BOSTON UNIVERSITY

GRADUATE SCHOOL OF ARTS AND SCIENCES

A Prospectus for Ph.D. Dissertation

Minimizing the Intent-to-Reality Gap

By

BASSEL EL MABSOUT

Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy
In the Department of Computer Science

COMMITTEE MEMBERS:

DR. RENATO MANCUSO *First Reader*

DR. SABRINA NEUMAN

DR. KATE SAENKO

DR. BINGZHUO ZHONG

January 15, 2025

1 Abstract

This dissertation research introduces and addresses the intent-to-reality gap in robot learning—the challenge of translating high-level intentions into deployable policies. Practitioners face two key obstacles: expressing their intentions as learning objectives (the intent-to-behavior gap) and maintaining policy behavior when moving from simulation to reality (the sim-to-real gap). Current approaches lack principled structure for specifying objectives and rely on optimization targets that break down during transfer, leading to catastrophic forgetting when policies fail to preserve learned behaviors.

I introduce Expressive Reinforcement Learning as a subfield focused on minimizing the intent-to-behavior gap through structured objective specification and composition. My early work demonstrated the promise of this design space by developing RE+AL, the first reinforcement learning framework to outperform classical PID controllers on quadrotor attitude control. Countless hours wrestling with brittle reward functions and watching otherwise-promising policies fail in spectacular ways taught me the need for more principled foundations. This led to the focus around objective specification and composition through which many methods were developed such as: (1) Conditioning for Action Policy Smoothness, allowing a robot actuating in reality to reduce its power consumption by 80% while maintaining good performance, (2) anchor critics for preventing catastrophic forgetting during sim-to-real transfer, and (3) an Algebraic Q-value Scalarization (AQS) method that minimizes the intent-to-behavior gap through intuitive objective composition producing a 600% improvement in sample efficiency. Through validation on increasingly complex robotic systems, I show how principled abstractions enable practitioners to directly express and compose their intentions, bridging the gap from intent to reality.

Contents

2 Problem Statement and Objectives

The fundamental challenge in robot learning lies in bridging the **intent-to-reality gap**—the disparity between a practitioner's intended robot behavior and what is achieved in reality (Def 3.2.1.2). While reinforcement learning offers powerful tools for developing complex controllers [4], [5], [6], [7], three critical problems limit its widespread adoption in the real-world [8], [9], [10]:

First, practitioners struggle to translate their high-level intentions into precise learning objectives. Current approaches rely on *brittle linear reward composition* and manual tuning [11], [12], [13], [14], [15], making it difficult to express and balance multiple objectives [16], [17]. This **intent-to-behavior gap** (Def 3.2.1.1) leads to policies that either fail to learn desired behaviors or require prohibitive engineering effort to achieve them [18].

Second, even when policies perform well in simulation, they often fail to preserve learned behaviors when deployed to real systems [19]. Traditional approaches to sim-to-real transfer treat it as a domain adaptation problem [20], leading to **catastrophic forgetting** [21] where policies maintain performance on common scenarios but fail in critical edge cases [22]. This challenge is particularly acute in robotics, where failures in rare but important scenarios can lead to system damage or safety violations [23].

Third, learned policies often exhibit *undesirable behaviors* that increase computational and energy demands [24]. High-frequency oscillations in control signals can cause system overheating and excessive power consumption [25], while complex neural architectures strain limited computational resources [26]. These challenges are particularly relevant for resource-constrained robotic systems that must operate efficiently in the real world.

I argue that addressing these challenges requires **fundamentally rethinking** (see Section 3.2) how practitioners express and compose their intentions throughout the learning process. Instead of focusing solely on optimization algorithms [27], [28], [29], [30], existing works provide motivation for developing principled frameworks for:

1. Specifying behavioral objectives in ways that capture intended relationships and trade-offs [31], [32], [33]
2. Preserving learned behaviors during transfer while enabling controlled adaptation [34], [35]
3. Ensuring efficient deployment through resource-aware policy design [24], [36]

My research addresses these challenges through four interconnected objectives:

1. **Structured Objective Specification:** Develop a principled framework for composing behavioral objectives that enables practitioners to directly express intended relationships and trade-offs. This includes both the mathematical foundations (Section 3.2.2) for combining objectives and practical tools for specifying complex behaviors Section 5.2.

2. **Intention-Preserving Transfer:** Create methods that maintain critical behaviors during sim-to-real transfer (Section 3.1.5) by treating adaptation as multi-objective optimization over

Q-values from both domains (Section 5.1). This ensures policies can adapt to reality while preserving behaviors learned in simulation.

3. **Resource-Aware Control:** Design techniques for learning controllers that are efficient in both computation and energy usage (Def 3.2.3.2). This spans from action smoothness for energy efficiency to architectural optimization for reduced inference cost (Section 4.3).
4. **Practical Validation:** Demonstrate the effectiveness of these approaches through systematic evaluation on increasingly complex robotic systems, from benchmark simulation scenarios to racing quadrotor attitude control. This includes developing open-source tools and system designs to enable broader adoption (Section 5.1).

Through these objectives, I aim to transform how practitioners develop robotic systems by providing principled ways to express intentions, preserve behaviors, and ensure efficient deployment. The following sections detail my progress toward these goals and outline the remaining work needed to advance this vision.

3 Definitions and Literature Review

3.1 Standard Fundamentals

3.1.1 Discrete-Time Markov Decision Processes

The standard formalization of sequential decision making [37], [38], defined by a tuple $\langle S, A, \mathbb{P}, R, \gamma \rangle$, where S is the state space, A is the action space, $\mathbb{P}(s_{t+1}|s_t, a_t)$ defines state transition probabilities, R specifies rewards, and γ is a discount factor. In robot learning, states typically represent sensor readings and physical configurations, while actions correspond to motor commands.

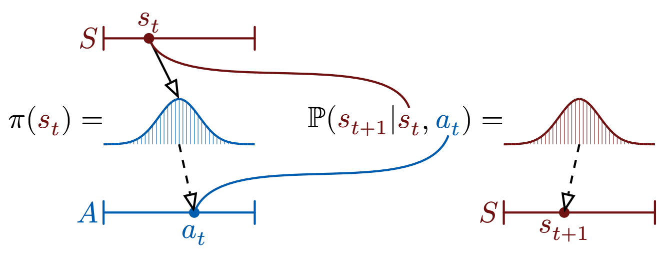

Figure 1: The Markov Decision Process showing how a state s_t in the state space S and action $a_t \sim \pi(s_t)$ determine the probability distribution $\mathbb{P}(s_{t+1}|s_t, a_t)$ over next states. Dashed arrows (- -▷) indicate sampling from a distribution.

3.1.2 Policies and Returns

A policy $\pi(s_t)$ defines the probability distribution over actions A in state s_t. The goal in RL is to find a policy that maximizes expected returns, defined as the discounted sum of rewards:

$\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t, a_t, s_{t+1})\right]$. The discount factor γ determines how much to prioritize immediate versus future rewards.

3.1.3 Value Functions and Q-Functions

A Q-value function Q^π represents the expected return when following policy π from state s_0 and action a_0:

$$Q^\pi(s_0, a_0) = \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t, a_t, s_{t+1}) \mid s_{t+1} \sim \mathbb{P}(s_{t+1}|s_t, a_t), a_t \sim \pi(s_t)\right] \tag{1}$$

Similarly, a value function V^π represents the expected return when in state s_0 and then following π:

$$V^\pi(s_0) = \mathbb{E}[Q^\pi(s_0, a_0) \mid a_0 \sim \pi(s_0)] \tag{2}$$

These functions form the basis of deep reinforcement learning [39], as they are commonly approximated by neural networks and represent the performance of a policy according to our notion of optimality, namely, maximizing the expected reward.

These functions are also central to my work in two key ways:

1. As a basis for objective composition in AQS, where normalized Q-values enable intuitive specification of behavioral trade-offs
2. As anchors for domain transfer, where simulation Q-values preserve critical behaviors during real-world adaptation

3.1.4 Multi-Objective RL

Multi-objective reinforcement learning extends the standard MDP framework to handle multiple reward signals simultaneously [16], [40]. Instead of a single scalar reward R, the agent receives a vector of rewards $\vec{R}$ corresponding to different objectives. This introduces the challenge of balancing competing objectives [33], forcing us to eventually make a choice of what objectives are more important than others, the scalarization function that combines them into a single objective is called the utility function. Much MORL work focuses on defining and finding Pareto-optimal policies [17] that make principled trade-offs between different goals.

3.1.5 Sim-to-Real Transfer

The process of deploying policies trained in simulation to real-world systems [19], encompassing both the technical challenges of domain adaptation [41] and the practical constraints of physical deployment [42]. This fundamental concept bridges the gap between idealized training environments and the complexities of real-world operation [20].

3.2 Introduced Conceptual Framework

3.2.1 Core Concepts

3.2.1.1 The Intent-to-Behavior Gap: The fundamental challenge of translating a practitioner's intended policy behavior into a mathematical specification that reliably produces that behavior when optimized [12], [18]. This gap exists in any reinforcement learning system, as practitioners must convert high-level intentions (like "move smoothly and efficiently") into concrete optimization objectives. Traditional approaches leave practitioners to define scalar rewards and combine objectives through linear scalarization, which often fails to capture the nuanced relationships between different behavioral aspects. More sophisticated specification techniques, such as algebraic composition methods or formal logic frameworks [31], [32], aim to minimize this gap by providing principled tools for expressing behavioral requirements.

3.2.1.2 The Intent-to-Reality Gap: A compound challenge in robot learning that combines the intent-to-behavior gap (the challenge of specifying desired behaviors through mathematical objectives) with the sim-to-real gap (the challenge of preserving these behaviors during real-world deployment) [19], [22]. This gap represents the full distance between a practitioner's intentions and the actual behavior of deployed systems [8], [9], [10], making it a central challenge in practical robot learning.

3.2.1.3 Expressive Reinforcement Learning (XRL): A subfield of reinforcement learning focused on minimizing the intent-to-behavior gap through principled frameworks for objective specification and composition. Key aspects include:

1. Algebraic approaches to objective composition that maintain semantic meaning, building on foundational work [43] (e.g., using power-mean operators for intuitive combination of Q-values as in AQS)
2. Formal methods for specifying policy behaviors [31] (e.g., temporal logic for constraint specification [32])

The key distinction from traditional multi-objective RL [16], [33] is the focus on providing practitioners with mathematically grounded tools for expressing objective composition. While early work established foundational principles [44], [45], modern approaches emphasize intuitive specification methods that preserve semantic intent [11], [17]. This addresses limitations in traditional reward engineering methods that often require significant domain expertise and manual tuning [13], [14], [15].

3.2.1.4 Behavioral Retention: The degree to which a policy maintains its learned behaviors when transferred to new domains or adapted to new conditions [21], [46]. We quantify this through multi-objective evaluation of Q values across domains, particularly focusing on performance in critical scenarios that may occur rarely but have high importance [23], [47]. This provides a more nuanced view than traditional average-case metrics [8], [9].

3.2.1.5 Behavioral Objectives: We use this term to encompass both the high-level intentions a practitioner wants to achieve and their formal expression as optimization targets. Our work demonstrates this through several key examples: smooth motor control achieved through temporal and spatial action similarity in CAPS, preservation of critical scenario performance through Q-value anchoring, resource efficiency through architectural optimization, and complex objective composition through algebraic Q-value operations in AQS. This structured approach distinguishes our objectives from simple reward functions, emphasizing their principled nature and composability.

3.2.2 Mathematical Framework

3.2.2.1 Power-Mean as a Logical Operator:

$$\overline{\mu}_p(x_1, ..., x_n) = \left(\frac{1}{n}\sum_{i=1}^{n} x_i^p\right)^{\frac{1}{p}} \tag{3}$$

where $x_i \in [0, 1]$ are the values being composed. The parameter $p \in \mathbb{R}$ controls the attraction to larger or smaller values, with several notable special cases:

Parameter	**Name**	**Operation**
$p \to -\infty$	Minimum	$\min(x_1, ..., x_n)$
$p = -1$	Harmonic Mean	$\frac{1}{\sum_{i=1}^{n} \frac{1}{x_i}}$
$p = 0$	Geometric Mean	$\left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}}$
$p = 1$	Arithmetic Mean	$\frac{1}{n}\sum_{i=1}^{n} x_i$
$p \to \infty$	Maximum	$\max(x_1, ..., x_n)$

For finite values, $p < 0$ biases toward the minimum value (pessimistic composition), while $p > 0$ biases toward the maximum (optimistic composition). The power-mean has several key properties:

1. **Range Preservation:** If $x_i \in [0, 1]$ then $\overline{\mu}_p(x_1, ..., x_n) \in [0, 1]$
2. **Commutativity:** $\overline{\mu}_p(x_1, ..., x_n) = \overline{\mu}_p\left(x_{\sigma(1)}, ..., x_{\sigma(n)}\right)$ for any permutation σ
3. **Idempotence:** $\overline{\mu}_p(x, ..., x) = x$
4. **Monotonicity in x:** If $x_i \leq x_j$ then $\overline{\mu}_p(x_1, ..., x_i, ..., x_n) \leq \overline{\mu}_p\left(x_1, ..., x_j, ..., x_n\right)$
5. **Monotonicity in p:** If $p < q$ then $\overline{\mu}_p(x_1, ..., x_n) \leq \overline{\mu}_q(x_1, ..., x_n)$
6. **$\wedge$ semantics at 0 and 1:** if $x_i \in \{0, 1\}$ and $p \leq 0$ then $\overline{\mu}_p(x_1, ..., x_n) = x_1 \wedge ... \wedge x_n$
7. **$\vee$ semantics at 0 and 1:** if $x_i \in \{0, 1\}$ and $p \leq 0$ then

$$1 - \overline{\mu}_p(1 - x_1, ..., 1 - x_n) = x_1 \vee ... \vee x_n$$

These properties are especially useful for use as a continuous logical operator. It allows for smooth interpolation between different logical operators with p being small representing notions similar as that of an "and", and p being large representing a notion of an "or". Being able to smoothly interpolate between these two extremes allows for a more intuitive specification of objective composition. And range preservation allows for the operator to be numerically stable.

3.2.3 Evaluation Metrics

3.2.3.1 Action Roughness: A quantitative measure of high-frequency components in a policy's action outputs, first introduced in "Regularizing Action Policies for Smooth Control with Reinforcement Learning" Section 4.2 (where we incorrectly termed it a "smoothness metric", as lower values indicate more smoothness). We measure action roughness through frequency analysis of control signals using:

$$\text{Roughness} = \sum_{i=1}^{n} \frac{X_i f_i}{\sum_{i=1}^{n} X_i f_s} \tag{4}$$

where X_i is the amplitude of the i^{th} frequency component f_i, and f_s is the sampling frequency. This metric provides the mean weighted normalized frequency of action outputs, with higher values indicating more high-frequency components in the control signal. In motor control applications, high roughness values can lead to catastrophic outcomes including motor burnout, while lower values indicate more sustainable actuation patterns. While originally introduced as a smoothness metric, we adopt the more precise term "roughness" here as the metric directly measures the presence of high-frequency components rather than their absence. This metric addresses a critical gap in the field where action signal quality was previously assessed through ad-hoc or qualitative methods, making it difficult to predict potential hardware damage before deployment.

3.2.3.2 Resource Efficiency: A comprehensive measure encompassing multiple dimensions of system efficiency [24], [36]:

1. **Power Efficiency:** Direct energy consumption by actuators and hardware, quantified through metrics like average power draw and peak current demands [25]. This includes both steady-state operation costs and transient spikes from high-frequency control signals [8].
2. **Computational Efficiency:** Resources required for policy execution [26], including:
   - Inference time: Latency between state observation and action selection
   - Memory usage: Both runtime memory and model parameter storage
   - Hardware utilization: CPU/GPU load and thermal considerations
3. **Training Efficiency:** Resources consumed during learning [48], [49]:
   - Sample efficiency: Number of environment interactions needed
   - Gradient steps: Computational work per data point
   - Wall-clock time: Total duration of training process
   - Memory requirements: Both for replay buffers and gradient computation

These metrics are particularly crucial for resource-constrained robotic systems [14], where inefficiencies in any dimension can make deployment impractical.

3.3 Related Work

This section reviews key developments in reinforcement learning that address the intent-to-reality gap in robot learning, organized around three fundamental challenges: objective specification, sim-to-real transfer, and energy-efficient control in the real world.

3.3.1 Objective Specification and Composition

Traditional reinforcement learning approaches often struggle to capture complex behavioral requirements, leading practitioners to develop increasingly sophisticated methods for reward specification. A compelling example is found in tokamak plasma control [11], where researchers implicitly developed a structured approach to composing control objectives to manage the extreme complexity of fusion reactor control. Their success in this challenging domain demonstrates both the necessity and natural emergence of formal frameworks for objective specification. However, their approach, while effective, required significant domain expertise to develop and lacked explicit formalization. This highlights a key challenge: how to make such structured specification approaches more accessible while maintaining their expressive power. Several research directions have emerged to address this challenge:

3.3.1.1 Multi-Objective RL: Classical approaches focus on finding Pareto-optimal policies [17], [33] for competing objectives, but often lack intuitive ways to specify trade-offs [16]. These methods typically rely on linear scalarization or constrained optimization [44], [45], which can be limiting when objectives have complex interactions. The tokamak control problem [11] exemplifies these limitations, where practitioners had to develop sophisticated objective structures implicitly to achieve the required control performance.

3.3.1.2 Formal Methods: Temporal logic frameworks [31] and propositional logic approaches [32] provide rigorous specifications but can be challenging for practitioners to use effectively with reinforcement learning. Recent work has explored more accessible formal methods that maintain mathematical rigor while improving usability [50]. This represents a step toward formalizing the kinds of structured approaches that emerge naturally in complex domains like tokamak control.

3.3.1.3 Expressive RL: Early foundational work explored algebraic approaches to objective composition [43], [51], establishing mathematical principles for combining objectives. Building on these foundations, modern approaches to structured reward design [13] have focused on maintaining semantic meaning during composition. These methods aim to make explicit the kinds of objective structures that practitioners develop implicitly, providing a formal foundation for composing objectives that preserves their semantic intent. This addresses a limitation of work that has focused on pure reward engineering [11], where the structured composition of objectives remained implicit in the domain expertise rather than being formally captured.

3.3.2 Sim-to-Real Transfer

The challenge of transferring policies from simulation to reality remains central to practical robot learning [19]. Current approaches broadly fall into two categories:

3.3.2.1 Environment-Focused Methods: Domain randomization [41] and adaptive architectures [14] attempt to bridge the reality gap through robust training. However, these approaches often struggle with accurately modeling complex real-world dynamics [52] and can lead to oscillatory control responses [8].

3.3.2.2 Policy-Focused Methods: While fine-tuning approaches can adapt to real systems, they struggle with catastrophic forgetting [21], where policies maintain high rewards on common scenarios but fail on rare, critical cases [46]. Recent work has shown that structured training environments [20] and multi-objective optimization across domains [22] can help maintain critical behaviors. Direct policy optimization through regularization [53] has also shown promise in improving transfer success.

3.3.3 Efficient Deployment

Practical deployment requires policies that are efficient in both computation and physical resource usage [24], [36]. Three key areas have emerged:

3.3.3.1 Control Optimization: Explicit regularization [54] and adaptive control methods [55], [56] help maintain smooth and efficient behavior. This builds on classical work in optimal control [57], while addressing the unique challenges of learned policies, particularly in reducing power consumption and system wear [53].

3.3.3.2 Architectural Efficiency: Recent advances in asymmetric actor-critic methods [49] and network compression [58] demonstrate that smaller networks can achieve comparable performance. Liquid neural networks [26], [59] show promise for adaptive control with reduced model complexity.

3.3.3.3 Resource-Aware Design: Joint optimization of computational and physical efficiency [14] has become crucial for practical robotics, particularly in embedded systems [24] where network dependencies can limit autonomy [36].

3.3.4 Open Challenges

Several fundamental challenges remain in bridging the intent-to-reality gap:

1. **Objective Specification:** While formal methods and expressive frameworks provide tools for composition [31], [32], specifying complex objectives in a way that reliably produces intended behaviors remains difficult [11], [12].
2. **Effective Transfer:** Current approaches lack effectiveness in behavioral preservation [21], [46], particularly for safety-critical scenarios [23]. This limits deployment in high-stakes applications [47].

This dissertation addresses these challenges through a unified approach that combines expressive objective specification which is used to achieve robust transfer and efficient deployment. Through frameworks like AQS, CAPS, and Anchor Critics, we demonstrate how structured approaches to reinforcement learning can minimize the intent-to-reality gap in practical robot learning.

4 Published Results

4.1 Foundations: How to Train Your Quadrotor [1]

Our journey began with a fundamental challenge in robotics: achieving reliable low-level control of quadrotor attitude that could outperform classical PID controllers. While PID controllers were straightforward to use, they couldn't learn from experience or adapt to the environment. Previous attempts to apply reinforcement learning to this problem faced two critical issues: poor transfer from simulation to reality (with only one in dozens of trained agents proving controllable on real drones) and unstable, non-smooth control leading to excessive power consumption and hardware wear.

Through careful analysis, we identified a fundamental limitation in how objectives were being specified: the standard practice of linear reward composition required constant manual tuning based on observed behavior, leading to brittle policies that failed to capture true behavioral objectives. This led us to develop RE+AL, which introduced multiplicative reward composition as an alternative that better preserved the intent of each objective, improving behavioral retention (Def 3.2.1.4) through more robust policy learning.

The results were transformative. We achieved the first successful application of reinforcement learning to outperform classical PID controllers on quadrotor attitude control. The approach yielded a 10$\times$ reduction in training time, dropping from 9 hours to under 50 minutes. Additionally, we achieved significant improvements in control signal quality, reducing oscillations from 330Hz to 130Hz while maintaining low tracking errors of 4.2 deg/s.

These improvements directly addressed resource efficiency (Def 3.2.3.2) through reduced training time and better control efficiency.

This early work revealed a crucial insight that would shape my subsequent research: the gap between practitioner intent and realized behavior wasn't just about improving simulators—it was about how we specified objectives to the learning system. This realization led me to develop the concepts of intent-to-behavior and intent-to-reality gaps, and the principles of Expressive Reinforcement Learning.

4.2 Conditioning for Action Policy Smoothness (CAPS) [2]

A critical but often unaddressed challenge in deploying learned policies to real robots is the prevalence of oscillatory control responses. While deep reinforcement learning enables training control policies for complex dynamical systems, these policies often exhibit problematic behaviors detrimental to system integrity. These oscillations manifest as:

- Potential hardware failures due to sustained stress
- Visible physical system oscillations affecting performance
- Increased power consumption and overheating from high-frequency control signals

The challenge is particularly acute in continuous control domains, where controller responses can vary infinitely within acceptable output limits. While we can help shape behavior using reward engineering, we identified that certain objectives - like smoothness - can be directly encoded into the policy optimization process itself. This led to a fundamental insight: by explicitly specifying these objectives through policy regularization, we could work in conjunction with reward signals to shape the desired outcomes.

Our Conditioning for Action Policy Smoothness (CAPS) addressed this by directly shaping the neural network's mapping from states to actions through two complementary objectives, implemented as regularization terms:

1. **Temporal smoothness**: Actions should maintain similarity with previous actions to ensure smooth transitions in controller outputs over time
2. **Spatial smoothness**: Similar states should map to similar actions, providing robustness against measurement noise and modeling uncertainties

We formalize these smoothness objectives through regularization terms:

4.2.0.1 Temporal Smoothness Loss:

$$\mathcal{L}_{\text{temporal}} = \frac{1}{T}\sum_{t=1}^{T} \|a_t - a_{t+1}\|_2 \tag{5}$$

where $\mathcal{L}_{\text{temporal}}$ measures the average change in actions over time, encouraging smooth transitions between consecutive control outputs.

4.2.0.2 Spatial Smoothness Loss:

$$\mathcal{L}_{\text{spatial}} = \mathop{\mathbb{E}}_{s_1,s_2}\left[\frac{\|a_1 - a_2\|_2}{\|s_1 - s_2\|_2}\right] \tag{6}$$

where $\mathcal{L}_{\text{spatial}}$ measures the average Lipschitz constant of the policy, ensuring similar states map to similar actions.

The total loss combines these terms with the policy objective:

4.2.0.3 CAPS Total Loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{policy}} + \alpha\,\mathcal{L}_{\text{temporal}} + \beta\,\mathcal{L}_{\text{spatial}} \tag{7}$$

where α and β are hyperparameters controlling the trade-off between objectives.

Key results include:

- 80% reduction in power consumption while maintaining task performance
- 96% improvement in policy roughness see (Def 3.2.3.1)
- Significant reduction in training time
- Successful flight-worthy controllers using simple reward structures

This work demonstrated a key principle: some objectives are better expressed directly through policy structure rather than through reward engineering. By directly encoding smoothness objectives into the policy structure, CAPS addresses simulation-to-reality transfer Section 3.1.5 by ensuring consistent behavior in reality and resource efficiency (Def 3.2.3.2) through reduced power consumption and hardware stress.

4.2.1 Impact

The success of CAPS in achieving smooth control has led to a large body of subsequent works that have been built on our notions of policy smoothness and on similar derivative notions. This exemplifies how well-structured specifications is desired and can simultaneously address multiple aspects of the intent-to-reality gap, from resource constraints to behavioral requirements.

4.3 Efficient Control through Actor-Critic Asymmetry [3]

While CAPS addressed energy efficiency through smoother actions, the computational cost of neural network inference remained a significant barrier to deployment. This led us to question a fundamental implicit assumption in actor-critic methods: that actors and critics should share similar neural network architectures.

A key insight emerged when we examined the different objectives these networks need to satisfy: the critic must develop rich representations to understand both system dynamics and reward structures, while the actor simply needs to learn a mapping that maximizes value. This suggested that the representational power needed to satisfy the actor's objectives might be much smaller than what's needed for the critic's objectives - yet the standard practice was to use identical architectures for both components. We thus developed a binary-search based algorithm that automatically searches for the smallest actors that can solve the tasks at hand.

Through systematic evaluation across multiple actor-critic algorithms and environments, we demonstrated that:

- Actors can be reduced by up to 99% in size while maintaining performance
- Average reduction of 77% in model parameters across tasks
- Successful validation across 4 popular actor-critic algorithms
- Consistent results across 9 different environments with varying dynamics

This work revealed a fundamental asymmetry in the representational requirements needed to satisfy actor versus critic objectives. This insight directly addresses resource efficiency (Def 3.2.3.2) by dramatically reducing computational requirements.

5 Current Work

5.1 Anchored Learning for On-the-Fly Adaptation

While direct on-robot training raises safety and cost concerns, making simulated training attractive, the transition from simulation to reality presents significant challenges [22], [60]. Real-world control scenarios can generate distributions of trajectories where critical scenarios occur only in the tails. This makes the expected discounted sum of rewards, and thus, the associated Q-value, an inadequate measure of agent performance across important tasks. Recent works have identified that traditional fine-tuning approaches often result in *catastrophic forgetting*, where policies lose their previously well-behaved performance when adapting to new environments [21], [46]. Indeed, we observed this phenomenon across every benchmark task we tested.

This challenge directly relates to behavioral retention (Def 3.2.1.4) - the ability of a policy to maintain its learned behaviors when transferred to new domains. Traditional approaches focus on average-case performance, but fail to preserve behavior in critical scenarios that occur rarely in real data but are crucial for safe operation.

We formalized this challenge through Q-value optimization:

5.1.0.1 Anchor Score:

$$O^{\pi}_{\text{retention}} = \mathop{\mathbb{E}}_{s \in S_{\text{critical}}} \left[Q^{\pi}_{\text{sim}}(s, a) \mid a \sim \pi(s) \right] \tag{8}$$

where $O_{\text{retention}}$ quantifies how well we satisfy the objective that the policy preserves behaviors learned in simulation across critical scenarios S_{critical}. We specifically use this value to “anchor” policies to the objective of satisfying good performance on critical scenarios in simulation.

5.1.0.2 Smoothness Loss:

$$O^{\pi}_{\text{real}} = \overline{\mu}_0 \left(Q^{\pi}_{\text{real}}, \frac{w_T}{w_T + \mathcal{L}_T}, \frac{w_S}{w_S + \mathcal{L}_S}, \frac{w_A}{w_A + \pi^{:-1}} \right) \tag{9}$$

where $\mathcal{L}_T$ and $\mathcal{L}_S$ are temporal and spatial smoothness terms from Equation 5 and Equation 6, Q_{π} is the critic’s Q-value, $\pi^{:-1}$ is the pre-activation policy output, and w_T, w_S, w_A are penalty thresholds that normalize the losses to [0,1]. The geometric mean ($\overline{\mu}_0$) ensures all objectives must be satisfied simultaneously while maintaining the [0,1] bounds established in Equation 11.

This composition proved more effective for preserving smoothness during transfer, achieving:

- 47% reduction in control jerk compared to additive composition
- 52% lower power consumption on real hardware
- Maintained tracking performance within 5% of simulation baseline

5.1.0.3 Combined Objective:

$$O^{\pi}_{\text{combined}} = \overline{\mu}_0(O^{\pi}_{\text{real}}, O^{\pi}_{\text{critical}}) \tag{10}$$

where $O^{\pi}_{\text{combined}}$ represents the overall objective that balances real-world adaptation with behavioral retention.

Building on our experience with CAPS, we also incorporate smoothness objectives multiplicatively:

This unifies our three key insights:
- Smoothness through direct policy regularization (CAPS)
- Behavioral retention through Q-value anchoring
- Objective composition through power-mean operators (AQS)

The multiplicative composition creates a natural AND relationship between objectives - both temporal and spatial smoothness must be satisfied simultaneously as well as performance on critical scenarios and performance in reality.

Thus we can preserve important behaviors learned in simulation while still allowing the policy to adapt to reality.

Another important aspect of this work is that we built an open-source system called SwaNNLake and a firmware called SwaNNFlight which enables on-board inference with efficient ground station communication for policy updates, making our approach viable for real-world applications requiring live adaptation and real-time control responses.

In this paper we show:
- 50% reduction in power consumption while maintaining stable flight
- Preservation of critical behaviors while adapting to real-world conditions
- Successful validation in both sim-to-sim and sim-to-real scenarios
- Experimental results which show how large of an issue catastrophic forgetting is even when the system dynamics are kept the same

5.2 Algebraic Q-value Scalarization (AQS)

Focusing specifically on the intent-to-behavior gap (Def 3.2.1.1), we developed an XRL (Def 3.2.1.3) method called AQS, a novel domain-specific language that allows practitioners to express how different objectives should interact using the power-mean operator. Rather than focusing on reward engineering, AQS enables direct specification of when one objective should take priority over another based on their current satisfaction levels. Our contributions include:
1. Using the power-mean as a logical operator over normalized Q-values
2. Q-value scalarization instead of traditional reward scalarization
3. Q-value normalization for stable learning across objectives
4. Integration with a new DDPG-based algorithm called Balanced Policy Gradient (BPG)

This approach fundamentally changes how we express objectives in reinforcement learning. Rather than trying to encode complex behaviors through reward engineering, AQS encourages practioners to consider simple definitions at the reward level and composition rules at the Q-value level. This provides an intuitive language for specifying how objectives should be prioritized. For example, if one objective is nearly satisfied while another is not, AQS can naturally express that the unsatisfied objective should take priority - similar to the semantics of an AND operator, but generalized to continuous values as described in (Def 3.2.2.1).

While the work in Section 4.2 and Section 5.1 showed the power of Q-values for preserving behaviors across domains, AQS demonstrates their potential in full generality for expressing complex objective relationships. By treating rewards as continuous constraints to be satisfied, we can then normalize Q-values so that we can treat them as continuous measures of objective satisfaction:

5.2.0.1 Q-value Normalization:

$$Q_{\text{norm}}(s, a) = \frac{Q(s, a)}{1 - \gamma} \tag{11}$$

assuming that $\forall_{s,a}, 1 \geq R(s, a) \geq 0$ The normalized Q-value Q_{norm} is bounded to [0,1], representing the relative satisfaction of an objective.

This normalization enables the use of power-mean operators (see Equation 3) to express logical relationships between objectives.

Our results demonstrated comprehensive improvements across multiple dimensions. We achieved up to 600% improvement in sample efficiency compared to Soft Actor Critic, along with substantial reductions in policy variability. This reduces the need for reward engineering and speeds up the iteration loop for developing new behaviors.

6 Timeline and Milestones

The completion of this dissertation involves finalizing two key publications and writing the dissertation itself:

January 2024:

- Present this dissertation proposal to committee (January 22nd)
- Submit AQS paper to ICML (January 30th)
  - Write example specifications for the various domains
  - Complete comparative analysis with recent MORL methods
  - Finish adding proofs and code release

February 2025:

- Submit Anchor Critics paper
  - Refine related works section and introduce the definition of the objectives framework as a contribution
  - Clean up SWANNFlight firmware documentation

- Prepare manuscript and supplementary materials

March - April 2025:

- Write and complete dissertation
  - Integrate existing published work (RE+AL, CAPS, Actor-Critic)
  - Develop unified theoretical framework
  - Formalize the concept of the intent-to-behavior gap
  - Present methodology for measuring progress on minimizing the aforementioned gaps
  - Write core chapters
  - Incorporate all paper contributions
  - Submit dissertation draft to committee
  - Dissertation defense (April)

This timeline builds on my already published work (RE+AL, CAPS, and Actor-Critic papers) while ensuring thorough completion of the remaining publications and dissertation writing.

Bibliography

[1] B. Mabsout, S. Mysore, K. Saenko, and R. Mancuso, "How to train your quadrotor: A framework for consistently smooth and responsive flight control via reinforcement learning," *ACM TCPS*, vol. 5, no. 4, pp. 1–24, 2021.

[2] B. Mabsout, S. Mysore, R. Mancuso, and K. Saenko, "Regularizing Action Policies for Smooth Control with Reinforcement Learning," in *IEEE ICRA*, 2021, pp. 1810–1816. doi: 10.1109/ICRA48506.2021.9561138.

[3] S. Mysore, B. E. Mabsout, R. Mancuso, and K. Saenko, "Honey. I Shrunk The Actor: A Case Study on Preserving Performance with Smaller Actors in Actor-Critic RL," *2021 IEEE Conference on Games (CoG)*, pp. 1–8, 2021, [Online]. Available: https://api.semanticschola r.org/CorpusID:235489652

[4] V. Mnih *et al.*, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[5] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[6] D. Silver *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[7] J. Schrittwieser *et al.*, "Mastering atari, go, chess and shogi by planning with a learned model," *Nature*, vol. 588, no. 7839, pp. 604–609, 2020.

[8] A. R. Mahmood, D. Korenkevych, G. Vasan, W. Ma, and J. Bergstra, “Benchmarking Reinforcement Learning Algorithms on Real-World Robots,” *Conference on Robot Learning*, 2018, [Online]. Available: http://arxiv.org/abs/1809.07731

[9] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, “Benchmarking Deep Reinforcement Learning for Continuous Control,” in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, in ICML'16. New York, NY, USA: JMLR.org, 2016, pp. 1329–1338.

[10] Q. Fu, Z. Han, J. Chen, Y. Lu, H. Wu, and Y. Wang, “Applications of reinforcement learning for building energy efficiency control: A review,” *Journal of Building Engineering*, vol. 50, p. 104165, 2022, doi: https://doi.org/10.1016/j.jobe.2022.104165.

[11] J. Degrave *et al.*, “Magnetic control of tokamak plasmas through deep reinforcement learning,” *Nature*, vol. 602, pp. 414–419, 2022, doi: 10.1038/s41586-021-04301-9.

[12] C. F. Hayes *et al.*, “A Brief Guide to Multi-Objective Reinforcement Learning and Planning,” in *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, 2023, pp. 1988–1990.

[13] A. Molchanov, T. Chen, W. Hönig, J. A. Preiss, N. Ayanian, and G. S. Sukhatme, “Sim-to-(Multi)-Real: Transfer of Low-Level Robust Control Policies to Multiple Quadrotors,” *International Conference on Intelligent Robots and Systems*, 2019, [Online]. Available: http://arxiv.org/abs/1903.04628

[14] J. Hwangbo, I. Sa, R. Siegwart, and M. Hutter, “Control of a Quadrotor With Reinforcement Learning,” *IEEE Robotics and Automation Letters*, vol. 2, pp. 2096–2103, 2017.

[15] A. Kendall *et al.*, “Learning to drive in a day,” in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8248–8254.

[16] D. M. Roijers, P. Vamplew, S. Whiteson, and R. Dazeley, “A survey of multi-objective sequential decision-making,” *Journal of Artificial Intelligence Research*, vol. 48, pp. 67–113, 2013.

[17] T. T. Nguyen, N. D. Nguyen, P. Vamplew, S. Nahavandi, R. Dazeley, and C. P. Lim, “A multi-objective deep reinforcement learning framework,” *Engineering Applications of Artificial Intelligence*, vol. 96, p. 103915, 2020, doi: https://doi.org/10.1016/j.engappai.2020.103915.

[18] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, “Concrete Problems in AI Safety.” 2016.

[19] W. Zhao, J. P. Queralta, and T. Westerlund, “Sim-to-real transfer in deep reinforcement learning for robotics: a survey,” in *IEEE SSCI*, 2020, pp. 737–744.

[20] F. Sadeghi, A. Toshev, E. Jang, and S. Levine, “Sim2Real View Invariant Visual Servoing by Recurrent Control,” *CoRR*, 2017.

[21] M. Wolczyk *et al.*, "Fine-tuning Reinforcement Learning Models is Secretly a Forgetting Mitigation Problem." [Online]. Available: https://arxiv.org/abs/2402.02868

[22] F. Muratore, F. Ramos, G. Turk, W. Yu, M. Gienger, and J. Peters, "Robot Learning From Randomized Simulations: A Review," *Frontiers in Robotics and AI*, vol. 9, Apr. 2022, doi: 10.3389/frobt.2022.799893.

[23] Q. Yang, T. D. Simão, N. Jansen, S. H. Tindemans, and M. T. J. Spaan, "Reinforcement Learning by Guided Safe Exploration," in *ECAI 2023*, IOS Press, 2023. doi: 10.3233/faia230598.

[24] S. Chinchali *et al.*, "Network offloading policies for cloud robotics: a learning-based approach," *Autonomous Robots*, vol. 45, no. 7, pp. 997–1012, Jul. 2021, doi: 10.1007/s10514-021-09987-4.

[25] W. Koch, R. Mancuso, and A. Bestavros, "Neuroflight: Next Generation Flight Control Firmware," *CoRR*, 2019, [Online]. Available: http://arxiv.org/abs/1901.06553

[26] R. Hasani, M. Lechner, A. Amini, D. Rus, and R. Grosu, "Liquid time-constant networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, pp. 7657–7666.

[27] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *CoRR*, 2017.

[28] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," *International Conference on Learning Representations*, 2016.

[29] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *International Conference on Machine Learning (ICML)*, 2018.

[30] S. Fujimoto, H. Hoof, and D. Meger, "Addressing Function Approximation Error in Actor-Critic Methods," in *International Conference on Machine Learning*, 2018, pp. 1587–1596.

[31] X. Li, C.-I. Vasile, and C. Belta, "Reinforcement learning with temporal logic rewards," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 3834–3839.

[32] K. Nottingham, A. Balakrishnan, J. Deshmukh, and D. Wingate, "Using logical specifications of objectives in multi-objective reinforcement learning," *arXiv preprint arXiv:1910.01723*, 2019.

[33] M. Sakawa and K. Kato, "Interactive decision-making for multiobjective linear fractional programming problems with block angular structure involving fuzzy numbers," *Fuzzy Sets and Systems*, vol. 97, no. 1, pp. 19–31, 1998, doi: https://doi.org/10.1016/S0165-0114(96)00352-1.

[34] K. Arndt, M. Hazara, A. Ghadirzadeh, and V. Kyrki, "Meta Reinforcement Learning for Sim-to-real Domain Adaptation," in *IEEE ICRA*, 2020, pp. 2725–2731. doi: 10.1109/ICRA40945.2020.9196540.

[35] A. Nagabandi *et al.*, “Learning to Adapt in Dynamic, Real-World Environments through Meta-Reinforcement Learning,” *arXiv: Learning*, 2019.

[36] M. Dordevic, M. Albonico, G. A. Lewis, I. Malavolta, and P. Lago, “Computation offloading for ground robotic systems communicating over WiFi - an empirical exploration on performance and energy trade-offs,” *Empirical Software Engineering*, vol. 28, no. 6, Oct. 2023, doi: 10.1007/s10664-023-10351-6.

[37] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.

[38] A. G. Barto, P. S. Thomas, and R. S. Sutton, “Some recent applications of reinforcement learning,” in *Proceedings of the Eighteenth Yale Workshop on Adaptive and Learning Systems*, 2017.

[39] V. Mnih *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, pp. 529–533, 2015, [Online]. Available: https://api.semanticscholar.org/CorpusID:205242740

[40] C. Hayes *et al.*, “A practical guide to multi-objective reinforcement learning and planning,” *Autonomous Agents and Multi-Agent Systems*, vol. 36, p. , 2022, doi: 10.1007/s10458-022-09552-y.

[41] F. Golemo, A. A. Taiga, A. Courville, and P.-Y. Oudeyer, “Sim-to-real transfer with neural-augmented robot simulation,” in *CoRL*, 2018, pp. 817–828.

[42] W. Yu, V. C. Kumar, G. Turk, and C. K. Liu, “Sim-to-real transfer for biped locomotion,” in *IEEE/RSJ IROS*, 2019, pp. 3503–3510.

[43] P. Hájek, “Product Logic, Gödel Logic (and Boolean Logic),” in *Metamathematics of Fuzzy Logic*, Dordrecht: Springer Netherlands, 1998, pp. 89–107. doi: 10.1007/978-94-011-5300-3_4.

[44] C. R. Shelton, “Balancing multiple sources of reward in reinforcement learning,” in *Advances in Neural Information Processing Systems*, 2001, pp. 1082–1088.

[45] A. Abels, D. M. Roijers, T. Lenaerts, A. Nowé, and D. Steckelmacher, “Dynamic weights in multi-objective deep reinforcement learning,” *arXiv preprint arXiv:1809.07803*, 2018.

[46] K. Binici, N. Trung Pham, T. Mitra, and K. Leman, “Preventing Catastrophic Forgetting and Distribution Mismatch in Knowledge Distillation via Synthetic Data,” in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2022, pp. 3625–3633. doi: 10.1109/WACV51458.2022.00368.

[47] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, “End-to-End Safe Reinforcement Learning through Barrier Functions for Safety-Critical Continuous Control Tasks,” *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, pp. 3387–3395, Jul. 2019, doi: 10.1609/aaai.v33i01.33013387.

[48] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

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

[50] Y.-C. Chang, N. Roohi, and S. Gao, "Neural lyapunov control," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA: Curran Associates Inc., 2019.

[51] P. J. Fleming and J. J. Wallace, "How not to lie with statistics: the correct way to summarize benchmark results," *Commun. ACM*, vol. 29, pp. 218–221, 1986.

[52] B. Pasik-Duncan, "Adaptive Control," *IEEE Control Syst.*, vol. 16, p. 87–, 1996.

[53] Q. Shen, Y. Li, H. Jiang, Z. Wang, and T. Zhao, "Deep Reinforcement Learning with Smooth Policy." 2020.

[54] Z. Liu, X. Li, B. Kang, and T. Darrell, "Regularization Matters in Policy Optimization." 2019.

[55] M. Schreier, "Modeling and Adaptive Control of a Quadrotor." p. , 2012. doi: 10.1109/ICMA.2012.6282874.

[56] Y. Song, A. Mavalankar, W. Sun, and S. Gao, "Provably Efficient Model-based Policy Adaptation," in *ICML*, 2020.

[57] J. G. Ziegler and N. B. Nichols, "Optimum Settings for Automatic Controllers," *Journal of Dynamic Systems, Measurement, and Control*, vol. 115, no. 2B, pp. 220–222, 1993.

[58] S. Han, H. Mao, and W. Dally, "Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding," *arXiv: Computer Vision and Pattern Recognition*, 2016.

[59] R. M. Hasani, M. Lechner, A. Amini, D. Rus, and R. Grosu, "Liquid Time-constant Recurrent Neural Networks as Universal Approximators," *CoRR*, 2018, [Online]. Available: http://arxiv.org/abs/1811.00321

[60] S. S. Sandha, L. Garcia, B. Balaji, F. Anwar, and M. Srivastava, "Sim2Real Transfer for Deep Reinforcement Learning with Stochastic State Transition Delays," in *Proceedings of the 2020 Conference on Robot Learning*, J. Kober, F. Ramos, and C. Tomlin, Eds., in Proceedings of Machine Learning Research, vol. 155. PMLR, 2021, pp. 1066–1083. [Online]. Available: https://proceedings.mlr.press/v155/sandha21a.html