

Reinforcement learning for long-horizon decision policies
in autonomous systems

by

Talha Zaidi

BS., University of Engineering and Technology Lahore, 2014

MS., Istanbul Medipol University, 2020

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computer Science
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2026

Abstract

Autonomous physical systems must make sequential decisions whose effects unfold over long temporal horizons. In domains such as space autonomy and embodied robotic control, these decisions are shaped by nonlinear dynamics, physical constraints, delayed feedback, limited interaction, fixed datasets, and imperfect decision context. A policy that is locally effective may still accumulate error, violate feasibility, or select actions unsupported by prior experience. This dissertation studies how reinforcement learning can produce long-horizon decision policies that remain feasible, controllable, support-aware, and robust across extended physical decision processes.

The central thesis is that reinforcement learning becomes more suitable for autonomous physical systems when its learning architecture reflects the temporal and physical structure of the task. The dissertation develops four structured approaches. First, cascaded reinforcement learning decomposes long-horizon precision control into coarse progress and terminal refinement. Second, attention-enhanced actor–critic learning improves feature relevance in complex physical dynamics. Third, support-preserving latent planning enables offline embodied agents to compose long-horizon behavior in a learned skill space while using conservative value estimation and deterministic generative execution to reduce unsupported extrapolation. Fourth, context-robust masked skill inference trains policies to recover missing latent decisions from partial temporal context, improving robustness when decision histories are degraded.

These methods are evaluated in two complementary settings: trajectory-level space autonomy and offline embodied robotic control. Across these settings, the dissertation shows that decomposition, attention, temporal abstraction, conservative value learning, generative execution, and masked inference address different aspects of the long-horizon feasibility gap. The broader contribution is a unified design perspective for reinforcement learning in autonomous physical systems: reliable long-horizon decision policies require learning structures aligned with horizon length, physical feasibility, data support, controllability, and context reliability.

Reinforcement learning for long-horizon decision policies
in autonomous systems

by

Syed Muhammad Talha Zaidi

BS., University of Engineering and Technology Lahore, 2014

MS., Istanbul Medipol University, 2020

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computer Science
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2026

Approved by:

Major Professor
Arslan Munir

Copyright

© Syed Muhammad Talha Zaidi 2026.

Abstract

Autonomous physical systems must make sequential decisions whose effects unfold over long temporal horizons. In domains such as space autonomy and embodied robotic control, these decisions are shaped by nonlinear dynamics, physical constraints, delayed feedback, limited interaction, fixed datasets, and imperfect decision context. A policy that is locally effective may still accumulate error, violate feasibility, or select actions unsupported by prior experience. This dissertation studies how reinforcement learning can produce long-horizon decision policies that remain feasible, controllable, support-aware, and robust across extended physical decision processes.

The central thesis is that reinforcement learning becomes more suitable for autonomous physical systems when its learning architecture reflects the temporal and physical structure of the task. The dissertation develops four structured approaches. First, cascaded reinforcement learning decomposes long-horizon precision control into coarse progress and terminal refinement. Second, attention-enhanced actor-critic learning improves feature relevance in complex physical dynamics. Third, support-preserving latent planning enables offline embodied agents to compose long-horizon behavior in a learned skill space while using conservative value estimation and deterministic generative execution to reduce unsupported extrapolation. Fourth, context-robust masked skill inference trains policies to recover missing latent decisions from partial temporal context, improving robustness when decision histories are degraded.

These methods are evaluated in two complementary settings: trajectory-level space autonomy and offline embodied robotic control. Across these settings, the dissertation shows that decomposition, attention, temporal abstraction, conservative value learning, generative execution, and masked inference address different aspects of the long-horizon feasibility gap. The broader contribution is a unified design perspective for reinforcement learning in autonomous physical systems: reliable long-horizon decision policies require learning structures aligned with horizon length, physical feasibility, data support, controllability, and context reliability.

Table of Contents

List of Figures	xii
List of Tables	xv
Acknowledgements	xviii
1 Introduction	1
1.1 Motivation and Dissertation Framing	1
1.2 Background and Context	3
1.2.1 Reinforcement Learning for Physical Decision Systems	3
1.2.2 The Long-Horizon Feasibility Gap	4
1.2.3 Representative Validation Contexts	5
1.3 Research Gap	6
1.4 Problem Statement	8
1.5 Thesis Statement	9
1.6 Research Questions	10
1.7 Dissertation Contributions	10
1.8 Scope of the Dissertation	12
1.9 Dissertation Organization	13
2 Background and Literature Review	14
2.1 Foundations of Long-Horizon Physical Decision-Making	14
2.1.1 Sequential Decision-Making in Autonomous Physical Systems	14
2.1.2 Reinforcement Learning Foundations	16

2.1.3	The Long-Horizon Decision Problem	17
2.2	Optimization and Deep Reinforcement Learning for Physical Autonomy	18
2.2.1	Classical Optimization and Control for Physical Decision Problems	18
2.2.2	Deep Reinforcement Learning for Physical Autonomy	19
2.2.3	Attention and Representation Learning in Reinforcement Learning	20
2.3	Offline Reinforcement Learning, Temporal Abstraction, and Support Preservation	21
2.3.1	Offline Reinforcement Learning and Distributional Shift	21
2.3.2	Temporal Abstraction and Latent Skill Learning	22
2.4	Generative and Context-Robust Decision Models	23
2.4.1	Generative Models for Decision-Making	23
2.4.2	Context Robustness and Masked Decision Modeling	24
2.5	Representative Validation Contexts	25
2.6	Synthesis: Design Principles for Long-Horizon RL in Autonomous Physical Systems	26
2.6.1	Decompose Long-Horizon Decisions into Structured Stages/Abstractions	26
2.6.2	Use Closed-Loop Policy Learning for Autonomy	26
2.6.3	Improve Feature Relevance through Attention and Representation Learning	26
2.6.4	Preserve Feasibility and Dataset Support	27
2.6.5	Maintain Controllability of Learned Latent Decisions	27
2.6.6	Improve Robustness under Incomplete or Degraded Context	27
2.7	Summary and Transition	28
3	Cascaded Reinforcement Learning for Long-Horizon Precision Control	29
3.1	Problem Setting and Preliminaries	30
3.1.1	Orbital Transfer Terminology	30
3.1.2	Orbit-Transfer Task	31
3.1.3	Initial and Target Conditions	32
3.2	Methodological Challenge and Design Principle	33
3.3	Technical Formulation	34

3.3.1	State Representation	34
3.3.2	Thrust Model and Action Parameterization	34
3.3.3	Orbital Dynamics and Eclipse Modeling	35
3.3.4	Reinforcement Learning Formulation	37
3.4	Learning Objective and Algorithm	38
3.4.1	Gradient-Aided Reward Function	38
3.4.2	Cascaded DRL Algorithm	39
3.5	Experimental Setup	40
3.6	Empirical Results and Analysis	42
3.6.1	Training and Learned Behavior	42
3.6.2	GTO Transfer Results	43
3.6.3	Physical Interpretation	45
3.6.4	Super-GTO Extension	46
3.6.5	Summary of Findings	48
3.7	Discussion	49
3.8	Summary and Transition	50
4	Attention-Enhanced Actor–Critic Learning for Complex Physical Dynamics	51
4.1	Problem Setting and Preliminaries	52
4.1.1	From Decomposition to Feature Relevance	52
4.1.2	Attention as Actor–Critic Feature Coupling	52
4.2	Methodological Challenge and Design Principle	53
4.3	Technical Formulation	54
4.3.1	State, Dynamics, and Perturbation Model	54
4.3.2	Action and Terminal Conditions	56
4.4	Learning Objective and Algorithm	57
4.4.1	Attention-Augmented Actor–Critic Update	57
4.5	Experimental Setup	59

4.6	Empirical Results and Analysis	60
4.6.1	Training Stability and Learned Policies	60
4.6.2	Transfer Performance	61
4.6.3	Attention Mechanism Ablation	64
4.6.4	Robustness Analysis	65
4.7	Discussion	66
4.8	Summary and Transition	67
5	Support-Preserving Latent Planning for Offline Robotic Control	68
5.1	Problem Setting and Preliminaries	69
5.1.1	Offline Latent-Space Control Setting	69
5.2	Methodological Challenge and Design Principle	70
5.2.1	The Support–Controllability Problem	70
5.2.2	GRALP Overview	72
5.2.3	Support-Preservation and Controllability Principle	73
5.3	Technical Formulation	74
5.3.1	Diffusion Decoder for Latent Skill Execution	74
5.3.2	Latent Controllability Regularization	76
5.3.3	Conservative Planning in Latent Space	78
5.4	Learning Objective and Algorithm	80
5.4.1	GRALP Training and Planning Algorithm	80
5.5	Experimental Setup	80
5.6	Empirical Results and Analysis	82
5.6.1	Main D4RL Performance	82
5.6.2	Contact-Rich RoboSuite Manipulation	83
5.6.3	Component Ablations	84
5.6.4	Mechanism Diagnostics	85
5.7	Discussion	87

5.8	Summary and Transition	88
6	Context-Robust Skill Inference for Long-Horizon Offline Control	90
6.1	Problem Setting and Preliminaries	91
6.1.1	Degraded Temporal Decision Context	91
6.2	Methodological Challenge and Design Principle	92
6.2.1	Local Shortcut Reliance in Sequence Planners	92
6.2.2	CRISP Overview	92
6.3	Technical Formulation	94
6.3.1	Masked Skill Inference	94
6.3.2	Advantage-Weighted Masked Learning and Behavior Anchoring	95
6.3.3	Causal Skill Inference and Value-Guided Refinement	96
6.4	Learning Objective and Algorithm	98
6.4.1	CRISP Training and Deployment Pipeline	98
6.5	Experimental Setup	98
6.6	Empirical Results and Analysis	99
6.6.1	Full-Context Performance	99
6.6.2	Context-Dropout Robustness	100
6.6.3	Delayed Key–Door Diagnostic	102
6.6.4	Mask Structure and Component Ablations	103
6.6.5	RoboSuite Results	104
6.7	Discussion	105
6.8	Summary and Transition	106
7	Conclusion and Future Directions	107
7.1	Dissertation Summary	107
7.2	Synthesis of Findings	108
7.3	Limitations	109

7.4	Future Directions	109
7.5	Closing Remarks	110
	Bibliography	111

List of Figures

3.1	Orbital parameters and orbits visualization	30
3.2	Episodic return and convergence time over training for the GTO-2 transfer. DRL-1 and DRL-2 represent the two cascaded reinforcement learning agents.	42
3.3	Predicted thrust angles and thrust values for the GTO-GEO transfer. The action variables are the thrust-direction angles α and β , while the thrust plots show the propulsion availability used by the environment.	43
3.4	Three-dimensional orbit-transfer trajectory for GTO-GEO, where one distance unit is equivalent to the GEO radius.	44
3.5	Orbital-element variation in one complete episode for GTO-1 and GTO-2 to GEO transfer. The plotted variables correspond directly to the terminal feasibility checks.	44
3.6	Orbital-element variation in one complete episode for the super-GTO to GEO transfer.	47
3.7	Three-dimensional orbit-transfer trajectory for super-GTO to GEO, where one distance unit is equivalent to the GEO radius.	48
4.1	SA3C methodology. The framework augments the standard actor-critic update by using critic-related features to modulate actor features through an attention module. The resulting attention-weighted action contributes an additional critic loss term.	57
4.2	Training scores for the GTO-to-GEO and super-GTO-to-GEO scenarios. DRL-1 and DRL-2 denote the first and second cascaded SA3C networks.	61
4.3	Predicted thrust-angle profiles for the GTO-1 to GEO scenario generated by the two cascaded SA3C agents.	62
4.4	Training scores for the cislunar transfer scenarios from super-GTO-2 to NRHO and from super-GTO-2 to patch point.	62

4.5	Orbital-element convergence for GTO and super-GTO transfers to GEO. The first row corresponds to DRL-1 and the second row corresponds to DRL-2.	63
4.6	Orbital-element convergence for the super-GTO-2 to NRHO-L2 9:2 transfer. The cislunar target requires convergence of five orbital elements rather than only the three GEO quantities.	64
4.7	Three-dimensional visualization of the transfer from super-GTO-2 to NRHO. . . .	64
4.8	Trained-agent performance under thrust uncertainties and initial-condition variations for the GTO-GEO setting.	65
5.1	GRALP pipeline. A Conditional Diffusion Autoencoder extracts latent skills from short action-state segments, conservative critics estimate value in latent space, and a causal Transformer predicts the next skill conditioned on recent context and return information.	72
5.2	Latent-action sensitivity with and without Z-force. With Z-force, decoded action changes grow with latent perturbations, while without Z-force the decoder shows weak sensitivity, indicating near-collapse of latent control authority.	77
5.3	Conservative value-guided planning on a multi-modal 2D ring bandit. The diagnostic shows dataset support, reward modes, conservative critic behavior, and CEM contraction toward a high-value supported region.	80
5.4	RoboSuite Pick-and-Place evaluation. The task requires grasping, transport, and placement under contact-rich manipulation dynamics.	84
5.5	PCA projections of Kitchen-Mixed latents for horizons $H \in \{1, 4, 12\}$. Increasing the skill horizon produces more structured latent organization, supporting stronger temporal abstraction.	86
5.6	Performance versus skill horizon H across representative tasks. Intermediate horizons provide the best balance between temporal abstraction and stable value estimation.	86

5.7	Trajectory stitching on AntMaze-Medium-Diverse. Gray traces show fragmented dataset coverage, colored trajectories are GRALP rollouts, and yellow stars mark skill transitions every $H = 12$ steps.	87
6.1	CRISP pipeline. A diffusion-based executor decodes latent skills into short action fragments, while a masked latent planner reconstructs hidden skill decisions from partial context and uses conservative value-guided refinement for support-aware deployment.	93
6.2	Context-dropout robustness and skill-horizon sensitivity. The left panel compares CRISP with the unmasked baseline under degraded temporal context. The right panel shows that intermediate skill horizons balance abstraction and value stability.	101
6.3	Delayed key-door diagnostic. The correct final decision depends on an early key cue, while a later decoy cue can create a local shortcut. Masked skill inference reduces reliance on the decoy and improves robustness when the shortcut is weakened or inverted.	102
6.4	RoboSuite manipulation tasks. CRISP controls a Franka Panda arm for Lift and Pick-and-Place, testing context-robust latent skill inference in contact-rich manipulation.	104

List of Tables

3.1	State and orbital conditions for the evaluated GTO, super-GTO, and GEO target cases.	32
3.2	Reward weights $W = [w_1; w_2; w_3]$ used in the first DRL network. The columns $W[:, 1]$, $W[:, 2]$, and $W[:, 3]$ correspond to a_{sm} , e , and i , respectively.	38
3.3	Convergence tolerances for the cascaded DRL networks.	41
3.4	Transfer-time comparison for GTO-GEO transfer cases. I_{sp} denotes specific impulse, while λ and P denote propulsion efficiency and power.	45
3.5	Reported revolution counts, mass-to-thrust ratios, and mass loss.	46
3.6	Transfer-time comparison for the super-GTO to GEO transfer case.	46
4.1	Initial orbital elements and spacecraft propulsion specifications for the SA3C transfer scenarios. The first three scenarios evaluate geocentric transfers to GEO, while super-GTO-2 supports the cislunar NRHO and patch-point transfers.	60
4.2	Target orbital elements and cascaded convergence tolerances for the SA3C transfer scenarios. DRL-1 defines the coarse convergence stage and DRL-2 defines the final target tolerance.	61
4.3	Transfer-time comparison for geocentric and cislunar scenarios. SA3C improves over the reported automated approaches in all listed cases while remaining close to traditional optimization references in several scenarios.	63
4.4	Ablation on the attention coefficient λ . The setting $\lambda = 1$ removes the attention-augmented critic loss; lower values increase the influence of attention.	65

5.1	Unified D4RL benchmark results. The table compares non-diffusion baselines, diffusion-based planners and policies, and GRALP across locomotion, navigation, sequential manipulation, and Adroit dexterous-control tasks. Best scores are bolded and second-best scores are underlined following the GRALP evaluation protocol. M/R/E denote Medium, Replay, and Expert; U/L/D denote Umaze, Large, and Diverse; P/Mxd/Cmp denote Partial, Mixed, and Complete; H/C/E denote Human, Cloned, and Expert.	83
5.2	RoboSuite success rates using offline human demonstrations. The results test whether support-preserving latent planning transfers to contact-rich manipulation with a 7-DoF Panda manipulator.	84
5.3	Component ablations on representative tasks. Removing conservative value learning, latent-controllability regularization, critic-guided refinement, or temporal abstraction degrades performance, showing that the method depends on the joint support–controllability design rather than a single isolated component.	85
6.1	Unified Evaluation Results on D4RL Benchmark. Comparison against non-diffusion (model-free/model-based) and diffusion-based (planner/policy) baselines. The maximum score in each row is bolded and the second highest is <u>underlined</u> . M/R/E : Medium/Replay/Expert; U/L/D : Umaze/Large/Diverse; P/Mxd/Cmp : Partial/Mixed/Complete; H/C/E : Human/Cloned/Expert.	100
6.2	Context-dropout robustness on AntMaze-Large-Diverse. At 40% context loss, CRISP retains 64.5% of its full-context performance, while the unmasked baseline retains 34.3%.	101
6.3	Delayed key–door diagnostic success rates. Random skill masking remains more robust than the unmasked sequencer when the decoy shortcut is weakened, inverted, or when context is dropped.	102
6.4	Adaptive masking ratio ablation. Sampling $\alpha \sim \mathcal{U}(0.2, 0.5)$ exposes the model to varied levels of context loss and outperforms no masking and fixed masking ratios.	103

6.5	Masking structure ablation. Random token masking outperforms contiguous block masking, indicating that scattered missing decisions more effectively discourage reliance on local interpolation.	103
6.6	Planning component ablations. Masked skill inference improves context robustness, but conservative value learning, latent controllability, and temporal abstraction remain necessary for reliable long-horizon offline control.	104
6.7	RoboSuite success rates over 10 seeds and 25 rollouts per seed.	104

Acknowledgments

I would like to express my deepest gratitude to my advisor, Dr. Arslan Munir, for his guidance, support, and encouragement throughout my Ph.D. journey. His mentorship has been instrumental in shaping my research direction, academic growth, and confidence as a researcher.

I am profoundly thankful to my wife, Bihter, for her love, patience, and constant support through every stage of this journey. I also owe immense gratitude to my mother and my family for their prayers, sacrifices, and unwavering belief in me. I especially remember my late father, Khurshid Ali Zaidi, whose values, guidance, and dreams continue to inspire me.

I am grateful to my brother, Ahsan, for his support and encouragement. My sincere thanks also go to my lab colleagues and friends, including Hayat, Ali, Shakib, and Abdullah, for their collaboration, discussions, and companionship throughout this work.

I would also like to thank all of my teachers and mentors, whose knowledge, encouragement, and guidance have shaped my academic path from the beginning.

This dissertation would not have been possible without the support, patience, and encouragement of all these people. I am deeply thankful to each of them.

Chapter 1

Introduction

1.1 Motivation and Dissertation Framing

Autonomous physical systems are increasingly expected to make decisions with limited human intervention while operating in environments governed by physical dynamics. These systems include autonomous agents that move through space, manipulate objects, follow constrained trajectories, interact with contact-rich environments, and execute decisions whose consequences unfold over extended time. Their autonomy is not defined merely by the ability to produce an action, but by the ability to choose actions that remain useful, feasible, and recoverable across a long sequence of future states.

The central difficulty addressed in this dissertation is the mismatch between the long-horizon nature of physical objectives and the local nature of most learning signals, feedback, and control decisions. In many physical systems, the effect of an action is not immediately visible. A decision that appears locally reasonable may gradually accumulate error, violate future feasibility, or prevent the system from reaching a distant objective. Conversely, a locally conservative action may preserve feasibility but fail to produce meaningful long-term progress. This tension makes long-horizon decision-making one of the defining challenges in physical autonomy.

This challenge appears across multiple forms of autonomous physical systems. In space autonomy, decision policies must guide systems through long-duration trajectory evolution where small

control choices accumulate over many future states. In embodied robotic autonomy, policies must compose short behaviors into extended task execution while maintaining consistency with physical contact, actuation limits, and available demonstrations. Although these domains differ substantially in dynamics, sensing, and data collection, they share a common decision-theoretic structure: a policy must produce sequential actions that remain physically meaningful over a horizon much longer than the immediate feedback available to the learner.

Reinforcement learning provides a natural framework for studying this problem because it directly models sequential decision-making under cumulative reward¹. In principle, reinforcement learning can learn feedback policies that reduce the need for repeated expert intervention at deployment. In practice, however, generic reinforcement learning is often insufficient for autonomous physical systems. Trial-and-error interaction may be expensive, unsafe, or impossible. Rewards may be sparse or delayed. The state space may be continuous, nonlinear, and sensitive to small errors. Learned policies may overfit to local cues, exploit simulation artifacts, or choose actions outside the physically supported region of prior experience.

The motivation for this dissertation is that long-horizon physical autonomy requires structured reinforcement learning. A policy must not only maximize return, but must do so while preserving feasibility, respecting temporal structure, and avoiding unsupported behavior. The dissertation studies this issue through two broad experimental settings: trajectory-level space autonomy and offline embodied control. The first setting emphasizes long-duration continuous decision-making under nonlinear dynamics and mission constraints. The second emphasizes learning from fixed datasets, where the policy must compose demonstrated behaviors without unsafe extrapolation. Together, these settings provide complementary evidence for a unified thesis: reinforcement learning for autonomous physical systems becomes more effective when the learning architecture reflects the temporal and physical structure of the decision problem.

1.2 Background and Context

1.2.1 Reinforcement Learning for Physical Decision Systems

A reinforcement learning problem is commonly formulated as a Markov decision process,

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma), \quad (1.1)$$

where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, P is the transition model, r is the reward function, and γ is the discount factor. A policy $\pi(a|s)$ selects actions from states, and the learning objective is to maximize expected cumulative return,

$$J(\pi) = \mathbb{E}_{\pi} \left[\sum_{t=0}^T \gamma^t r(s_t, a_t) \right]. \quad (1.2)$$

In physical decision systems, these abstract variables correspond to real dynamical quantities. States may describe orbital elements, robot configurations, object poses, velocities, mass properties, or task context. Actions may represent thrust directions, motor torques, joint commands, or temporally extended skills. The transition model is not merely a data-generating process; it is the physical evolution of the system under dynamics, actuation limits, numerical integration, contact behavior, or simulator-defined interaction.

This physical grounding changes the learning problem. The policy must remain compatible with the evolution of the system. A physically infeasible or unsupported action can move the system into a state where later decisions become ineffective. Thus, the objective is not simply to learn a high-return mapping from states to actions. The objective is to learn a long-horizon decision policy whose intermediate decisions preserve the ability to complete the task.

Actor-critic reinforcement learning is especially relevant to this dissertation because it separates policy learning from value estimation. The actor proposes actions, while the critic estimates the long-term value of those actions. Off-policy actor-critic methods such as Soft Actor-Critic are useful in continuous control because they can learn from replayed experience and support stochastic exploration. However, the dissertation argues that actor-critic learning alone is not enough for

long-horizon physical autonomy. The learning process must also account for temporal abstraction, feature relevance, support constraints, and robustness to imperfect context.

1.2.2 The Long-Horizon Feasibility Gap

The dissertation is centered on what may be called the long-horizon feasibility gap. This gap occurs when a policy must optimize objectives that unfold over many future decisions, but the learning process receives only local, delayed, incomplete, or distribution-limited information. The gap has four main components.

First, long-horizon tasks create difficult credit assignment. The reward received at a later time may depend on decisions made much earlier, but the learning algorithm may not receive a clear signal about which decision was responsible. Second, errors compound across time. A small deviation can grow into a large terminal failure, especially in nonlinear physical systems. Third, feasibility is path-dependent. It is not enough for the terminal state to be desirable; the intermediate trajectory must remain inside a region where actions are meaningful, safe, or recoverable. Fourth, physical systems often restrict data collection. Some systems can be trained only in simulation before deployment, while others must learn from fixed datasets because real-world exploration is unsafe or costly.

This gap explains why purely local policies and unstructured deep reinforcement learning often struggle in physical autonomy. A policy that reacts only to immediate state information may fail to preserve long-term reachability. A policy trained only to imitate short behavior fragments may fail to compose them into a longer plan. A policy optimized without support constraints may select actions that look valuable under a learned critic but are not physically grounded in the available data. A policy trained under complete context may become brittle when parts of the decision history are missing or corrupted.

The methods studied in this dissertation address this gap through structured reinforcement learning. Cascaded learning decomposes a difficult long-horizon objective into staged decision policies. Attention-enhanced actor-critic learning improves feature selection in complex continuous dynamics. Temporal abstraction reduces the effective planning horizon by representing short

behavior fragments as higher-level decisions. Conservative value learning restricts planning toward supported regions of experience. Generative execution produces expressive local behavior while keeping long-horizon planning in a more tractable representation. Masked skill inference encourages non-local temporal reasoning when complete decision history cannot be assumed.

Physical systems are more constrained than many purely digital decision problems because their states evolve through dynamics that cannot be arbitrarily reset or ignored. A poor recommendation in a digital system may reduce user engagement, but the next interaction may remain largely independent. A poor decision in a physical system can change the future state distribution, consume limited resources, violate constraints, or make recovery difficult.

This distinction creates several methodological requirements. First, policy learning must account for feasibility throughout the trajectory. Second, the learned action representation must preserve control authority, meaning that changes in the decision variable should produce meaningful changes in physical behavior. Third, learning from data must account for support. In offline settings, a learned value function may assign high value to actions never demonstrated in the dataset, leading to unsafe extrapolation. Fourth, long-horizon autonomy must tolerate imperfect decision context. Real systems may encounter sensing uncertainty, truncated histories, execution drift, or mismatch between training and deployment conditions.

These requirements motivate the dissertation’s broader view of reinforcement learning. Reinforcement learning is not treated as a generic replacement for optimization or control. Instead, it is treated as a decision-learning framework that must be shaped by physical constraints. The central question is not whether reinforcement learning can be applied to physical systems, but under what structural conditions it can learn policies that remain feasible, compositional, and useful over long horizons.

1.2.3 Representative Validation Contexts

The dissertation studies the long-horizon feasibility gap in two application contexts. The first is trajectory-level space autonomy, where policies are trained to guide long-duration orbital transfer behavior under nonlinear dynamics and mission constraints. This setting tests whether rein-

forcement learning can reduce dependence on repeated expert-guided optimization and support autonomous decision-making after training.

The second is offline embodied control, where policies must learn from fixed datasets of demonstrated behavior without further online exploration. This setting tests whether reinforcement learning can compose long-horizon behavior while avoiding unsupported extrapolation. It also tests whether learned policies remain reliable when decision context is incomplete or degraded.

These applications are not treated as separate dissertation themes. They are used as complementary testbeds for a common theoretical problem. The space autonomy setting emphasizes long-horizon continuous control under simulated physical dynamics. The embodied control setting emphasizes support-constrained long-horizon composition under fixed data. Together, they evaluate whether structured reinforcement learning can produce long-horizon decision policies for autonomous physical systems.

1.3 Research Gap

The research gap addressed in this dissertation is theoretical before it is application-specific. Existing methods in optimization, reinforcement learning, and offline control address parts of the long-horizon physical decision problem, but they do not fully resolve the combined requirements of temporal reach, feasibility preservation, limited interaction, and robust behavior composition.

The first gap is the lack of scalable decision structures for long-horizon physical objectives. Traditional optimization and control methods can produce strong solutions when the problem is well specified and sufficient expert effort is available. However, many physical autonomy problems require repeated decision-making, replanning, or rapid inference under changing conditions. Methods that depend heavily on expert initialization, iterative numerical solving, or problem-specific handcrafted guidance are difficult to convert into autonomous policies that can act efficiently after deployment. The theoretical gap is therefore not simply computational cost. It is the absence of learned decision structures that preserve long-horizon feasibility while reducing dependence on human-guided optimization.

The second gap is weak temporal credit assignment in standard reinforcement learning. Physical

objectives often depend on actions whose effects appear much later. Generic actor-critic learning can optimize long-term return in principle, but in complex physical systems it may suffer from sample inefficiency, unstable value estimates, sparse rewards, and sensitivity to reward shaping. Without additional structure, the policy may learn local progress signals without reliably satisfying delayed terminal objectives. This creates a need for reinforcement learning methods that make long-horizon structure more learnable through decomposition, feature selection, or abstraction.

The third gap is the support-feasibility problem in offline policy learning. When interaction is restricted and the learner has access only to a fixed dataset, the policy must avoid decisions outside the data-supported region. Conservative offline RL methods address value overestimation through pessimism or behavior regularization²⁻⁴. However, excessive conservatism can limit behavior composition, while insufficient conservatism can produce unsafe extrapolation. The unresolved issue is how to preserve support while still enabling the policy to compose useful long-horizon behavior.

The fourth gap is the tension between expressivity and controllability in generative decision models. Generative models can represent rich multimodal behavior, and diffusion-based methods have become influential for trajectory and policy modeling⁵⁻⁸. However, when generative sampling is coupled directly to long-horizon planning, inference may become expensive and errors may accumulate. When expressive decoders ignore latent decisions, the high-level planner loses control authority. A central theoretical gap is how to use generative models for expressive execution while keeping planning stable, support-aware, and controllable.

The fifth gap is brittleness under imperfect decision context. Sequence models and trajectory prediction methods can capture long-range dependencies^{9;10}, but they often assume that complete history is available and reliable. Physical deployment rarely satisfies this assumption. Histories may be truncated, sensors may be noisy, and execution may drift from the nominal trajectory. Policies that rely on short-range correlations may perform well under clean evaluation but fail when context is degraded. This motivates training objectives that force policies to infer missing decisions from non-local temporal structure rather than relying only on adjacent information.

These gaps form one broader research problem: learning long-horizon decision policies for autonomous physical systems requires methods that jointly address horizon length, feasibility,

support, controllability, and context robustness. The dissertation evaluates this claim in space trajectory autonomy and offline embodied control, but the intellectual gap is not limited to either domain.

1.4 Problem Statement

Autonomous physical systems must make sequential decisions whose consequences unfold over long horizons, while their learning processes are constrained by nonlinear dynamics, limited feedback, restricted interaction, and physical feasibility requirements. Existing optimization, control, and reinforcement learning methods each address parts of this challenge, but they often struggle when long-horizon planning, autonomous inference, support preservation, and robust behavior composition are required simultaneously.

The problem addressed in this dissertation is the *long-horizon physical decision-policy learning problem*: learning policies for autonomous physical systems that can produce temporally extended behavior, maintain feasibility under physical dynamics, and remain reliable when interaction is expensive, data are fixed, rewards are delayed, or decision context is incomplete.

Formally, let an autonomous physical system evolve according to

$$s_{t+1} \sim P(s_{t+1}|s_t, a_t), \quad (1.3)$$

where $s_t \in \mathcal{S}$ is the system state, $a_t \in \mathcal{A}$ is the action, and P is induced by physical dynamics, simulation, or an offline dataset. The goal is to learn a policy π that maximizes long-horizon return,

$$\max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^T \gamma^t r(s_t, a_t) \right], \quad (1.4)$$

subject to feasibility and support constraints,

$$(s_t, a_t) \in \mathcal{F}, \quad t = 0, \dots, T, \quad (1.5)$$

where $\mathcal{F}$ denotes the set of physically feasible, constraint-satisfying, or data-supported state-action decisions.

The difficulty is that $\mathcal{F}$ is rarely available as a simple analytic object. In some settings, feasibility is defined through nonlinear dynamics, terminal tolerances, and actuation constraints. In others, feasibility is implicit in a fixed dataset of demonstrated behavior. The learned policy must therefore infer how to act over long horizons without violating the physical or statistical support needed for reliable execution. This is the central problem of the dissertation.

1.5 Thesis Statement

This dissertation argues that reinforcement learning can learn long-horizon decision policies for autonomous physical systems when policy learning is structured to preserve feasibility across time. Specifically, long-horizon physical decision-making becomes more tractable when reinforcement learning incorporates staged decomposition, attention-based feature selection, temporal abstraction, support-preserving latent planning, conservative value learning, generative execution, and context-robust inference. These structures allow learned policies to reason beyond immediate actions while remaining grounded in the dynamics, constraints, and data support of the physical system.

The dissertation supports this thesis through two complementary lines of evidence. In trajectory-level space autonomy, cascaded and attention-enhanced actor-critic methods are used to learn long-duration decision policies under nonlinear orbital dynamics and mission constraints. In offline embodied control, latent skill planning, diffusion-based execution, conservative critics, and masked skill inference are used to compose long-horizon behavior from fixed datasets while reducing unsupported extrapolation and brittleness under degraded context. Across both settings, the dissertation demonstrates that the central issue is not merely applying deep reinforcement learning, but structuring reinforcement learning so that long-horizon decisions remain feasible, controllable, and reliable.

1.6 Research Questions

This dissertation is guided by the following research questions.

1. How can long-horizon physical decision problems be decomposed into learnable stages without losing the precision required for terminal feasibility?
2. How can actor-critic reinforcement learning be modified so that the policy focuses on the most decision-relevant features in complex continuous dynamics?
3. How can temporal abstraction reduce the effective horizon of physical control problems while preserving enough control authority for long-horizon planning?
4. How can offline reinforcement learning compose new long-horizon behavior from fixed datasets while avoiding unsupported or physically unsafe extrapolation?
5. How can learned decision policies remain reliable when the temporal context available at deployment is incomplete, degraded, or inconsistent with training assumptions?

These questions are intentionally framed at the level of learning theory and decision structure. The spacecraft and robotics studies serve as concrete evaluations of these questions in two demanding forms of autonomous physical systems.

1.7 Dissertation Contributions

The dissertation makes the following contributions.

1. **A unified formulation of long-horizon physical decision-policy learning.** The dissertation frames trajectory-level space autonomy and offline embodied control as instances of a shared problem: learning long-horizon decision policies under physical feasibility, delayed feedback, and limited interaction. This formulation provides the conceptual foundation for connecting the dissertation chapters without reducing them to separate paper summaries.

2. **A staged reinforcement learning approach for long-horizon precision control.** The dissertation studies cascaded reinforcement learning as a way to decompose difficult physical decision problems into sequential learning stages. This contribution addresses the problem of reaching distant objectives that require both coarse long-range progress and fine terminal precision. It is evaluated in multirevolution space trajectory planning, where a cascaded Soft Actor-Critic framework and gradient-aided reward design are used to guide long-duration transfer behavior^{11–13}.
3. **An attention-enhanced actor-critic mechanism for complex physical dynamics.** The dissertation develops an attention-based actor-critic structure that improves the interaction between policy features and value features during learning. The broader contribution is a method for improving feature relevance in continuous long-horizon control, where not all state-action dimensions contribute equally at every decision point. This idea is evaluated in trajectory-level space autonomy under geocentric and cislunar dynamical settings¹⁴.
4. **A support-preserving latent planning framework for offline physical control.** The dissertation studies temporal abstraction through latent skill planning, where long-horizon decisions are made in a learned skill space rather than directly at the action level. The contribution is a planning structure that separates expressive short-horizon execution from high-level decision-making. Diffusion is used for deterministic action execution, while planning and value estimation remain in latent space under conservative constraints. This is evaluated in offline embodied control benchmarks involving navigation, sequential manipulation, dexterous control, and contact-rich manipulation¹⁵.
5. **A context-robust skill inference objective for long-horizon behavior composition.** The dissertation introduces masked skill inference as a way to train long-horizon policies to reconstruct missing latent decisions from partial temporal context. The contribution is not only robustness to missing history, but a learning objective that discourages brittle short-range correlations and promotes non-local temporal reasoning. This is evaluated through offline embodied control under controlled context degradation¹⁶.

6. A cross-domain synthesis of structure in reinforcement learning for physical autonomy.

Across the dissertation, the methods show that different physical decision regimes require different forms of structure. Staged decomposition addresses terminal precision. Attention addresses feature relevance. Latent abstraction addresses temporal scale. Conservative value learning addresses support and feasibility. Generative execution addresses expressive local behavior. Masked inference addresses imperfect context. The synthesis contribution is the identification of these mechanisms as complementary responses to the long-horizon feasibility gap in autonomous physical systems.

1.8 Scope of the Dissertation

The scope of this dissertation is reinforcement learning for long-horizon decision policies in autonomous physical systems. The dissertation focuses on two classes of systems: trajectory-level space autonomy and offline embodied control. These settings are selected because they expose complementary aspects of the same central problem. The first emphasizes long-duration continuous decision-making under nonlinear dynamics and mission constraints. The second emphasizes behavior composition from fixed datasets under support and context limitations.

The dissertation does not claim that a single reinforcement learning algorithm solves all forms of physical autonomy. It also does not claim to replace the full engineering pipeline required for safety-critical deployment. Instead, it studies how reinforcement learning methods can be structured so that learned policies become more suitable for long-horizon physical decision-making.

The space autonomy studies focus on learned decision policies for trajectory planning after an offline training phase. The emphasis is on reducing dependence on repeated expert-guided numerical optimization during inference, while recognizing that final mission deployment would still require validation under higher-fidelity dynamics, uncertainty analysis, and operational constraints.

The embodied control studies focus on offline reinforcement learning from fixed datasets. The emphasis is on support-preserving long-horizon composition, not unrestricted online exploration. The methods are evaluated in benchmark environments and robotic simulation settings that expose navigation, manipulation, dexterous control, and contact-rich execution challenges.

1.9 Dissertation Organization

The dissertation is organized as follows:

Chapter 2 presents background and related work on reinforcement learning, actor-critic methods, long-horizon decision-making, physical autonomy, trajectory optimization, offline reinforcement learning, generative models for control, and latent skill planning.

Chapter 3 presents the staged reinforcement learning framework for long-horizon precision control. The chapter develops the cascaded learning formulation, reward design, physical state representation, and evaluation in long-duration trajectory-level space autonomy.

Chapter 4 presents the attention-enhanced actor-critic framework for continuous physical decision-making. The chapter explains how attention is incorporated into the actor-critic learning process and evaluates its role in complex trajectory-planning scenarios.

Chapter 5 presents the support-preserving latent planning framework for offline physical control. The chapter introduces latent skill learning, deterministic generative execution, conservative value estimation, and long-horizon planning in latent space.

Chapter 6 presents the context-robust skill inference framework for long-horizon offline control. The chapter develops masked skill inference, analyzes incomplete decision context, and evaluates robustness under controlled context degradation.

Chapter 7 concludes the dissertation by synthesizing the findings across the two application settings, discussing limitations, and outlining future directions for reinforcement learning in autonomous physical systems. These directions include stronger feasibility guarantees, better uncertainty handling, improved sim-to-real validation, and deeper integration of learning-based policies with model-based physical reasoning.

Chapter 2

Background and Literature Review

Chapter 1 introduced the long-horizon physical decision-policy learning problem and argued that autonomous physical systems require reinforcement learning methods that preserve physical feasibility, controllability, support, and context robustness. This chapter provides the theoretical and methodological background for that argument. It proceeds from general foundations to structured methods. It begins with sequential decision-making in physical systems, then reviews reinforcement learning, the long-horizon decision problem, classical optimization and control, deep reinforcement learning, attention-based representation learning, offline reinforcement learning, temporal abstraction, generative decision models, and masked decision modeling. The final sections connect these ideas to the dissertation’s two representative validation contexts and synthesize the literature into design principles for long-horizon reinforcement learning in autonomous physical systems.

2.1 Foundations of Long-Horizon Physical Decision-Making

2.1.1 Sequential Decision-Making in Autonomous Physical Systems

Physical autonomy requires sequential decisions whose consequences propagate forward through dynamics. At each decision epoch, the system observes a state, selects an action, transitions to a new state, and accumulates consequences that may become visible only after many future steps. A state may include positions, velocities, orbital elements, robot configurations, object poses,

mass properties, contact conditions, or a learned representation of recent history. An action may correspond to a thrust direction, motor command, torque, gripper motion, or a temporally extended skill. A trajectory is the resulting sequence

$$\tau = (s_0, a_0, s_1, a_1, \dots, s_T, a_T), \quad (2.1)$$

where $s_t \in \mathcal{S}$ is the state, $a_t \in \mathcal{A}$ is the action, and T is the decision horizon. A policy π maps available decision information to actions, either deterministically as $a_t = \pi(s_t)$ or stochastically as $a_t \sim \pi(\cdot \mid s_t)$.

The quality of a long-horizon decision policy cannot be judged from a single action. A locally useful action may reduce immediate error while making future recovery difficult. A locally conservative action may preserve safety but fail to produce enough progress toward a distant objective. Policies are therefore commonly evaluated through cumulative return,

$$G_t = \sum_{k=t}^T \gamma^{k-t} r(s_k, a_k), \quad (2.2)$$

where $r(s_k, a_k)$ is the reward and $\gamma \in [0, 1]$ is a discount factor. In physical systems, however, return is only a partial measure of policy quality. The trajectory must also remain compatible with dynamics, actuation limits, terminal tolerances, safety restrictions, and, in offline learning, the support of the data used to train the policy.

This makes autonomous physical systems harder than many purely digital decision problems. A poor decision in a digital environment may reduce a score or recommendation quality, but the next interaction often remains recoverable. A poor physical action can consume limited resources, violate a constraint, produce unstable contact, move the system outside the reliable training distribution, or create a state from which later actions have little corrective authority. The result is a decision problem in which physical feasibility is a trajectory-level requirement rather than a final-state property.

Several recurring properties make these systems difficult for learning. Their dynamics are often nonlinear. Their constraints may be hard, path-dependent, or only indirectly observable. Their

rewards may be delayed. Their training interaction may be restricted because trial-and-error on a real system is unsafe or expensive. Their state estimates may be uncertain. Their errors may compound over time. These properties motivate the structured reinforcement learning methods reviewed in the rest of this chapter.

2.1.2 Reinforcement Learning Foundations

Reinforcement learning formalizes sequential decision-making through interaction between an agent and an environment¹. A standard formulation is the Markov decision process (MDP),

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma), \quad (2.3)$$

where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P(s' | s, a)$ is the transition distribution, $r(s, a)$ is the reward function, and γ is the discount factor. The Markov assumption states that the current state contains sufficient information to predict the next-state distribution given the current action. In physical systems, this assumption depends on whether the state representation includes the relevant velocities, hidden modes, contact states, mass properties, and history-dependent variables.

For a policy $\pi(a | s)$, the state-value function is

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^T \gamma^t r(s_t, a_t) \mid s_0 = s \right], \quad (2.4)$$

and the action-value function is

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{t=0}^T \gamma^t r(s_t, a_t) \mid s_0 = s, a_0 = a \right]. \quad (2.5)$$

These functions satisfy Bellman consistency relations, such as

$$Q^\pi(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim P, a' \sim \pi} [Q^\pi(s', a')]. \quad (2.6)$$

Bellman recursion is central to RL because it converts a long-horizon objective into local boot-

strapped updates. In long-horizon physical systems, this same recursion can become unreliable when value estimates are learned from sparse rewards, insufficient exploration, or actions outside the training distribution.

Model-based RL uses or learns a transition model and then plans through that model. It can be sample efficient and can exploit known dynamics, but model error may compound over long horizons. Model-free RL learns policies or value functions directly from interaction data. It can avoid explicit model identification and can yield fast inference after training, but often requires many samples and careful reward design. In simulation-trained physical control, these categories often overlap: the training environment may use a physics model, while the deployed policy is model-free.

Actor-critic methods are especially important for continuous physical control. The actor represents the policy, and the critic estimates state or state-action values that guide policy improvement. Soft Actor-Critic (SAC) is a widely used off-policy actor-critic algorithm that optimizes an entropy-regularized objective¹⁷,

$$J(\pi) = \mathbb{E}_{\pi} \left[\sum_{t=0}^T \gamma^t (r(s_t, a_t) + \alpha \mathcal{H}(\pi(\cdot | s_t))) \right], \quad (2.7)$$

where $\mathcal{H}$ is policy entropy and α controls the strength of entropy regularization. The entropy term encourages exploration and discourages premature collapse to a narrow action distribution. SAC also uses replay buffers, target networks, and clipped double-Q estimation, which make it useful for continuous simulated control. In this dissertation, SAC provides a foundation for staged DRL trajectory planning, while later work extends actor-critic learning through attention and structured decision representations.

2.1.3 The Long-Horizon Decision Problem

The long-horizon decision problem arises when current actions influence outcomes many steps later. In such settings, the learner must assign credit to decisions whose consequences are delayed, indirect, or visible only at the end of an episode. This credit-assignment problem becomes more

severe when rewards are sparse, when success requires a precise sequence of intermediate decisions, or when early deviations reveal their effect only after recovery becomes difficult.

Sparse and delayed rewards are common in autonomous physical systems. A task may provide a meaningful success signal only when a distant target is reached, a final tolerance is satisfied, or a multi-step physical behavior is completed. Reward shaping can provide denser feedback, but shaping is not a neutral design choice. If the shaped reward overemphasizes local progress, a policy may improve the proxy objective while failing the true physical task. If it is too weak, learning may become slow or unstable.

Compounding error is another defining issue. A small deviation at one time step shifts the next state. The policy then acts from this shifted state, creating the possibility of further deviation. Over long horizons, this feedback loop can turn small errors into terminal failure. In nonlinear physical systems, sensitivity to state, contact, timing, or environmental conditions can amplify this effect.

These issues form the long-horizon feasibility gap introduced in Chapter 1. The gap is the mismatch between objectives that unfold over many future decisions and learning signals that are local, delayed, incomplete, or distribution-limited. Reward maximization alone does not close this gap because a high predicted return does not guarantee that intermediate states remain feasible, that actions remain supported, or that future correction remains possible. The rest of the chapter uses this gap as shorthand rather than redefining it in every section.

2.2 Optimization and Deep Reinforcement Learning for Physical Autonomy

2.2.1 Classical Optimization and Control for Physical Decision Problems

Classical optimization and control provide the mathematical foundation for many physical decision problems. Optimal control seeks a control signal that drives a dynamical system toward an objective while satisfying dynamics and constraints. Direct methods transcribe the continuous problem into a finite-dimensional nonlinear program, often through collocation, multiple shooting, or control

parameterization^{8;18–20}. Indirect methods derive necessary conditions for optimality and solve the resulting boundary-value problem. Both approaches can produce precise solutions when the model, boundary conditions, and initialization are well specified.

These methods have clear strengths. They explicitly encode dynamics and constraints, provide interpretable problem formulations, and can achieve high-accuracy solutions^{21;22}. They are therefore indispensable for physical system design and for validating learned methods. Feedback guidance and model-predictive ideas add closed-loop correction by updating decisions using the current state. This improves responsiveness compared with fixed open-loop plans.

Their limitations become important when the goal is rapid autonomous inference. Many long-horizon physical decision problems are nonlinear, nonconvex, and sensitive to initialization^{23;24}. Direct and indirect solvers may require expert-designed initial guesses, iterative numerical solving, careful scaling, and problem-specific tuning. In trajectory-level space autonomy, low-thrust transfer problems illustrate these limitations because traditional solvers can be accurate but often depend on expert initialization and repeated computation^{25;26}. This dissertation treats learning-based policies as complementary to these methods. Classical optimization supplies physical grounding and baselines; reinforcement learning supplies a way to learn closed-loop decision policies that can act efficiently after training.

2.2.2 Deep Reinforcement Learning for Physical Autonomy

Deep reinforcement learning extends RL with neural function approximation, making it possible to represent policies and value functions over continuous, nonlinear, and high-dimensional state-action spaces. For autonomous physical systems, DRL is attractive because it can learn from repeated simulation and then deploy a policy through fast neural inference. This shifts much of the computational burden into an offline training phase and can reduce the need for repeated online optimization.

DRL is also attractive because it learns closed-loop policies. Instead of producing one fixed open-loop trajectory, a trained policy can map the current state to an action. This is useful when the system may deviate from a nominal path or when rapid action selection is required. Prior work

has explored reinforcement learning for low-thrust transfer, orbit raising, cislunar guidance, and related trajectory-planning tasks [11;27–29](#).

The benefits of DRL come with significant limitations. Training can be sample inefficient. Reward shaping can bias the learned behavior. Actor-critic updates can become unstable when value estimates are inaccurate. Policies may generalize poorly outside the training distribution. Exploration can be unsafe unless it occurs in simulation or under strong constraints. These issues are not incidental; they are direct consequences of the long-horizon feasibility gap in Section [2.1.3](#).

Structured DRL methods address these difficulties by making the learning problem more aligned with the physical decision structure. Staged or cascaded RL decomposes distant objectives into learnable phases. Attention-guided actor-critic learning improves feature relevance in complex dynamics. These ideas are developed in later chapters, but their motivation is methodological: long-horizon physical control benefits when policy learning is given structure rather than treated as generic trial-and-error.

2.2.3 Attention and Representation Learning in Reinforcement Learning

In continuous physical control, not all state and action dimensions matter equally at every timestep. The variables that determine useful progress early in a trajectory may differ from the variables that determine terminal precision, constraint satisfaction, or recovery later. A policy representation that treats all features uniformly may waste capacity or learn correlations that are locally predictive but globally weak.

Attention mechanisms provide a learned way to assign different importance to different parts of a representation [30](#). In RL, attention can help a policy or critic emphasize features that are most relevant to the current decision. This is not merely a question of interpretability. In actor-critic learning, the actor selects actions while the critic evaluates their long-term value. If these networks focus on different or irrelevant features, policy improvement can become inefficient. Attention can act as an architectural bias that improves the interaction between action selection and value estimation.

Prior work has used attention in RL for visual control, multi-agent coordination, feature se-

lection, and actor-critic architectures^{31–37}. For this dissertation, the important point is narrower: attention is useful when complex dynamics make feature relevance state-dependent and temporally variable. The attention-enhanced actor-critic chapter uses this idea to study how policy features and critic features can be coupled more effectively in continuous physical decision-making¹⁴.

2.3 Offline Reinforcement Learning, Temporal Abstraction, and Support Preservation

2.3.1 Offline Reinforcement Learning and Distributional Shift

Offline reinforcement learning learns a policy from a fixed dataset,

$$\mathcal{D} = \{(s_t, a_t, r_t, s_{t+1})\}, \quad (2.8)$$

without further interaction with the environment. This setting is important for autonomous physical systems because online exploration may be unsafe, expensive, slow, or physically irreversible. A robot should not learn by repeatedly failing in damaging contact configurations. A safety-critical system should not test arbitrary actions during deployment.

The central difficulty is distributional shift. The dataset is generated by one or more behavior policies, but the learned policy may choose actions that are rare or absent in the dataset. A learned critic may then extrapolate incorrectly. If it assigns overly optimistic values to out-of-distribution actions, policy improvement can push the policy toward unsupported behavior. This is the offline version of the feasibility gap: the policy must improve beyond imitation while remaining grounded in the behavior distribution that makes evaluation reliable.

Several families of methods address this issue. Behavior-regularized methods constrain the learned policy toward the behavior distribution, as in Batch-Constrained Q-learning². Conservative value-learning methods penalize high values assigned to actions outside the dataset distribution, as in Conservative Q-Learning³. Implicit Q-Learning avoids explicit maximization over unseen actions by extracting policies from value-weighted behavior data⁴. Other methods combine behav-

ior cloning with value-guided improvement³⁸. Sequence-modeling approaches recast control as conditional trajectory prediction, as in Decision Transformer and Trajectory Transformer^{9;10}.

The central tradeoff is worth stating directly. A policy that stays too close to the dataset may be safe but unable to compose new long-horizon behavior. A policy that deviates too much may be expressive but unsafe or unsupported. The offline-control chapters of this dissertation study this tradeoff through support-preserving latent planning, conservative value learning, and generative execution.

2.3.2 Temporal Abstraction and Latent Skill Learning

Action-level policies struggle with long horizons because they must make many low-level decisions before receiving meaningful feedback. Temporal abstraction reduces this burden by grouping sequences of low-level actions into higher-level units, often called options, skills, subgoals, or macro-actions^{5;6}. A high-level policy then reasons over temporally extended decisions rather than individual actions.

In latent skill learning, trajectory fragments are encoded into continuous latent variables. A fragment of horizon H can be written as

$$\tau_t^H = (s_t, a_t, \dots, s_{t+H-1}, a_{t+H-1}), \quad (2.9)$$

and encoded into a latent skill z_t . A low-level executor maps (s_t, z_t) into an action fragment, while a high-level planner selects a sequence of latent skills. The effective high-level horizon is reduced because each skill represents multiple action steps.

This abstraction is especially useful in offline RL because fixed datasets often contain locally coherent motion fragments. A latent skill model can expose these fragments as reusable decision units, allowing a planner to compose them into longer behavior. Prior hierarchical and latent-skill approaches use abstractions to improve long-horizon reasoning, planning, and compositional control^{39–43}.

Temporal abstraction also introduces risks. Skills may be too rigid to adapt. Latents may drift

outside dataset support. A decoder may ignore the latent variable, producing posterior collapse and eliminating high-level control authority. A useful latent skill space must therefore satisfy two conditions: selected skills should remain close to data-supported behavior, and changes in latent decisions should produce meaningful changes in decoded behavior. These conditions correspond to support preservation and controllability, which are central to the dissertation’s offline control chapters.

2.4 Generative and Context-Robust Decision Models

2.4.1 Generative Models for Decision-Making

Generative models are useful in decision-making because physical behavior is often multimodal. The same state may admit several plausible actions depending on task intent, contact strategy, or future subgoal. A unimodal policy may average these alternatives and produce weak behavior. A generative model can represent multiple valid possibilities.

Diffusion models represent complex continuous distributions through a forward noising process and a learned reverse denoising process⁷. In decision-making, diffusion can be used as a trajectory planner that generates future action sequences^{5;6}, as a policy model that samples actions from a learned behavior distribution^{44–48}, or as a short-horizon skill executor that decodes a latent decision into a local action fragment.

Diffusion-based decision models face long-horizon concerns. Full trajectory sampling can be computationally expensive at inference. Stochastic denoising can complicate value estimation if the same high-level decision produces inconsistent action fragments. Generated trajectories can accumulate error when executed over long horizons. Generative models can also produce actions that are plausible under the model but not reliable under the true physical system.

Design principle. Use generative models for expressive short-horizon execution, but keep long-horizon planning and value estimation structured in latent space.

This principle is central to the dissertation’s offline control work. Diffusion provides expressive local behavior, while planning and conservative value estimation operate over latent skills. This

separation reduces inference burden, improves the stability of value-guided planning, and makes support preservation easier to enforce at the decision level¹⁵.

2.4.2 Context Robustness and Masked Decision Modeling

Long-horizon policies often depend on temporal context. A decision may require knowledge of earlier states, previous skills, task progress, or intended return. Sequence models can use this information effectively, but they can also become brittle when deployment context differs from training context. Histories may be truncated, sensors may be noisy, buffers may be incomplete, and execution may drift from nominal trajectories.

The failure mode is often shortcut learning. A model may learn to predict the next decision from adjacent tokens or local temporal correlations instead of using the broader structure of the task. This can work under clean full-context evaluation, but it fails when context is missing or corrupted. Context robustness therefore requires training objectives that discourage over-reliance on local shortcuts.

Masked modeling provides one such objective. Parts of a sequence are hidden during training, and the model learns to reconstruct them from the remaining context. This encourages non-local temporal reasoning because the model cannot simply copy adjacent information. Masked modeling has been studied for trajectory and decision modeling as a way to improve representation learning and robustness⁴⁹.

For long-horizon control, the masked units can be states, actions, trajectory tokens, or latent skills. Masking latent skills is particularly aligned with temporal abstraction because each token represents a compositional decision unit rather than a single low-level action. The context-robust skill inference chapter uses this idea to train a planner that reconstructs missing latent decisions from partial context while remaining constrained by conservative latent value learning¹⁶.

2.5 Representative Validation Contexts

The dissertation evaluates these design principles in two carefully chosen domains that expose different facets of the long-horizon feasibility gap. Trajectory-level space autonomy tests long-duration continuous control, where small action errors accumulate over extended physical evolution and terminal feasibility depends on many earlier decisions. This setting directly stresses staged decomposition, closed-loop policy learning, and feature relevance under nonlinear dynamics. Low-thrust trajectory problems are a strong validation context because traditional direct and indirect methods provide high-quality baselines, while their dependence on expert initialization and iterative solving motivates learned policies for rapid autonomous inference^{11;25;26}.

Offline embodied control tests a different facet of the same theory: behavior composition from fixed data. Here, the central risk is not only long horizon, but unsupported extrapolation. The policy must compose useful behavior without leaving the dataset-supported region where value estimates and generated actions remain reliable. This setting directly stresses support preservation, latent controllability, generative execution, and context robustness. Benchmarks such as D4RL and RoboSuite provide standardized environments for evaluating navigation, sequential manipulation, dexterous control, and contact-rich manipulation^{50;51}.

Together, these domains provide convergent evidence rather than disconnected applications. One domain emphasizes long-duration continuous dynamics and terminal precision. The other emphasizes fixed-data learning, behavior composition, and degraded context. If the same design principles are useful across both, the dissertation’s argument becomes stronger: long-horizon reinforcement learning for autonomous physical systems requires structure that matches the temporal and feasibility constraints of the decision problem.

2.6 Synthesis: Design Principles for Long-Horizon RL in Autonomous Physical Systems

The literature reviewed in this chapter suggests six design principles for reinforcement learning in autonomous physical systems. These principles provide the conceptual bridge from the background literature to the technical chapters of the dissertation.

2.6.1 Decompose Long-Horizon Decisions into Structured Stages/Abstractions

Long-horizon tasks often contain intermediate structure. A distant terminal objective may be easier to reach through staged refinement. A long robotic task may be easier to complete through temporally extended skills. Decomposition reduces credit-assignment difficulty by introducing intermediate decision scales. The risk is that poorly chosen stages or skills can create artificial bottlenecks. The useful form of decomposition is therefore not arbitrary task splitting, but structure that reduces horizon length while preserving terminal feasibility. Chapter 3 instantiates this principle through cascaded reinforcement learning for long-horizon precision control.

2.6.2 Use Closed-Loop Policy Learning for Autonomy

Autonomous systems require policies that can map current decision information to actions without solving a full optimization problem at every deployment step. Closed-loop policy learning provides this capability. A trained policy can react to the current state and produce actions efficiently. This does not eliminate the need for optimization and control theory. It shifts repeated search into offline training while retaining physical modeling for validation, baselines, and constraint design.

2.6.3 Improve Feature Relevance through Attention and Representation Learning

Physical dynamics involve interacting variables whose relevance changes over time. Attention and representation learning provide mechanisms for emphasizing the features that matter for the current

decision. In actor-critic learning, this matters because the actor and critic must coordinate action selection and value evaluation. Chapter 4 instantiates this principle through attention-enhanced actor-critic learning for continuous physical decision-making.

2.6.4 Preserve Feasibility and Dataset Support

Physical feasibility must be preserved across the trajectory. In simulation-trained settings, feasibility may be represented through dynamics, constraints, terminal tolerances, and reward penalties. In offline settings, feasibility is often implicit in dataset support. Conservative value learning, behavior regularization, and latent-space constraints reduce unsupported decisions. The central challenge is preserving support without eliminating useful long-horizon composition and is defined in Chapter 5.

2.6.5 Maintain Controllability of Learned Latent Decisions

Latent planning is useful only if latent decisions control decoded behavior. If the executor ignores the latent variable, high-level planning loses authority. If small latent changes produce unstable action changes, value-guided refinement becomes unreliable. Controllability therefore requires a latent representation in which changes in planned skills produce meaningful and stable changes in execution. This principle is central to the dissertation’s support-preserving latent planning framework.

2.6.6 Improve Robustness under Incomplete or Degraded Context

Long-horizon decision policies often assume that history is complete and reliable. Physical deployment may violate this assumption. Context robustness requires policies that can act when decision histories are partial, noisy, or inconsistent with training conditions. Masked decision modeling encourages non-local temporal reasoning and reduces reliance on brittle local correlations. Chapter 6 instantiates this principle through masked skill inference for long-horizon offline control.

2.7 Summary and Transition

This chapter developed the theoretical and methodological background for reinforcement learning for long-horizon decision policies in autonomous physical systems. It reviewed sequential decision-making, reinforcement learning foundations, the long-horizon decision problem, classical optimization and control, deep reinforcement learning, attention, offline reinforcement learning, temporal abstraction, generative models, and masked decision modeling. The central thread across these topics is the long-horizon feasibility gap: learned policies must optimize extended objectives while preserving physical feasibility, support, controllability, and robustness.

The synthesis of the literature leads to six design principles: decompose long-horizon decisions into stages or abstractions, use closed-loop policy learning for autonomy, improve feature relevance through attention or representation learning, preserve feasibility and dataset support, maintain controllability of learned latent decisions, and improve robustness under incomplete or degraded context. The following chapters instantiate these principles in different methodological forms. The next chapter develops the first technical contribution: staged reinforcement learning for long-horizon precision control.

Chapter 3

Cascaded Reinforcement Learning for Long-Horizon Precision Control

Long-horizon precision control presents a design tension: policies must make coarse progress across large state regions while eventually satisfying strict terminal tolerances. This chapter addresses this tension through cascaded/staged reinforcement learning, where separate policies handle global transfer and terminal refinement. The chapter therefore serves as the first technical instantiation of the decomposition principle introduced in [Chapter 2](#).

The method studied here is staged reinforcement learning, implemented as a cascaded deep reinforcement learning framework. The central idea is simple but important: a first policy learns to drive the system toward the neighborhood of the target, and a second policy learns to refine the state to satisfy tighter terminal conditions. This is not a generic hierarchy added for architectural complexity. It is a response to a specific learning difficulty: in precision-control tasks, the requirements for global progress and terminal accuracy are often different enough that a single policy may struggle to satisfy both efficiently.

The validation setting is low-thrust spacecraft orbit transfer. This domain is useful for the dissertation because it exposes the long-horizon feasibility gap in a physically transparent way. A solar-electric spacecraft changes its orbit gradually over many revolutions, but success is determined only when the final orbit satisfies strict tolerances on semimajor axis, eccentricity, and inclination.

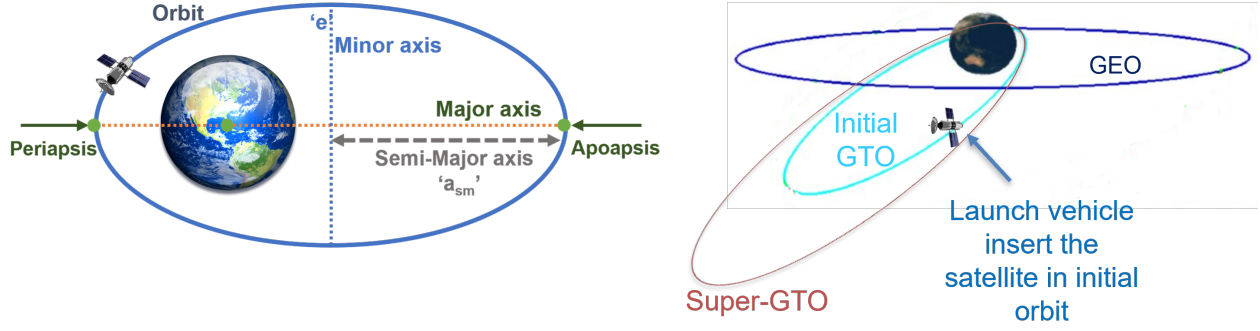


Figure 3.1: *Orbital parameters and orbits visualization*

The chapter shows how a cascaded Soft Actor-Critic (SAC) framework can use staged learning, orbital-element reward shaping, and variable action resolution to compute time-efficient transfers from geostationary transfer orbit (GTO) and super-GTO to geosynchronous equatorial orbit (GEO).

3.1 Problem Setting and Preliminaries

3.1.1 Orbital Transfer Terminology

This section introduces the orbital terminology needed to read the chapter from a computer science or reinforcement learning perspective. An orbit is the path followed by a spacecraft under gravity and other forces. For Earth-orbiting satellites, a useful orbit is usually described by geometric quantities rather than by raw Cartesian coordinates alone. Three of the most important quantities in this chapter are semimajor axis, eccentricity, and inclination.

The semimajor axis, denoted a_{sm} , describes the size of the orbit. For a circular orbit, it is essentially the orbital radius. Eccentricity, denoted e , describes the shape of the orbit. A circular orbit has $e = 0$, while larger eccentricity corresponds to a more elongated ellipse. Inclination, denoted i , describes the tilt of the orbital plane relative to Earth’s equatorial plane (Figure 3.1). A geosynchronous equatorial orbit requires the spacecraft to be at the correct orbital size, nearly circular, and aligned with the equator. In the experiments of this chapter, this target orbit is GEO, with $a_{sm} = 42164$ km, $e = 0$, and $i = 0$.

A geostationary transfer orbit, or GTO, is an elliptical transfer orbit commonly used after launch

as an intermediate orbit before reaching GEO. A super-GTO is a higher-energy transfer orbit with a larger semimajor axis and typically a higher eccentricity than a standard GTO. Transferring from GTO or super-GTO to GEO is important because many communication and Earth-observation satellites ultimately need to operate near GEO, but they may be inserted initially into a transfer orbit rather than directly into the final circular orbit.

The propulsion setting in this chapter is low thrust. Electric propulsion is efficient, but it produces small thrust compared with chemical propulsion. Instead of making a few large impulsive maneuvers, a low-thrust spacecraft gradually changes its orbit over many revolutions. This makes the task naturally long-horizon: each action has a small effect, and the final outcome depends on the accumulation of many thrust-direction decisions.

The action learned by the reinforcement learning policy is the thrust direction. The thrust magnitude is determined by the spacecraft propulsion system, while the policy decides where to point the thrust vector at each decision step. During eclipse, the spacecraft passes through Earth's shadow. Since the propulsion system depends on solar power, the spacecraft is assumed to coast during eclipse intervals, and the thrust magnitude is set to zero. This means that the same commanded thrust direction may have no effect when the spacecraft is in shadow, which adds a physical constraint to the decision process.

Terminal tolerance matters because reaching a visually similar orbit is not sufficient. A spacecraft may appear close to GEO in a trajectory plot but still fail mission requirements if the semimajor axis, eccentricity, or inclination remains outside acceptable limits. The staged framework in this chapter is designed around this distinction: the first stage moves the spacecraft near the desired orbit, while the second stage improves the terminal orbital-element accuracy.

3.1.2 Orbit-Transfer Task

The chapter evaluates staged reinforcement learning on two classes of orbit-raising problems: GTO-to-GEO transfer and super-GTO-to-GEO transfer. The GTO-to-GEO evaluation contains three cases. GTO-1 and GTO-2 include eclipse-induced coasting and use different spacecraft parameters. GTO-3 assumes constant thrust throughout the transfer so that it can be compared with

Table 3.1: *State and orbital conditions for the evaluated GTO, super-GTO, and GEO target cases.*

Scenario	Position	h	h_x	h_y	e_x	e_y	ϕ	a_{sm} (km)	e	i (deg)
GTO-1	Initial	67288.41965	0	-32107.258	0.7306	0	0	24364	0.7306	28.5
	Target	29640.2292	0	0	0	0	free	42164	0	0
	Conv.	–	–	–	–	–	–	$\leq \pm 0.2$	$\leq 5 \times 10^{-5}$	≤ 0.08
GTO-2	Initial	67246.21689	0	-30529.143	0.7310	0	0	24364	0.7310	27
	Target	29640.2292	0	0	0	0	free	42164	0	0
	Conv.	–	–	–	–	–	–	$\leq \pm 0.08$	$\leq 8.7 \times 10^{-5}$	≤ 0.08
GTO-3	Initial	68196.3519	0	-8311.044	0.725	0	0	24505	0.725	28.5
	Target	29640.2292	0	0	0	0	free	42164	0	0
	Conv.	–	–	–	–	–	–	$\leq \pm 12$	$\leq 9 \times 10^{-4}$	≤ 0.08
Super-GTO	Initial	70532.88179	0	-26991.765	0.8705	0	0	51525	0.8705	22.5
	Target	29640.2292	0	0	0	0	free	42164	0	0
	Conv.	–	–	–	–	–	–	$\leq \pm 0.5$	$\leq 6.9 \times 10^{-5}$	≤ 0.08

the single-agent reinforcement learning result of Kwon et al.²⁹. The super-GTO case evaluates the same cascaded approach from a higher and more eccentric initial orbit.

The objective is to minimize transfer time while satisfying the final GEO conditions. The policy controls the thrust direction, and the environment propagates the spacecraft state under the low-thrust dynamics. A transfer is successful only when the semimajor axis, eccentricity, and inclination simultaneously satisfy the required tolerances.

3.1.3 Initial and Target Conditions

Table 3.1 reports the state and orbital conditions used in the experiments. The state variables $(h, h_x, h_y, e_x, e_y, \phi)$ are used by the dynamics and reinforcement learning environment, while the orbital elements (a_{sm}, e, i) define the terminal success criteria. The value of ϕ is free at the target because the final orbit is specified by its orbital geometry rather than by a fixed spacecraft phase.

The convergence rows in Table 3.1 show the precision level required at the end of the transfer. For GTO-1, GTO-2, and super-GTO, the semimajor-axis tolerance is below one kilometer, while the eccentricity tolerance is on the order of 10^{-5} . These requirements make the task more demanding than simply approaching the visual neighborhood of GEO.

3.2 Methodological Challenge and Design Principle

The methodological contribution of this chapter is staged reinforcement learning for tasks that combine long-duration progress with terminal precision. Let π_1 denote a first-stage policy and π_2 denote a second-stage policy. The first policy is trained to reach a coarse target set,

$$s_T^{(1)} \in \mathcal{G}_1, \quad (3.1)$$

where $\mathcal{G}_1$ is a neighborhood of the desired final condition. The second policy is trained from this neighborhood toward a stricter target set,

$$s_T^{(2)} \in \mathcal{G}_2, \quad \mathcal{G}_2 \subset \mathcal{G}_1. \quad (3.2)$$

The two policies therefore solve related but distinct problems. The first policy is responsible for global progress across a large state region. The second policy is responsible for local refinement near the target.

This decomposition is useful in low-thrust orbit transfer because the transfer has two different learning regimes. Far from GEO, the agent must learn a long sequence of thrust-direction decisions that reduce the large initial mismatch in orbital elements. Near GEO, the agent must make smaller corrections while satisfying tight tolerance bounds. A single policy can be asked to solve both, but the reward scale, action resolution, and state distribution required for the two regimes are not identical. The cascaded design allows the learning process to allocate separate policies to these roles.

In the implementation used in this chapter, DRL-1 denotes the first SAC policy and DRL-2 denotes the second SAC policy. DRL-1 is trained from the initial orbit and terminates when the spacecraft reaches a coarse neighborhood of GEO. DRL-2 is trained from that neighborhood and terminates only when the final tolerance conditions are satisfied. The final trajectory is produced by executing DRL-1 first and then switching to DRL-2 for terminal refinement.

3.3 Technical Formulation

3.3.1 State Representation

The spacecraft dynamics are written in terms of six orbital-state variables,

$$\mathbf{s} = \begin{bmatrix} h & h_x & h_y & e_x & e_y & \phi \end{bmatrix}^T. \quad (3.3)$$

Here, h is the magnitude of the specific angular momentum, h_x and h_y are angular-momentum components in the inertial frame, e_x and e_y are components of the eccentricity vector in a rotated orbital frame, and ϕ is a true-anomaly-like angle specifying the spacecraft position along the orbit. This state representation follows the low-thrust orbit-raising formulation used by Sreesawet and Dutta²⁵.

3.3.2 Thrust Model and Action Parameterization

The thrust magnitude is determined by the solar-electric propulsion model

$$F = \frac{2\lambda P}{g_0 I_{sp}}, \quad (3.4)$$

where P is the available solar-array power, λ is propulsion efficiency, g_0 is standard gravitational acceleration, and I_{sp} is specific impulse. The thrust magnitude is constant during sunlit portions of the transfer and becomes zero during eclipse intervals.

The reinforcement learning action consists of two thrust-direction angles,

$$\mathbf{a} = \begin{bmatrix} \alpha & \beta \end{bmatrix}^T, \quad \alpha \in [-\pi, \pi], \quad \beta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]. \quad (3.5)$$

Using these angles, the thrust components in the orbital frame are

$$\mathbf{u} = \begin{bmatrix} F_r & F_n & F_h \end{bmatrix} = F \begin{bmatrix} -\sin \alpha \cos \beta \\ \cos \alpha \cos \beta \\ \cos \beta \end{bmatrix}. \quad (3.6)$$

The policy therefore learns a continuous two-dimensional action, while the physical model converts that action into a thrust vector.

3.3.3 Orbital Dynamics and Eclipse Modeling

The transition model propagates the orbital state under the selected thrust direction. The continuous-time dynamics are

$$\begin{bmatrix} \dot{h} \\ \dot{h}_x \\ \dot{h}_y \\ \dot{e}_x \\ \dot{e}_y \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \frac{\mu^2 B^2}{h^3} \end{bmatrix} \begin{bmatrix} h \\ h_x \\ h_y \\ e_x \\ e_y \\ \phi \end{bmatrix} + \mathbf{G} \begin{bmatrix} F_r \\ F_n \\ F_h \end{bmatrix}, \quad (3.7)$$

where μ is Earth's gravitational parameter. The auxiliary quantities

$$A = e_x \sin \phi - e_y \cos \phi, \quad B = 1 + e_x \cos \phi + e_y \sin \phi \quad (3.8)$$

appear in the force-mapping matrix $\mathbf{G}$. This matrix maps the thrust components into rates of change of the orbital-state variables:

$$\mathbf{G} = \begin{bmatrix} 0 & \frac{h^2}{\mu m B} & \frac{\mu^2 B^2}{h^3} \\ 0 & \frac{h h_x}{\mu m B} & \frac{h^2 \sqrt{h^2 - h_x^2 - h_y^2} B^2}{\mu m B \sqrt{h^2 - h_y^2}} \sin \phi + \frac{h h_x h_y}{\mu m B \sqrt{h^2 - h_y^2}} \cos \phi \\ 0 & \frac{h h_y}{\mu m B} & -\frac{h \sqrt{h^2 - h_y^2}}{\mu m B} \cos \phi \\ \frac{h \sin \phi}{\mu m} & \frac{2h \cos \phi}{\mu m} + \frac{h A \sin \phi}{\mu m B} & \frac{h e_y h_y}{\mu m B \sqrt{h^2 - h_y^2}} \sin \phi \\ -\frac{h \cos \phi}{\mu m} & \frac{2h \sin \phi}{\mu m} + \frac{h A \cos \phi}{\mu m B} & -\frac{h e_x h_y}{\mu m B \sqrt{h^2 - h_y^2}} \sin \phi \\ 0 & 0 & -\frac{h h_y}{\mu m B \sqrt{h^2 - h_y^2}} \sin \phi \end{bmatrix}. \quad (3.9)$$

The model includes the onboard thrust as the active non-Keplerian force. Orbital perturbations and radiation-induced solar-array degradation are not included in the reported comparisons. These assumptions follow the comparison setting used by the automated orbit-raising baseline and define the scope of the results in this chapter.

The eclipse model determines when solar-electric thrust is unavailable. Let (X_I, Y_I, Z_I) be the spacecraft position in the inertial frame and let R_E be Earth's radius. The cylindrical Earth-shadow model declares eclipse when

$$X_I < 0, \quad (3.10)$$

$$\sqrt{Y_I^2 + Z_I^2} < R_E. \quad (3.11)$$

When both conditions hold, the thrust magnitude is set to zero and the spacecraft coasts. This is important for the learning problem because the action selected by the policy has no propulsive effect during eclipse. The agent must therefore learn under dynamics in which control authority

changes along the trajectory.

3.3.4 Reinforcement Learning Formulation

The orbit-transfer problem is formulated as a Markov decision process with transition tuple

$$\mathcal{M} = \{\mathbf{s}_t, \mathbf{a}_t, r_{t+1}, \mathbf{s}_{t+1}, d_t\}, \quad (3.12)$$

where $\mathbf{s}_t$ is the orbital state, $\mathbf{a}_t$ is the thrust-direction action, r_{t+1} is the reward, $\mathbf{s}_{t+1}$ is the propagated orbital state, and d_t is the termination indicator. The transition function integrates the dynamics in Section 3.3 using the action selected by the policy and the eclipse model when applicable.

At every time step, the environment checks three terminal conditions. The eccentricity condition is

$$0 \leq \left[e = \sqrt{e_x^2 + e_y^2} \right] \leq e_{\text{tol}}. \quad (3.13)$$

The semimajor-axis condition is

$$a_{\text{sm}}^{\text{tar}} - a_{\text{sm}}^{\text{tol}} \leq \left[a_{\text{sm}} = \frac{h^2}{\mu(1 - e_x^2 - e_y^2)} \right] \leq a_{\text{sm}}^{\text{tar}} + a_{\text{sm}}^{\text{tol}}. \quad (3.14)$$

The inclination condition is

$$0 \leq \left[i = \sqrt{\frac{h_x^2 + h_y^2}{h}} \right] \leq i_{\text{tol}}. \quad (3.15)$$

A successful episode requires simultaneous satisfaction of Eqs. (3.13) to (3.15).

SAC is used as the base reinforcement learning algorithm because the action space is continuous and the problem requires substantial exploration over thrust-angle profiles^{52;53}. The SAC background was introduced in Chapter 2; here, SAC functions as the policy-optimization engine inside the staged framework. The methodological focus of this chapter is not the SAC update itself, but the way two SAC agents are organized into a cascaded precision-control procedure.

Table 3.2: Reward weights $W = [w_1; w_2; w_3]$ used in the first DRL network. The columns $W[:, 1]$, $W[:, 2]$, and $W[:, 3]$ correspond to a_{sm} , e , and i , respectively.

Weight	GTO to GEO			Super-GTO to GEO		
	$W[:, 1]$	$W[:, 2]$	$W[:, 3]$	$W[:, 1]$	$W[:, 2]$	$W[:, 3]$
w_1	1×10^3	2×10^3	3×10^2	1×10^3	4×10^3	3×10^2
w_2	1×10^{-2}	1.9×10^{-7}	3×10^{-5}	1×10^{-2}	1.9×10^{-9}	3×10^{-5}
w_3	5×10^2	7×10^2	3×10^2	5×10^2	2×10^3	3×10^2

3.4 Learning Objective and Algorithm

3.4.1 Gradient-Aided Reward Function

A sparse terminal reward is poorly suited to this task because most early training trajectories do not reach the final tolerance set. The agent therefore needs intermediate feedback that indicates whether the orbit is moving toward the desired physical state. The reward function uses a potential-based shaping term computed from the orbital-element errors.

Let s_t denote the reward variables and let s_{tar} denote their target values. The potential function is

$$\Phi(s_t) = -w_1 (|s_t - s_{\text{tar}}|) + \sum w_2 \exp(-w_3 |s_t - s_{\text{tar}}|), \quad (3.16)$$

and the shaped reward is

$$r' = \Phi(s') - \Phi(s) - \tau - 5(\xi) + 100(d). \quad (3.17)$$

The parameter $\tau = 0.005$ penalizes long episodes. The boundary flag ξ penalizes violations of allowable orbital-element ranges. The done flag d becomes one only when the terminal target is reached.

The important choice is that the reward is computed using the orbital elements (a_{sm}, e, i) rather than only the raw state variables in Eq. (3.3). This aligns the learning signal with the physical quantities used to define terminal feasibility. The exponential term provides stronger shaping as the spacecraft approaches the target, which helps the agent receive useful feedback during terminal refinement. This shaping improves learning guidance, but it does not provide a formal guarantee

of global optimality.

Table 3.2 reports the weights used for the first DRL network. For the second DRL network, the weights associated with a_{sm} and e are increased by a factor of three. This reflects the role of DRL-2 as a terminal-refinement policy, where small errors in semimajor axis and eccentricity dominate the remaining feasibility requirement.

3.4.2 Cascaded DRL Algorithm

The complete training and inference process is summarized in Algorithm 1. The algorithm combines the three procedural components of the original implementation: SAC training, segment-size selection, and environment propagation. The first stage trains DRL-1 from the initial orbit. The second stage selects a successful DRL-1 policy and uses the resulting terminal neighborhood to define the start of the refinement problem. The third stage trains DRL-2 with tighter tolerances. The final stage executes the two policies sequentially during inference.

The segment-size rule used by DRL-1 is based on the spacecraft's proximity to the target:

$$n = \begin{cases} 0.1, & e \leq 0.01, i \leq 0.1, a_{\text{sm}} \in \{a_{\text{sm}}^{\text{tar}} \pm 200 \text{ km}\}, \\ 1, & e \leq 0.01, i \leq 0.1, a_{\text{sm}} \in \{a_{\text{sm}}^{\text{tar}} \pm 2100 \text{ km}\}, \\ 10, & \text{otherwise.} \end{cases} \quad (3.18)$$

The segment size n is measured in degrees of revolution. A larger segment size means the policy updates less frequently, which keeps early training episodes shorter. A smaller segment size gives the policy finer control near the target. DRL-2 uses the high-resolution setting because it begins near the final orbit.

Table 3.3 reports the coarse and refined tolerance values associated with the two DRL networks.

The tolerance progression in Table 3.3 is the main operational expression of the staged design. DRL-1 is allowed to terminate in a broader region, while DRL-2 is evaluated against substantially tighter final bounds.

Algorithm 1 Cascaded reinforcement learning for long-horizon precision control

Require: Initial state s_0 , dynamics model, reward function, DRL-1 tolerance set $\mathcal{G}_1$, DRL-2 tolerance set $\mathcal{G}_2$, SAC hyperparameters

Ensure: Trained policies π_1 and π_2 , and cascaded transfer trajectory

```
1: Initialize SAC actor, critic, target critic, and replay buffer for DRL-1
2: for each DRL-1 training episode do
3:   Set  $s_t \leftarrow s_0$ 
4:   while episode is not terminated do
5:     Sample thrust-angle action  $a_t \sim \pi_1(\cdot|s_t)$ 
6:     Select segment size  $n$  using the current  $a_{\text{sm}}$ ,  $e$ , and  $i$ 
7:     Propagate the spacecraft dynamics over the selected segment
8:     Compute  $r_{t+1}$  using Eq. (3.17)
9:     Store  $(s_t, a_t, r_{t+1}, s_{t+1}, d_t)$  in replay buffer
10:    Update SAC actor, critic, and target networks from replay
11:    Set  $s_t \leftarrow s_{t+1}$ 
12:  end while
13: end for
14: Select the DRL-1 checkpoint with the minimum successful convergence time
15: Define the DRL-2 training region from the terminal neighborhood reached by DRL-1
16: Initialize SAC actor, critic, target critic, and replay buffer for DRL-2
17: for each DRL-2 training episode do
18:   Initialize  $s_t$  near the DRL-1 terminal region
19:   while episode is not terminated do
20:     Sample thrust-angle action  $a_t \sim \pi_2(\cdot|s_t)$ 
21:     Propagate the dynamics using high-resolution action updates
22:     Compute  $r_{t+1}$  using Eq. (3.17)
23:     Store transition and update SAC networks from replay
24:     Set  $s_t \leftarrow s_{t+1}$ 
25:   end while
26: end for
27: Execute  $\pi_1$  from  $s_0$  until the coarse target set  $\mathcal{G}_1$  is reached
28: Switch to  $\pi_2$  and continue until the final target set  $\mathcal{G}_2$  is reached
29: Return the cascaded trajectory, transfer time, terminal elements, revolution count, and mass loss
```

3.5 Experimental Setup

The experiments are conducted in simulation using Python 3.7 and PyTorch with an implementation based on OpenAI stable-baselines style components. The reported hardware includes an Intel Xeon E5-1620 v4 CPU at 3.5 GHz, an NVIDIA GeForce GTX 1080 GPU, and 32 GB of RAM. Training is performed over repeated episodes until stable reward behavior is observed for successful transfers.

Table 3.3: *Convergence tolerances for the cascaded DRL networks.*

Scenario	Network	a_{sm} (km)	e	i (deg)
GTO-1	Initial	24364	0.7306	28.5
	DRL-1	$\leq \pm 55$	≤ 0.01	≤ 0.1
	DRL-2	$\leq \pm 0.2$	$\leq 5 \times 10^{-5}$	≤ 0.08
GTO-2	Initial	24364	0.7310	27
	DRL-1	$\leq \pm 55$	≤ 0.1	≤ 0.1
	DRL-2	$\leq \pm 0.08$	$\leq 8.7 \times 10^{-5}$	≤ 0.08
GTO-3	Initial	24505	0.725	28.5
	DRL-1	$\leq \pm 55$	≤ 0.01	≤ 0.1
	DRL-2	$\leq \pm 12$	$\leq 9 \times 10^{-4}$	≤ 0.08
Super-GTO	Initial	51525	0.8705	22.5
	DRL-1	$\leq \pm 10$	≤ 0.002	≤ 0.1
	DRL-2	$\leq \pm 0.5$	$\leq 6.9 \times 10^{-5}$	≤ 0.08

The reported experiments use a maximum of 900 training episodes, and the selected trained weights are chosen based on minimum successful convergence time during inference.

The baselines include both automated and traditional trajectory-planning methods. The main automated baseline is the method of Sreesawet and Dutta²⁵, which is relevant because it does not require a user-provided initial guess. The GTO-3 case compares against the single-agent RL method of Kwon et al.²⁹. Traditional numerical references include direct and optimization-based approaches reported in the low-thrust orbit-raising literature^{20;54–60}. These comparisons are interpreted carefully because shadow geometry, perturbation models, spacecraft assumptions, and optimization formulations can differ across studies.

The evaluation metrics are transfer time, terminal tolerance satisfaction, orbital-element convergence, training convergence, revolution count, and mass loss. Transfer time evaluates the efficiency of the policy. Terminal tolerance and orbital-element convergence evaluate physical feasibility. Training convergence indicates whether the staged policies learn reliably under the shaped reward. Revolution count and mass loss connect the learned behavior back to the physical transfer process.

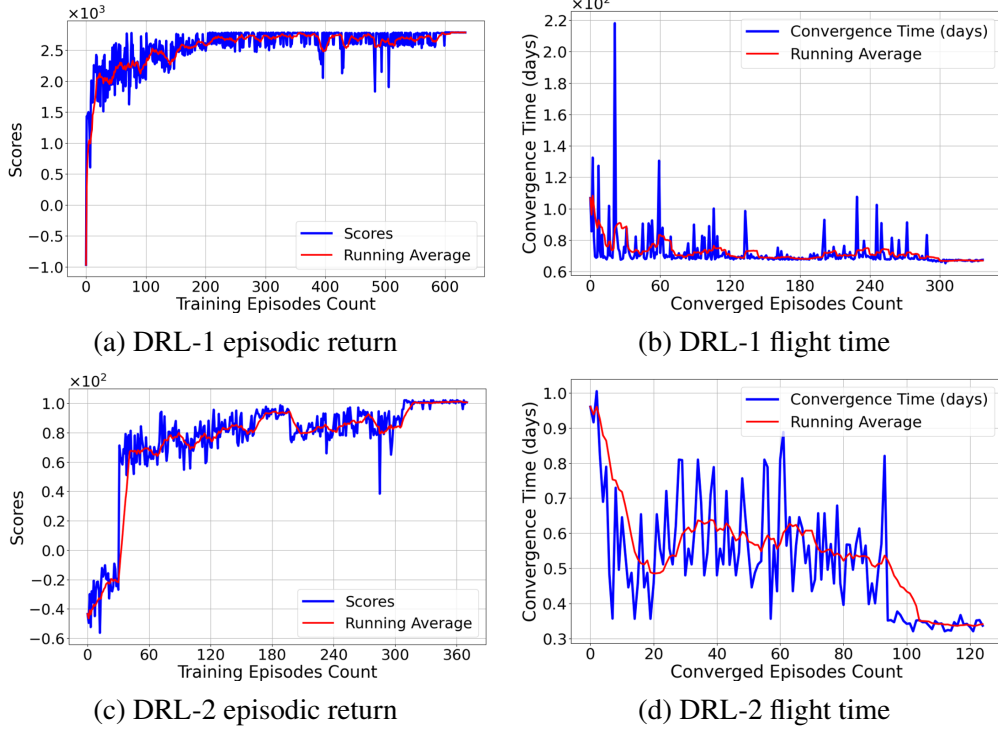


Figure 3.2: *Episodic return and convergence time over training for the GTO-2 transfer. DRL-1 and DRL-2 represent the two cascaded reinforcement learning agents.*

3.6 Empirical Results and Analysis

3.6.1 Training and Learned Behavior

The first empirical question is whether both stages can learn stable policies under the shaped reward and variable action-resolution strategy. Figure 3.2 reports the GTO-2 training behavior for the two cascaded agents. The episodic return stabilizes as training progresses, and the flight time of successful episodes decreases. This matters because the staged method depends on both policies learning reliably: DRL-1 must learn the long orbit-raising phase, while DRL-2 must learn a shorter but more precise terminal correction.

The separation between the two stages is visible in the scale of the learned behavior. DRL-1 accounts for most of the transfer duration and learns the broad orbit-raising behavior. DRL-2 operates over a much shorter terminal segment, where the remaining objective is high-resolution convergence rather than global orbital change. The training curves therefore support the design assumption behind the cascade: the two policies learn different parts of the same long-horizon

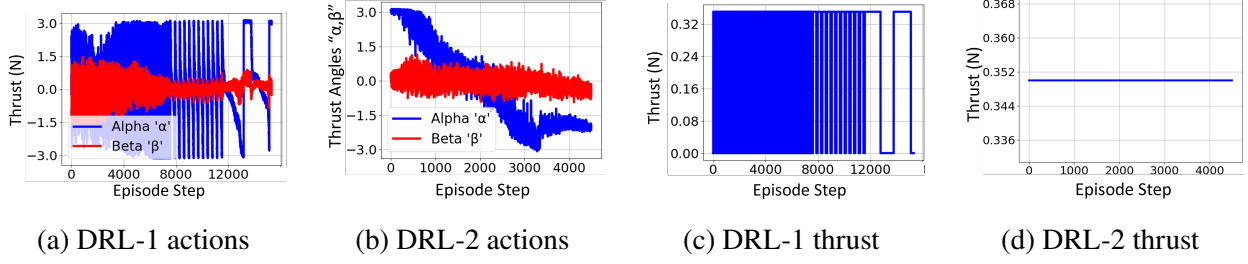


Figure 3.3: Predicted thrust angles and thrust values for the GTO-GEO transfer. The action variables are the thrust-direction angles α and β , while the thrust plots show the propulsion availability used by the environment.

precision-control problem.

Figure 3.3 shows the predicted thrust-angle actions and thrust values for the GTO-GEO case. The action variables are α and β , the two angles that orient the thrust vector. The early part of DRL-1 contains larger action variation because the spacecraft is far from the target and the segment size is coarser. Near the target, action updates become finer. DRL-2 then performs the terminal correction at high resolution.

These plots are not only diagnostic. They show that the learned policy operates through physically meaningful control inputs and that the environment imposes thrust availability according to the propulsion and eclipse assumptions. The policy is therefore learning under the same actuation structure used to evaluate the trajectory.

3.6.2 GTO Transfer Results

The GTO-to-GEO cases provide the main evidence for staged precision control. Figure 3.4 shows a three-dimensional trajectory generated by the learned policy. The spacecraft begins on an elliptical transfer orbit and gradually approaches GEO. The trajectory illustrates the accumulated effect of many small thrust-direction decisions, but terminal feasibility must be evaluated through the orbital elements rather than through geometry alone.

Figure 3.5 reports the orbital-element histories for the GTO-1 and GTO-2 cases. These plots track semimajor axis, eccentricity, and inclination, which are the same quantities used in the terminal feasibility checks. The first row corresponds to DRL-1, which reduces the large initial

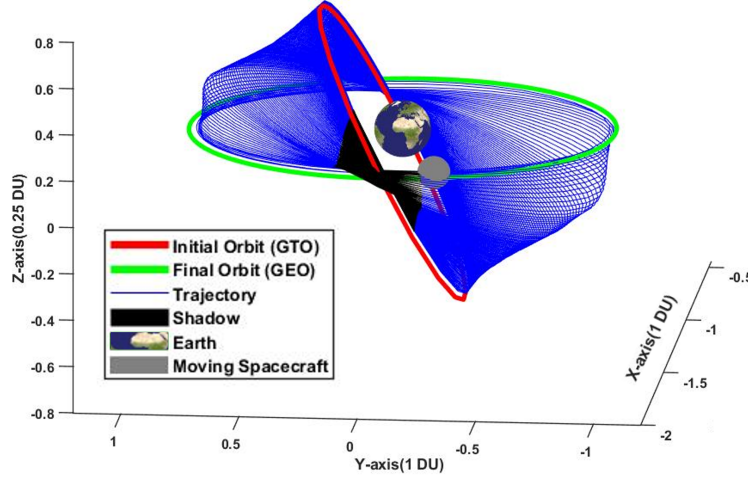


Figure 3.4: Three-dimensional orbit-transfer trajectory for GTO-GEO, where one distance unit is equivalent to the GEO radius.

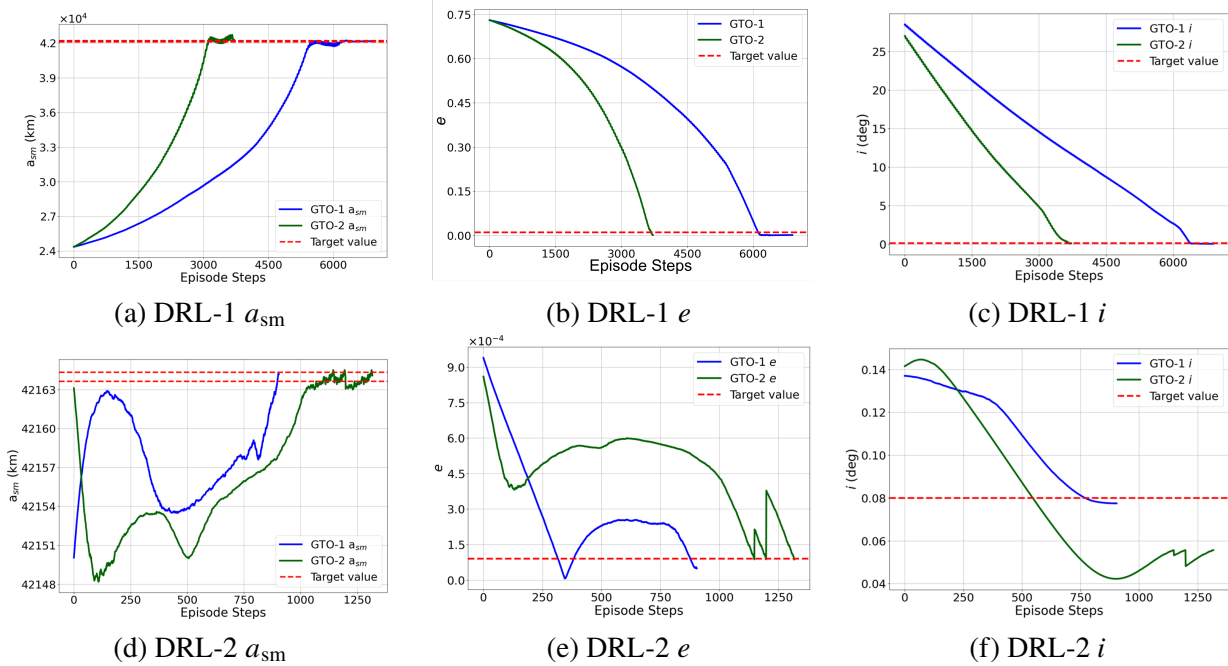


Figure 3.5: Orbital-element variation in one complete episode for GTO-1 and GTO-2 to GEO transfer. The plotted variables correspond directly to the terminal feasibility checks.

orbital mismatch. The second row corresponds to DRL-2, which refines the remaining error near the target.

The tolerance refinement is most evident when Figure 3.5 is read together with Table 3.3. In GTO-1, DRL-1 reaches a semimajor-axis tolerance of ± 55 km, while DRL-2 refines this to ± 0.2

Table 3.4: Transfer-time comparison for GTO-GEO transfer cases. I_{sp} denotes specific impulse, while λ and P denote propulsion efficiency and power.

Case	Spacecraft parameters	Thrust	Transfer Time (days)			Automated baseline	Traditional reference
			DRL-1	DRL-2	Total		
GTO-1	$I_{sp} = 1800$ s, $\lambda = 55\%$, $P = 5$ kW, $m = 1200$ kg	$F = 0.31$, no thrust in shadow	116.75	0.25	117.00	139.76 ²⁵	121.22 ⁵⁴ ; 119.20 ⁵⁸ ; 112.11–118.36 ⁵⁵ ; 118.29 ⁵⁶ ; 118.36 ²⁰ ; 119.35–120.19 ⁶⁰
GTO-2	$I_{sp} = 3300$ s, $\lambda = 65\%$, $P = 5$ kW $m = 450$ kg	$F = 0.2007$, no thrust in shadow	70.08	0.31	70.38	83.43 ²⁵ 83.43 ²⁵	65.9 ⁵⁴ 65.9 ⁵⁴
GTO-3	$I_{sp} = 3000$ s, $m = 2000$ kg	$F = 0.35$, constant thrust	145.19	1.24	146.43	150.4 ²⁹	–

km. In the same case, eccentricity improves from 0.01 to 5×10^{-5} . In GTO-2, the final semimajor-axis tolerance is ± 0.08 km. These results show that the second stage is not merely shortening the last part of the transfer; it is materially improving terminal precision.

Table 3.4 compares transfer times for the three GTO-to-GEO cases. The cascaded DRL method improves over the automated baseline in all three cases. Relative to traditional numerical references, the method is competitive in some settings and slower in others. This mixed comparison is important for a defensible dissertation claim: the staged RL method improves automated policy-based transfer computation under the modeled assumptions, but it does not establish global optimality over all numerical optimal-control methods.

For GTO-1, the cascaded DRL transfer time is 117.00 days compared with 139.76 days for the automated baseline. For GTO-2, the cascaded result is 70.38 days compared with 83.43 days for the automated baseline. For GTO-3, the cascaded result is 146.43 days compared with 150.4 days for the single-agent RL baseline, corresponding to the reported 2.64% improvement. The more important dissertation-level result is the combination of improved automated transfer time and tighter terminal convergence through staged refinement.

3.6.3 Physical Interpretation

Table 3.5 summarizes revolution counts, mass-to-thrust ratios, and mass loss. These quantities help interpret transfer time in physical terms. The learned policy is not being evaluated on an abstract reward alone; it is producing a physical transfer whose duration depends on thrust level, spacecraft

Table 3.5: *Reported revolution counts, mass-to-thrust ratios, and mass loss.*

Scenario	Revolutions	Mass-to-thrust ratio	Mass loss (kg)	Interpretation
GTO-1	178.07	3870	173.84	Requires more revolutions than GTO-2 because of the larger mass-to-thrust ratio.
GTO-2	103.81	2242	36.29	Completes the transfer with fewer revolutions and lower mass loss than GTO-1.
GTO-3	223.16	5714	150.47	Requires the largest number of revolutions among the GTO cases, consistent with the highest reported mass-to-thrust ratio.
Super-GTO	74.51	2988	84.56	Completes in fewer revolutions than GTO-1 despite the higher initial eccentricity and semimajor axis.

mass, initial orbit shape, and eclipse exposure.

The comparison between GTO-1 and GTO-2 illustrates the role of physical parameters. GTO-1 has a higher mass-to-thrust ratio and requires 178.07 revolutions, while GTO-2 requires 103.81 revolutions. GTO-3 has an even higher mass-to-thrust ratio and requires 223.16 revolutions. These results reinforce the chapter’s main interpretation: staged RL is embedded in a physical decision process, and its behavior must be evaluated through both learning metrics and physical transfer quantities.

3.6.4 Super-GTO Extension

The super-GTO case evaluates whether the cascaded method remains useful from a more eccentric and higher semimajor-axis initial orbit. The reward weights for eccentricity are increased relative to the GTO cases, reflecting the greater difficulty of converging from the super-GTO initial condition. Table 3.6 reports the transfer-time comparison.

Table 3.6: *Transfer-time comparison for the super-GTO to GEO transfer case.*

Case	Spacecraft parameters	Thrust	Transfer Time (days)			Automated baseline	Traditional reference
			DRL-1	DRL-2	Total		
Super-GTO	$I_{sp} = 3300$ s, $\lambda = 65\%$, $P = 10$ kW $m = 1200$ kg	$F = 0.4015$, no thrust in shadow	80.60	0.27	80.87	82.11 ²⁵	72.60–75.42 ⁵⁴

The qualitative comparison reported in Table 3.6 remains clear: the cascaded DRL method improves over the automated baseline and remains slower than the traditional numerical reference

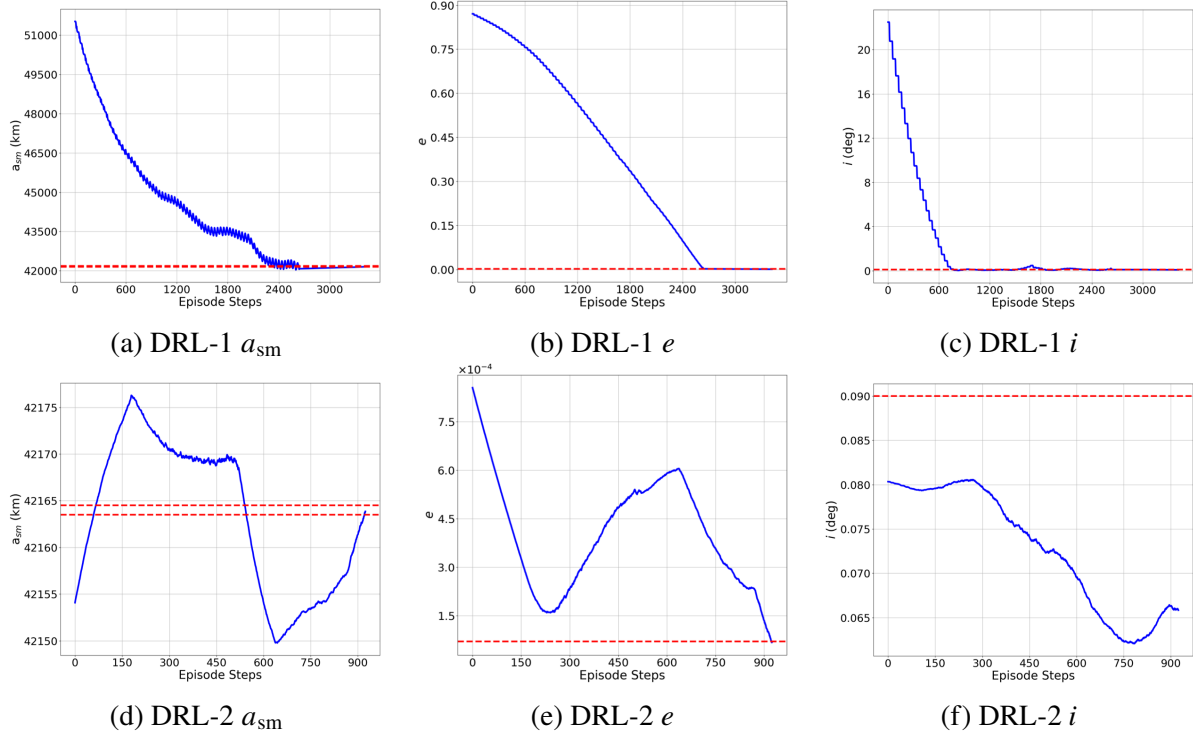


Figure 3.6: *Orbital-element variation in one complete episode for the super-GTO to GEO transfer.*

under its reported assumptions.

Figure 3.6 shows the super-GTO orbital-element convergence. The first stage accounts for most of the transfer and brings the spacecraft near the final orbit. The second stage completes the high-precision terminal refinement. The reported total number of revolutions is 74.51, with 73.63 revolutions for DRL-1 and 0.88 revolutions for DRL-2.

Figure 3.7 shows the corresponding three-dimensional transfer trajectory. The larger initial orbit and the shadow region make the setting physically distinct from the GTO cases. The result demonstrates that the staged framework remains applicable beyond a single GTO initial condition.

The super-GTO case strengthens the empirical evidence because it stresses eccentricity convergence from a more challenging initial orbit. At the same time, the comparison with the faster traditional reference prevents overstatement. The appropriate conclusion is that cascaded DRL improves automated transfer computation and terminal convergence under the modeled conditions, while high-fidelity numerical optimal-control methods remain important references.

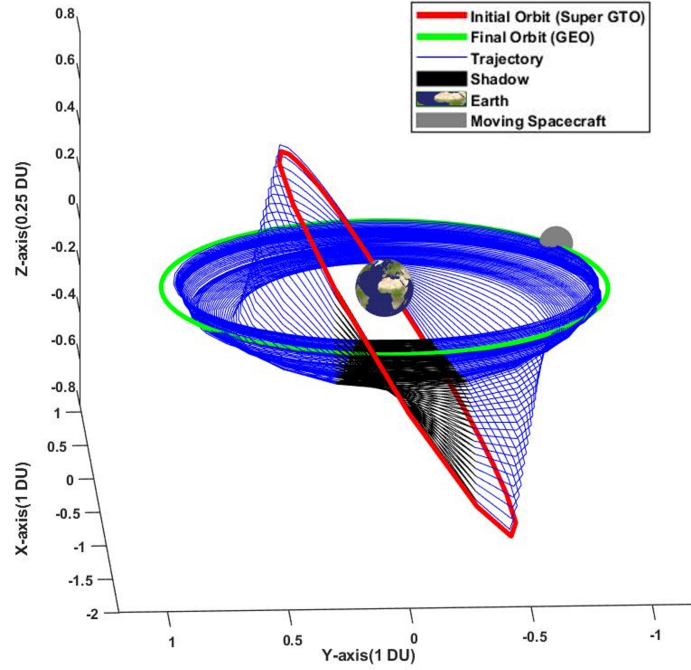


Figure 3.7: *Three-dimensional orbit-transfer trajectory for super-GTO to GEO, where one distance unit is equivalent to the GEO radius.*

3.6.5 Summary of Findings

The results support the staged-learning argument in three ways. First, the DRL-1 and DRL-2 tolerance progression shows that the second policy materially improves terminal precision rather than merely continuing the first policy. Second, the transfer-time tables show that the cascaded policies improve over the selected automated baselines in the reported GTO and super-GTO comparisons. Third, the revolution-count and mass-loss results show that the learned behavior remains tied to the physical characteristics of the transfer, including thrust level, spacecraft mass, eclipse exposure, and initial orbital geometry.

Taken together, the empirical evidence supports a measured conclusion: staged reinforcement learning can produce time-efficient and terminally precise automated transfers under the modeled low-thrust dynamics. The results do not remove the need for classical trajectory optimization or higher-fidelity validation, but they show that decomposition is a useful structure for learning long-horizon decision policies in a representative autonomous physical system.

3.7 Discussion

This chapter contributes empirical evidence that staged reinforcement learning can produce time-efficient, terminally precise transfers under representative low-thrust dynamics. The evidence supports the broader dissertation thesis through three findings. First, staged decomposition enables terminal refinement that would be difficult for a single policy to learn directly across the full transfer horizon. This is reflected in the DRL-1 to DRL-2 tolerance progression in Table 3.3. Second, the learned policies provide closed-loop inference after offline simulation training, reducing the need to solve a new numerical trajectory-optimization problem at every deployment decision point. Third, reward shaping based on mission-relevant orbital elements aligns the learning signal with the physical quantities that determine terminal feasibility.

The chapter is strongest when interpreted as a method-led contribution rather than as a claim that learned policies dominate all trajectory-optimization approaches. Classical optimization remains essential for high-fidelity mission design, verification, and constraint analysis. The contribution here is that a cascaded RL structure can automate a difficult long-horizon decision process and produce competitive transfer times while satisfying strict terminal tolerances under the modeled assumptions.

The scope boundaries also point to natural future work. Higher-fidelity studies should incorporate orbital perturbations, solar-array degradation, navigation uncertainty, operational constraints, and model mismatch. Additional work is also needed to study sensitivity to reward weights, the choice of intermediate tolerance regions, and generalization across broader mission classes. These extensions would move the method closer to operational relevance while preserving the main insight of the chapter: long-horizon physical decision policies benefit when learning is structured around the different precision demands of the task.

The chapter also clarifies why another structure is needed in the next stage of the dissertation. Staging helps with horizon length and terminal precision, but it does not directly change how the actor and critic represent decision-relevant features. The policies still rely on the representational capacity of standard neural networks. Chapter 4 addresses this complementary issue by studying attention-enhanced actor-critic learning for complex physical dynamics.

3.8 Summary and Transition

This chapter presented staged reinforcement learning for long-horizon precision control. The method decomposes low-thrust orbit transfer into two SAC-based stages: DRL-1 for coarse long-horizon transfer and DRL-2 for terminal refinement. The chapter introduced the orbital-transfer terminology needed to understand the validation setting, defined the physical state and action parameterization, described the low-thrust dynamics and eclipse model, formulated the reinforcement learning problem, developed the gradient-aided reward function, and presented the cascaded training and inference procedure.

The empirical results show that the staged method improves terminal feasibility and produces time-efficient automated transfers under the modeled assumptions. The GTO and super-GTO experiments demonstrate that the second policy materially tightens terminal tolerances, while the transfer-time comparisons show improvement over selected automated baselines. These results support the dissertation thesis that reinforcement learning for autonomous physical systems benefits from structures that match the long-horizon physical decision problem.

Chapter 4 builds on this conclusion by considering a different structural need. While this chapter used staged decomposition to address horizon length and terminal precision, the next chapter examines attention-enhanced actor-critic learning as a mechanism for improving feature relevance and sample efficiency in complex physical dynamics.

Chapter 4

Attention-Enhanced Actor–Critic Learning for Complex Physical Dynamics

Long-horizon precision control presents a design tension: policies must make coarse progress across large state regions while eventually satisfying strict terminal tolerances. Chapter 3 addressed this tension through staged reinforcement learning, where separate policies handle global transfer and terminal refinement. A representational challenge still remains inside each stage: which state-action features matter most for long-term value?

Standard actor–critic learning can treat state and action features too uniformly, even when their relevance changes across the trajectory. This chapter addresses that limitation through attention-enhanced actor–critic learning. The method, Single-Agent Attention Actor–Critic (SA3C), introduces an attention mechanism that links actor features with critic features, allowing policy learning to emphasize features that are more relevant to value estimation in complex physical dynamics. The aerospace settings are used to validate the method, not to define the contribution. The evaluation includes geocentric orbit-raising cases that provide continuity with Chapter 3, as well as cislunar transfer cases that introduce stronger third-body gravitational effects and more demanding terminal conditions.

4.1 Problem Setting and Preliminaries

4.1.1 From Decomposition to Feature Relevance

Staged decomposition reduces the temporal burden of a long-horizon task by separating broad progress from terminal refinement. It does not, however, determine how an actor or critic should use the features available within each stage. In continuous physical control, feature relevance is state dependent and phase dependent. A variable that strongly influences long-term value early in a trajectory may become less important near the target, while a variable that is nearly secondary at the start may dominate terminal feasibility later.

A simple example is inclination during GTO-to-GEO transfer. Early in the trajectory, when eccentricity is high, large-scale orbital reshaping may dominate the transfer. Near GEO, when eccentricity has already been reduced, inclination and other terminal constraints can become more influential. In cislunar transfer, the situation becomes more complex because lunar motion, third-body gravity, and epoch alignment can make timing and orbital orientation increasingly important. If actor–critic learning treats all features uniformly, the policy may require more samples to discover which variables matter at each phase.

SA3C addresses this issue by making the actor–critic update more representation-aware. The actor proposes actions through its feature representation, while the critic evaluates the long-term value of actions. The attention module uses critic-related features to reweight actor features before generating an attention-influenced action. The purpose is not to introduce attention as a generic neural-network component, but to make actor–critic learning more selective in a physical decision problem where the importance of different variables is state dependent and phase dependent.

4.1.2 Attention as Actor–Critic Feature Coupling

Actor–critic learning separates action selection from value estimation. The actor maps states to actions, while the critic evaluates how useful those actions are for long-term return. This separation is powerful, but it also creates a coordination problem. If the actor and critic learn internal features that are weakly aligned, policy improvement can be inefficient. The actor may produce actions

from features that are locally plausible, while the critic may estimate value from different features that better reflect long-horizon outcomes.

Attention provides a mechanism for improving this interaction. In SA3C, the attention module uses actor and critic features to construct attention-weighted actor features. These features produce an attention-influenced action, which is then evaluated by the critic. The resulting loss augments the standard SAC critic loss. The attention mechanism therefore influences learning at the point where action-generation features and value-estimation features meet.

The physical motivation is feature relevance under changing dynamics. During a transfer, the policy may need to emphasize semimajor axis, eccentricity, inclination, orientation variables, mass, time, or epoch alignment depending on the current phase. In cislunar settings, third-body gravity can make small differences in timing and state orientation more important than they would be in a simpler geocentric case. Attention is used here as a structural bias that allows actor–critic learning to adapt feature emphasis as the trajectory evolves.

4.2 Methodological Challenge and Design Principle

The physical decision problem in this chapter is minimum-time or near-minimum-time low-thrust trajectory planning. The agent controls thrust direction, while the spacecraft evolves under an astrodynamical transition model. Chapter 3 already introduced the basic GTO-to-GEO setting, so this chapter only recalls the geocentric cases when needed for comparison. The main extension is that the same attention-enhanced framework is evaluated not only on geocentric orbit raising but also on cislunar transfer scenarios.

A cislunar transfer occurs in the Earth–Moon region, where the gravitational influence of both bodies shapes the spacecraft trajectory. This is more sensitive than the geocentric two-body setting because the Moon moves during the transfer and its gravity can significantly change the trajectory. A near-rectilinear halo orbit (NRHO) is a lunar-region orbit useful for future deep-space infrastructure because it remains linked to the Moon while extending into cislunar space. The NRHO target in this chapter is an NRHO-L2 orbit with a 9:2 synodic resonance. The patch-point target is an intermediate state located on a stable manifold leading toward the NRHO. In practical

terms, it is a carefully selected terminal state from which the natural dynamics can help guide the spacecraft toward the desired lunar-region trajectory.

The evaluation therefore tests two levels of complexity. The geocentric cases test whether attention improves actor–critic learning in long-duration Earth-centered orbit raising. The cislunar cases test whether the same mechanism remains useful when the dynamics include multibody gravitational effects and terminal conditions beyond semimajor axis, eccentricity, and inclination.

4.3 Technical Formulation

4.3.1 State, Dynamics, and Perturbation Model

The state representation extends the orbital-element formulation used in Chapter 3 by including spacecraft mass and time:

$$\mathbf{s} = \begin{bmatrix} h & h_X & h_Y & e_x & e_y & m & t \end{bmatrix}^T. \quad (4.1)$$

The first five variables describe the slowly varying orbital geometry, while m and t track propellant use and elapsed transfer time. The dynamics are discretized using fourth-order Runge–Kutta integration over small control intervals. The control is treated as piecewise constant over each integration step.

The continuous dynamics use ϕ as the independent variable. The orbital-element, mass, and time evolution are written as

$$\begin{aligned} \frac{d}{d\phi} \begin{bmatrix} h \\ h_X \\ h_Y \\ e_x \\ e_y \end{bmatrix} &= \frac{1}{\dot{\phi}} \mathbf{C} \begin{bmatrix} F_r \\ F_n \\ F_h \end{bmatrix}, \\ \frac{d}{d\phi} \begin{bmatrix} m \\ t \end{bmatrix} &= \frac{1}{\dot{\phi}} \begin{bmatrix} -F_T/(g_0 I_{sp}) \\ 1 \end{bmatrix}. \end{aligned} \quad (4.2)$$

The force-mapping matrix $\mathbf{C}$ follows the same orbital-element structure as the matrix used in

Chapter 3, while the physical force model is richer in this chapter. Three additional effects shape the transfer dynamics.

First, Earth oblateness is modeled through the J_2 perturbation:

$$\mathbf{F}_{J_2} = -\mu m J_2 R_E^2 \left[\frac{3z}{r_{sc}^5} \hat{\mathbf{K}} + \left(\frac{3}{2r_{sc}^4} - \frac{15z^2}{2r_{sc}^6} \right) \frac{\mathbf{r}_{sc}}{\|\mathbf{r}_{sc}\|} \right]. \quad (4.3)$$

Second, lunar and solar third-body gravity are included as

$$\begin{aligned} \mathbf{F}_{third} = m \left[\mu_{Moon} \left(\frac{\mathbf{r}_{Moon} - \mathbf{r}_{sc}}{\|\mathbf{r}_{Moon} - \mathbf{r}_{sc}\|^3} - \frac{\mathbf{r}_{Moon}}{\|\mathbf{r}_{Moon}\|^3} \right) \right. \\ \left. + \mu_{\odot} \left(\frac{\mathbf{r}_{\odot} - \mathbf{r}_{sc}}{\|\mathbf{r}_{\odot} - \mathbf{r}_{sc}\|^3} - \frac{\mathbf{r}_{\odot}}{\|\mathbf{r}_{\odot}\|^3} \right) \right]. \end{aligned} \quad (4.4)$$

Third, eclipse is modeled using a smoothed apparent-disk approximation. Let a_D denote the apparent angular separation between the Sun and occulting body as seen from the spacecraft, and let a_{SR} and a_{BR} denote the apparent radii of the Sun and occulting body. The sunlight factor is

$$\gamma_L = \frac{1}{1 + \exp[-c_s(a_D - c_t(a_{BR} + a_{SR}))]}. \quad (4.5)$$

Here, $\gamma_L = 1$ corresponds to full sunlight, while $\gamma_L = 0$ corresponds to no sunlight and therefore no solar-electric thrust. The smoothed form avoids a hard discontinuity at shadow entry and exit.

The thrust vector is

$$\mathbf{F}_T = T_{max} \begin{bmatrix} \sin \alpha \cos \beta \\ \cos \alpha \cos \beta \\ \sin \beta \end{bmatrix} \gamma_L, \quad (4.6)$$

where

$$T_{max} = \frac{2\lambda_e P}{g_0 I_{sp}}. \quad (4.7)$$

The total noncentral force used by the transition model is

$$\begin{bmatrix} F_r & F_n & F_h \end{bmatrix}^T = \mathbf{F}_T + \mathbf{F}_{J_2} + \mathbf{F}_{third}. \quad (4.8)$$

This formulation keeps the learned action identical in form to Chapter 3, namely thrust direction, while expanding the physical environment to include the perturbations needed for cislunar transfers.

4.3.2 Action and Terminal Conditions

The action is the thrust-angle vector

$$\mathbf{a} = \begin{bmatrix} \alpha & \beta \end{bmatrix}^T, \quad (4.9)$$

with $\alpha \in [-\pi, \pi]$ and $\beta \in [-\pi/2, \pi/2]$. The predicted thrust angles remain fixed between action-prediction time steps.

Terminal conditions are evaluated using Keplerian orbital elements derived from the state. The eccentricity condition is

$$e_{tar} - e_{tol} \leq \left[\|\mathbf{e}\| = \sqrt{e_x^2 + e_y^2} \right] \leq e_{tar} + e_{tol}. \quad (4.10)$$

The semimajor-axis condition is

$$a_{sm}^{tar} - a_{sm}^{tol} \leq \left[a_{sm} = \frac{h^2}{\mu (1 - \|\mathbf{e}\|^2)} \right] \leq a_{sm}^{tar} + a_{sm}^{tol}. \quad (4.11)$$

The inclination condition is

$$i_{tar} - i_{tol} \leq \left[i = \arccos \left(\frac{h_z}{h} \right) \right] \leq i_{tar} + i_{tol}. \quad (4.12)$$

For cislunar targets, right ascension of the ascending node Ω , argument of periapsis ω , and target epoch are also included in the convergence criteria. This is one reason the cislunar cases are a stronger test of attention-enhanced learning: the policy must satisfy more terminal conditions under more sensitive dynamics.

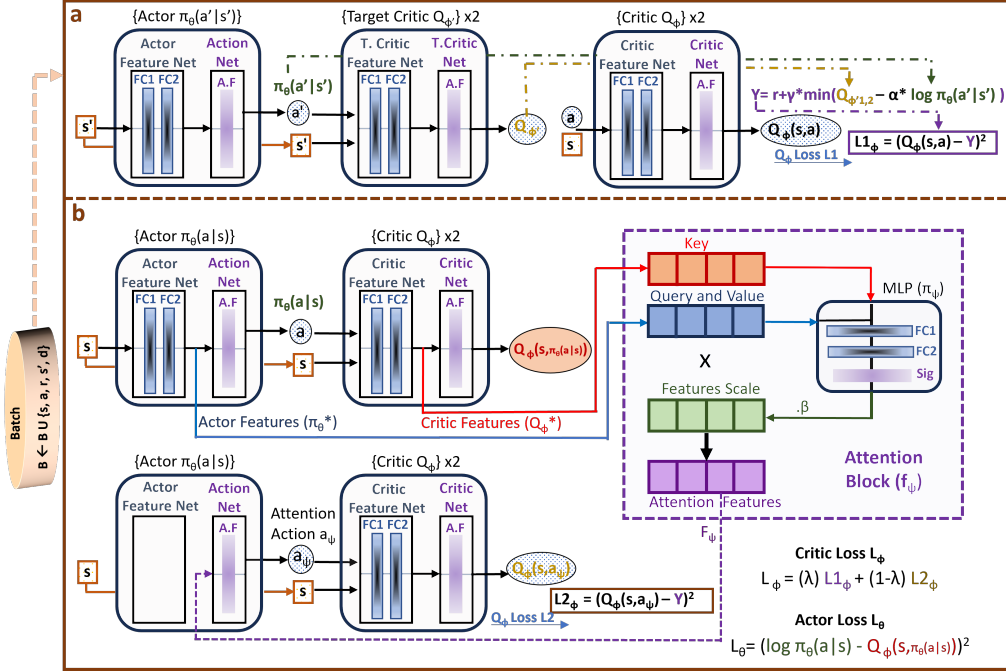


Figure 4.1: SA3C methodology. The framework augments the standard actor–critic update by using critic-related features to modulate actor features through an attention module. The resulting attention-weighted action contributes an additional critic loss term.

4.4 Learning Objective and Algorithm

4.4.1 Attention-Augmented Actor–Critic Update

SA3C builds on SAC by adding an attention-influenced critic update. The standard critic loss is

$$L1_{\phi} = \mathbb{E} (Q_{\phi}(s, a) - Y)^2, \quad (4.13)$$

where the target is

$$Y = r + \gamma(1 - d) \left(\min_{i=1,2} Q'_{\phi_i}(s', a') - \alpha \log \pi_{\theta}(a' | s') \right). \quad (4.14)$$

The actor network is decomposed into a feature extractor π_{θ}^* and an action head π_{θ}^- , so that

$$\pi_{\theta}(s) = \pi_{\theta}^- (\pi_{\theta}^*(s)). \quad (4.15)$$

The critic is similarly decomposed into a feature network Q_ϕ^* and a scalar-output network Q_ϕ^- :

$$Q_\phi(s, a) = Q_\phi^- \left(Q_\phi^*(s, a) \right). \quad (4.16)$$

The attention mechanism uses the actor feature representation as the Query and Value and the critic feature representation as the Key. Its feature-weighting function is

$$f_\psi(\text{Query}, \text{Key}) = \sigma(\text{Query}, \text{Key}^T) \beta \text{Value}, \quad (4.17)$$

where σ is the sigmoid function and β is a scaling factor. The resulting attention feature is

$$F_\psi = f_\psi \left(\pi_\theta^*, Q_\phi^*(s, \pi_\theta(a|s)) \right). \quad (4.18)$$

The attention-weighted action is then

$$a_\psi = \pi_\theta^-(F_\psi). \quad (4.19)$$

This action is evaluated by the critic, producing an attention-augmented critic loss

$$\mathbb{L}2_\phi = \mathbb{E} \left(Q_\phi(s, a_\psi) - Y \right)^2. \quad (4.20)$$

The total critic loss is

$$\mathbb{L}_\phi = \lambda \mathbb{L}1_\phi + (1 - \lambda) \mathbb{L}2_\phi, \quad (4.21)$$

where λ controls the balance between the standard critic loss and the attention-augmented loss.

The actor loss is

$$\mathbb{L}_\theta = \left(\log \pi_\theta(a|s) - Q_\phi(s|\pi_\theta(a|s)) \right)^2. \quad (4.22)$$

The attention network is trained through the change in actor loss before and after the attention-influenced critic update. This update encourages attention features that improve the actor's value-guided learning (Figure 4.1). The practical role of λ is evaluated empirically in Section 4.6.3.

Algorithm 2 Single-Agent Attention Actor–Critic for long-horizon physical decision policies

Require: Environment, actor network π_θ , critic networks Q_{ϕ_1}, Q_{ϕ_2} , target critics, attention module f_ψ , replay buffer $\mathcal{D}$, training hyperparameters

Ensure: Trained attention-enhanced actor–critic policy

- 1: Initialize actor, critics, target critics, attention module, and replay buffer
 - 2: **for** each training episode **do**
 - 3: Initialize spacecraft state s_0 for the selected transfer scenario
 - 4: **while** terminal condition is not satisfied and episode limit is not reached **do**
 - 5: Sample thrust-direction action $a_t \sim \pi_\theta(\cdot|s_t)$
 - 6: Propagate the spacecraft state using the transition model and thrust direction
 - 7: Compute shaped reward r_{t+1} and terminal flag d_t
 - 8: Store $(s_t, a_t, r_{t+1}, s_{t+1}, d_t)$ in replay buffer $\mathcal{D}$
 - 9: Sample a minibatch from $\mathcal{D}$
 - 10: Compute SAC target Y using Eq. (4.14)
 - 11: Compute standard critic loss $\mathbb{L}_\phi$ using Eq. (4.13)
 - 12: Compute actor and critic feature representations π_θ^* and Q_ϕ^*
 - 13: Compute attention features F_ψ and attention-weighted action a_ψ using Eqs. (4.18) and (4.19)
 - 14: Compute attention-augmented critic loss $\mathbb{L}_{2\phi}$ using Eq. (4.20)
 - 15: Update critics using the combined loss $\mathbb{L}_\phi$ in Eq. (4.21)
 - 16: Update actor using Eq. (4.22)
 - 17: Update attention module through its effect on the attention-influenced actor–critic losses
 - 18: Update target critic networks
 - 19: Set $s_t \leftarrow s_{t+1}$
 - 20: **end while**
 - 21: **end for**
 - 22: Select inference weights based on the shortest successful convergence time
 - 23: Return the trained SA3C policy and resulting transfer trajectory
-

Algorithm 2 summarizes the SA3C training procedure. The structure follows the SAC training loop, with the additional attention-augmented critic branch from Eqs. (4.18)–(4.21).

4.5 Experimental Setup

The evaluation includes five transfer scenarios. The geocentric cases are GTO-1 to GEO, GTO-2 to GEO, and super-GTO to GEO. The cislunar cases are super-GTO-2 to NRHO-L2 9:2 and super-GTO-2 to a patch point. Table 4.1 lists the initial orbital elements and propulsion parameters. Table 4.2 lists the target conditions and convergence tolerances for DRL-1 and DRL-2. The comparison frameworks are automated sequential methods, previously reported DRL-based methods,

Table 4.1: *Initial orbital elements and spacecraft propulsion specifications for the SA3C transfer scenarios. The first three scenarios evaluate geocentric transfers to GEO, while super-GTO-2 supports the cislunar NRHO and patch-point transfers.*

Initial Orbit	a_{sm} (km)	e	i (deg)	Ω (deg)	ω (rad)	mass (kg)	I_{sp} (s)	Thrust (N)	Epoch
GTO-1	24364	0.7306	28.53	0	0	1200	1800	0.31	22 Mar 2000
GTO-2	24364	0.7310	27	99	0	450	3300	0.20	1 Jan 2000
Super-GTO	51525	0.8705	22.5	0	0	1200	3300	0.40	20 Mar 2020
Super-GTO-2	44364	0.65	27	11.3	0	1000	1500	1	Free

and traditional optimization-based methods. The automated comparisons are important because they do not require user-provided initial trajectory guesses. Traditional optimization references are included because they provide strong trajectory-quality benchmarks, although they may require initial guesses and iterative numerical solving. The evaluation metrics are training stability, transfer time, terminal convergence, attention-coefficient sensitivity, and robustness under bounded thrust or initial-state variations.

4.6 Empirical Results and Analysis

4.6.1 Training Stability and Learned Policies

The first validation question is whether the attention-enhanced agents learn stable policies under the proposed dynamics and reward structure. Figure 4.2 reports the episodic return for the geocentric cases. DRL-1 and DRL-2 correspond to the two cascaded agents. The stabilization of the score curves indicates that the agents learn repeatable policies for both coarse transfer and terminal refinement.

Figure 4.3 shows the thrust-angle profiles for the GTO-1 to GEO case. The profiles demonstrate that the learned policy produces continuous thrust-direction commands rather than discrete maneuver templates. This matters for the chapter’s argument because the attention mechanism is embedded inside a policy that directly controls physically meaningful action variables.

For the cislunar scenarios, Figure 4.4 shows that the training scores also increase and stabilize. This is important because the cislunar cases include stronger third-body effects and additional terminal requirements, making convergence harder than in the geocentric cases.

Table 4.2: Target orbital elements and cascaded convergence tolerances for the SA3C transfer scenarios. DRL-1 defines the coarse convergence stage and DRL-2 defines the final target tolerance.

Target Orbit	Condition	a_{sm} (km)	e	i (deg)	Ω (rad)	ω (rad)	Epoch
GEO	Target	42164	0	0	–	–	Free
	DRL-1	$\leq \pm 55$	≤ -0.015	≤ -1	–	–	Free
	DRL-2	$\leq \pm 1$	$\leq -8 \times 10^{-4}$	≤ -0.08	–	–	Free
NRHO-L2 9:2	Target	376118	0.126	18.15	21.18	4.158	21 Aug 2022
	DRL-1	$\leq \pm 1000$	≤ -0.15	≤ -0.1	$\leq +1$	≤ -0.7	$\leq$ target
	DRL-2	$\leq \pm 3$	$\leq -6 \times 10^{-4}$	≤ -0.05	$\leq +0.01$	≤ -0.35	target
Patch Point	Target	230271.8	0.39	28.25	-0.12	4.44	06 Nov 2019
	DRL-1	$\leq \pm 5500$	$\leq +0.05$	≤ -1.5	$\leq +0.61$	$\leq +0.61$	$\leq$ target
	DRL-2	$\leq \pm 1$	$\leq +3 \times 10^{-3}$	≤ -0.6	$\leq +0.26$	$\leq +0.20$	target

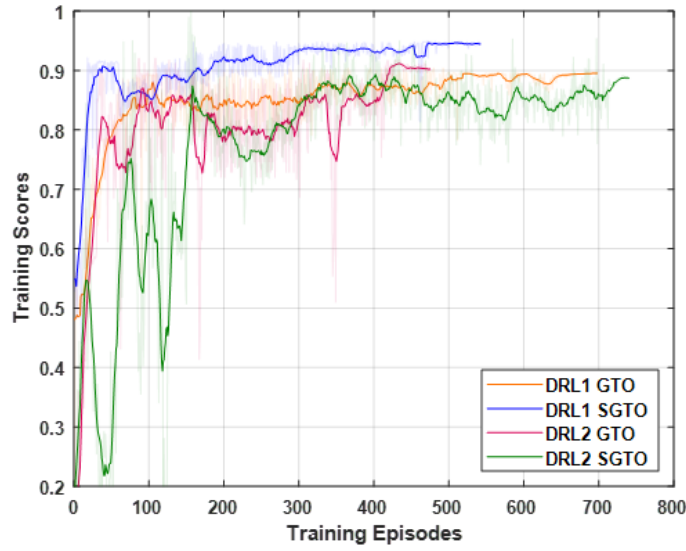


Figure 4.2: Training scores for the GTO-to-GEO and super-GTO-to-GEO scenarios. DRL-1 and DRL-2 denote the first and second cascaded SA3C networks.

4.6.2 Transfer Performance

The second validation question is whether SA3C improves transfer quality relative to automated alternatives while remaining competitive with traditional optimization references. Table 4.3 summarizes the transfer-time comparisons across geocentric and cislunar scenarios.

For the geocentric cases, SA3C improves over the automated baselines and improves on the corresponding cascaded DRL results from Chapter 3. GTO-1 reaches GEO in 114.20 days, compared with 117 days for the previous cascaded DRL result and 139.76 days for the sequential automated method. GTO-2 reaches GEO in 69.0 days, compared with 70.83 days and 83.43 days

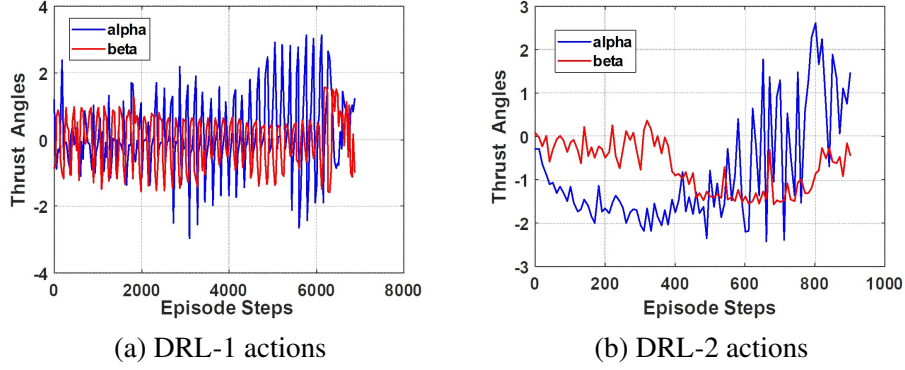


Figure 4.3: Predicted thrust-angle profiles for the GTO-1 to GEO scenario generated by the two cascaded SA3C agents.

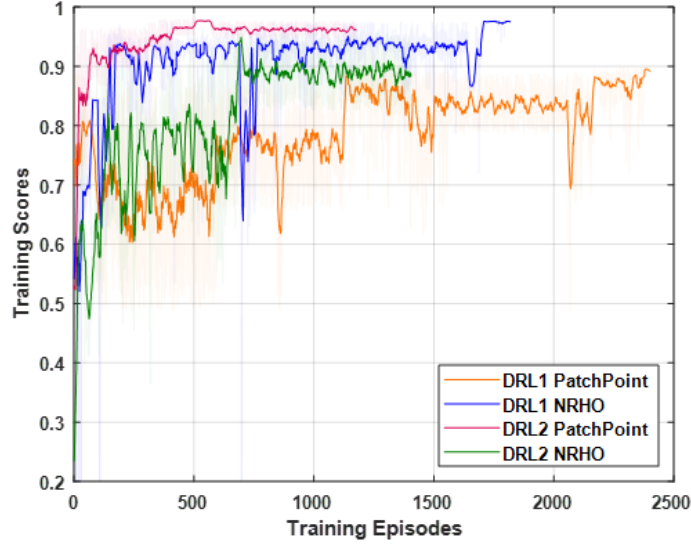


Figure 4.4: Training scores for the cislunar transfer scenarios from super-GTO-2 to NRHO and from super-GTO-2 to patch point.

for the corresponding automated references. Super-GTO reaches GEO in 79.9 days, compared with 80.87 days and 82.11 days. These results support the claim that attention improves the policy-learning process beyond staging alone.

Figure 4.5 shows the orbital-element convergence for the geocentric cases. The plotted quantities are semimajor axis, eccentricity, and inclination, which are the terminal quantities used to determine GEO convergence. The figure is important because transfer time alone is insufficient; the learned policy must also satisfy the physical terminal conditions.

The cislunar results test a different form of difficulty. The NRHO transfer reaches the target

Table 4.3: Transfer-time comparison for geocentric and cislunar scenarios. SA3C improves over the reported automated approaches in all listed cases while remaining close to traditional optimization references in several scenarios.

Initial orbit	Target	SA3C transfer time (days)			Automated approaches	ap-approaches	Traditional optimization approaches
		DRL-1	DRL-2	Total			
GTO-1	GEO	113.69	0.514	114.20	139.76 ²⁵ ; 117 ¹¹		121.22 ⁵⁴ ; 112–118.36 ⁵⁵ ; 118.62 ²⁰ ; 119–120.19 ⁶⁰ ; 119.20 ⁵⁸ ; 118.37 ⁶¹
GTO-2	GEO	68.5	0.5	69.0	83.43 ²⁵ ; 70.83 ¹¹		65.9 ⁵⁸ ; 66.8 ⁶¹ ; 66.6–67.0 ¹⁹
Super-GTO	GEO	79.73	0.26	79.9	82.11 ²⁵ ; 80.87 ¹¹		75.42 ⁵⁴ ; 76 ⁶¹
Super-GTO-2	NRHO	30.83	0.57	31.4	45.4 ⁶¹		29.02 ⁶² ; 28.74 ⁶¹
Super-GTO-2	Patch Point	27.4	0.2	27.6	36.3 ⁶¹		25.05 ⁶¹

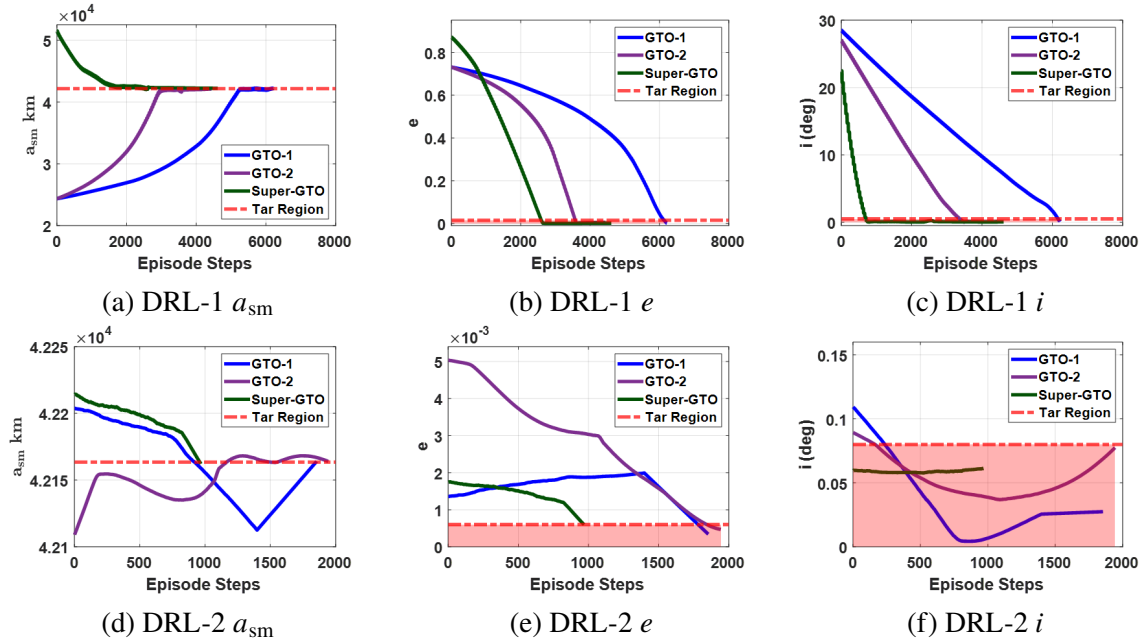


Figure 4.5: Orbital-element convergence for GTO and super-GTO transfers to GEO. The first row corresponds to DRL-1 and the second row corresponds to DRL-2.

in 31.4 days, compared with 45.4 days for the automated cislunar baseline and 29.02 and 28.74 days for traditional optimization references. The patch-point transfer reaches the target in 27.6 days, compared with 36.3 days for the automated sequential result and 25.05 days for the traditional optimization reference. These results show that the attention-enhanced actor–critic framework remains effective beyond the two-body geocentric setting.

Figure 4.6 and Figure 4.7 provide complementary evidence. The convergence plots show

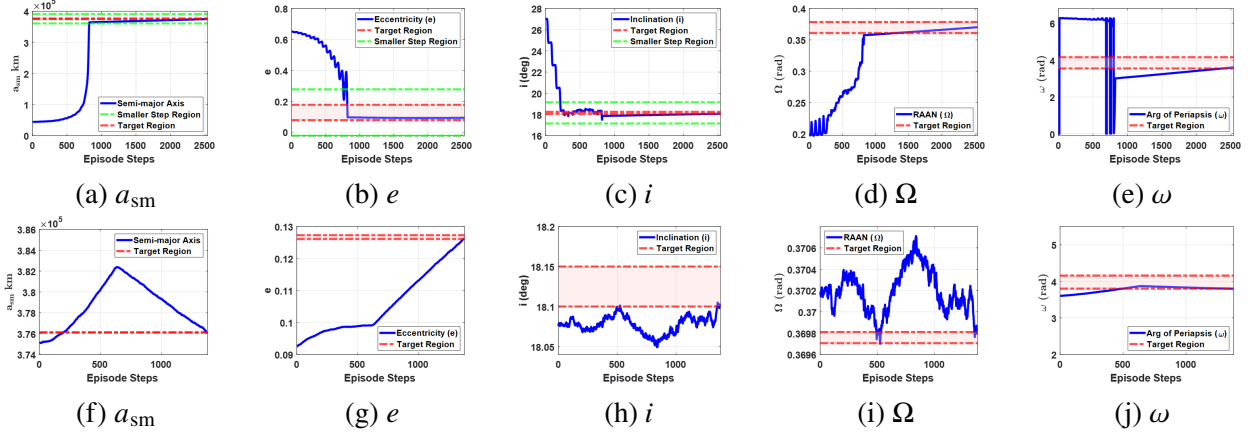


Figure 4.6: *Orbital-element convergence for the super-GTO-2 to NRHO-L2 9:2 transfer. The cislunar target requires convergence of five orbital elements rather than only the three GEO quantities.*

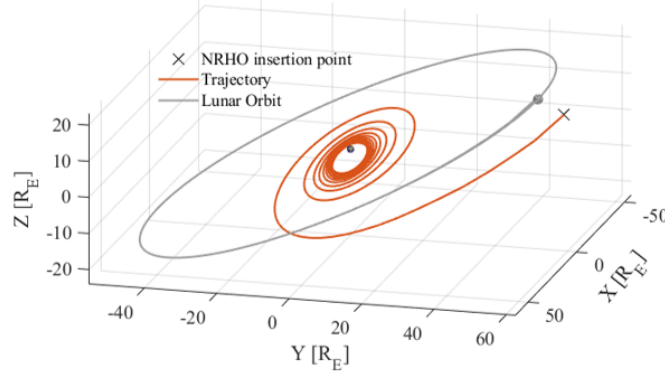


Figure 4.7: *Three-dimensional visualization of the transfer from super-GTO-2 to NRHO.*

terminal satisfaction in the orbital elements, while the trajectory visualization shows the learned transfer through the cislunar region. Together, they demonstrate that SA3C produces a complete trajectory, not only a numerically isolated terminal state.

4.6.3 Attention Mechanism Ablation

The ablation on λ directly tests the attention mechanism. Recall from Eq. (4.21) that $\lambda = 1$ removes the attention-augmented critic loss, reducing the critic update to the standard SAC form. Table 4.4 compares the best selected λ values with the $\lambda = 1$ setting.

The ablation supports the chapter’s methodological claim. In all reported scenarios, the attention-influenced setting improves over $\lambda = 1$ or enables convergence where $\lambda = 1$ does not

Table 4.4: Ablation on the attention coefficient λ . The setting $\lambda = 1$ removes the attention-augmented critic loss; lower values increase the influence of attention.

Scenario	Optimal λ	Transfer time with optimal λ	Transfer time with $\lambda = 1$
GTO-1 to GEO	0.7	114.20	117
GTO-2 to GEO	0.7	69	70.83
S-GTO to GEO	0.85	79.9	80.87
S-GTO-2 to NRHO	0.67	31.4	31.6
S-GTO-2 to Patch Point	0.6	27.6	No-Convergence

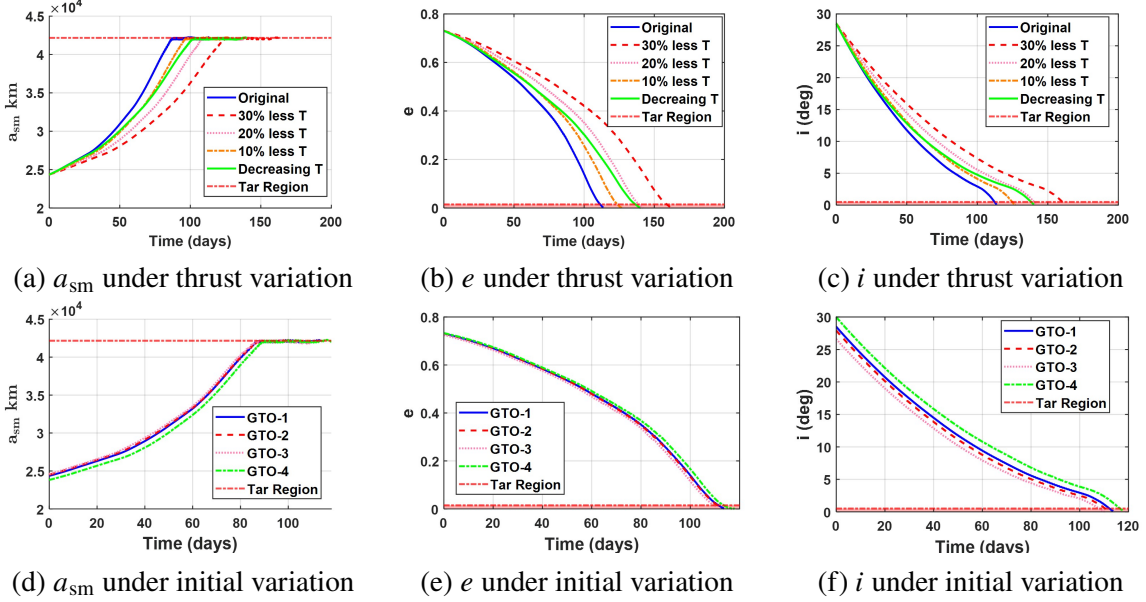


Figure 4.8: Trained-agent performance under thrust uncertainties and initial-condition variations for the GTO-GEO setting.

converge. The patch-point case is especially important because the standard SAC setting does not converge, while the attention-enhanced setting reaches the 27.6-day transfer reported in Table 4.4. This convergence contrast supports the role of attention in difficult cislunar cases.

4.6.4 Robustness Analysis

The robustness analysis evaluates the trained GTO-1 DRL-1 agent under thrust reductions and perturbed initial conditions. Figure 4.8 shows convergence behavior under nominal thrust, reduced thrust levels, linearly degrading thrust, and bounded initial-state variations. The result does not establish general robustness under all mission uncertainties, but it shows that the trained policy can adapt within the tested perturbation ranges without retraining.

This result matters because long-horizon physical decision policies should not collapse under small deviations from nominal assumptions. The bounded robustness evidence complements the main transfer results by showing that the learned policy retains corrective behavior under selected variations in thrust and initial orbital elements.

4.7 Discussion

This chapter contributes empirical evidence that attention-enhanced actor–critic learning can improve long-horizon physical decision policies when feature relevance changes across the trajectory. Three findings support the broader dissertation thesis. First, SA3C improves transfer efficiency relative to reported automated baselines across the evaluated geocentric and cislunar scenarios. Second, the attention ablation shows that the attention-augmented critic loss improves performance relative to standard SAC settings and enables convergence in the reported patch-point case. Third, the cislunar results demonstrate that the method extends beyond geocentric two-body transfer to multibody dynamics with more restrictive terminal conditions.

Together with Chapter 3, these findings show two complementary forms of structure for simulation-trained reinforcement learning in physical autonomy. Staged decomposition addresses horizon management and terminal precision. Attention-enhanced actor–critic learning addresses feature relevance inside policy optimization. Both structures are useful because long-horizon physical decision policies must both organize decisions across time and identify which features matter at each phase of the trajectory.

The scope of the evidence is simulation-based. The results depend on the selected mission cases, state representation, reward design, attention architecture, and hyperparameter choices. These boundaries point to natural extensions: richer perturbation models, navigation uncertainty, broader transfer families, automated tuning of λ , and tighter integration with high-fidelity optimal-control pipelines.

This chapter completes the dissertation’s simulation-trained trajectory-policy contributions. Chapter 3 and Chapter 4 both assume that policies can be trained through simulation. The next part of the dissertation shifts to offline embodied control, where additional online exploration is

restricted and the central challenge becomes support-preserving long-horizon behavior composition from fixed datasets.

4.8 Summary and Transition

This chapter presented SA3C, an attention-enhanced actor–critic framework for long-horizon physical decision policies. The method augments SAC by linking actor features and critic features through an attention module, producing an attention-influenced action and an additional critic loss term. The goal is to improve feature relevance during policy learning in complex physical dynamics.

The evaluation included geocentric transfers to GEO and cislunar transfers to NRHO and patch-point targets. The results show improved automated transfer times relative to reported automated baselines, convergence in cislunar scenarios involving third-body effects, and improved performance over standard SAC settings in the attention-coefficient ablation. These findings support the dissertation argument that reinforcement learning for autonomous physical systems benefits from architecture-level structure matched to the decision problem.

Chapter 5 shifts from simulation-trained trajectory policies to offline robotic control. In that setting, long-horizon decision-making must be learned from fixed data, and the central challenge becomes support-preserving behavior composition rather than simulated exploration.

Chapter 5

Support-Preserving Latent Planning for Offline Robotic Control

Chapters 3 and 4 studied structured reinforcement learning in settings where simulation interaction is available during training. Chapter 3 used staged decomposition to separate coarse progress from terminal refinement. Chapter 4 used attention-enhanced actor-critic learning to improve feature relevance in complex physical dynamics. Both chapters addressed long-horizon decision policies by shaping the learning process around the structure of the physical task.

This chapter studies a stricter version of the same dissertation problem. The agent no longer receives unrestricted interaction with a simulator during policy learning. Instead, it must learn from a fixed dataset of prior behavior. The policy must compose long-horizon behavior from data that may contain only local fragments of successful execution, while avoiding actions and skill choices that leave the demonstrated support. The central challenge is therefore not only horizon length, but the combination of horizon length and restricted data support.

This fixed-data setting changes the role of structure. In interactive simulation, the agent can explore, fail, and improve over repeated rollouts. In offline control, unsupported exploration during learning is unavailable, and unsupported decisions at deployment can lead to unsafe or physically meaningless behavior. A useful long-horizon decision policy must therefore satisfy two requirements at the same time. It must be expressive enough to compose demonstrated behaviors

into extended task solutions, and conservative enough to avoid extrapolating beyond the behavior distribution that makes its value estimates and decoded actions reliable.

The domain used in this chapter is offline embodied control. The methods are evaluated on navigation, sequential manipulation, dexterous manipulation, locomotion, and contact-rich manipulation benchmarks. The domain changes from trajectory-level autonomy to embodied control, but the dissertation thesis remains the same: long-horizon decision policies for autonomous physical systems require structure matched to the decision problem. Here, that structure is support-preserving latent planning.

The methodological shift is substantial. In Chapters 3 and 4, the training process can repeatedly query a simulator and improve from new experience. In this chapter, the dataset is fixed before learning begins. The policy can no longer discover new transitions through exploration, and the critic cannot verify the value of arbitrary skill choices by executing them during training. This makes the learned representation itself part of the feasibility mechanism. A latent space that is poorly organized, weakly controlled, or poorly constrained will not merely reduce prediction accuracy. It can make long-horizon planning unreliable because the planner will optimize over decisions whose physical consequences are unsupported by data.

5.1 Problem Setting and Preliminaries

5.1.1 Offline Latent-Space Control Setting

Chapter 2 introduced offline reinforcement learning, temporal abstraction, and generative decision models. This chapter uses those concepts without repeating their general background. The setting is a fixed dataset

$$\mathcal{D} = \{(s_t, a_t, r_t, s_{t+1})\}, \quad (5.1)$$

collected by an unknown behavior policy. No additional environment interaction is used during training. The dataset is decomposed into fixed-length trajectory segments,

$$\tau_t = (s_t, a_t, \dots, s_{t+H}, a_{t+H}), \quad (5.2)$$

where H is the skill horizon. Each segment is treated as evidence for a temporally extended behavior unit. Instead of planning directly over individual actions, the method learns a latent variable z that represents a short action-state segment. A high-level policy then selects latent skills, and a low-level decoder converts each selected skill into an H -step action sequence.

This construction reduces the effective decision horizon. A task that requires many low-level actions can be represented by a smaller sequence of latent decisions. For example, a manipulation task that requires reaching, contact formation, transport, and placement can be treated as a sequence of skill-level decisions rather than a long stream of individual motor commands. The same logic applies to navigation, where local motion fragments can be composed into a path through a maze.

The abstraction is useful only if the latent space remains support-aligned and controllable. If the latent representation is unstructured, a planner may select skills that do not correspond to meaningful behavior. If the decoder ignores the latent variable, the planner cannot influence the action sequence. If the critic overestimates unsupported latents, the planner may prefer skills that look valuable numerically but are not grounded in the dataset. The rest of the chapter explains how the method enforces support preservation and controllability at the skill level.

5.2 Methodological Challenge and Design Principle

5.2.1 The Support–Controllability Problem

The conceptual center of this chapter is the support–controllability problem. Offline long-horizon control requires planning decisions that remain close to the data distribution while still producing meaningful changes in behavior. These two requirements can conflict.

Support preservation means that planned behavior should remain near demonstrated behavior. In a fixed dataset, the learner has evidence only for states, actions, and local transitions that appear in the data or are close to them. If a learned policy selects actions far outside this support, the critic may assign unreliable values, the decoder may produce physically implausible motions, and the robot may enter states where the dataset provides no useful guidance. In physical systems, this is not only a statistical issue. Unsupported actions can correspond to unstable contact, collision,

geometric constraint violation, or failure to complete a manipulation sequence.

Controllability means that high-level decisions must actually influence the decoded low-level actions. A latent skill planner is useful only if changing the latent decision changes the executed action sequence in a predictable and meaningful way. If the decoder ignores the latent variable and reconstructs actions mainly from the current state, the high-level planner loses control authority. This failure mode is closely related to posterior collapse in latent-variable models. The planner may appear to operate in a latent space, but the selected latent has little effect on the physical behavior.

The tradeoff is therefore precise. Too much conservatism keeps the policy near the dataset but may prevent useful long-horizon composition. Too much expressivity allows the planner to generate new behavior but increases the risk of unsupported or unsafe extrapolation. A strong offline control method must not choose one side of this tradeoff blindly. It must preserve support while maintaining enough controllability to compose behaviors across long horizons.

This tradeoff is especially important for long-horizon tasks because planning errors accumulate at the level of behavior fragments. If one unsupported latent is selected, the decoded action sequence can move the robot into a state where the next latent decision is also outside the data-supported regime. Conversely, if the planner is restricted to imitate only local dataset fragments without recombination, it may never solve tasks that require stitching together partial demonstrations. The support–controllability problem is therefore the fixed-data counterpart of the long-horizon feasibility gap: the policy must keep each intermediate decision physically grounded while still producing a sequence that achieves a distant objective.

This chapter develops GRALP, a Generative Representation framework for Action refinement and Latent Planning, as a structured response to this tradeoff. The method uses latent skills as high-level decisions, a deterministic diffusion decoder for short-horizon execution, conservative critics for support-aware value estimation, and a Transformer planner with latent-space refinement. The key separation is that diffusion is used for execution, not for long-horizon planning. Planning and value estimation remain in latent space, where support constraints can be imposed directly on the decisions the planner makes.

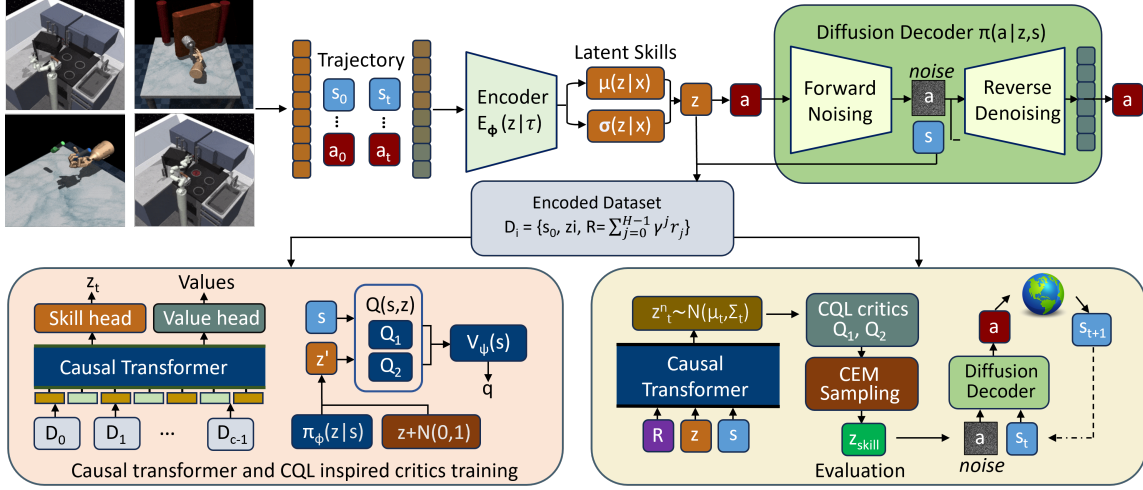


Figure 5.1: *GRALP pipeline. A Conditional Diffusion Autoencoder extracts latent skills from short action-state segments, conservative critics estimate value in latent space, and a causal Transformer predicts the next skill conditioned on recent context and return information.*

5.2.2 GRALP Overview

GRALP decomposes offline control into two levels. The low-level component learns a latent skill representation and a deterministic action decoder. The high-level component learns to plan over latent skills using conservative value estimates. Figure 5.1 shows the overall pipeline.

The pipeline has five main components. First, a Conditional Diffusion Autoencoder encodes short trajectory segments into latent skills. Second, a deterministic diffusion decoder maps a selected latent skill and current state into a short action sequence. Third, conservative critics estimate value directly in latent space, reducing overestimation for unsupported skill choices. Fourth, a causal Transformer predicts latent skills from recent states, previous skills, and optional return-to-go information. Fifth, the Cross-Entropy Method (CEM) refines latent candidates using conservative values before the selected latent is decoded into actions.

These components correspond to distinct parts of the support–controllability problem. The encoder and prior define the skill distribution induced by the dataset. The decoder provides local action execution. The controllability regularizers keep the decoder responsive to latent decisions. The conservative critics define which latents are valuable without encouraging unsupported extrapolation. The Transformer and CEM planner provide long-horizon composition at the skill level. The architecture is therefore not a collection of disconnected modules; each component plays a role

in keeping planning both expressive and grounded.

The architectural separation is the core design choice. Diffusion is not used as a full trajectory planner at inference. It is used only to decode a selected latent skill into local action execution. Long-horizon decision-making, value estimation, and candidate refinement occur in latent space. This separation matters because value learning over full action trajectories can become unstable and expensive, especially when diffusion sampling is stochastic and repeated over long horizons. By keeping planning in latent space and execution in short-horizon action space, GRALP aims to retain expressive behavior generation without turning diffusion into the planning mechanism itself.

5.2.3 Support-Preservation and Controllability Principle

The support–controllability problem can be expressed through a local relationship between latent deviation and decoded action deviation. Let

$$f_s : \mathcal{Z} \rightarrow \mathcal{A}^H \quad (5.3)$$

denote the decoder that maps a latent skill z to an H -step action sequence at state s . Let $\mathcal{Z}_{\mathcal{D}}(s)$ denote the dataset-supported latents near state s , encoded from trajectory segments whose start states lie in a local neighborhood.

Assumption 1 (Local Decoder Smoothness). For each state s , the decoder f_s is L_s -Lipschitz continuous in a neighborhood of dataset-supported latents.

Proposition 1 (Support–Controllability Bound). For any planned latent z and state s ,

$$\|f_s(z) - f_s(z_{\mathcal{D}}^*)\|_2 \leq L_s \cdot \delta_s(z, s), \quad (5.4)$$

where

$$\delta_s(z, s) = \min_{z_{\mathcal{D}} \in \mathcal{Z}_{\mathcal{D}}(s)} \|z - z_{\mathcal{D}}\|_2, \quad (5.5)$$

and $z_{\mathcal{D}}^*$ is the nearest dataset-supported latent.

Proof sketch. By Assumption 1,

$$\|f_s(z) - f_s(z')\|_2 \leq L_s \|z - z'\|_2 \quad (5.6)$$

for latents in the local neighborhood. Choosing $z' = z_{\mathcal{D}}^*$ gives

$$\|f_s(z) - f_s(z_{\mathcal{D}}^*)\|_2 \leq L_s \|z - z_{\mathcal{D}}^*\|_2 = L_s \delta_s(z, s). \quad (5.7)$$

This establishes Eq. (5.4).

The proposition should be interpreted as a local design principle, not as a global safety guarantee. It states that decoded action deviation can be controlled locally by keeping planned latents near dataset-supported latents and by maintaining a decoder whose latent sensitivity is meaningful. Small $\delta_s(z, s)$ reduces the risk of unsupported action generation. A non-degenerate L_s ensures that changing the latent variable produces a corresponding change in decoded behavior. GRALP operationalizes these two requirements through conservative latent value learning and latent-controllability regularization.

The proposition also clarifies why support preservation and controllability cannot be optimized independently. If $\delta_s(z, s)$ is unconstrained, the planner may choose latents whose decoded actions are far from demonstrated behavior. If L_s collapses toward zero, the planner may remain near dataset latents but lose the ability to steer the decoder. The useful regime is the middle ground: planned latents stay close enough to dataset support to be reliable, while the decoder remains sensitive enough for high-level planning decisions to matter.

5.3 Technical Formulation

5.3.1 Diffusion Decoder for Latent Skill Execution

Trajectory segments are encoded into latent skills using a Conditional Diffusion Autoencoder. The encoder $q_\phi(z \mid \tau)$ produces a Gaussian posterior over latent skills, while a state-conditional

prior $p_\omega(z \mid s)$ regularizes the latent space and provides proposals for planning. The decoder p_θ reconstructs action sequences conditioned on the state and selected latent skill.

The diffusion process is used only for short-horizon action reconstruction. For a trajectory-segment representation x_0 , the forward diffusion step can be written as

$$x_k = \sqrt{\bar{\alpha}_k} x_0 + \sqrt{1 - \bar{\alpha}_k} \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I), \quad (5.8)$$

where $\bar{\alpha}_k$ controls the signal-to-noise ratio at diffusion step k . The action-level forward noising process is

$$q(a_k \mid a_{k-1}) = \mathcal{N}\left(\sqrt{1 - \beta_k} a_{k-1}, \beta_k I\right), \quad (5.9)$$

with learned reverse process

$$p_\theta(a_{k-1} \mid a_k, s, z) = \mathcal{N}(\mu_\theta(a_k, k, s, z), \Sigma_\theta(k)). \quad (5.10)$$

At inference, deterministic DDIM sampling is used to produce consistent latent-to-action mappings,

$$x_{k-1} = g_\theta(x_k, k). \quad (5.11)$$

This deterministic execution is important for value-guided planning. If the same latent skill decoded into highly variable action sequences, a critic trained in latent space would evaluate an unstable decision. Deterministic decoding makes the latent decision more reproducible and therefore more compatible with conservative value estimation.

Latent and state information are injected into the diffusion decoder through Feature-wise Linear Modulation (FiLM),

$$h' = \gamma_\theta(k, s, z) \odot h + \beta_\theta(k, s, z). \quad (5.12)$$

The diffusion model is trained with the noise-prediction objective

$$\mathcal{L}_{\text{diff}} = \mathbb{E} \left[\|\epsilon - \epsilon_\theta(a_k, k, s, z)\|_2^2 \right]. \quad (5.13)$$

The result is an execution module that can generate expressive local behavior while remaining trained on empirical trajectory segments. The important limitation is also clear: the decoder alone does not solve long-horizon planning. It only executes the short skill selected by the latent planner.

This is the main difference between GRALP and diffusion-as-planner approaches. A trajectory-level diffusion planner samples or denoises long action sequences at inference time, which can be computationally expensive and difficult to couple with value estimation. GRALP instead treats diffusion as a local executor. Once a latent skill is chosen, the decoder deterministically produces the corresponding action fragment. This keeps the stochastic generative process out of the high-level planning loop and makes the value function responsible for evaluating compact latent decisions rather than full action trajectories.

5.3.2 Latent Controllability Regularization

The diffusion decoder is expressive enough to reconstruct action sequences, but expressivity alone is not sufficient for planning. If the decoder can reconstruct actions from the state while ignoring the latent skill, then the high-level planner has little influence over the executed behavior. GRALP mitigates this collapse through state dropout and a latent-action sensitivity regularizer.

State dropout is applied inside the decoder’s FiLM conditioning. Randomly masking state inputs with probability p_{drop} forces the decoder to rely more strongly on the latent skill. This makes it harder for the model to bypass z and reconstruct actions only from state context.

The second mechanism is a geometric regularizer called Z-force. Let $\pi_{\text{low}}(a \mid s, z)$ denote the action sequence decoded by the diffusion model. The regularizer encourages local sensitivity to latent perturbations by penalizing small Jacobian norms:

$$\mathcal{L}_{\text{force}} = -\log (\|\nabla_z \pi_{\text{low}}(a \mid s, z)\|_2 + \varepsilon) . \quad (5.14)$$

The log form strongly penalizes near-zero sensitivity while saturating for large sensitivity values. This reduces the risk that the decoder ignores the latent variable while avoiding unbounded pressure toward overly sensitive mappings.

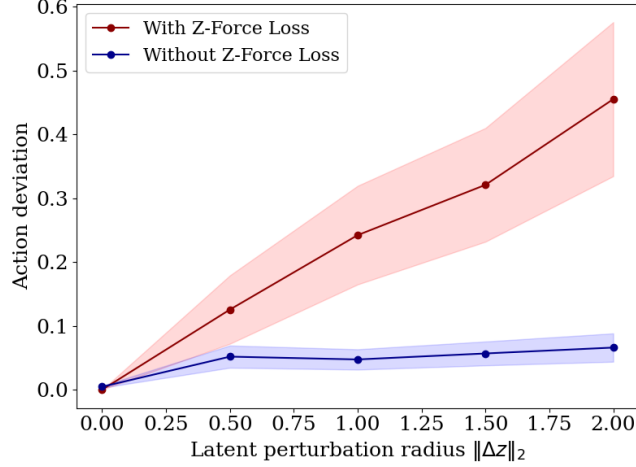


Figure 5.2: *Latent-action sensitivity with and without Z-force. With Z-force, decoded action changes grow with latent perturbations, while without Z-force the decoder shows weak sensitivity, indicating near-collapse of latent control authority.*

The full regularization objective is

$$\mathcal{L}_{\text{reg}} = \beta_{\text{KL}} D_{\text{KL}}(q_{\phi}(z | \tau) \| p_{\omega}(z | s)) + \lambda \mathcal{L}_{\text{force}}. \quad (5.15)$$

The KL term regularizes the latent posterior toward the state-conditioned prior, while the Z-force term preserves decoder sensitivity to latent decisions.

Figure 5.2 illustrates the effect of this regularization. With Z-force, changes in latent skill produce a substantially larger action response than without Z-force. The reported sensitivity reaches approximately 0.47 at $\|\Delta z\|_2 = 2.0$ with Z-force, compared with approximately 0.07 without it. This diagnostic directly supports the controllability side of the support–controllability principle.

The practical implication is that the planner can search in latent space with some confidence that the search variable has behavioral meaning. Without this property, CEM refinement and value-based latent selection could optimize an inactive or weakly coupled variable. Z-force does not prove global controllability, but it empirically mitigates collapse and improves the local sensitivity needed for skill-level planning.

5.3.3 Conservative Planning in Latent Space

Once latent skills are learned, planning becomes the problem of selecting latent codes whose decoded action fragments remain feasible and useful. This planning problem is still offline. The planner cannot freely evaluate arbitrary latent skills through new interaction, and the critic can overestimate unsupported latents. GRALP therefore performs value learning and candidate refinement directly in latent space under conservative constraints.

Each latent skill executes an H -step decoded fragment, yielding cumulative return

$$R_{t:t+H-1} = \sum_{h=0}^{H-1} \gamma^h r_{t+h}. \quad (5.16)$$

The horizon-aligned Bellman target is

$$y = R_{t:t+H-1} + \gamma^H V(s_{t+H}). \quad (5.17)$$

The factor γ^H is important because a latent skill represents H low-level action steps. Using the low-level discount at the high-level decision boundary would misalign the critic with the temporal abstraction.

The latent action-value function is written as

$$Q(s, z) = R_z + \gamma^H V(s'), \quad (5.18)$$

where s' is the successor state after executing the decoded skill. Two Q-networks and a value network are trained for stability.

To suppress overestimation for unsupported skills, conservative value learning is applied in latent space. The critic objective is

$$\begin{aligned} \mathcal{L}_{\text{CQL}}(Q_i) = & \alpha \left(\mathbb{E}_{z \sim \mu(z)} [Q_i(s, z)] - \mathbb{E}_{z \sim \mathcal{D}} [Q_i(s, z)] \right) \\ & + \frac{1}{2} \mathbb{E} \left[(Q_i(s, z) - \hat{Q}_{\text{target}})^2 \right]. \end{aligned} \quad (5.19)$$

The proposal distribution $\mu(z)$ targets regions near dataset support through Gaussian perturbations around encoder latents, local variations of dataset skills, and samples from a proposal head trained to match the planner’s predictions. The purpose is not to make all out-of-dataset behavior impossible, but to penalize overly optimistic values for unsupported latent choices. This distinction matters because long-horizon composition may require selecting skill sequences that were not observed end-to-end. The conservative critic should discourage unsupported latent actions while preserving useful ranking among feasible skills. In other words, conservatism should constrain extrapolation without collapsing the planner into pure behavior cloning.

High-level planning is performed by a causal Transformer that conditions on recent states, previous skills, and optional return-to-go information. The planner is trained with advantage-weighted regression. The advantage is

$$A(s, z) = \min(Q_1, Q_2) - V(s), \quad (5.20)$$

and the corresponding weight is

$$w(s, z) = \exp\left(\frac{A(s, z)}{\tau}\right). \quad (5.21)$$

This emphasizes high-value skills while keeping learning anchored to dataset-supported latent decisions.

At inference, CEM refines the Transformer’s latent proposal directly in skill space. Starting from the Transformer’s Gaussian proposal, CEM samples latent candidates, scores them using the conservative critic, fits a new Gaussian to elite samples, and repeats this refinement for three iterations. The selected latent is then decoded through deterministic DDIM into an action sequence.

Figure 5.3 provides a simple diagnostic of conservative value-guided planning. The critic assigns high values to in-support regions and suppresses off-support regions, while CEM contracts candidate latents toward high-value supported areas. This figure is not intended as a benchmark result; it visualizes the mechanism that the conservative critic is supposed to enforce in larger control tasks.

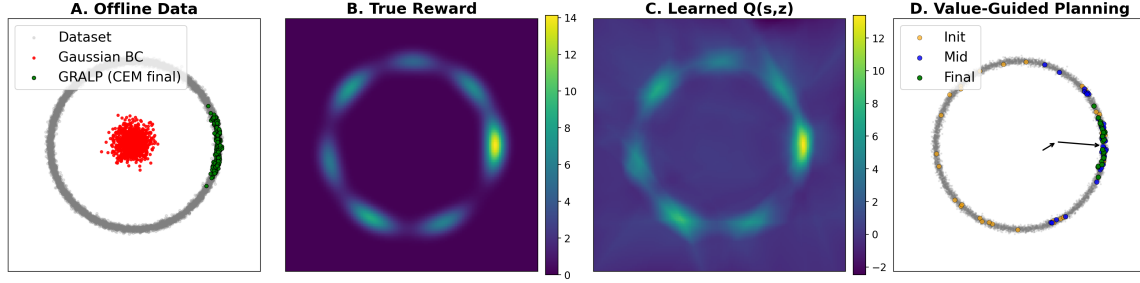


Figure 5.3: *Conservative value-guided planning on a multi-modal 2D ring bandit. The diagnostic shows dataset support, reward modes, conservative critic behavior, and CEM contraction toward a high-value supported region.*

5.4 Learning Objective and Algorithm

5.4.1 GRALP Training and Planning Algorithm

Algorithm 3 summarizes the training and inference pipeline. The procedure has three training stages: skill-model training, encoded dataset construction, and conservative latent planner training. During inference, the planner predicts and refines a latent skill, and the deterministic decoder executes the corresponding action fragment.

5.5 Experimental Setup

The evaluation spans five benchmark groups. D4RL locomotion tasks test smooth continuous control, where long-horizon skill composition is less central but stability remains important. Maze2D and AntMaze navigation tasks test the ability to compose local motion fragments into goal-directed trajectories under sparse rewards. Kitchen tasks test sequential manipulation, where the agent must order multiple subtasks correctly. Adroit tasks test high-dimensional dexterous manipulation with contact-rich dynamics. RoboSuite tasks test contact-rich manipulation with human demonstrations using a 7-DoF Panda manipulator.

The baseline categories include behavior cloning and model-free offline RL, conservative offline RL, sequence modeling, model-based and planning methods, diffusion planners, and diffusion policies. Representative baselines include BC, BCQ², CQL³, IQL⁴, Decision Transformer⁹,

Algorithm 3 GRALP training and latent-space planning pipeline

Require: Offline dataset $\mathcal{D}$, skill horizon H , latent dimension d_z

Ensure: Trained skill model, conservative latent critics, planner, and deterministic decoder

- 1: **Stage 1: Train diffusion skill model**
 - 2: Initialize encoder $q_\phi(z | \tau)$, prior $p_\omega(z | s)$, and decoder $p_\theta(a | s, z)$
 - 3: **for** $t = 1$ to T_1 **do**
 - 4: Sample trajectory segment $\tau \sim \mathcal{D}$
 - 5: Sample latent skill $z \sim q_\phi(z | \tau)$
 - 6: Sample diffusion timestep and apply state dropout
 - 7: Update (ϕ, ω, θ) using $\mathcal{L}_{\text{diff}}$ and $\mathcal{L}_{\text{reg}}$
 - 8: **end for**
 - 9: **Stage 2: Construct encoded dataset**
 - 10: Freeze q_ϕ and encode trajectory segments into latent transitions D_z
 - 11: Normalize latent skills if required by the planner implementation
 - 12: **Stage 3: Train conservative latent critics and planner**
 - 13: Initialize critics Q_1, Q_2, V and Transformer planner π_ψ
 - 14: **for** $t = 1$ to T_2 **do**
 - 15: Sample latent transition batch (s, z, R_z, s') from D_z
 - 16: Update V using expectile regression
 - 17: Update Q_1, Q_2 using horizon target in Eq. (5.17) and conservative loss in Eq. (5.19)
 - 18: Update π_ψ using advantage-weighted regression with Eqs. (5.20)–(5.21)
 - 19: Polyak update target networks
 - 20: **end for**
 - 21: **Inference**
 - 22: Use π_ψ to propose the next latent skill from recent states, previous skills, and optional return-to-go
 - 23: Refine latent candidates with CEM using conservative critic scores
 - 24: Decode the selected latent skill with deterministic DDIM and execute the resulting action sequence
-

MOReL⁶³, Trajectory Transformer¹⁰, Diffuser⁵, Decision Diffuser⁶, QGPO⁴⁷, IDQL⁴⁶, and DQL. The D4RL benchmark is cited as⁵⁰; RoboSuite is cited as⁵¹.

The comparison is intentionally organized by benchmark family rather than by a single aggregate number. The dissertation question is not whether one method dominates every task. It is whether support-preserving latent planning helps specifically where long-horizon composition, sparse reward, and dataset support are central. Locomotion, navigation, Kitchen, Adroit, and RoboSuite stress different aspects of the same fixed-data decision problem.

The implementation uses a two-stage training protocol: diffusion-based action decoder learning, followed by latent critic and causal Transformer planner optimization. The encoder uses a two-

layer GRU with hidden size 256, critics are three-layer MLPs with width 256, and the planner is an eight-layer Transformer with eight attention heads. Skill horizon is selected from $H \in \{8, 12\}$ depending on the domain. Results are averaged over ten random seeds with twenty-five evaluation rollouts per seed.

5.6 Empirical Results and Analysis

5.6.1 Main D4RL Performance

Table 5.1 reports the full D4RL benchmark comparison. The table provides the empirical basis for evaluating the support–controllability claim across qualitatively different offline-control regimes. The result pattern is more informative than a single aggregate number: GRALP is strongest on benchmark groups where long-horizon composition and support preservation are central, while remaining competitive on locomotion tasks where smooth action-level control is often sufficient.

The pattern in Table 5.1 is aligned with the support–controllability thesis. GRALP achieves the highest average performance on Navigation, Sequential Kitchen, Adroit cloned/expert, and Adroit human-demonstration groups. It remains competitive but not uniformly best on Locomotion, where smooth reactive control is often sufficient and temporal abstraction provides less advantage. This distinction is important because it keeps the claim appropriately scoped: GRALP is strongest where long-horizon composition, sparse rewards, and dataset support are central.

Navigation tasks provide the clearest example. Maze2D and AntMaze require the agent to connect local movement fragments into a goal-directed path. A method that imitates too closely may fail to produce a complete route, while unsupported extrapolation can lead to invalid transitions. Kitchen tasks create a different form of composition, where the agent must execute subtasks in a useful order. Adroit tasks test whether temporal abstraction can coexist with fine-grained contact control. The full benchmark table therefore supports the chapter’s main claim more directly than a compact average alone: support-preserving latent planning is most valuable when the policy must compose reliable behavior from incomplete fixed-data coverage.

Table 5.1: Unified D4RL benchmark results. The table compares non-diffusion baselines, diffusion-based planners and policies, and GRALP across locomotion, navigation, sequential manipulation, and Adroit dexterous-control tasks. Best scores are bolded and second-best scores are underlined following the GRALP evaluation protocol. M/R/E denote Medium, Replay, and Expert; U/L/D denote Umaze, Large, and Diverse; P/Mxd/Cmp denote Partial, Mixed, and Complete; H/C/E denote Human, Cloned, and Expert.

	Task Name	Non-Diffusion							Diffusion-based										(Mean $\pm$ Std)
		Model-Free				Model-Based			Diffusion as Planner				Diffusion as Policy				OURS		
		BC	CQL	IQL	DT	MOREL	TT	TAP	LDCQ	Diffuser	DD	HDMI	QGPO	SfBC	IDQL	DQL	DQL@1		
Locomotion	Hopper-M	52.9	58.5	66.3	67.6	95.4	61.1	63.4	66.2	58.5	79.3	76.4	98.0	57.1	65.4	82.4	64.1	96.26 $\pm$ 0.4	
	Hopper-M-R	18.1	95.0	94.7	82.7	93.6	91.5	87.3	86.2	96.8	100.0	99.6	96.9	86.2	92.1	100.7	63.1	88.41 $\pm$ 1.8	
	Hopper-M-E	52.5	105.4	91.5	107.6	108.7	110.0	105.5	111.3	107.2	111.8	113.5	108.0	108.6	108.6	107.2	100.6	111.90 $\pm$ 2.3	
	HCheetah-M	42.6	44.0	47.4	42.6	42.1	46.9	45.0	42.8	44.2	49.1	48.0	54.1	45.9	51.0	50.6	47.8	51.23 $\pm$ 0.3	
	HCheetah-M-R	36.6	45.5	44.2	36.6	40.2	41.9	40.8	41.8	42.2	39.3	44.9	47.6	37.1	45.9	47.5	44.0	41.92 $\pm$ 0.42	
	HCheetah-M-E	55.2	91.6	86.7	86.8	53.3	95.0	91.8	90.2	79.8	90.6	92.1	93.5	91.6	<u>95.9</u>	96.1	94.8	90.13 $\pm$ 0.9	
	Walker2d-M	75.3	72.5	78.3	74.0	77.8	79.0	64.9	69.4	79.7	82.5	79.9	86.0	53.1	82.5	<u>85.1</u>	82.0	78.42 $\pm$ 0.8	
	Walker2d-M-R	26.0	77.2	73.9	66.6	49.8	82.6	66.8	68.5	61.2	75.0	80.7	84.4	65.1	<u>85.1</u>	94.3	75.6	83.94 $\pm$ 1.8	
	Walker2d-M-E	107.5	108.8	109.6	108.1	95.6	101.9	107.4	109.3	108.4	108.8	107.9	<u>110.7</u>	109.6	108.6	109.7	108.9	111.61 $\pm$ 0.7	
Average		51.87	77.19	76.96	74.70	72.94	78.80	74.77	76.18	75.30	81.60	82.60	86.58	72.70	81.68	<u>86.30</u>	75.60	83.76	
Navigation	Maze2d-U	3.8	5.7	47.4	27.3	0	-	-	<u>134.2</u>	113.9	116.2	120.1	-	73.9	57.9	-	-	143.14 $\pm$ 4.0	
	Maze2d-M	30.0	5.0	34.9	32.1	1.3	-	-	<u>125.3</u>	121.5	122.3	121.8	-	73.8	89.5	-	-	145.08 $\pm$ 4.9	
	Maze2d-L	5.0	12.5	58.6	18.1	0	-	-	150.1	123.0	125.9	128.6	-	74.4	90.0	-	-	<u>149.92 $\pm$ 2.9</u>	
	AntMaze-U-D	45.6	80.3	62.2	53.0	0	-	69.8	81.4	72.2	49.2	73.7	74.4	85.3	80.2	66.2	56.4	90.75 $\pm$ 3.2	
	AntMaze-M-D	0.0	53.7	70.0	0.0	-	0.6	85	68.9	45.5	24.6	-	83.6	82.0	<u>84.8</u>	78.6	14.8	87.75 $\pm$ 4.2	
	AntMaze-L-D	0.0	14.9	47.5	0.0	-	0	-	<u>57.7</u>	22.0	7.5	71.5	64.8	45.5	67.9	56.6	1.6	52.35 $\pm$ 2.5	
Average		14.06	28.68	53.43	22.78	0.32	-	-	<u>102.94</u>	83.01	74.28	-	-	72.48	78.38	-	-	111.83	
Sequential	Kitchen-P	38.0	49.8	46.3	0.0	-	-	-	<u>67.8</u>	55.7	56.5	-	66.0	47.9	66.7	60.5	-	67.96 $\pm$ 3.14	
	Kitchen-Mxd	51.5	51.0	51.0	2.5	-	-	-	<u>62.3</u>	52.5	51.8	-	45.5	45.4	66.5	62.6	-	66.70 $\pm$ 2.8	
	Kitchen-Cmp	65.0	43.8	62.5	15.0	-	-	-	62.5	-	-	-	62.8	77.9	62.5	84.0	-	<u>81.06 $\pm$ 4.3</u>	
	Average	51.5	48.2	53.26	5.83	-	-	-	64.4	-	-	-	58.1	57.06	65.23	<u>69.03</u>	-	71.63	
Adroit	Pen-C	37.0	39.2	37.3	23.5	-	11.4	57.4	47.7	61.7	47.7	48.3	54.2	-	82.3	28.3	49.3	<u>81.96 $\pm$ 2.5</u>	
	Pen-E	85.1	107.0	-	110.0	-	72.0	127.4	121.2	99.7	107.6	109.5	119.1	-	<u>137.8</u>	133.5	112.6	146.20 $\pm$ 3.9	
	Door-C	0.0	0.4	1.6	-0.1	-	-0.1	11.7	1.1	0.1	9.0	9.3	<u>11.2</u>	-	4.4	-0.3	10.6	2.23 $\pm$ 1.2	
	Door-E	34.9	101.5	-	95.5	-	94.1	104.8	<u>105.1</u>	103	94.0	85.9	98.8	-	105.0	104.8	93.7	108.03 $\pm$ 0.6	
	Relocate-C	-0.3	-0.1	-0.1	-0.3	-	-0.1	-0.2	-0.2	0.0	<u>0.15</u>	-0.1	-0.2	-	0.0	0.1	-0.2	0.27 $\pm$ 0.06	
	Relocate-E	101.3	95.0	-	15.0	-	10.3	105.8	104.0	102.2	87.5	91.3	102.5	-	107.0	<u>108.5</u>	95.2	109.93 $\pm$ 2.1	
	Hammer-C	0.8	2.1	1.6	0.2	-	0.5	1.2	2.8	1.2	0.9	1.0	1.1	-	<u>3.5</u>	0.2	1.1	4.26 $\pm$ 1.3	
	Hammer-E	125.6	86.7	-	89.7	-	15.5	127.6	45.8	103.1	106.7	111.8	123.2	-	127.6	<u>128.3</u>	114.8	132.35 $\pm$ 1.16	
	Average	48.05	53.96	55.66	41.69	-	25.45	66.96	53.44	58.86	56.70	57.13	63.74	-	<u>70.95</u>	62.925	59.6375	73.15	
Adroit	Pen-H	34.4	37.5	71.3	6.3	-	36.4	<u>76.5</u>	74.1	-	64.1	66.2	73.9	-	-	72.8	66.0	80.92 $\pm$ 3.5	
	Door-H	0.5	<u>9.9</u>	4.3	3.9	-	0.1	8.8	11.8	-	6.9	7.1	8.5	-	-	-	8.0	7.23 $\pm$ 2.3	
	Relocate-H	0.0	0.2	0.1	0.2	-	0.0	0.2	0.3	-	0.2	0.1	0.2	-	-	-	0.2	0.24 $\pm$ 0.9	
	Hammer-H	1.2	4.4	1.4	0.5	-	0.8	1.4	1.5	-	1.0	1.2	1.4	-	-	-	1.3	<u>3.55 $\pm$ 1.1</u>	
	Average	9.03	13.00	19.28	2.72	-	9.33	21.73	<u>21.93</u>	-	18.05	18.65	21	-	-	-	18.86	22.98	

5.6.2 Contact-Rich RoboSuite Manipulation

Beyond D4RL, the evaluation includes RoboSuite Lift and Pick-and-Place using offline human demonstrations. Lift tests single-stage grasp acquisition under contact constraints. Pick-and-Place adds sequencing: the robot must grasp, transport, and place an object. These tasks are useful because they involve a different embodiment and action space from the D4RL domains.

Table 5.2: *RoboSuite success rates using offline human demonstrations. The results test whether support-preserving latent planning transfers to contact-rich manipulation with a 7-DoF Panda manipulator.*

Task	Success rate	Control challenge
Lift	$96.6\% \pm 2.1\%$	Single-stage grasp acquisition with precise contact constraints.
Pick-and-Place	$94.3\% \pm 3.4\%$	Multi-stage grasping, transport, and placement.



Figure 5.4: *RoboSuite Pick-and-Place evaluation. The task requires grasping, transport, and placement under contact-rich manipulation dynamics.*

Across ten seeds with twenty-five rollouts per seed, GRALP achieves a success rate of $96.6\% \pm 2.1\%$ on Lift and $94.3\% \pm 3.4\%$ on Pick-and-Place. These results indicate that the latent skill abstraction can support contact-rich manipulation beyond the D4RL task families.

The RoboSuite results are particularly relevant for the dissertation because they test the method under contact-rich embodied control rather than only navigation or locomotion. The high success rates support the claim that support-preserving latent planning can remain effective when physical contact and sequential manipulation are required.

5.6.3 Component Ablations

Table 5.3 reports ablations on representative tasks spanning sparse-reward navigation, compositional manipulation, dexterous control, and Maze2D navigation. Each ablation removes one part of the support–controllability design.

The ablations support three conclusions. First, removing Z-force or state dropout weakens

Table 5.3: *Component ablations on representative tasks. Removing conservative value learning, latent-controllability regularization, critic-guided refinement, or temporal abstraction degrades performance, showing that the method depends on the joint support–controllability design rather than a single isolated component.*

Ablation	AntMaze-M-D	Kitchen-Mxd	Pen-C	Maze2d-L
GRALP (Full)	87.75	66.14	81.96	149.90
W/O Z-force	63.82	40.75	68.38	115.46
W/O State Dropout	65.19	32.46	71.22	133.36
W/O Both	40.48	29.48	51.02	95.85
W/O CQL	37.67	35.65	44.72	73.64
Critic Ignore	61.19	43.28	48.33	135.49
$H = 1$	34.28	30.03	42.71	118.02

latent controllability, particularly on tasks where decoded behavior must respond meaningfully to planned skills. Second, removing CQL produces severe degradation, especially on AntMaze and Maze2D, showing that conservative support preservation is central in sparse-reward navigation. Third, setting $H = 1$ collapses the method toward action-level planning and consistently reduces performance, indicating that temporal abstraction is necessary for the long-horizon tasks targeted by this chapter.

The relative magnitude of the drops is also informative. The W/O CQL ablation is especially damaging in navigation, where unsupported skill selection can quickly move the agent into poor regions of the maze. The W/O Z-force and W/O State Dropout variants show that representation quality matters even when value learning is conservative. The Critic Ignore ablation shows that the Transformer proposal alone is not enough; latent-space refinement guided by conservative values is an important part of the planning pipeline.

5.6.4 Mechanism Diagnostics

The diagnostic figures provide qualitative evidence for the mechanisms behind the benchmark results. Figure 5.5 shows PCA projections of Kitchen-Mixed latents for different skill horizons. At $H = 1$, the latent representation is diffuse and overlapping. At $H = 4$, partial structure appears. At $H = 12$, the representation forms more separated clusters, indicating stronger temporal abstraction.

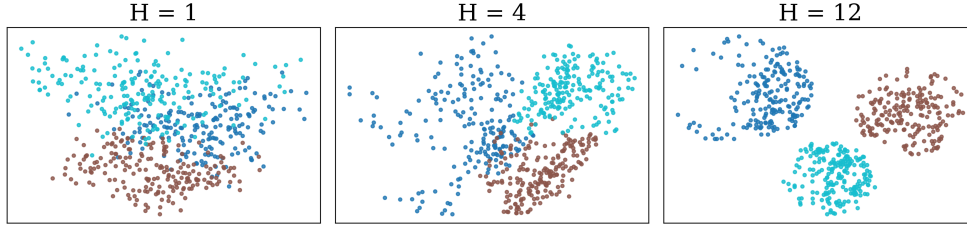


Figure 5.5: PCA projections of Kitchen-Mixed latents for horizons $H \in \{1, 4, 12\}$. Increasing the skill horizon produces more structured latent organization, supporting stronger temporal abstraction.

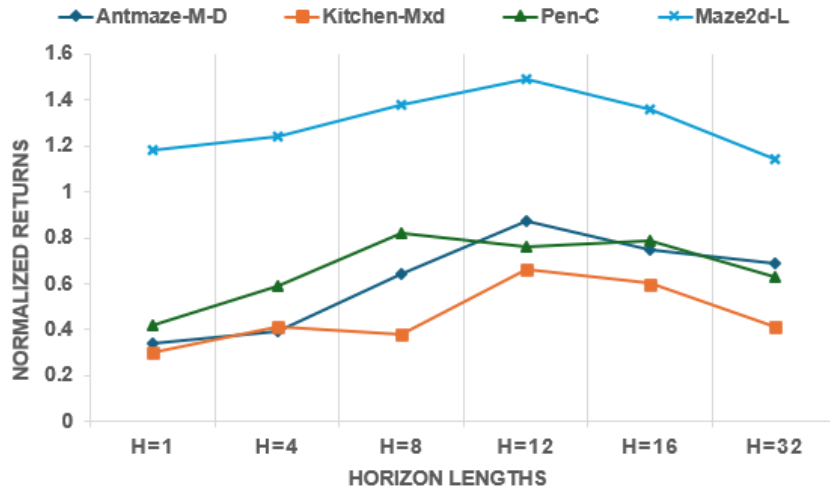


Figure 5.6: Performance versus skill horizon H across representative tasks. Intermediate horizons provide the best balance between temporal abstraction and stable value estimation.

This supports the use of moderate skill horizons for compositional planning.

Figure 5.6 shows that performance peaks at intermediate horizons across representative tasks. Short horizons provide insufficient abstraction, while overly long horizons increase critic variance and reduce stable planning. This illustrates an abstraction-stability tradeoff: useful skill horizons must be long enough to reduce the planning horizon but short enough for value learning to remain reliable.

Figure 5.7 visualizes long-horizon composition in AntMaze-Medium-Diverse. The gray traces show fragmented dataset coverage, while the colored trajectories show GRALP rollouts. The successful trajectory demonstrates that the planner can compose locally demonstrated fragments into a maze-spanning behavior that was not necessarily present as a single demonstration.

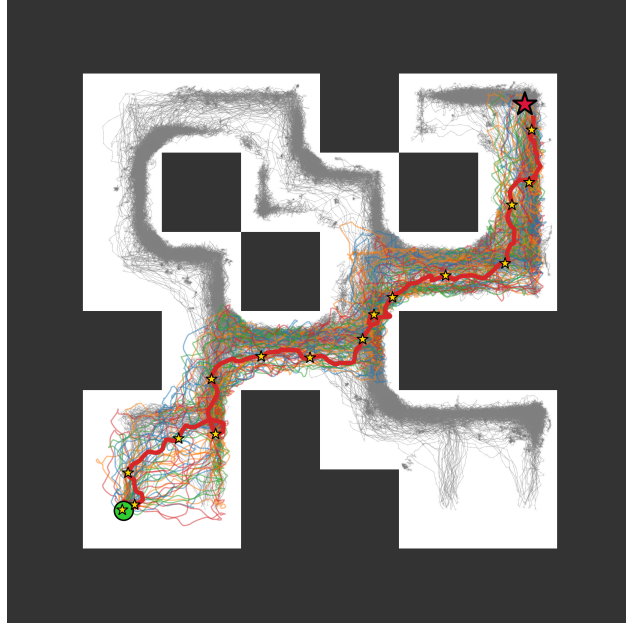


Figure 5.7: *Trajectory stitching on AntMaze-Medium-Diverse. Gray traces show fragmented dataset coverage, colored trajectories are GRALP rollouts, and yellow stars mark skill transitions every $H = 12$ steps.*

Together, the diagnostics connect the quantitative gains to the chapter’s central mechanism. Conservative critics reduce unsupported planning, Z-force preserves latent controllability, and intermediate-horizon latent skills enable long-horizon behavior composition.

5.7 Discussion

This chapter adds a new layer to the dissertation’s argument. Chapter 3 showed how staged decomposition can manage horizon length and terminal precision when simulation interaction is available. Chapter 4 showed how attention can improve feature relevance inside actor–critic learning. This chapter addresses the fixed-data case, where long-horizon decision learning must occur without online exploration and without leaving the support of demonstrated behavior.

The evidence supports three main claims. First, support preservation and controllability must be treated jointly. Conservative critics alone can reduce unsupported decisions, but without a controllable latent decoder the planner loses authority over execution. Conversely, an expressive decoder without support constraints can produce unsupported behavior. Second, separating dif-

fusion execution from latent planning provides a stable structure for offline long-horizon control. Diffusion handles local action generation, while the planner and critics operate over latent skills. Third, the method is strongest in tasks where long-horizon composition is essential, including navigation, sequential manipulation, and dexterous control.

The scope boundaries are also important. The evaluation is based on benchmark and simulation environments, not physical robot deployment. Performance depends on dataset quality, since a fixed dataset must contain enough local behavior fragments for the latent planner to compose. The skill horizon and latent dimension require tuning. Conservative critic strength and CEM refinement parameters can influence performance. The support–controllability proposition is local and does not provide a formal global safety guarantee. Sim-to-real validation remains necessary before making claims about physical deployment.

These limitations define natural future directions rather than weakening the central contribution. A stronger version of the framework would incorporate uncertainty-aware critics, adaptive skill horizons, formal support estimation, and real-robot validation. It could also integrate richer state representations for vision-based control. Another important direction is to reduce sensitivity to hand-selected hyperparameters, especially skill horizon, CQL strength, latent dimension, and CEM refinement settings. These choices influence the balance between support preservation and skill expressivity.

Within the dissertation, the immediate next question is different: GRALP assumes that the planner receives sufficient temporal context to choose the next latent skill. This assumption is reasonable for the baseline offline-control setting, but it may be fragile under deployment conditions where histories are truncated, sensors are noisy, or previously executed skills are not fully observed. Chapter 6 studies what happens when that context is incomplete, corrupted, or degraded.

5.8 Summary and Transition

This chapter developed support-preserving latent planning for offline robotic control. The method addresses the fixed-data version of the long-horizon decision-policy problem, where the agent must compose extended behavior without collecting new interaction data and without selecting

unsupported actions. GRALP addresses this through a Conditional Diffusion Autoencoder for latent skill execution, controllability regularization to prevent latent collapse, conservative critics for support-aware value learning, a causal Transformer planner, and CEM refinement in latent space.

The empirical results show that GRALP performs strongest on benchmark groups where long-horizon composition is central, including navigation, sequential manipulation, and Adroit control, while remaining competitive on locomotion. RoboSuite results show high success rates on contact-rich manipulation from human demonstrations. Ablations and diagnostics show that the gains depend on the combined use of support preservation, latent controllability, and temporal abstraction.

Chapter 6 builds on this latent skill planning framework by asking whether a planner can remain reliable when temporal context is missing, corrupted, or incomplete. This shifts the dissertation from support-preserving offline planning to context-robust long-horizon decision-making.

Chapter 6

Context-Robust Skill Inference for Long-Horizon Offline Control

Chapter 5 developed support-preserving latent planning for offline robotic control. Its central argument was that a long-horizon policy trained from fixed data must remain close to demonstrated behavior while retaining enough latent controllability to compose useful behavior. This chapter addresses the next limitation of that framework. Even when planning remains support-aware, the planner still assumes that the temporal context used to choose the next skill is sufficiently reliable. In autonomous physical systems, that assumption can fail.

A learned offline planner often conditions on a history of recent states, previous skills, return information, or task progress. In controlled benchmark evaluation, this history is usually complete and clean. In deployment, the available history may be truncated, corrupted, delayed, partially missing, or misleading because execution has drifted away from the nominal trajectory. A policy that performs well under complete context can therefore fail when the temporal evidence needed to select the next skill is incomplete. This is not only a robustness add-on; it is a long-horizon decision-policy problem. If a policy cannot infer the correct next decision from partial context, it may choose a locally plausible skill that breaks the global task sequence.

This chapter develops Context-Robust Inference of Skills for Planning (CRISP), a framework for long-horizon offline control under degraded temporal context. The method retains the latent skill

infrastructure introduced in Chapter 5: a diffusion-based skill executor, conservative latent critics, and latent-space refinement. It then adds a new training objective, masked skill inference, that hides subsets of latent skill decisions during training and requires the planner to reconstruct them from the remaining context. By training the sequencer to recover missing decisions, the method discourages brittle dependence on adjacent tokens and promotes non-local temporal reasoning.

The chapter is therefore not a repetition of the support-preserving planning chapter. Chapter 5 addressed the question of how to avoid unsupported behavior in fixed-data control. This chapter asks whether the high-level planner can still choose appropriate skills when the decision history itself is incomplete or degraded. The distinction is important: reliable long-horizon decision policies require not only feasible action generation, but also inference mechanisms that remain dependable when the evidence available for decision-making is imperfect.

6.1 Problem Setting and Preliminaries

6.1.1 Degraded Temporal Decision Context

Sequence-based planners are attractive for long-horizon offline control because they can condition on previous states, previous skills, and desired return. This temporal context helps the model infer what has already been accomplished and what decision should come next. The weakness is that a sequence model can learn shortcuts. If the next skill is often predictable from the immediately previous skill, the model may rely on local correlations rather than on the broader task structure. This can produce strong validation performance under clean trajectories while leaving the planner fragile when history is missing or corrupted.

The problem is easiest to see in compositional tasks. A robot may need to complete a sequence of subtasks in the correct order, or a navigation agent may need to choose a skill based on information observed earlier in the path. The most recent token may be a poor guide when the correct decision depends on a distant cue. If the model has learned to depend on adjacent tokens, it can fail when those tokens are removed, noisy, or misleading. The failure is not that the chosen skill is necessarily outside the dataset; the skill may still be data-supported. The failure is that the planner selects the

wrong supported skill because it has not inferred the correct long-range context.

6.2 Methodological Challenge and Design Principle

6.2.1 Local Shortcut Reliance in Sequence Planners

When temporal decision context is degraded—such as when past observations are missing, corrupted, or uninformative—a sequence planner’s ability to maintain the correct long-horizon objective is severely tested. This challenge highlights a critical distinction in offline control: the difference between context robustness and support preservation. Support preservation constrains the planner to choose skills that remain close to demonstrated behavior. Context robustness concerns whether the planner chooses the appropriate supported behavior when the history is incomplete. A policy can be support-preserving but context-fragile: it may choose a valid-looking skill from the dataset, but the wrong one for the current long-horizon objective. Conversely, a context-aware planner without support control may infer the correct intent but select latents whose decoded actions are not reliable. Long-horizon offline control requires both.

CRISP addresses the context component by treating missing decisions as part of the training problem rather than only as a test-time perturbation. During training, the model sees partial skill sequences and learns to reconstruct masked skill tokens from the visible context. This objective forces the planner to use non-local information and reduces the incentive to rely exclusively on short-range correlations. At deployment, the policy acts causally using only available past context, but it has been trained to reason under missing information.

6.2.2 CRISP Overview

CRISP is a hierarchical offline reinforcement learning framework for context-robust skill inference. The method uses the same general separation of roles established in Chapter 5: short-horizon execution is performed by a deterministic diffusion-based skill decoder, while long-horizon decision-making occurs in latent skill space. The new contribution is the training of the high-level skill

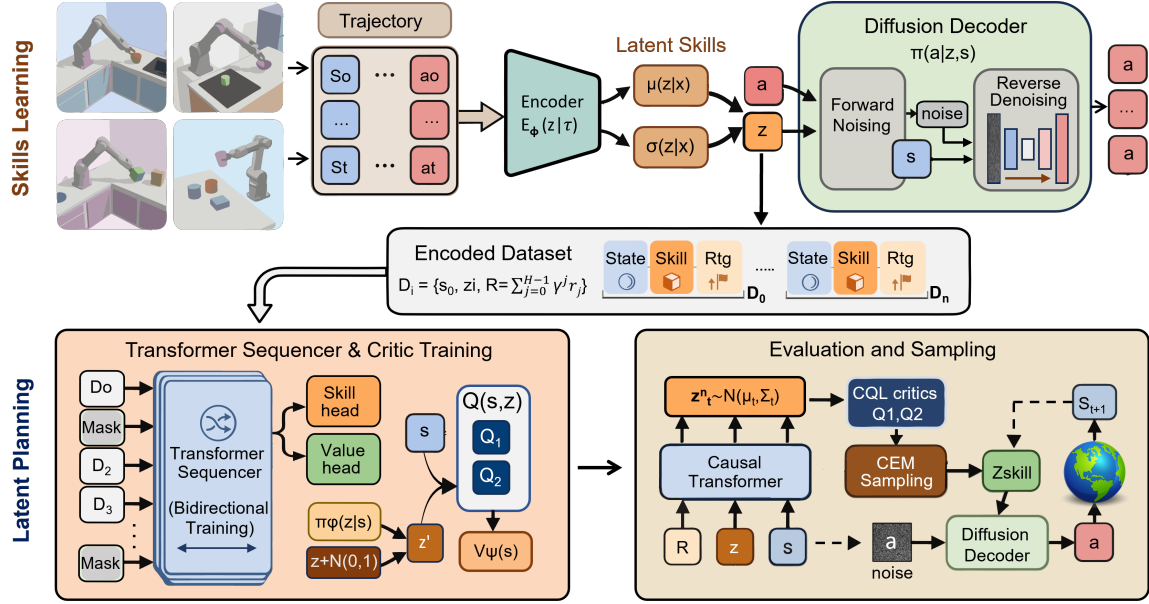


Figure 6.1: *CRISP pipeline. A diffusion-based executor decodes latent skills into short action fragments, while a masked latent planner reconstructs hidden skill decisions from partial context and uses conservative value-guided refinement for support-aware deployment.*

sequencer through masked skill inference.

The pipeline has four components. First, fixed-length trajectory segments are encoded into latent skills using a Conditional Diffusion Autoencoder. Second, a deterministic diffusion decoder maps selected latent skills into executable action fragments. Third, conservative latent critics estimate the value of candidate skills while penalizing unsupported latent proposals. Fourth, a transformer-based skill sequencer predicts latent skills from temporal context and is trained to reconstruct masked skill tokens. The conservative critics and deterministic decoder preserve the fixed-data feasibility mechanisms from Chapter 5; the masked sequencer introduces robustness to incomplete decision context.

Figure 6.1 summarizes the architecture. The important design point is that masking operates on latent skill decisions rather than raw action tokens. A masked raw action may be recoverable from local smoothness, especially in low-level control. A masked skill represents a temporally extended decision. Reconstructing it requires the model to infer which behavior fragment belongs at that point in the long-horizon sequence. This aligns the training objective with the actual decision variable used during planning.

6.3 Technical Formulation

6.3.1 Masked Skill Inference

Let a trajectory be decomposed into skill-level tokens. Each token corresponds to a fixed-horizon trajectory segment encoded into a latent skill $z_i \in \mathbb{R}^{d_z}$. A transformer sequencer receives a context window containing states, previous skills, and optional return-to-go information. In ordinary next-skill prediction, the model is trained to predict the next skill from a complete causal history. CRISP modifies this objective by masking a subset of skill tokens during training.

For each training sequence, a binary mask $m_i \in \{0, 1\}$ is sampled over valid skill positions. A masked position has $m_i = 1$ and is hidden from the skill stream. The sequencer predicts a Gaussian distribution for each skill token,

$$\pi_\psi(z_i \mid C_{\text{partial}}) = \mathcal{N}(\mu_i, \text{diag}(\sigma_i^2)), \quad (6.1)$$

where C_{partial} denotes the visible partial context. The masked negative log-likelihood (NLL) objective is

$$\mathcal{L}_{\text{mask}} = \frac{\sum_i m_i \text{NLL}(z_i; \mu_i, \sigma_i)}{\sum_i m_i + \epsilon}. \quad (6.2)$$

Only masked positions contribute to the loss. This forces the model to reconstruct missing decisions rather than simply fitting every token under complete information.

Masking latent skills differs from masking raw actions in two ways. First, each latent skill represents a short action fragment, so reconstructing a masked token corresponds to recovering a compositional decision rather than an instantaneous motor command. Second, the skill representation is the same decision space used by the planner and conservative critic. Training the sequencer to infer missing skills therefore directly improves the robustness of the planning variable, not only the robustness of a low-level action predictor.

CRISP uses random token masking as the main training mechanism and evaluates contiguous block masking as an ablation. Random masking scatters missing positions across the sequence, which makes local interpolation less reliable and encourages the sequencer to use distributed

temporal information. Contiguous block masking removes a span of consecutive tokens; it can test robustness to sustained missing history, but it may also allow reconstruction from boundary tokens around the gap. In the main training configuration, the masking ratio is sampled adaptively as

$$\alpha \sim \mathcal{U}(0.2, 0.5), \quad (6.3)$$

so that the model encounters both mild and moderate context degradation during training.

A practical issue arises when the skill stream is shifted. If the previous skill conditions prediction of the next skill, then naively masking position i can still reveal the target skill as the shifted input at position $i + 1$. CRISP prevents this leakage by hiding both the masked prediction position and its shifted carrier. This ensures that reconstruction depends on genuine context inference rather than copying the answer from an adjacent token.

During training, the model can use visible context from the sequence to learn stronger representations. During deployment, however, the policy is causal: it uses only available past context and does not access future skill tokens. The masked objective therefore functions as a training-time regularizer for context robustness, while the deployed policy remains compatible with online execution.

6.3.2 Advantage-Weighted Masked Learning and Behavior Anchoring

Masked reconstruction alone would train the sequencer to imitate missing skills from the dataset. CRISP additionally uses conservative value estimates to emphasize useful skills while retaining behavior anchoring. The conservative critics are trained in latent space as in Chapter 5, so the discussion here focuses on how their advantages shape the masked skill objective.

For a state-skill pair (s, z) , the conservative advantage is defined as

$$A(s, z) = \min(Q_1(s, z), Q_2(s, z)) - V(s), \quad (6.4)$$

with an exponential weight

$$w(s, z) = \exp\left(\frac{A(s, z)}{\tau_{\text{AWR}}}\right). \quad (6.5)$$

The advantage-weighted masked loss is

$$\mathcal{L}_{\text{AWR}} = \frac{\sum_i m_i w_i \text{NLL}(z_i; \mu_i, \sigma_i)}{\sum_i m_i + \epsilon}. \quad (6.6)$$

This objective gives greater weight to masked skills that the conservative critic estimates as useful. Because the advantage is computed using pessimistic value estimates, the weighting favors high-value skills without encouraging arbitrary drift away from the offline dataset.

CRISP also includes a behavior-anchoring term on masked positions:

$$\mathcal{L}_{\text{beh}} = \beta_{\text{beh}} \frac{\sum_i m_i D_{\text{KL}}(\pi_{\psi}(z_i | C_{\text{partial}}) \parallel q_{\phi}(z_i | \tau_i))}{\sum_i m_i + \epsilon}. \quad (6.7)$$

Here, $q_{\phi}(z_i | \tau_i)$ is the skill encoder distribution for the corresponding trajectory segment. This term discourages masked reconstruction from drifting too far from the encoded offline behavior. The combined sequencer objective can be written as

$$\mathcal{L}_{\text{seq}} = \mathcal{L}_{\text{AWR}} + \mathcal{L}_{\text{beh}}. \quad (6.8)$$

The loss couples three requirements: reconstruct missing skills, emphasize skills with conservative value support, and remain anchored to the offline behavior distribution.

6.3.3 Causal Skill Inference and Value-Guided Refinement

At deployment, the planner receives the available past context and predicts a Gaussian distribution over the next latent skill. It does not use future tokens. This causal execution constraint is essential because the masked objective is used only during training. The purpose of training-time masking is to make the sequencer less dependent on complete histories, not to assume access to future information.

The predicted latent distribution is refined using conservative value guidance in latent space. Candidate skills are sampled from the planner proposal, scored with

$$Q_{\min}(s, z) = \min(Q_1(s, z), Q_2(s, z)), \quad (6.9)$$

and updated by the Cross-Entropy Method. The CRISP configuration uses 5 CEM iterations and 64 candidates. The selected latent is then decoded into an H -step action sequence using deterministic DDIM decoding. The diffusion executor is not used to generate synthetic training transitions; it is invoked for execution after the latent decision has been selected.

The latent value functions use horizon-aligned Bellman targets. For an encoded transition (s, z, R_H, s') , the target is

$$y = R_H + \gamma^H V_{\text{target}}(s'), \quad (6.10)$$

where

$$R_H = \sum_{h=0}^{H-1} \gamma^h r_{t+h}. \quad (6.11)$$

The conservative critic objective is

$$\begin{aligned} \mathcal{L}_{\text{cons}}(Q_i) = & \alpha_{\text{CQL}} \left(\mathbb{E}_{z \sim \mu(z)} [Q_i(s, z)] - \mathbb{E}_{(s, z) \sim D_z} [Q_i(s, z)] \right) \\ & + \frac{1}{2} \mathbb{E}_{(s, z) \sim D_z} \left[(Q_i(s, z) - y)^2 \right]. \end{aligned} \quad (6.12)$$

The proposal distribution $\mu(z)$ includes latent samples near dataset support, including perturbations of dataset latents, state-conditional prior samples, and sequencer proposals. This formulation retains support-aware value estimation from Chapter 5, while the masked objective changes how the sequencer learns to infer skills under incomplete context.

Algorithm 4 CRISP training and context-robust skill inference pipeline

Require: Offline dataset $\mathcal{D}$, skill horizon H , latent dimension d_z , context window, masking-ratio distribution

Ensure: Skill executor, masked skill sequencer, conservative latent critics

- 1: Train the Conditional Diffusion Autoencoder on trajectory segments from $\mathcal{D}$
 - 2: Learn encoder $q_\phi(z \mid \tau)$, state-conditional prior $p_\omega(z \mid s)$, and deterministic diffusion decoder $p_\theta(a \mid s, z)$
 - 3: Freeze q_ϕ and encode fixed-horizon segments into $D_z = \{(s, z, R_H, s')\}$
 - 4: Initialize latent critics Q_1, Q_2, V and masked skill sequencer π_ψ
 - 5: **for** each Stage-2 training iteration **do**
 - 6: Sample a context-window batch from D_z
 - 7: Update V using expectile regression and update Q_1, Q_2 using Eq. (6.10) and Eq. (6.12)
 - 8: Sample masking ratio α and binary mask m over valid skill positions
 - 9: Hide masked skill tokens and shifted carrier tokens to prevent leakage
 - 10: Predict Gaussian skill distributions with $\pi_\psi(z_i \mid C_{\text{partial}})$
 - 11: Compute conservative advantages using Eq. (6.4)
 - 12: Update π_ψ using Eq. (6.8)
 - 13: Polyak update target networks
 - 14: **end for**
 - 15: During deployment, use causal context to propose the next latent skill
 - 16: Refine latent candidates using CEM and $Q_{\min}(s, z)$
 - 17: Decode the selected latent using deterministic DDIM and execute the resulting action fragment
-

6.4 Learning Objective and Algorithm

6.4.1 CRISP Training and Deployment Pipeline

Algorithm 4 summarizes the training and deployment pipeline. The algorithm emphasizes the differences from Chapter 5: the skill executor and latent critics remain infrastructure, while the sequencer is trained through masked advantage-weighted inference.

6.5 Experimental Setup

The evaluation uses D4RL and RoboSuite benchmarks, but the chapter focuses on the aspects most relevant to context robustness. The D4RL tasks span locomotion, navigation, sequential manipulation, and dexterous manipulation⁵⁰. Navigation and Kitchen are especially relevant because they require long-horizon composition from fragmented data. Adroit adds contact-rich

dexterous manipulation, while locomotion tests whether the method remains competitive when short-horizon feedback control is dominant. RoboSuite evaluates contact-rich manipulation with a Franka Panda arm⁵¹.

Baselines are grouped into model-free offline RL, model-based or planning methods, diffusion planners, diffusion policies, unmasked latent sequencers, and masking variants. The most important comparisons for this chapter are not only full-context benchmark scores, but also context-dropout tests. In context-dropout evaluation, portions of the past latent skill history are removed at inference. This controlled perturbation approximates degraded high-level decision memory, such as truncated buffers, missing history, delayed sensing, or execution drift. It does not directly corrupt raw observations; it isolates the planner’s dependence on temporal decision context.

CRISP is trained in two stages. Stage 1 learns the skill executor and encoder. Stage 2 trains the masked skill sequencer and conservative latent critics on the encoded dataset. The transformer sequencer uses 8 layers and 8 attention heads, while the critics are 3-layer MLPs with width 256. The masking ratio follows $\alpha \sim \mathcal{U}(0.2, 0.5)$, and skill horizon is selected in the range $H \in \{8, 12\}$ depending on the domain.

6.6 Empirical Results and Analysis

6.6.1 Full-Context Performance

Full-context evaluation verifies that the masked objective does not sacrifice ordinary benchmark performance. Table 6.1 provides the detailed D4RL evaluation. The table follows the same D4rl evaluation pattern as in Chapter 5 for latent planning. Here, the relevant question is whether CRISP remains competitive while adding context robustness.

The strongest full-context gains occur in Navigation, Sequential Manipulation, and Adroit. These are the settings where long-horizon composition and decision context matter most. Locomotion remains competitive but not category-leading, which is consistent with the interpretation from Chapter 5: smooth locomotion rewards reactive control more than long-range skill sequencing. The full-context results establish that masking improves robustness without turning the method

Table 6.1: Unified Evaluation Results on D4RL Benchmark. Comparison against non-diffusion (model-free/model-based) and diffusion-based (planner/policy) baselines. The maximum score in each row is **bolded** and the second highest is underlined. *M/R/E*: Medium/Replay/Expert; *U/L/D*: Umaze/Large/Diverse; *P/Mxd/Cmp*: Partial/Mixed/Complete; *H/C/E*: Human/Cloned/Expert.

Task Name	Non-Diffusion						Diffusion-based									
	Model-Free			Model-Based			Diffusion as Planner				Diffusion as Policy				OURS	
	IQL	BC	CQL	TAP	MOREL	TT	DD	HDMI	LDCQ	Diffuser	IDQL	DQL	DQL@1	QGPO	SfBC	(Mean $\pm$ Std)
Locomotion	Hopper-M	66.3	52.9	58.5	63.4	95.4	61.1	79.3	76.4	66.2	58.5	65.4	82.4	64.1	98.0	57.1 97.34 $\pm$ 0.6
	Hopper-M-R	94.7	18.1	95.0	87.3	93.6	91.5	<u>100.0</u>	99.6	86.2	96.8	92.1	100.7	63.1	96.9	86.2 94.26 $\pm$ 1.7
	Hopper-M-E	91.5	52.5	105.4	105.5	108.7	110.0	111.8	<u>113.5</u>	107.2	107.2	108.6	107.2	100.6	108.0	113.9 $\pm$ 2.6
	HCheetah-M	47.4	42.6	44.0	45.0	42.1	46.9	49.1	48.0	42.8	44.2	51.0	50.6	47.8	54.1	45.9 50.93 $\pm$ 0.69
	HCheetah-M-R	44.2	36.6	45.5	40.8	40.2	41.9	39.3	44.9	41.8	42.2	45.9	<u>47.5</u>	44.0	47.6	37.1 42.54 $\pm$ 0.84
	HCheetah-M-E	86.7	55.2	91.6	91.8	53.3	95.0	90.6	92.1	90.2	79.8	<u>95.9</u>	96.1	94.8	93.5	91.6 92.13 $\pm$ 0.5
	Walker2d-M	78.3	75.3	72.5	64.9	77.8	79.0	82.5	79.9	69.4	79.7	82.5	<u>85.1</u>	82.0	86.0	53.1 77.42 $\pm$ 0.7
	Walker2d-M-R	73.9	26.0	77.2	68.5	49.8	82.6	75.0	80.7	66.8	61.2	<u>85.1</u>	94.3	75.6	84.4	65.1 84.14 $\pm$ 0.9
	Walker2d-M-E	109.6	107.5	108.8	107.4	95.6	101.9	108.8	107.9	109.3	108.4	108.6	109.7	108.9	<u>110.7</u>	109.6 112.24 $\pm$ 0.8
Average		76.96	51.87	77.19	74.76	72.94	78.80	81.60	82.60	76.18	75.30	81.68	<u>85.95</u>	75.60	86.58	72.70 84.94
Navigation	Maze2d-U	47.4	3.8	5.7	-	0	-	116.2	120.1	<u>134.2</u>	113.9	57.9	-	-	-	73.9 146.64 $\pm$ 3.12
	Maze2d-M	34.9	30.0	5.0	-	1.3	-	122.3	121.8	<u>125.3</u>	121.5	89.5	-	-	-	73.8 147.74 $\pm$ 2.56
	Maze2d-L	58.6	5.0	12.5	-	0	-	125.9	128.6	150.1	123.0	90.0	-	-	-	74.4 148.2 $\pm$ 3.1
	AntMaze-U-D	62.2	45.6	80.3	69.8	0	-	49.2	73.7	81.4	72.2	80.2	66.2	56.4	74.4	<u>85.3</u> 92.18 $\pm$ 2.8
	AntMaze-M-D	70.0	0.0	53.7	85	-	0.6	24.6	-	68.9	45.5	<u>84.8</u>	78.6	14.8	83.6	88.43 $\pm$ 3.7
	AntMaze-L-D	47.5	0.0	14.9	-	-	0	7.5	71.5	57.7	22.0	<u>67.9</u>	56.6	1.6	64.8	45.5 61.21 $\pm$ 1.76
Average		53.43	14.06	28.68	-	-	-	74.28	-	<u>102.94</u>	83.01	78.38	-	-	-	72.48 114.03
Sequential	Kitchen-P	46.3	38.0	49.8	-	-	-	56.5	-	<u>67.8</u>	55.7	66.7	60.5	-	66.0	47.9 68.24 $\pm$ 2.84
	Kitchen-Mxd	51.0	51.5	51.0	-	-	-	51.8	-	<u>62.3</u>	52.5	<u>66.5</u>	62.6	-	45.5	45.4 69.84 $\pm$ 1.7
	Kitchen-Cmp	62.5	65.0	43.8	-	-	-	-	-	62.5	-	<u>62.5</u>	84.02	-	62.8	77.9 <u>83.12</u> $\pm$ 3.6
	Average	53.26	51.5	48.2	-	-	-	-	-	64.4	-	65.23	<u>69.03</u>	-	58.1	57.06 73.73
Adroit	Pen-C	37.3	37.0	39.2	57.4	-	11.4	47.7	48.3	47.7	61.7	82.3	28.3	49.3	54.2	- 87.34 $\pm$ 2.8
	Pen-E	-	85.1	107.0	127.4	-	72.0	107.6	109.5	121.2	99.7	<u>137.8</u>	133.5	112.6	119.1	- 146.84 $\pm$ 3.4
	Door-C	1.6	0.0	0.4	11.7	-	-0.1	9.0	9.3	1.1	0.1	4.4	-0.3	10.6	<u>11.2</u>	- 3.43 $\pm$ 0.92
	Door-E	-	34.9	101.5	104.8	-	94.1	94.0	85.9	<u>105.1</u>	103	105.0	104.8	93.7	98.8	- 107.03 $\pm$ 0.71
	Relocate-C	-0.1	-0.3	-0.1	-0.2	-	-0.1	0.15	-0.1	-0.2	0.0	0.0	0.1	-0.2	-0.2	- 0.31 $\pm$ 0.08
	Relocate-E	-	101.3	95.0	105.8	-	10.3	87.5	91.3	104.0	102.2	107.0	<u>108.5</u>	95.2	102.5	- 111.48 $\pm$ 2.3
	Hammer-C	1.6	0.8	2.1	1.2	-	0.5	0.9	1.0	2.8	1.2	<u>3.5</u>	0.2	1.1	1.1	- 3.92 $\pm$ 0.94
	Hammer-E	-	125.6	86.7	127.6	-	15.5	106.7	111.8	45.8	103.1	127.6	<u>128.3</u>	114.8	123.2	- 134.82 $\pm$ 1.24
	Average	-	48.05	53.96	66.96	-	25.45	56.70	57.13	53.44	58.86	<u>70.95</u>	62.925	59.6375	63.74	- 74.39
Hammer	Pen-H	71.3	34.4	37.5	<u>76.5</u>	-	36.4	64.1	66.2	74.1	-	-	72.8	66.0	73.9	- 82.36 $\pm$ 3.1
	Door-H	4.3	0.5	9.9	8.8	-	0.1	6.9	7.1	11.8	-	-	-	8.0	8.5	- 6.98 $\pm$ 1.56
	Relocate-H	0.1	0.0	0.2	0.2	-	0.0	0.2	0.1	0.3	-	-	-	0.2	0.2	- 0.29 $\pm$ 0.6
	Hammer-H	1.4	1.2	4.4	1.4	-	0.8	1.0	1.2	1.5	-	-	-	1.3	1.4	- <u>4.12</u> $\pm$ 1.1
Average		19.28	9.03	13.00	21.73	-	9.33	18.05	18.65	<u>21.93</u>	-	-	-	18.86	21	- 23.43

into a narrow robustness-only model.

6.6.2 Context-Dropout Robustness

Context-dropout evaluation is the centerpiece of the chapter. It directly tests whether the sequencer can continue selecting useful skills when parts of its high-level history are unavailable. The evaluation is conducted on AntMaze-Large-Diverse, a sparse-reward navigation task where the agent must compose local motion fragments into a long path.

Table 6.2 shows that the performance gap grows as context is degraded. With complete context,

Table 6.2: Context-dropout robustness on AntMaze-Large-Diverse. At 40% context loss, CRISP retains 64.5% of its full-context performance, while the unmasked baseline retains 34.3%.

Method	0% dropout	40% dropout	Retention
CRISP	61.21	39.49	64.5%
No masking	54.93	18.86	34.3%
Gap	+6.28	+20.63	—

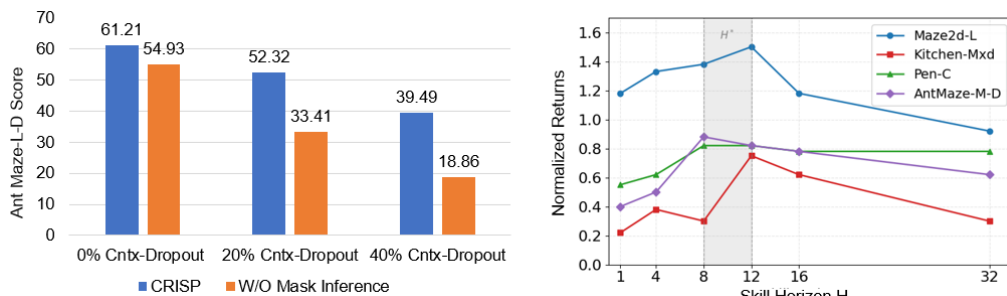


Figure 6.2: Context-dropout robustness and skill-horizon sensitivity. The left panel compares CRISP with the unmasked baseline under degraded temporal context. The right panel shows that intermediate skill horizons balance abstraction and value stability.

CRISP exceeds the unmasked baseline by 6.28 points. Under 40% context loss, the gap increases to 20.63 points. This pattern supports the central claim of the chapter: masking improves the sequencer’s ability to infer useful skills from partial temporal information. The method does not merely improve average performance; it changes the failure mode under missing history.

Figure 6.2 visualizes context-dropout robustness together with horizon sensitivity. The left panel shows that the masked model degrades more gracefully as context is removed. The right panel shows that intermediate horizons provide the best tradeoff between temporal abstraction and value-estimation stability.

This result is particularly important because the perturbation is applied to the high-level decision history rather than to raw observations. The test isolates a failure mode of sequence planners: dependence on complete recent skill context. CRISP is trained to reconstruct missing skill tokens, so its stronger performance under dropout is directly tied to the masked skill inference objective.

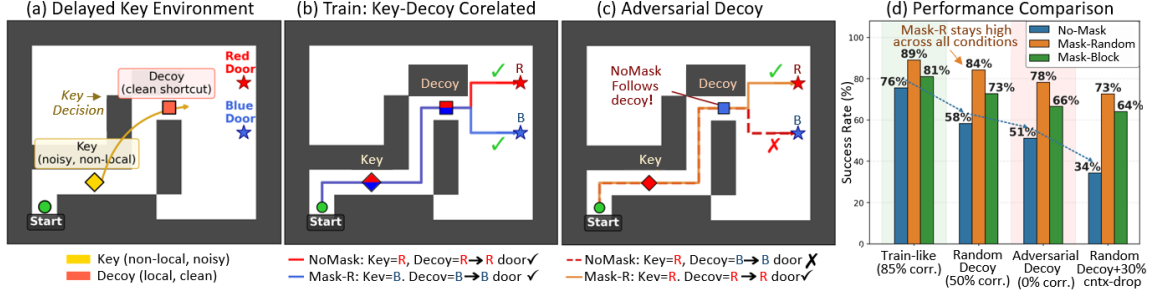


Figure 6.3: Delayed key–door diagnostic. The correct final decision depends on an early key cue, while a later decoy cue can create a local shortcut. Masked skill inference reduces reliance on the decoy and improves robustness when the shortcut is weakened or inverted.

Table 6.3: Delayed key–door diagnostic success rates. Random skill masking remains more robust than the unmasked sequencer when the decoy shortcut is weakened, inverted, or when context is dropped.

Evaluation condition	No masking	Mask-Random	Mask-Block
Train-like decoy correlation	76%	89%	81%
Random decoy correlation	58%	84%	73%
Adversarial decoy correlation	51%	78%	66%
30% context dropout	34%	73%	64%

6.6.3 Delayed Key–Door Diagnostic

The delayed key–door task provides a controlled diagnostic for shortcut reliance. The agent observes an early key cue that determines the correct final door. A later decoy cue is spuriously correlated with the correct door during training. A planner that relies on the decoy can perform well under train-like conditions but fail when the decoy becomes random or adversarial. This diagnostic isolates non-local temporal reasoning more clearly than a large benchmark table.

Table 6.3 shows that the unmasked model collapses when the local decoy no longer predicts the correct door. Mask-Random remains substantially stronger under random and adversarial decoy settings and retains 73% success under 30% context dropout. Because all methods use the same data and architecture class, the difference is attributable to the masking objective. The diagnostic supports the interpretation that random masking reduces reliance on local shortcuts and encourages decisions based on information distributed across the trajectory.

Table 6.4: Adaptive masking ratio ablation. Sampling $\alpha \sim \mathcal{U}(0.2, 0.5)$ exposes the model to varied levels of context loss and outperforms no masking and fixed masking ratios.

Task	Uniform	No masking	Fixed 0.2	Fixed 0.5	Fixed 0.7
AntMaze-L-D	61.21	54.93	56.18	51.27	48.83
Hopper-M-R	94.26	88.38	90.02	87.61	82.48
Kitchen-Mixed	69.84	64.25	65.10	61.87	51.54
Pen-Cloned	87.34	82.24	86.27	80.53	73.51

Table 6.5: Masking structure ablation. Random token masking outperforms contiguous block masking, indicating that scattered missing decisions more effectively discourage reliance on local interpolation.

Task	Random mask	Contiguous mask	Gain
AntMaze-L-D	61.21	57.43	+3.78
Kitchen-Mixed	69.84	63.11	+6.73
Pen-Cloned	87.34	84.80	+2.54
Average gain	—	—	+4.35

6.6.4 Mask Structure and Component Ablations

The ablation results test whether robustness comes specifically from masking and whether the fixed-data planning infrastructure remains necessary. Table 6.4 compares adaptive masking with fixed masking ratios and no masking. The adaptive ratio $\alpha \sim \mathcal{U}(0.2, 0.5)$ performs best across all listed tasks.

The fixed-ratio trend is informative. Very low masking allows the model to keep relying on local continuity. Very high masking removes too much useful context. The adaptive range exposes the model to both easier and harder reconstruction conditions, producing more reliable performance.

Table 6.5 shows that random masking is stronger than contiguous masking. The result is consistent with the chapter thesis. A contiguous block can sometimes be reconstructed by interpolating from neighboring boundary tokens. Random masking scatters missing positions throughout the context window, making it harder for the sequencer to rely only on local continuity.

Table 6.6 confirms that context robustness does not replace support preservation or controllability. Removing CQL sharply reduces performance, especially on AntMaze-L-D, because the planner loses support-aware value guidance. Removing z -force weakens latent controllability, making de-

Table 6.6: *Planning component ablations. Masked skill inference improves context robustness, but conservative value learning, latent controllability, and temporal abstraction remain necessary for reliable long-horizon offline control.*

Ablation	AntMaze-L-D	Hopper-M-R	Kitchen-M	Pen-C
CRISP (Full)	61.21	94.26	69.84	87.34
W/O z -force	47.35	77.23	53.73	73.58
W/O CQL	40.54	58.48	51.33	79.49
$H = 1$	38.84	76.29	48.37	72.85

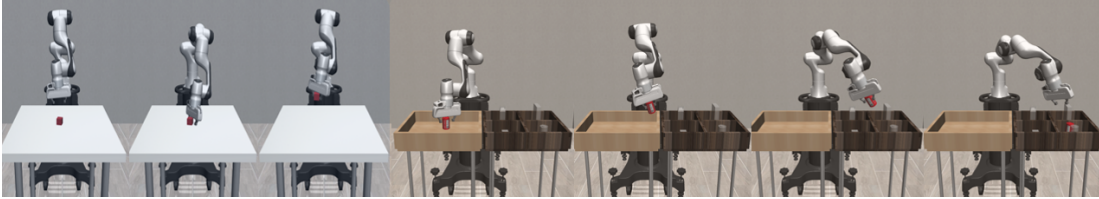


Figure 6.4: *RoboSuite manipulation tasks. CRISP controls a Franka Panda arm for Lift and Pick-and-Place, testing context-robust latent skill inference in contact-rich manipulation.*

Table 6.7: *RoboSuite success rates over 10 seeds and 25 rollouts per seed.*

Task	Success rate
Lift	97.2% $\pm$ 1.7%
Pick-and-Place	95.2% $\pm$ 2.6%

coded behavior less responsive to high-level decisions. Setting $H = 1$ collapses the abstraction toward action-level prediction and is especially harmful on long-horizon tasks. These ablations connect Chapter 5 and this chapter: support-preserving latent planning remains necessary, while masked skill inference adds robustness to degraded context.

6.6.5 RoboSuite Results

RoboSuite evaluates whether the method remains effective in contact-rich manipulation with a different robot embodiment. The tasks use a 7-DoF Franka Panda arm rather than the Adroit Shadow Hand. Lift requires grasping and raising an object, while Pick-and-Place requires grasping, transporting, and placing a can in the correct bin.

The RoboSuite results show that the method remains effective beyond D4RL trajectories. They

do not establish real-robot deployment, but they provide evidence that masked latent planning with deterministic skill execution can handle multi-stage contact-rich manipulation under a different morphology and action interface.

6.7 Discussion

This chapter adds the fourth technical layer to the dissertation’s argument. Chapter 3 used decomposition to manage horizon length and terminal precision. Chapter 4 used attention to improve feature relevance in actor–critic learning. Chapter 5 developed support-preserving latent planning for fixed-data control. This chapter shows that even when planning is support-aware, the decision process can remain fragile if the temporal context is assumed to be complete and reliable.

The empirical evidence supports three conclusions. First, masked skill inference improves robustness to degraded high-level context, as shown by the 40% context-dropout evaluation. Second, random masking is more effective than contiguous block masking in the evaluated tasks, suggesting that scattered reconstruction pressure better discourages local shortcut reliance. Third, context robustness depends on the same infrastructure that enabled support-preserving planning: conservative critics, latent controllability, and temporal abstraction remain necessary.

The scope of the evidence should be stated carefully. The context-dropout test is a controlled proxy for degraded decision history; real sensor corruption, state-estimation errors, asynchronous observations, and execution drift may differ in structure. The method depends on the quality of the learned skill representation and on the selected skill horizon. Masking-ratio choices affect performance, and the current evaluation remains in benchmark and simulation environments. The method does not provide a formal robustness guarantee or a real-robot safety guarantee. These limitations define important future work: evaluating masked skill inference under richer partial observability, real sensor noise, online adaptation, and physical robot deployment.

The chapter nevertheless strengthens the dissertation thesis. Long-horizon decision policies in autonomous physical systems require structure matched to the decision problem. In fixed-data control, the structure must preserve support. Under degraded history, the structure must also make the planner less dependent on complete local context. CRISP addresses this by making missing

decision context part of the training objective.

6.8 Summary and Transition

This chapter developed context-robust skill inference for long-horizon offline control. It began from a limitation left open by support-preserving latent planning: a policy may remain within dataset support but still fail when its decision history is incomplete or corrupted. CRISP addresses this limitation by training a skill sequencer with masked skill inference, advantage-weighted reconstruction, behavior anchoring, and causal deployment with conservative value-guided refinement.

The results show that CRISP remains competitive under full-context evaluation and is substantially more robust under controlled context dropout. The delayed key-door diagnostic further isolates the mechanism, showing that masking reduces reliance on local shortcuts and improves non-local temporal reasoning. Ablations confirm that masked inference, support preservation, latent controllability, and temporal abstraction each contribute to reliable long-horizon offline control.

This chapter completes the dissertation’s technical progression. The next chapter synthesizes the design principles developed across trajectory-level autonomy and offline embodied control, discusses limitations, and outlines future directions for reinforcement learning for long-horizon decision policies in autonomous physical systems.

Chapter 7

Conclusion and Future Directions

7.1 Dissertation Summary

This dissertation studied reinforcement learning for long-horizon decision policies in autonomous physical systems. The central problem was that physical autonomy requires more than selecting locally useful actions. A learned policy must preserve the possibility of successful future action across extended temporal horizons, nonlinear dynamics, physical constraints, limited feedback, restricted interaction, and imperfect decision context. This requirement creates the long-horizon feasibility gap developed in Chapter 1: the objectives of physical systems unfold over many future decisions, while the learning process often receives local, delayed, incomplete, or distribution-limited information.

The dissertation argued that reinforcement learning becomes more suitable for autonomous physical systems when the learning process is structured around the temporal and physical nature of the decision problem. This argument was developed through two representative validation settings. The first setting considered trajectory-level physical autonomy, where learned policies must guide long-duration transfers under continuous dynamics and terminal precision requirements. The second setting considered offline embodied control, where policies must compose long-horizon behavior from fixed datasets without unsafe extrapolation. Although these settings differ in dynamics, data access, and evaluation criteria, they expose the same underlying challenge: learned policies

must remain feasible, controllable, and reliable over horizons longer than the immediate learning signal.

7.2 Synthesis of Findings

The technical chapters collectively show that no single reinforcement learning mechanism is sufficient for long-horizon physical autonomy. Instead, different forms of structure address different parts of the long-horizon feasibility gap. Staged decomposition reduces the burden of learning a single policy that must handle both broad progress and strict terminal precision. Attention-enhanced actor-critic learning improves the policy’s ability to emphasize decision-relevant features when dynamics and terminal requirements change across a trajectory. Temporal abstraction through latent skills reduces the effective planning horizon by moving long-horizon reasoning from individual actions to higher-level behavior units.

The offline-control chapters extend the same thesis to settings where additional interaction is unavailable. In this regime, feasibility is not only a property of dynamics and constraints, but also a property of data support. Support-preserving latent planning addresses this by keeping long-horizon decisions in a learned skill space where conservative value learning can discourage unsupported extrapolation. Deterministic diffusion decoders provide expressive short-horizon execution while avoiding the instability of using generative sampling as the entire planning mechanism. Context-robust masked skill inference then addresses a further weakness of sequence-based decision policies: their tendency to rely on complete and clean temporal context. By training the planner to infer missing latent decisions, the policy becomes less dependent on brittle local correlations.

Taken together, these findings support the dissertation’s central claim. Reinforcement learning for autonomous physical systems should not be viewed as a generic replacement for optimization, control, or physical modeling. It is more defensible as a structured decision-learning framework. The structure must be chosen according to the failure mode being addressed: horizon length, feature relevance, support preservation, latent controllability, generative execution, or context robustness. The dissertation contributes a unified view in which these mechanisms are complementary responses to the same long-horizon physical decision-policy learning problem.

7.3 Limitations

The trajectory-autonomy studies are simulation-based and rely on modeled dynamics, selected transfer scenarios, reward design, and specified terminal conditions. They do not establish flight readiness, global optimality, or formal safety guarantees. Higher-fidelity effects, uncertainty, operational constraints, navigation errors, and broader mission families require further study before deployment-level claims can be made.

The offline embodied-control studies are evaluated in benchmark and simulation environments. These settings are useful for controlled comparison, but they do not fully capture the sensing uncertainty, hardware variation, contact complexity, and recovery requirements of real robotic deployment. The methods also depend on choices such as skill horizon, latent dimension, conservative critic strength, masking ratio, and planner refinement settings. These choices affect the balance between expressivity and support preservation. In addition, the dissertation provides empirical evidence and design principles rather than formal guarantees of global safety, feasibility, or optimality.

These limitations do not weaken the main conclusion. They clarify the level at which the dissertation contributes. The work demonstrates that structured reinforcement learning can improve long-horizon physical decision policies under representative modeled conditions. It also identifies the mechanisms that appear most important for making such policies more feasible, controllable, support-aware, and context-robust.

7.4 Future Directions

Several directions follow naturally from this dissertation. First, future work should develop stronger feasibility and safety guarantees for learned long-horizon policies. This includes integrating constraint satisfaction, reachability analysis, control barrier functions, or verification tools with learned policies so that physical feasibility can be monitored and enforced more explicitly.

Second, uncertainty-aware reinforcement learning is needed for autonomous physical systems. The methods in this dissertation can be extended with uncertainty estimates over dynamics, value

functions, latent skills, and context reliability. Such uncertainty could guide when a policy should act autonomously, when it should replan, and when it should defer to model-based reasoning or human supervision.

Third, stronger sim-to-real and real-world validation is essential. For trajectory autonomy, this means higher-fidelity dynamics, perturbation modeling, uncertainty analysis, and integration with established mission-design pipelines. For embodied control, this means hardware experiments, realistic sensing, contact-rich manipulation, and robustness to execution drift.

Fourth, future systems should more tightly integrate learned policies with model-based physical reasoning. Optimization and control methods remain essential for precision, interpretability, and constraint handling. A promising direction is to use reinforcement learning for fast policy inference and behavior proposal, while using model-based reasoning for verification, correction, and safety-aware refinement.

Finally, the broader direction of embodied foundation models suggests new opportunities for long-horizon physical reasoning. Large sequence models and multimodal policies may provide rich priors for perception, language-conditioned goals, and task knowledge. However, this dissertation suggests that such models will still need physical structure: support awareness, controllable action representations, temporal abstraction, and mechanisms for reasoning under incomplete context.

7.5 Closing Remarks

This dissertation does not claim that one reinforcement learning algorithm solves autonomous physical decision-making. Its contribution is a design principle supported across multiple technical settings: long-horizon physical autonomy requires reinforcement learning methods whose structure reflects the horizon length, feasibility constraints, data support, controllability requirements, and context reliability of the system. When learning is organized around these properties, reinforcement learning becomes a more credible framework for building decision policies that can act over extended horizons in autonomous physical systems.

Bibliography

- [1] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- [2] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning (ICML)*, pages 2052–2062. PMLR, 2019.
- [3] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems (NeurIPS)*, 33:1179–1191, 2020.
- [4] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.
- [5] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning (ICML)*, 2022.
- [6] Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022.
- [7] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- [8] Maryam Norouzi, Mingxi Zhou, and Chengzhi Yuan. A novel adaptive formation control strategy for teams of unmanned vehicles under complete dynamic uncertainty. In *AIAA AVIATION FORUM AND ASCEND 2025*, page 3192, 2025.

- [9] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems (NeurIPS)*, 34: 15084–15097, 2021.
- [10] Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems (NeurIPS)*, 34:1273–1286, 2021.
- [11] Syed Muhammad Talha Zaidi, Pardhasai Chadavada, Hayat Ullah, Arslan Munir, and Atri Dutta. Cascaded deep reinforcement learning-based multi-revolution low-thrust spacecraft orbit-transfer. *IEEE Access*, 2023.
- [12] Atri Dutta, Adrian Arustei, Matthew Chace, Pardhasai Chadavada, James Steck, Talha Zaidi, and Arslan Munir. Machine learning assisted low-thrust orbit-raising: A comparative assessment of a sequential algorithm and deep reinforcement learning approach. In *AIAA SCITECH 2024 Forum*, page 1669, 2024.
- [13] Syed Muhammad Talha Zaidi, Adrian Arustei, Arslan Munir, and Atri Dutta. Automated trajectory planning: A cascaded deep reinforcement learning approach for low-thrust spacecraft orbit-raising. *IEEE Aerospace and Electronic Systems Magazine*, 2025.
- [14] Syed Muhammad Talha Zaidi, Adrian Arustei, Arslan Munir, and Atri Dutta. Single-agent attention actor-critic: A deep reinforcement learning-based solution for low-thrust spacecraft trajectory optimization. *IEEE Transactions on Aerospace and Electronic Systems*, 2025.
- [15] Talha Zaidi, Arslan Munir, and Sardar Ali Abbas. Gralp: A generative representation framework for action refinement and latent planning in offline robotic control. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2026. Accepted.
- [16] Talha Zaidi, Arslan Munir, and Syed Ahasan Masud Zaidi. Crisp: Context-robust skill inference for long-horizon offline robotic control. In *2026 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2026.

- [17] 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. Pmlr, 2018.
- [18] John T Betts, Stephen L Campbell, and Karmethia C Thompson. Solving optimal control problems with control delays using direct transcription. *Applied Numerical Mathematics*, 108:185–203, 2016.
- [19] Craig A Kluever and Steven R Oleson. Direct approach for computing near-optimal low-thrust earth-orbit transfers. *Journal of Spacecraft and Rockets*, 35(4):509–515, 1998.
- [20] Bruce A Conway. *Spacecraft trajectory optimization*, volume 29. Cambridge University Press, 2010.
- [21] Talha Zaidi, Asmaa Samy, Mehmet Kocatürk, and Hasan Fehmi Ateş. Learned vs. hand-crafted features for deep learning based aperiodic laboratory earthquake time-prediction. In *2020 28th signal processing and communications applications conference (SIU)*, pages 1–4. IEEE, 2020.
- [22] Hayat Ullah, Syed Muhammad Talha Zaidi, and Arslan Munir. Improving adversarial robustness through adaptive learning-driven multi-teacher knowledge distillation. *arXiv preprint arXiv:2507.20996*, 2025.
- [23] Syed Ahsan Masud Zaidi, Lior Shamir, William Hsu, Scott Dietrich, and Talha Zaidi. Graze: Grounded refinement and motion-aware zero-shot event localization. *arXiv preprint arXiv:2604.01383*, 2026.
- [24] Muhammad Haris Khan, Syed Muhammad talha Zaidi, Salman Khan, and Fahad Shahbaz Khan. Mode-guided feature augmentation for domain generalization. In *BMVC*, page 176, 2021.
- [25] Suwat Sreesawet and Atri Dutta. Fast and robust computation of low-thrust orbit-raising trajectories. *Journal of Guidance, Control, and Dynamics*, 41(9):1888–1905, 2018.

- [26] Harry Holt, Roberto Armellin, Nicola Baresi, Yoshi Hashida, Andrea Turconi, Andrea Scorsoglio, and Roberto Furfaro. Optimal q-laws via reinforcement learning with guaranteed stability. *Acta astronautica*, 187:511–528, 2021.
- [27] Daniel Miller and Richard Linares. Low-thrust optimal control via reinforcement learning. In *29th AAS/AIAA Space Flight Mechanics Meeting*, volume 168, pages 1817–1834. American Astronautical Society Ka’anapali, Hawaii, 2019.
- [28] Christopher J Sullivan and Natasha Bosanac. Using reinforcement learning to design a low-thrust approach into a periodic orbit in a multi-body system. In *AIAA Scitech 2020 Forum*, page 1914, 2020.
- [29] Hyeokjoon Kwon, Snyoll Oghim, and Hyochoon Bang. Autonomous guidance for multi-revolution low-thrust orbit transfer via reinforcement learning. *AAS 21*, 315, 2021.
- [30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [31] Alexander Mott, Daniel Zoran, Mike Chrzanowski, Daan Wierstra, and Danilo Jimenez Rezende. Towards interpretable reinforcement learning using attention augmented agents. *Advances in neural information processing systems*, 32, 2019.
- [32] Shariq Iqbal and Fei Sha. Actor-attention-critic for multi-agent reinforcement learning. In *International conference on machine learning*, pages 2961–2970. PMLR, 2019.
- [33] Elaheh Barati and Xuewen Chen. An actor-critic-attention mechanism for deep reinforcement learning in multi-view environments. *arXiv preprint arXiv:1907.09466*, 2019.
- [34] Yiying Li, Yongxin Yang, Wei Zhou, and Timothy Hospedales. Feature-critic networks for heterogeneous domain generalization. In *International Conference on Machine Learning*, pages 3915–3924. PMLR, 2019.

- [35] Chengmin Zhou, Bingding Huang, Haseeb Hassan, and Pasi Fränti. Attention-based advantage actor-critic algorithm with prioritized experience replay for complex 2-d robotic motion planning. *Journal of Intelligent Manufacturing*, 34(1):151–180, 2023.
- [36] Shengchao Yan, Lukas König, and Wolfram Burgard. Single-agent actor critic for decentralized cooperative driving. *arXiv preprint arXiv:2403.11914*, 2024.
- [37] Alana de Santana Correia and Esther Luna Colombini. Attention, please! a survey of neural attention models in deep learning. *Artificial Intelligence Review*, 55(8):6037–6124, 2022.
- [38] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems (NeurIPS)*, 34:20132–20145, 2021.
- [39] Wenhao Li, Xiangfeng Wang, Bo Jin, and Hongyuan Zha. Hierarchical diffusion for offline decision making. In *International Conference on Machine Learning (ICML)*, pages 20035–20064. PMLR, 2023.
- [40] Siddarth Venkatraman, Shivesh Khaitan, Ravi Tej Akella, Jeff Dolan, and Glen Berseth. Reasoning with latent diffusion in offline reinforcement learning. *International Conference on Learning Representations (ICLR)*, 2024.
- [41] Jinning Li, Chen Tang, Masayoshi Tomizuka, and Wei Zhan. Hierarchical planning through goal-conditioned offline reinforcement learning. *IEEE Robotics and Automation Letters (RA-L)*, 7(4):10216–10223, 2022.
- [42] Ajay Mandlekar, Fabio Ramos, Byron Boots, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Dieter Fox. Iris: Implicit reinforcement without interaction at scale for learning control from offline robot manipulation data. In *International Conference on Robotics and Automation (ICRA)*, pages 4414–4420, 2020.
- [43] Erick Rosete, Oier Mees, Gabriel Kalweit, Joschka Boedecker, and Wolfram Burgard. Latent plans for task-agnostic offline reinforcement learning. In *Conference on Robot Learning (CoRL)*, pages 1838–1849. PMLR, 2023.

- [44] Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2023.
- [45] Huayu Chen, Cheng Lu, Chengyang Ying, Hang Su, and Jun Zhu. Offline reinforcement learning via high-fidelity generative behavior modeling. *arXiv preprint arXiv:2209.14548*, 2022.
- [46] Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. Idql: Implicit q-learning as an actor-critic method with diffusion policies. *International Conference on Learning Representations (ICLR)*, 2023.
- [47] Cheng Lu, Huayu Chen, Jianfei Chen, Hang Su, Chongxuan Li, and Jun Zhu. Contrastive energy prediction for exact energy-guided diffusion sampling in offline reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 22825–22855. PMLR, 2023.
- [48] C Chi, Z Xu, S Feng, E Cousineau, Y Du, B Burchfiel, R Tedrake, and S Song. Diffusion policy: Visuomotor policy learning via action diffusion. the intl. *Journal of Robotics Research*, 2023.
- [49] Philipp Wu, Arjun Majumdar, Kevin Stone, Yixin Lin, Igor Mordatch, Pieter Abbeel, and Aravind Rajeswaran. Masked trajectory models for prediction, representation, and control. In *International Conference on Machine Learning*, pages 37607–37623. PMLR, 2023.
- [50] J.Fu, A.Kumar, O.Nachum, G.Tucker, S.Levine. D4rl:datasets for deep data-driven reinforcement learning. *preprint arXiv:2004.07219*, 2020.
- [51] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Abhishek Martín-Martín, Kevin Lin, Soroush Maddukuri, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.
- [52] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

- [53] RA Khan, A Asad, MS Jamal, and SA Zaidi. Olympus field development optimization study-kfupm. In *EAGE/TNO Workshop on OLYMPUS Field Development Optimization*, pages cp–562. European Association of Geoscientists & Engineers, 2018.
- [54] Kathryn F Graham and Anil V Rao. Minimum-time trajectory optimization of low-thrust earth-orbit transfers with eclipsing. *Journal of Spacecraft and Rockets*, 53(2):289–303, 2016.
- [55] Craig A Kluever and B Conway. Low-thrust trajectory optimization using orbital averaging and control parameterization. *Spacecraft Trajectory Optimization*, pages 112–138, 2010.
- [56] Robert Falck and John Dankanich. Optimization of low-thrust spiral trajectories by collocation. In *AIAA/AAS Astrodynamics Specialist Conference*, page 4423, 2012.
- [57] Syed Ahsan Masud Zaidi, William Hsu, and Scott Dietrich. Vits for action classification in videos: An approach to risky tackle detection in american football practice videos. *arXiv preprint arXiv:2604.01318*, 2026.
- [58] Jackson L Shannon, Martin T Ozimek, Justin A Atchison, and M Christine. Q-law aided direct trajectory optimization for the high-fidelity, many-revolution low-thrust orbit transfer problem. *Advances in the Astronautical Sciences*, 168:781–800, 2019.
- [59] Sanam Nayab, Sohail Raza Chohan, Aqsa Jameel, Syed Rehan Shah, Syed Ahsan Masud Zaidi, Aditya Nath Jha, and Kamran Siddique. Advancing remote and continuous cardiovascular patient monitoring through a novel and resource-efficient iot-driven framework. *arXiv preprint arXiv:2505.03409*, 2025.
- [60] Harry Holt, Roberto Armellin, Andrea Scorsoglio, and Roberto Furfaro. Low-thrust trajectory design using closed-loop feedback-driven control laws and state-dependent parameters. In *AIAA Scitech 2020 Forum*, page 1694, 2020.
- [61] Adrian Arustei and Atri Dutta. Direct optimization of low-thrust orbit-raising maneuvers using adjoint sensitivities. *Acta Astronautica*, 219:965–981, 2024.

- [62] Sandeep K Singh, Brian D Anderson, Ehsan Taheri, and John L Junkins. Low-thrust transfers to southern 1 2 near-rectilinear halo orbits facilitated by invariant manifolds. *Journal of Optimization Theory and Applications*, 191(2):517–544, 2021.
- [63] Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems (NeurIPS)*, 33:21810–21823, 2020.