suggesting that complex temporal behaviors can emerge from simple local dynamics without the architectural complexity typically associated with memory systems in artificial intelligence.

4.8.3 Interpretability and Biological Validation

The interpretability of GRN controllers represents a significant advantage over black-box neural network approaches. The gene expression patterns provide meaningful insights into system operation that can be related to specific phases of the manipulation task and understood in terms of biological analogs. This interpretability is crucial for building trust in autonomous systems and enabling effective human-robot collaboration.

The biological validation aspect is equally important. By demonstrating that computational implementations of biological principles can achieve superior performance on challenging tasks, our work provides indirect validation of biological theories about gene regulation and cellular control.

This bidirectional validation, where computational success validates biological understanding and biological principles inform computational design, suggests a promising future for bio-inspired approaches that goes beyond superficial mimicry to capture fundamental organizational principles. The clear functional specialization observed in evolved GRN controllers, with distinct gene populations handling sensory processing, motor coordination, and regulatory control, mirrors the modular organization found in biological developmental systems [36].

4.8.4 Scalability and Future Applications

The scalability of the GRN approach depends on several factors that merit careful consideration. The computational complexity scales roughly linearly with the number of genes, making it feasible to implement larger networks for more complex tasks. However, the evolutionary optimization becomes more challenging as the

parameter space grows, potentially requiring more sophisticated search strategies or hybrid approaches that combine evolution with gradient-based fine-tuning.

The sparse connectivity helps mitigate scalability concerns by keeping the number of regulatory interactions manageable even as network size grows, with biological evidence suggesting that this sparsity can be maintained at larger scales without sacrificing functionality.

Future applications could extend beyond manipulation to domains such as locomotion, navigation, and multi-robot coordination. The hierarchical control capabilities and natural temporal dynamics of GRNs make them particularly suitable for tasks requiring coordination across multiple actuators and time scales. The approach might also prove valuable for online adaptation scenarios where controllers must modify their behavior based on changing environmental conditions or hardware failures. The inherent robustness and self-organizing properties of GRNs could enable controllers that adapt gracefully to unexpected situations without explicit retraining. Similarly, incorporating GRN principles into reinforcement learning frameworks might address the sample efficiency limitations that currently hinder practical deployment of learning-based robot control.

4.8.5 Real-World Deployment Considerations

While our simulation results demonstrated GRN effectiveness within controlled environments, real-world deployment would face additional challenges including varying lighting conditions, sensor calibration drift, mechanical wear, and timing constraints not present in simulation environments. Real robotic systems experience dynamic illumination changes and visual artifacts that could affect the sparse visual features processed by our CNN encoder. Physical sensors exhibit gradual calibration drift over extended operation, while robot actuators experience mechanical wear and changing dynamics that could impact the assumed consistent actuator response characteristics in our motor command generation.

4.9 Conclusion

This chapter has presented, to our knowledge, the first comprehensive application of Gene Regulatory Networks to vision-based robot control, demonstrating that biological regulatory principles can provide improvements in training efficiency while maintaining competitive performance on challenging manipulation tasks.

Key Contributions

1. **Novel GRN-Based Control Architecture**: First comprehensive application of gene regulatory networks to vision-based robotic manipulation, establishing a new paradigm for bio-inspired control.

2. **Evolutionary Optimization Framework**: Specialized evolutionary algorithm adapted for GRN parameters with biologically motivated constraints, achieving efficient search through constrained parameter spaces.

3. **Efficiency Improvements**: 13.7× reduction in training time compared to established baselines while achieving competitive or superior performance (57.5% success rate).

4. **Robustness Validation**: Maintained 91.8% of clean performance under noisy conditions, demonstrating resilience to sensory perturbations.

Comprehensive ablation studies confirmed that performance improvements stem from fundamental biological principles rather than superficial implementation details.

The natural emergence of temporal dynamics, hierarchical control, and functional specialization demonstrates how sophisticated behaviors can arise from simple regulatory principles.

The next chapter will provide comprehensive comparative analysis integrating insights from both major contributions, examining their complementary strengths and exploring possibilities for synergistic combination.

CHAPTER 5

COMPARATIVE ANALYSIS, INTEGRATION, AND FUTURE DIRECTIONS

5.1 Introduction

The preceding chapters presented two distinct yet complementary bio-inspired approaches to adaptive robotics: the neuromodulatory coordination framework (Chapter 3) for sequential multi-task learning and the gene regulatory network architecture (Chapter 4) for efficient vision-based control. While these approaches draw inspiration from different biological systems and address different computational challenges, their success reveals convergent principles that characterize effective biological control architectures.

This chapter provides a comprehensive comparative analysis examining fundamental mechanisms, design principles, and computational advantages underlying both approaches. By identifying common themes and complementary strengths, we establish foundations for understanding when and how different bio-inspired principles apply to autonomous system design.

Both presented approaches share fundamental characteristics distinguishing them from conventional methods: strategic use of biological constraints as optimization guides, multi-scale temporal dynamics enabling sophisticated coordination, hierarchical organization emerging naturally from biological principles, and robust performance degrading gracefully under perturbations. However, they also exhibit complementary strengths suggesting opportunities for synergistic integration.

5.2 Methodological Comparison and Architectural Analysis

5.2.1 Temporal Dynamics and Multi-Scale Coordination

Both bio-inspired approaches implement temporal dynamics operating across multiple time scales, but achieve this through fundamentally different mechanisms. The neuromodulatory framework explicitly coordinates four distinct temporal scales through specialized modules, with time constants ranging from rapid norepinephrine responses ($\tau_{\text{NE}} = 0.05$) to slower serotonin regulation ($\tau_{\text{5HT}} = 0.5$). This design directly implements known temporal hierarchies of biological neuromodulatory systems.

In contrast, the GRN approach achieves multi-scale temporal processing through distributed effects of individual gene decay rates, ranging from rapidly responding genes ($\delta_i = 0.05$) to slowly decaying memory genes ($\delta_i = 1.0$). Rather than explicit coordination between temporal scales, the GRN approach allows temporal coordination to emerge from regulatory interactions between genes with different temporal characteristics.

The architectural implications differ significantly. The neuromodulatory framework requires explicit design of temporal coordination mechanisms and careful balance between competing influences from different modules. The GRN approach achieves temporal coordination naturally through regulatory matrix structure, but provides less direct control over specific temporal relationships.

Experimental analysis reveals both approaches successfully implement temporal reasoning capabilities with different strengths. The neuromodulatory framework excels at coordinating learning dynamics across tasks with different temporal requirements, as demonstrated by adaptive modulation of learning rates based on current context. The GRN approach excels at implementing complex temporal sequences within individual tasks, shown by coordinating multi-phase manipulation behaviors without explicit sequence programming.

5.2.2 Constraint-Guided Optimization and Search Space Structure

As established in Chapters 3 and 4, both approaches validate that strategic biological constraints enhance optimization. Rather than repeating this analysis, we focus on how constraint implementation differs between approaches and enables their potential integration.

The neuromodulatory framework implements constraints primarily at the algorithmic level, constraining learning parameter adaptation to follow biologically-motivated coordination principles. The bounded nature of neuromodulator states ($s_m \in [0, 1]$) and specific temporal dynamics (τ_m values) focus the adaptation process on parameter regions proven effective in biological systems.

The GRN approach implements constraints primarily at the architectural level, constraining both network structure through sparse connectivity and dynamics through bounded gene expression ($x_i \in [0, 1]$). These structural constraints reduce effective dimensionality of the optimization problem while maintaining sufficient representational power for complex control tasks.

Table 5.1: Constraint Types and Implementation Mechanisms

Constraint Category	**BIARM Framework**	**GRN Approach**	**Biological Basis**
Parameter Bounds	Neuromodulator states $\in [0, 1]$	Gene expression $\in [0, 1]$	Concentration limits
Temporal Dynamics	Exponential moving averages with τ_m	Decay processes with δ_i	Molecular kinetics
Network Structure	Fixed PNN architecture + adaptive modulation	20% evolved sparse regulatory matrix	Evolved connectivity
Optimization Space	Learning parameter modulation	Full network architecture	Different scales
Coordination	Explicit inter-module coordination	Emergent regulatory cascades	System-level organization

Table 5.1 reveals complementary constraint strategies: BIARM constrains the *adaptation process* while maintaining architectural flexibility, whereas GRN constrains the *architecture itself* while allowing behavioral emergence.

5.3 Applications and Limitations

5.3.1 Application Domain

The success in respective domains provides insight into robotic applications where bio-inspired methods offer the greatest advantages.

The BIARM Framework demonstrates particular effectiveness in sequential multi-task learning scenarios where knowledge transfer and adaptation are crucial. It

shows promise for lifelong learning robotics, where domestic robots continuously acquire new skills while maintaining previously learned capabilities. The framework is also well-suited for multi-domain service robots, which are systems operating across multiple environments with different task demands. Additionally, it proves valuable in human-robot collaboration scenarios requiring adaptation based on human feedback and changing collaboration patterns.

The GRN Approach demonstrates exceptional effectiveness for vision-based control tasks requiring rapid deployment and robust execution. It is particularly applicable to industrial automation, specifically manufacturing systems requiring rapid deployment without extensive training periods. The approach is also beneficial in resource-constrained environments such as space and underwater robotics where training time and computational efficiency are critical. Furthermore, it suits safety-critical applications, which are systems requiring interpretable dynamics and predictable behavior for verification and validation.

5.3.2 Limitations and Possible Failure Modes

Critical assessment of limitations is essential for future research direction and practical deployment considerations.

The BIARM Framework faces several significant limitations. The computational overhead of 52–87% runtime may be prohibitive for resource-constrained applications, necessitating hardware acceleration or selective module activation. The framework currently exhibits architecture dependency, being tied to Progressive Neural Networks architecture, which means generalization to other architectures requires investigation. While the framework demonstrates robustness, it shows hyperparameter sensitivity in that optimal time constants (τ_m) require domain-specific tuning. There is also a scalability ceiling concern, as the quadratic growth of lateral connections with task count may limit applicability beyond approximately 10 tasks without architectural modifications.

The GRN Approach similarly has notable limitations. It demonstrates limited post-deployment adaptation flexibility after evolutionary optimization. The approach exhibits evolutionary sensitivity, as performance can vary with initialization, with the observed success rate confidence interval of [51.2%, 63.8%] suggesting inherent variability. Additionally, there is a simulation-reality gap since all results were obtained in simulation, meaning real-world deployment faces sensor noise, lighting variations, and mechanical wear not fully captured in simulated environments.

Both approaches share several common limitations. They exhibit domain specificity, with BIARM tested only on discrete-action environments and GRN tested

only on a single manipulation task. Finally, there are comparison constraints, as the different problem domains prevent direct quantitative comparison between the two approaches.

5.4 Integration Possibilities and Hybrid Architectures

5.4.1 Conceptual Framework for Integration

The complementary strengths suggest promising avenues for integration leveraging advantages of both whilst mitigating individual limitations. The conceptual foundation rests on observation that biological systems routinely coordinate multiple adaptive mechanisms operating at different temporal and organizational scales.

Consider how biological organisms coordinate gene regulation with neuromodulation: gene regulatory networks establish basic architectural framework and functional capabilities of neural circuits, whilst neuromodulatory systems dynamically adjust circuit operation based on behavioural context and environmental demands. This hierarchical coordination enables both stability provided by evolved architectural constraints and flexibility required for adaptive behaviour.

5.4.2 Hierarchical Integration Architecture

We propose hierarchical integration where GRN-based controllers provide base-level motor control whilst neuromodulatory systems enable adaptive parameter tuning. Since both approaches have been validated independently in previous chapters, this section outlines the integration proposal.

The integrated system extends the base GRN dynamics from Chapter 4 to include neuromodulatory influence:

$$x_i(t+1) = x_i(t) + \Delta t \cdot [f_i(s, x, \mathbf{m}) - \delta_i(m_{\text{DA}})x_i(t) + \eta_i(t)] \tag{5.1}$$

where $\mathbf{m} = [m_{\text{DA}}, m_{\text{5HT}}, m_{\text{NE}}, m_{\text{ACh}}] \in [0,1]^4$ represents the neuromodulatory state vector. The system state consists of 50 gene expression levels plus 4 neuromodulator concentrations, yielding a 54-dimensional dynamic state.

The neuromodulatory modules modulate key GRN parameters as follows:

$$\delta_i(m_{\text{DA}}) = \delta_i^{\text{base}} \cdot \exp(\gamma_{\text{DA}} \cdot (m_{\text{DA}} - 0.5)) \tag{5.2}$$

$$\theta_i(m_{\text{5HT}}) = \theta_i^{\text{base}} \cdot (1 + \gamma_{\text{5HT}} \cdot (m_{\text{5HT}} - 0.5)) \tag{5.3}$$

$$w_{ij}^{(r)}(m_{\text{NE}}, m_{\text{ACh}}) = w_{ij}^{\text{base}} \cdot (1 + \gamma_{\text{NE}} \cdot (m_{\text{NE}} - 0.5) + \gamma_{\text{ACh}} \cdot (m_{\text{ACh}} - 0.5)) \tag{5.4}$$

Equation 5.2 modulates gene decay rates δ_i through dopamine, where δ_i^{base} represents the evolved baseline decay rate for gene i and γ_{DA} controls the modulation strength. The exponential form ensures decay rates remain positive whilst allowing bidirectional adjustment around the baseline. Equation 5.3 adjusts activation thresholds θ_i via serotonin, where θ_i^{base} is the baseline threshold and γ_{5HT} determines the modulation magnitude. Equation 5.4 modulates regulatory interaction strengths $w_{ij}^{(r)}$ through norepinephrine and acetylcholine, where w_{ij}^{base} represents the evolved baseline weight between genes i and j, and γ_{NE}, γ_{ACh} control their respective modulation contributions.

These formulations follow the neuromodulatory principles established in Chapter 3, adapted to modulate GRN parameters rather than neural network learning rates.

The primary implementation challenges include managing the increased computational cost, ensuring stable interaction between the GRN and neuromodulatory components, and determining appropriate modulation parameter ranges (γ values) for each neuromodulator. These considerations, along with empirical validation of the proposed integration, represent important directions for future work.

5.5 Broader Implications for Bio-Inspired Robotics

The success of both individual approaches and potential for integration reveals fundamental design principles guiding future development of bio-inspired robotic systems.

Principle 1: Constraints as Optimization Guides Biological constraints should be viewed as optimization guides rather than limitations. Both approaches demonstrate carefully chosen constraints reduce search space dimensionality while maintaining representational power, focus optimization on biologically validated parameter regions, and provide inherent stability preventing pathological behaviors.

Principle 2: Multi-Scale Temporal Organization Effective adaptive systems require coordination across multiple temporal scales. Implementation should

match temporal constants to functional requirements, enable emergent coordination through appropriate coupling mechanisms, and leverage temporal hierarchy for both stability and flexibility.

Principle 3: Hierarchical Functional Organization Complex behaviors emerge from hierarchical organization of simple components. Design should implement specialized modules with clear functional roles, enable emergent coordination through appropriate interaction mechanisms, and maintain modularity while allowing information flow between levels.

Principle 4: Robustness Through Bounded Dynamics Biological systems achieve robustness through bounded state variables and dissipative dynamics. Artificial systems should implement bounded parameter spaces preventing extreme values, use decay processes providing natural stability, and incorporate noise enabling exploration while filtering perturbations.

5.6 Research Questions Revisited

Having presented the experimental results, comparative analysis, and broader implications of both contributions, this section returns to the six research questions posed in the Introduction and provides direct answers grounded in the empirical and theoretical findings of this thesis.

RQ1: How can mechanisms governing biological development, from gene regulatory networks to neuromodulatory systems, be translated into computational frameworks that enhance learning efficiency and robustness in robotic systems?

Both contributions demonstrate that effective translation requires abstraction at the level of organisational principles rather than molecular detail. For BIARM, the key abstraction is the coordinated multi-timescale structure of neuromodulatory interaction, implemented through four interdependent modules with module-specific adaptation rates and cascaded dependency chains. For the GRN controller, the key abstractions are bounded gene expression dynamics, sparse regulatory connectivity, Hill function activation, and molecular decay, all implemented through discrete-time updates that preserve the essential computational properties of their biological counterparts. In both cases, the translation preserves functional behaviour rather than biological mechanism, which is what enables measurable performance advantages in artificial systems.

RQ2: Can neuromodulatory principles be effectively implemented in artificial systems to overcome the limitations of fixed-parameter learning algorithms?

The experimental results presented in Chapter 3 provide a direct affirmative answer. BIARM achieves performance improvements ranging from 1.06% to 155.94% over baseline Progressive Neural Networks across four Gymnasium environments, with an average knowledge retention rate of 98.0% compared to 93.2% for the fixed-parameter baseline. The LunarLander-v3 result is particularly illustrative: the baseline PNN consistently failed to learn an effective policy (-417.17 ± 412.62), while BIARM achieved stable positive performance (233.37 ± 34.38), demonstrating that coordinated neuromodulatory adaptation can resolve learning failures that fixed-parameter approaches cannot overcome.

RQ3: Can biological constraints serve as optimisation guides that focus the search process on robust solutions rather than limiting system capabilities?

The GRN ablation study provides the clearest answer to this question. The bounded gene expression interval $[0, 1]$, the sparse regulatory matrix, and the Hill function activation collectively reduce the effective dimensionality of the evolutionary search space while preserving sufficient representational power for complex manipulation. The resulting 13.7× reduction in training time compared to PPO, achieved without sacrificing final performance, confirms that biological constraints accelerate optimisation rather than impeding it. The BIARM framework corroborates this finding at the algorithmic level, where biologically motivated adaptation rates and dependency structures consistently outperformed unconstrained fixed-parameter baselines.

RQ4: What role can gene regulatory networks play in creating efficient and robust robot controllers?

Chapter 4 demonstrates that GRNs can serve as complete vision-based robot controllers, achieving a 57.5% success rate on the KukaDiverseObjectEnv benchmark while retaining 91.8% of clean performance under sensory noise perturbations. Beyond raw performance, GRNs offer three specific advantages for robotic control: natural temporal memory through gene decay dynamics without explicit recurrent architectures, inherent robustness through bounded state variables that prevent extreme responses to noisy inputs, and interpretable regulatory structure that can be analysed in terms of biologically meaningful gene interaction patterns. These properties collectively position GRNs as a promising control paradigm for resource-constrained and safety-critical robotic applications.

RQ5: How do bio-inspired constraints and mechanisms affect training efficiency, robustness, and generalisation capabilities compared to conventional approaches?

Across both contributions, bio-inspired constraints consistently improved all three dimensions simultaneously. Training efficiency was improved by 13.7× for the

GRN controller and by substantial variance reduction for BIARM, with the LunarLander standard deviation decreasing from 412.62 to 34.38. Robustness was demonstrated through GRN noise tolerance (91.8% performance retention under $\mathcal{N}(0, 0.03^2)$ visual perturbations) and through BIARM's superior knowledge retention across sequential tasks. Generalisation was evidenced by consistent improvements across four structurally distinct Gymnasium environments for BIARM, and by successful grasping across over 1000 diverse object types for the GRN controller. Critically, these gains were achieved without increasing model capacity, confirming that the advantages stem from structural biological principles rather than from additional parameters.

RQ6: Can the demonstrated principles be generalised into a coherent framework for developmental evolutionary robotics that guides adaptive system design across multiple domains?

Chapter 5 has identified four convergent design principles shared by both contributions, constraints as optimisation guides, multi-scale temporal organisation, hierarchical functional organisation, and robustness through bounded dynamics, that hold independently of the specific biological system being abstracted. The proposed hierarchical integration architecture (Section 5.4.2), in which GRN-based controllers provide base-level motor control while neuromodulatory modules provide adaptive parameter tuning, demonstrates that these principles are compositional and can be combined into a unified framework. While empirical validation of the integrated architecture remains a direction for future work, the theoretical coherence of the framework and the consistency of results across two independent bio-inspired implementations support the viability of developmental evolutionary robotics as a principled design methodology rather than a collection of isolated techniques.

5.7 Future Research Directions

Several immediate research directions emerge directly from this work:

Real-World Hardware Implementation While simulation results provide compelling evidence for effectiveness, validation on physical robotic systems remains essential. Challenges include: sensor noise and calibration drift not fully captured in simulation, actuator dynamics and mechanical wear affecting controller performance, real-time computational constraints requiring optimization, and safety considerations for learning-based control systems. The biological robustness principles underlying both approaches (bounded dynamics, intrinsic noise tolerance, distributed decision-making) may provide inherent advantages when adapting to real-world uncertainties.

Extended Task Complexity Both approaches should be evaluated on more complex tasks requiring longer-horizon planning, multi-object manipulation, and coordination between multiple robots. Extensions could include: hierarchical task decomposition and planning, coordination between multiple GRN or neuromodulatory controllers, and integration with symbolic reasoning systems for high-level planning.

Hybrid System Implementation The proposed integration architectures require empirical validation through: systematic comparison of different integration approaches, ablation studies isolating contributions of each component, and evaluation across diverse task domains and environmental conditions.

CONCLUSION

In the dynamic landscape of autonomous robotics, where the fusion of biological principles, evolutionary computation, and machine learning is reshaping intelligent system design, the imperative for efficient, robust, and interpretable control mechanisms has never been more pronounced. The research presented in this dissertation underscores the multifaceted advantages of incorporating developmental principles into robotic systems and introduces innovative solutions to enhance their performance, efficiency, and reliability.

This thesis demonstrates that biological development principles, when systematically translated into computational frameworks, can outperform traditional approaches. The formalization rests on three foundational principles: constrained optimization spaces prove more efficient than unbounded searches, multi-factor coordination outperforms isolated mechanisms, and temporal dynamics emerge naturally from regulatory mechanisms without explicit memory architectures. The mathematical formulation of gene regulatory networks (GRNs) for robotic control bridges computational biology and autonomous systems through discrete-time gene expression dynamics with Hill function activation, biological noise, and sparse connectivity constraints. The Bio-Inspired Adaptive Rate Modulation (BIARM) framework coordinates four computational modules inspired by neuromodulatory systems; dopamine, serotonin, norepinephrine, and acetylcholine, demonstrating substantial advantages over single-factor approaches.

From the development of GRN-based controllers to the BIARM framework, the research efforts outlined in this work have focused on demonstrating the practical advantages of biological inspiration. These contributions reveal that biological constraints consistently improved optimization performance. Bio-inspired systems

maintained strong performance under noisy conditions, while the substantial variance reduction observed across all tasks indicates improved training stability addressing persistent challenges in neural network optimization.

In conclusion, this dissertation contributes to developmental evolutionary robotics as a promising approach to autonomous systems, with potential applications in bio-inspired technologies for efficient and robust deployment scenarios. By leveraging biological constraints as optimization advantages and demonstrating coordinated multi-factor adaptation, this research provides algorithmic optimizations that may inform future advancements in robotic learning.

Outlook

Building upon the insights gained from the demonstrated advantages of biological constraints, several promising avenues for further research and development emerge.

Scalable bio-inspired architectures: As robotic tasks grow in complexity, there is a pressing need for scalable bio-inspired solutions that can effectively address multi-modal, long-horizon challenges. Future research efforts could focus on investigating how the demonstrated principles scale to more sophisticated systems, exploring deeper integration of developmental mechanisms and expanding the complexity of gene regulatory architectures while maintaining computational tractability.

Interdisciplinary collaboration: The intersection of computational biology, neuroscience, evolutionary computation, and robotics presents rich opportunities for interdisciplinary collaboration. Future research endeavors could leverage insights from diverse fields to develop holistic solutions.

Interpretability and transparency: With increasing deployment of autonomous systems in safety-critical applications, there is growing demand for interpretable control mechanisms that enable human oversight and trust. Future research could explore advanced visualization techniques for gene expression dynamics, develop formal verification methods for bio-inspired controllers.

Real-world deployment and evaluation: While simulations demonstrations are essential, real-world impact ultimately depends on practical deployment in diverse operational settings. Future research efforts should prioritize implementation in actual robotic platforms, allowing for empirical evaluation of performance under realistic conditions including hardware constraints, sensor noise, and environmental unpredictability that extend beyond simulation benchmarks.

BIBLIOGRAPHY

[1] Bruno Siciliano and Oussama Khatib, editors. *Springer Handbook of Robotics*. Springer International Publishing, 2nd edition, 2016.

[2] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[3] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, Cambridge, MA, 2005.

[4] Rodney A Brooks. A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, 2(1):14–23, 1986.

[5] Ronald C Arkin. *Behavior-Based Robotics*. MIT Press, Cambridge, MA, 1998.

[6] Dario Floreano and Claudio Mattiussi. *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies*. MIT Press, Cambridge, MA, 2008.

[7] Rolf Pfeifer and Josh Bongard. *How the Body Shapes the Way We Think: A New View of Intelligence*. MIT Press, Cambridge, MA, 2006.

[8] Stefano Nolfi and Dario Floreano. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press, Cambridge, MA, 2000.

[9] Hod Lipson and Jordan B Pollack. Automatic design and manufacture of robotic lifeforms. *Nature*, 406(6799):974–978, 2000.

[10] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[11] David Silver, Aja Huang, Chris J Maddison, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[12] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, et al. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[13] Marc Peter Deisenroth, Gerhard Neumann, and Jan Peters. A survey on policy search for robotics. *Foundations and Trends in Robotics*, 2(1-2):1–141, 2013.

[14] Sergey Levine, Peter Pastor, Alex Krizhevsky, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *International Journal of Robotics Research*, 37(4-5):421–436, 2018.

[15] Hyeonwoo Lee, Takuya Seno, Seunghoon Kim, and Peter R Wurman. A champion-level vision-based reinforcement learning agent for competitive racing in gran turismo 7. *arXiv preprint arXiv:2504.09021*, 2025.

[16] Tao Chen, Ben Abbatematteo, Yiran Geng, and Sergey Levine. Deep reinforcement learning for real-world robotics: A survey. *Annual Review of Control, Robotics, and Autonomous Systems*, 8:1–31, 2025.

[17] Rami Al Homsi, Antoine Youssef, and Daniel Asmar. Comparative analysis of deep q-learning algorithms for robotic object throwing. *Frontiers in Robotics and AI*, 12:1567211, 2025.

[18] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

[19] Peter Henderson, Riashat Islam, Philip Bachman, et al. Deep reinforcement learning that matters. In *AAAI Conference on Artificial Intelligence*, pages 3207–3214, 2018.

[20] Nick Jakobi, Phil Husbands, and Inman Harvey. Noise and the reality gap: The use of simulation in evolutionary robotics. In *European Conference on Artificial Life*, pages 704–720. Springer, 1995.

[21] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2022.

[22] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, et al. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[23] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:1–72, 2017.

[24] Anthony M Zador. A critique of pure learning and what artificial neural networks can learn from animal brains. *Nature Communications*, 10(1):3770, 2019.

[25] Scott F Gilbert and David Epel. *Ecological Developmental Biology: Integrating Epigenetics, Medicine, and Evolution*. Sinauer Associates, Sunderland, MA, 2009.

[26] Mary Jane West-Eberhard. *Developmental Plasticity and Evolution*. Oxford University Press, Oxford, UK, 2003.

[27] Dario Floreano, Peter Dürr, and Claudio Mattiussi. Neuroevolution: From architectures to learning. *Evolutionary Intelligence*, 1(1):47–62, 2008.

[28] Kenji Doya. Metalearning and neuromodulation. *Neural Networks*, 15(4-6):495–506, 2002.

[29] Eve Marder. Neuromodulation of neuronal circuits: Back to the future. *Neuron*, 76(1):1–11, 2012.

[30] Cornelia I Bargmann. Beyond the connectome: How neuromodulators shape neural circuits. *BioEssays*, 34(6):458–465, 2012.

[31] Uri Alon. *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Chapman and Hall/CRC, Boca Raton, FL, 2nd edition, 2019.

[32] Peter Dayan. Twenty-five lessons from computational neuromodulation. *Neuron*, 76(1):240–256, 2012.

[33] Melanie Mitchell. Why ai is harder than we think. *Artificial Intelligence*, 323:103941, 2023.

[34] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[35] Anirudh Goyal and Yoshua Bengio. Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A*, 478(2266):20210068, 2022.

[36] Eric H Davidson. *The Regulatory Genome: Gene Regulatory Networks in Development and Evolution*. Academic Press, Burlington, MA, 2006.

[37] Chourouk Guettas and Foudil Cherif. Bio-inspired adaptive rate modulation for multi-task learning in progressive neural networks. *Informatica*, 49(21):437–448, 2025.

[38] Chourouk Guettas, Foudil Cherif, Ammar Muthanna, Mohammad Hammoudeh, and Abdelkader Laouid. Gene regulatory networks for enhanced vision-based robot control: A bio-inspired approach. *Sensors*, 26(6):1742, 2026.

[39] Mark Towers, Jordan K. Terry, Ariel Kwiatkowski, John U. Balis, Gianluca de Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Arjun KG, Markus Krimmel, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Andrew Tan Jin Shen, and Omar G. Younis. Gymnasium: A standardized interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

[40] Guy Karlebach and Ron Shamir. Modelling and analysis of gene regulatory networks. *Nature Reviews Molecular Cell Biology*, 9(10):770–780, 2008.

[41] Lacramioara Bintu, Nicolas E Buchler, Hernan G Garcia, et al. Transcriptional regulation by the numbers: models. *Current Opinion in Genetics & Development*, 15(2):116–124, 2005.

[42] Rob Phillips, Jane Kondev, Julie Theriot, and Hernan G Garcia. *Physical Biology of the Cell*. CRC Press, 2012.

[43] Alvaro Sanchez and Ido Golding. Transcriptional control of cell fate decisions. *Nature Reviews Genetics*, 21(11):701–714, 2020.

[44] Albert-László Barabási and Zoltán N Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113, 2004.

[45] Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, et al. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

[46] Uri Alon. Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8(6):450–461, 2007.

[47] Shmoolik Mangan and Uri Alon. Structure and function of the feed-forward loop network motif. *Proceedings of the National Academy of Sciences*, 100(21):11980–11985, 2003.

[48] Béla Novák and John J Tyson. Design principles of biochemical oscillators. *Nature Reviews Molecular Cell Biology*, 9(12):981–991, 2008.

[49] Timothy S Gardner, Charles R Cantor, and James J Collins. Construction of a genetic toggle switch in escherichia coli. *Nature*, 403(6767):339–342, 2000.

[50] Joseph S Takahashi. Transcriptional architecture of the mammalian circadian clock. *Nature Reviews Genetics*, 18(3):164–179, 2017.

[51] Michael B Elowitz, Arnold J Levine, Eric D Siggia, and Peter S Swain. Stochastic gene expression in a single cell. *Science*, 297(5584):1183–1186, 2002.

[52] Gábor Balázsi, Alexander van Oudenaarden, and James J Collins. Cellular decision making and biological noise: from microbes to mammals. *Cell*, 144(6):910–925, 2011.

[53] Mitchell Lewis. Allostery and the lac operon. *Journal of Molecular Biology*, 425(13):2309–2316, 2013.

[54] Thomas Gregor, David W Tank, Eric F Wieschaus, and William Bialek. Probing the limits to positional information. *Cell*, 130(1):153–164, 2007.

[55] Michael C Avery and Jeffrey L Krichmar. Neuromodulatory systems and their interactions: A review. *Frontiers in Neural Circuits*, 11:108, 2017.

[56] Claire O'Callaghan, Ishan C. Walpola, and James M. Shine. Neuromodulation of the mind-wandering brain state: the interaction between neuromodulatory tone, sharp wave-ripples and spontaneous thought. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 376(1817):20190699, 2021.

[57] Wolfram Schultz. Dopamine reward prediction error coding. *Dialogues in Clinical Neuroscience*, 18(1):23–32, 2016.

[58] Wolfram Schultz, Peter Dayan, and P Read Montague. A neural substrate of prediction and reward. *Science*, 275(5306):1593–1599, 1997.

[59] Gary Aston-Jones and Jonathan D Cohen. An integrative theory of locus coeruleus–norepinephrine function: adaptive gain and optimal performance. *Annual Review of Neuroscience*, 28:403–450, 2005.

[60] Michael E. Hasselmo. The role of acetylcholine in learning and memory. *Current Opinion in Neurobiology*, 16(6):710–715, 2006.

[61] Katie A. Ferguson and Jessica A. Cardin. Mechanisms underlying gain modulation in the cortex. *Nature Reviews Neuroscience*, 21(2):80–92, 2020.

[62] Ricardo Medina-Coss y León, Elí Lezama, Inmaculada Márquez, and Mario Treviño. Adrenergic modulation of cortical gain and sensory processing in the mouse visual cortex. *Brain Sciences*, 15(4):406, 2025.

[63] Nils Lundqvist, Mateusz Garbulowski, Thomas Hillerton, and Erik L. L. Sonnhammer. Topology-based metrics for finding the optimal sparsity in gene regulatory network inference. *Bioinformatics*, 41(5):btaf120, 2025.

[64] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2nd edition, 2018.

[65] John Schulman, Filip Wolski, Prafulla Dhariwal, et al. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[66] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870, 2018.

[67] Thomas Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, Oxford, UK, 1996.

[68] Agoston E Eiben and James E Smith. *Introduction to evolutionary computing*. Springer, 2nd edition, 2015.

[69] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165, 1989.

[70] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999.

[71] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. *European Conference on Computer Vision (ECCV)*, pages 532–547, 2018.

[72] German I Parisi, Ronald Kemker, Jose L Part, et al. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

[73] John Cichon and Wen-Biao Gan. Branch-specific dendritic ca2+ spikes cause persistent synaptic plasticity. *Nature*, 520(7546):180–185, 2015.

[74] Yadin Dudai, Avi Karni, and Jan Born. The restless engram: consolidations never end. *Annual Review of Neuroscience*, 38:465–486, 2015.

[75] David Meunier, Renaud Lambiotte, and Edward T Bullmore. Modular and hierarchically modular organization of brain networks. *Frontiers in Neuroscience*, 4:200, 2010.

[76] Claudio Mattiussi and Dario Floreano. Analog genetic encoding for the evolution of circuits and networks. *IEEE Transactions on Evolutionary Computation*, 11(5):596–607, 2007.

[77] Michał Joachimczak and Borys Wróbel. Evolving gene regulatory networks for real time control of foraging behaviours. In *Artificial Life Conference Proceedings 14*, pages 348–355. MIT Press, 2014.

[78] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems*, pages 2204–2212, 2014.

[79] Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, et al. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016.

[80] Inman Harvey, Phil Husbands, and Dave Cliff. Seeing the light: Artificial evolution, real vision. In *From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, pages 392–401. MIT Press, 1994.

[81] Karl Sims. Evolving virtual creatures. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, pages 15–22, 1994.

[82] Kenneth O Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 2002.

[83] Joel Lehman and Kenneth O Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223, 2011.

[84] Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*, 2015.

[85] Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, 2015.

[86] Jørgen Nordmoen, Eivind Samuelsen, and Kyrre Glette. Map-elites enables powerful stepping stones and diversity for modular robotics. *Frontiers in Robotics and AI*, 8:639173, 2021.

[87] Yanan Liu, Yanteng Sun, Bing Xue, Mengjie Zhang, Gary G Yen, and Kay Chen Tan. Evolutionary neural architecture search for automated deep learning. *IEEE Transactions on Evolutionary Computation*, 27(4):786–800, 2023.

[88] Matthew C Fontaine, Julian Togelius, Stefanos Nikolaidis, and Amy K Hoover. Differentiable quality diversity. In *Advances in Neural Information Processing Systems*, volume 34, pages 10040–10052, 2021.

[89] Adam Faldor and Antoine Cully. Map-elites with descriptor-conditioned gradients and archive distillation into a single policy. *arXiv preprint arXiv:2303.03832*, 2023.

[90] Agoston E Eiben, Nicolas Bredeche, Andries Haasdijk, and Eliseo Ferrante. Real-world robot evolution: why would it (not) work? *Frontiers in Robotics and AI*, 8:696452, 2021.

[91] Eliseo Ferrante, Jon Timmis, Alan FT Winfield, and Danesh Tarapore. Evolution of self-organized task specialization in robot swarms. *PLoS Computational Biology*, 11(8):e1004273, 2015.

[92] Kenneth O Stanley, David B D'Ambrosio, and Jason Gauci. A hypercube-based encoding for evolving large-scale neural networks. *Artificial Life*, 15(2):185–212, 2009.

[93] Tim Salimans, Jonathan Ho, Xi Chen, et al. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

[94] Nikolaus Hansen. The cma evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.

[95] Adam Gaier and David Ha. Discovering representations for black-box optimization. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pages 103–111, 2020.

[96] Antoine Ligot, Ken Hasselmann, and Mauro Birattari. Automatic modular design of robot swarms using behavior trees as a control architecture. *PeerJ Computer Science*, 6:e314, 2020.

[97] Manon Flageat and Antoine Cully. Empirical analysis of pga-map-elites for neuroevolution in uncertain domains. *Genetic Programming and Evolvable Machines*, 24(2):15, 2023.

[98] Matthew C Fontaine and Stefanos Nikolaidis. Covariance matrix adaptation map-annealing. *arXiv:2205.10752*, 2023.

[99] Matteo De Carlo, Jacintha Zeeuwen, Eliseo Ferrante, and Agoston E Eiben. Heritability and selection in robot evolution. *Artificial Life*, pages 1–21, 2024.

[100] Timothée Anne, Luca Grillotti, and Antoine Cully. Parametric map-elites for multi-task adaptation. *arXiv preprint arXiv:2402.01275*, 2024.

[101] Mikkel Jørgensen, Jørgen Nordmoen, Fay Veenstra, Kai Olav Ellefsen, and Kyrre Glette. Llm-guided evolutionary search for complex robot behavior synthesis. *arXiv preprint arXiv:2409.16890*, 2024.

[102] Arun Mallya and Svetlana Lazebnik. PackNet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018.

[103] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pages 3987–3995. PMLR, 2017.

[104] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. iCaRL: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[105] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[106] Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations*, 2021.

[107] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[108] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022.

[109] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. DualPrompt: Complementary prompting for rehearsal-free continual learning. *arXiv preprint arXiv:2204.04799*, 2022.

[110] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*, pages 4548–4557. PMLR, 2018.

[111] Liyuan Wang, Mingtian Zhang, Zhongfan Jia, Qian Li, Chenglong Bao, Kaisheng Ma, Jun Zhu, and Yi Zhong. AFEC: Active forgetting of negative transfer in continual learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 22379–22391, 2021.

[112] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017.

[113] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. PODNet: Pooled outputs distillation for small-tasks incremental learning. In *European Conference on Computer Vision*, pages 86–102. Springer, 2020.

[114] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2L: Contrastive continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9516–9525, 2021.

[115] Xinting Hu, Kaihua Tang, Chunyan Miao, Xian-Sheng Hua, and Hanwang Zhang. Distilling causal effect of data in class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3957–3966, 2021.

[116] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: A strong, simple baseline. In *Advances in Neural Information Processing Systems*, volume 33, pages 15920–15930, 2020.

[117] Matteo Boschini, Lorenzo Bonicelli, Pietro Buzzega, Angelo Porrello, and Simone Calderara. Class-incremental continual learning into the extended DER-verse. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[118] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[119] Chenshen Wu, Luis Herranz, Xialei Liu, Joost van de Weijer, Bogdan Raducanu, et al. Memory replay GANs: Learning to generate new categories without forgetting. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[120] Mengyao Zhai, Lei Chen, Frederick Tung, Jiawei He, Megha Nawhal, and Greg Mori. Lifelong GAN: Continual learning for conditional image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2759–2768, 2019.

[121] Xialei Liu, Chenshen Wu, Mikel Menta, Luis Herranz, Bogdan Raducanu, Andrew D Bagdanov, Shangling Jui, and Joost van de Weijer. Generative feature replay for class-incremental learning. In *CVPR Workshops*, pages 226–227, 2020.

[122] Tyler L Hayes, Kushal Kafle, Robik Shrestha, Manoj Acharya, and Christopher Kanan. REMIND your neural network to prevent catastrophic forgetting. In *European Conference on Computer Vision*, pages 466–483. Springer, 2020.

[123] Grégoire Petit, Adrian Popescu, Hugo Schindler, David Picard, and Bertrand Delezoide. FeTrIL: Feature translation for exemplar-free class-incremental learning. *arXiv preprint arXiv:2211.13131*, 2022.

[124] Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3762–3773. PMLR, 2020.

[125] Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training networks in null space of feature covariance for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 184–193, 2021.

[126] Khurram Javed and Martha White. Meta-learning representations for continual learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[127] Shawn Beaulieu, Lapo Frati, Thomas Miconi, Joel Lehman, Kenneth O Stanley, Jeff Clune, and Nick Cheney. Learning to continually learn. In *ECAI 2020*, pages 992–1001. IOS Press, 2020.

[128] Gunshi Gupta, Karmesh Yadav, and Liam Paull. Look-ahead meta learning for continual learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 11588–11598, 2020.

[129] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. Understanding the role of training regimes in continual learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 7308–7320, 2020.

[130] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Dilan Gorur, Razvan Pascanu, and Hassan Ghasemzadeh. Linear mode connectivity in multitask and continual learning. In *International Conference on Learning Representations*, 2021.

[131] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

[132] Divyam Madaan, Jaehong Yoon, Yuanchun Li, Yunxin Liu, and Sung Ju Hwang. Representational continuity for unsupervised continual learning. In *International Conference on Learning Representations*, 2021.

[133] Quang Pham, Chenghao Liu, and Steven Hoi. DualNet: Continual learning, fast and slow. In *Advances in Neural Information Processing Systems*, volume 34, pages 16131–16144, 2021.

[134] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. CODA-Prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11909–11919, 2023.

[135] Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. SLCA: Slow learner with classifier alignment for continual learning on a pre-trained model. *arXiv preprint arXiv:2303.05118*, 2023.

[136] Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in superposition. In *Advances in Neural Information Processing Systems*, volume 33, pages 15173–15184, 2020.

[137] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*, 2018.

[138] Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. DyTox: Transformers for continual learning with dynamic token expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9285–9295, 2022.

[139] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3366–3375, 2017.

[140] Rahul Ramesh and Pratik Chaudhari. Model zoo: A growing brain that learns continually. In *International Conference on Learning Representations*, 2021.

[141] Liyuan Wang, Xingxing Zhang, Qian Li, Jun Zhu, and Yi Zhong. CoSCL: Cooperation of small continual learners is stronger than a big one. In *European Conference on Computer Vision*, pages 254–271. Springer, 2022.

[142] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. *Neural Networks: Tricks of the Trade*, pages 437–478, 2012.

[143] Tom Schaul, Sixin Zhang, and Yann LeCun. No more pesky learning rates. In *International Conference on Machine Learning*, pages 343–351, 2013.

[144] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011.

[145] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4(2):26–31, 2012.

[146] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. Published in ICLR 2015.

[147] Matthew D Zeiler. Adadelta: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[148] Thomas Degris, Martha White, and Richard S Sutton. Off-policy actor-critic. *arXiv preprint arXiv:1205.4839*, 2012.

[149] Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte carlo gradient estimation in machine learning. *Journal of Machine Learning Research*, 21(132):1–62, 2020.

[150] Arsalan Sharifnassab, Saber Salehkaleybar, and Richard S. Sutton. Metaoptimize: A framework for optimizing step sizes and other meta-parameters. *arXiv preprint arXiv:2402.02342*, 2024.

[151] Tim van Erven, Wouter M. Koolen, and Dirk van der Hoeven. Metagrad: Adaptation using multiple learning rates in online learning. *arXiv preprint arXiv:2102.06622*, 2021.

[152] Kang Zhou, Qing Gui, Yue Zhang, Yuan Gao, Kai Yu, et al. Metaslrcl: A self-adaptive learning rate and curriculum learning approach for neural text classification with imbalanced data. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2039–2050, 2022.

[153] Yiping Liang, Lulu Tian, Xu Zhang, Xiao Zhang, and Libing Bai. Multi-dimensional adaptive learning rate gradient descent optimization algorithm for network training in magneto-optical defect detection. *International Journal of Network Dynamics and Intelligence*, 2024.

[154] Angela J Yu and Peter Dayan. Uncertainty, neuromodulation, and attention. *Neuron*, 46(4):681–692, 2005.

[155] Wolfgang Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997.

[156] Wulfram Gerstner and Werner M Kistler. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, Cambridge, UK, 2002.

[157] Samanwoy Ghosh-Dastidar and Hojjat Adeli. Spiking neural networks. *International Journal of Neural Systems*, 19(4):295–308, 2009.

[158] Herbert Jaeger. The "echo state" approach to analysing and training recurrent neural networks. *GMD Report 148, German National Research Center for Information Technology*, 2001.

[159] Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, 2002.

[160] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[161] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3856–3866, 2017.

[162] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, et al. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[163] Andrea Soltoggio, Peter Dürr, Claudio Mattiussi, and Dario Floreano. Evolutionary advantages of neuromodulated plasticity in dynamic, reward-based scenarios. *Artificial Life*, pages 569–576, 2008.

[164] Jean-Marc Fellous and Christiane Linster. The role of neuromodulation in cortical plasticity. *Advances in Neural Information Processing Systems*, 11:333–340, 1999.

[165] Thomas Miconi, Jeff Clune, and Kenneth O. Stanley. Differentiable plasticity: training plastic neural networks with backpropagation. *Proceedings of the 35th International Conference on Machine Learning*, 80:3559–3568, 2018.

[166] Jürgen Schmidhuber. Evolutionary principles in self-referential learning. *Diploma Thesis, Institut für Informatik, Technische Universität München*, 1987.

[167] Jie Mei, Alejandro Rodriguez-Garcia, Daigo Takeuchi, Gabriel Wainstein, Nina Hubig, Yalda Mohsenzadeh, and Srikanth Ramaswamy. Improving the adaptive and continuous learning capabilities of artificial neural networks: Lessons from multi-neuromodulatory dynamics. *arXiv preprint arXiv:2501.06762*, 2025. Version 2.

[168] Ziyu Wang, Rui Chen, and Ming Liu. Neuromodulated meta-learning. *arXiv preprint arXiv:2411.06746*, 2024.

[169] Eseoghene Ben-Iwhiwhu, Jonathan Dick, Nicholas A. Ketz, Praveen K. Pilly, and Andrea Soltoggio. Context meta-reinforcement learning via neuromodulation. *Neural Networks*, 152:70–79, 2022.

[170] Guillaume Bellec, Franz Scherr, Anand Subramoney, Elias Hajek, Darjan Salaj, Robert Legenstein, and Wolfgang Maass. A solution to the learning dilemma for recurrent networks of spiking neurons. *Nature Communications*, 11(1):3625, 2020.

[171] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[172] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey Levine. QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *Conference on Robot Learning*, pages 651–673, 2018.

[173] Arjun Raj and Alexander van Oudenaarden. Nature, nurture, or chance: Stochastic gene expression and its consequences. *Cell*, 135(2):216–226, 2008.

[174] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[175] Saurabh Kumar, Harshit Sampson, and Avijit Behera. Benchmarking of deep reinforcement learning algorithms for vision-based robotics. *arXiv preprint arXiv:2201.04224*, 2022.

[176] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations, 2021.