Online Active Preference Learning from Sparse Expert Feedback in Continuous Control

by

Aabishkar Timalsina

A thesis accepted and approved in partial fulfillment of the

requirements for the degree of

Master of Science

in Computer Science

Thesis Committee:

Thanh Nguyen, Chair

Jee Choi, Member

University of Oregon

Spring 2026

© 2026 Aabishkar Timalsina
All rights reserved.

THESIS ABSTRACT

Aabishkar Timalsina

Master of Science in Computer Science

Title: Online Active Preference Learning from Sparse Expert Feedback in Continuous Control

In continuous-control reinforcement learning, direct reward design can be difficult when desired behavior is multi-dimensional, and expert feedback is often too limited to query exhaustively. This thesis develops an online active-preference learning framework that combines a fixed expert log-density oracle, a preference-supervised SAC-style training loop, and modular query-acquisition methods. Within this framework, pairwise labels are generated by a frozen SAC expert that ranks candidate actions by their log-density at the queried state, so the supervision reflects state-conditioned behavioral alignment with the expert rather than human preference or reward identification.

The evaluation studies two acquisition families on Hopper-v4: a BADGE-inspired method based on gradient-embedding diversity and a DBAL-style method based on uncertainty and diversity. Under the headline protocol of $K = 8$ preference-pair queries every 100 environment steps over a 1,000,000-step horizon, the BADGE-inspired method outperforms its matched random baseline, with mean return 1897 versus 1593 and a paired bootstrap difference of +304 (95% CI: [86, 476]). The DBAL-style comparison remains inconclusive, with mean return 1708 versus 1693 and a paired difference of +15 (95% CI: [−281, 309]). Mechanism-support analyses suggest that candidate-pool design, multi-anchor querying, and BADGE-specific

disagreement filtering materially affect outcomes within the reported Hopper-v4 configurations.

Taken together, the results support a positive within-family effect for the BADGE-inspired pipeline under the headline protocol, while the DBAL-style effect remains unresolved. The findings are scoped to Hopper-v4 under the reported density-oracle setup and motivate broader validation across environments, oracle types, and query-budget regimes.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Professor Thanh Nguyen, for her guidance, patience, and support throughout this thesis. Her feedback and direction shaped both this work and my experience as a researcher. I am also thankful to Professor Jee Choi for serving on my committee and for his valuable time and perspective.

I am truly grateful to my parents and my brother for their unconditional love, encouragement, and sacrifices. As a first-generation student, I am amazed by their unwavering faith in my pursuit of higher education. I am also thankful to my maternal uncles, who always inspired and encouraged me to pursue higher academic goals.

I am grateful to my friends for their conversations and kindness, which helped me stay grounded throughout graduate school. I am especially grateful to Bibek Gautam and Aayush Dahal for their unwavering compassion and support throughout this journey.

Finally, I am thankful to everyone who contributed, directly or indirectly, to this thesis and to my time at the University of Oregon.

TABLE OF CONTENTS

Chapter Page

I. INTRODUCTION 12

1.1. Chapter Overview 12

1.2. Motivation and Problem Setting 12

1.3. Literature Gap 13

1.4. Research Questions 15

1.5. Contributions 15

1.6. Scope and Caveats 17

1.7. Evaluation Stance 17

1.8. Thesis Map 18

II. RELATED WORK 19

2.1. Chapter Overview 19

2.2. Preference-Based Reinforcement Learning and Feedback 19

2.3. Batch Active Learning and Selective Querying 22

2.4. Actor–Critic Methods for Continuous Control 23

2.5. Synthesis 24

2.6. Positioning and Contribution 25

III. BACKGROUND AND NOTATION 28

3.1. Chapter Overview 28

3.2. Markov Decision Processes and Continuous Control 28

3.3. SAC-Style Actor–Critic Substrate as Implemented 29

3.4. Pairwise Preference Labels 30

3.5. Batch Active Learning Vocabulary 31

Chapter Page

3.6. Notation Summary . . . 33

IV. METHODOLOGY . . . 35

4.1. Overview . . . 35

4.2. The Expert Oracle and Identification . . . 37

4.3. The Shared Training Loop . . . 39

4.4. DBAL-Style Acquisition . . . 41

4.5. BADGE-Inspired Acquisition . . . 43

4.6. Candidate-Pool Construction and Multi-Anchor Querying . . . 46

V. EXPERIMENTS AND RESULTS . . . 49

5.1. Chapter Overview . . . 49

5.2. Evaluation Protocol and Claim Tiers . . . 49

5.2.1. Environment and Training Conditions . . . 49

5.2.2. Performance Metric . . . 50

5.2.3. Primary Comparisons . . . 50

5.2.4. Evidence Tiers . . . 51

5.3. Implementation and Reproducibility . . . 52

5.4. Primary Within-Family Results . . . 53

5.4.1. BADGE-Inspired Active Method vs. Matched Random . . . 53

5.4.2. DBAL-Style Active Method vs. Matched Random . . . 54

5.4.3. Primary Learning Curves . . . 54

5.5. Descriptive Cross-Family Comparison . . . 57

5.6. Mechanism Results . . . 57

5.6.1. Candidate-Pool Composition and Anchor Exposure . . . 58

5.6.2. Disagreement Gate . . . 59

5.7. Summary . . . 61

Chapter Page

VI. DISCUSSION, LIMITATIONS, AND CONCLUSION 62

6.1. Reading the Evidence 62

6.2. What the Supervision Signal Identifies 63

6.3. Why the Primary Contrasts Split Across Families 64

6.3.1. BADGE-Inspired Pipeline 64

6.3.2. DBAL-Style Pipeline 65

6.3.3. Mechanism Primitives and the Shared Scaffold 67

6.3.4. Synthesis 68

6.4. Scope, Limitations, and Validity 69

6.5. Future Work 70

6.6. Conclusion 72

Appendix Overview 74

APPENDICES

A. EVALUATION, REPRODUCIBILITY, AND STATISTICS 75

A.1. Primary-Pair Definitions, Metric, and Curve Policy 75

A.2. Claim-to-Evidence Map 76

A.3. Reproducibility and Artifact Provenance 77

A.3.1. Software Stack and Launchers 77

A.3.2. Log Paths and Primary Aggregates 78

A.3.3. Fair Random-Pool Provenance 78

A.4. Bootstrap Procedure 79

A.5. Mechanism and Gate Reference Tables 79

B. IMPLEMENTATION REFERENCE 81

B.1. SAC-Variant Training Loop 81

B.2. DBAL Acquisition Hyperparameters 82

Chapter Page

B.3. BADGE-Select Procedure and Hyperparameters 82

B.3.1. Hyperparameter Values for the Primary BADGE Row 82

B.4. Candidate-Pool Construction and Schedule Variants 83

B.5. Oracle Log-Density Computation 83

B.6. Oracle Checkpoint Selection 83

REFERENCES CITED 85

LIST OF FIGURES

Figure Page

1. Online active-preference learning framework as a cycle. The learner's current policy π_θ observes a state s_t; a candidate pool of state-conditioned action pairs is constructed at that anchor; the acquisition criterion selects a batch $\mathcal{B}$ of K pairs; the frozen expert oracle π^E (gray fill) returns binary preference labels by log-density comparison; labels accumulate in the preference buffer $\mathcal{D}_{\text{pref}}$; preference-supervised SAC-style updates use minibatches from the buffer to update π_θ. Rounded rectangles are processes; gray fill marks external or stored components. 37

2. Hopper-v4 learning curves for the BADGE-inspired primary pair under the headline protocol (K=8, query interval 100 steps, H=1,000,000, n=10 seeds). Solid lines show the mean across seeds; shaded bands show ±1 standard deviation. Curves are interpolated onto a fixed 10,000-step grid using last-observation carry-forward. 55

3. Hopper-v4 learning curves for the DBAL-style primary pair under the headline protocol (K=8, query interval 100 steps, H=1,000,000, n=10 seeds). Solid lines show the mean across seeds; shaded bands show ±1 standard deviation. Curves are interpolated onto a fixed 10,000-step grid using last-observation carry-forward. 56

LIST OF TABLES

Table Page

1. Adjacent related-work settings versus the setting studied in this thesis. Rows highlight design axes that are especially important for interpreting the methods and experiments that follow. . . . 27

2. Notation for Chapters IV and V, defined in this chapter. 34

3. Designated primary within-family comparisons. Both pairs used $K = 8$ preference pairs per round, 100-step query interval, a 1,000,000-step training horizon, and seeds 1–10. 51

4. Evidence classes and their inferential role. 52

5. Primary within-family results (headline protocol). 53

6. Mechanism-support configurations. 58

7. Disagreement-gate 2×2 mean returns. 60

8. Designated primary within-family comparisons (Hopper-v4, headline protocol). Row IDs are the `ABLATION_SLUG` identifiers used in the experiment configuration. 75

9. Claim-to-evidence map. † = designated primary row. 77

10. Designated matched-random primary configurations (headline protocol). . 78

11. Primary row aggregates (mean of per-seed means). 78

12. Mechanism-support rows (10^6 steps, ten seeds). † = headline primary at K=8, AL=100. 79

13. Disagreement-gate 2×2 (BADGE). All cells at K=8, AL=100. 80

14. Shared training-loop hyperparameters (primary rows). 81

15. DBAL-style acquisition hyperparameters (primary row). 82

16. BADGE-Select hyperparameters (primary row). 83

17. Oracle selection (Hopper-v4 expert runs, last-100 episode means). 84

CHAPTER I

INTRODUCTION

1.1 Chapter Overview

This thesis develops an online preference-learning framework for continuous control and evaluates two acquisition families inside that framework on the Hopper-v4 locomotion task. Section 1.2 motivates the problem from reinforcement learning through preference-based supervision to the Hopper-v4 density-oracle setting. Section 1.3 states the literature gap and points to Chapter II for the full contrast. Section 1.4 states the three research questions and their evidence tiers. Section 1.5 summarizes the methodological, empirical, and engineering contributions. Section 1.6 records the preference-supervised actor–critic caveat, empirical bounds, and non-goals. Section 1.7 states the evaluation metric, inferential procedure, and appendix claim-to-evidence map. Section 1.8 maps the remaining chapters.

1.2 Motivation and Problem Setting

In the reinforcement learning (RL) paradigm, an agent learns by interacting with an environment, taking actions, and receiving feedback about the consequences of those actions (Sutton & Barto, 2018). Through this process, the agent improves a control policy toward behavior that achieves high return over the long term. The feedback is generally provided by a scalar reward function. In continuous-control tasks, where desired behavior may depend on competing qualities such as balance, smoothness, progress, and stability, designing a reward function can be difficult. In such settings, it may be easier to recognize good behavior than to encode it as a numeric reward.

Preference-based reinforcement learning offers an alternative form of supervision. Instead of requiring a hand-engineered reward for every state–action pair, the learner

receives comparative feedback: given two alternatives, a teacher or expert indicates which one is better (Christiano et al., 2017; Wirth, Akrour, Neumann, & Fürnkranz, 2017). This shifts the problem from designing a reward function to collecting useful comparisons. However, it does not remove the cost of supervision. The agent still must decide which comparisons are worth querying.

That decision is crucial in the online setting studied in this thesis, where expert feedback is limited and the agent cannot query every possible comparison it encounters during training. Comparison labels are requested during training rather than collected once in a fixed offline dataset. The central question is therefore how to spend a finite query budget during learning: which candidate comparisons should be queried at each round? This thesis holds the query schedule fixed at a designated headline protocol and treats the acquisition rule that selects comparisons within that schedule as the object of study.

The concrete setting is online preference learning on Hopper-v4, a continuous-control locomotion task, under a density-based oracle: a frozen soft actor–critic (SAC) expert ranks candidate action pairs by log-density, not human judgment. These labels encourage behavioral alignment with the expert's action likelihood rather than human preference, trajectory-level success, or direct reward identification. Chapter IV specifies the shared training loop and the two acquisition families; Section 1.4 states the empirical questions.

1.3 Literature Gap

The closest adjacent literatures are preference-based reinforcement learning, batch active learning, and entropy-regularized actor–critic methods for continuous control. Preference-based RL provides the supervision and has shown that comparative feedback can train policies (Christiano et al., 2017; Wirth et al., 2017);

however, the literature is dominated by reward-model-based learning anchored on trajectory segments, rather than on state-conditioned action pairs on a fixed query schedule during training. Active preference learning has also been developed for reward function learning from human comparisons of trajectories, including in the batch setting (Biyik & Sadigh, 2018; Sadigh, Dragan, Sastry, & Seshia, 2017). In these literatures, query selection is usually organized around reward function learning. Consequently, the policy is improved based on the learned reward model or preference distribution. This thesis instead combines query selection from state-conditioned action pairs and policy update inside one online loop through preference-supervised critics without an intermediate reward model.

Batch active learning supplies selection principles for choosing informative and non-redundant queries, including uncertainty-then-diversity shortlisting and gradient-embedding batch selection (Ash, Zhang, Krishnamurthy, Langford, & Agarwal, 2020; Zhdanov, 2019). However, these methods were developed and implemented in pool-based supervised classification tasks with vector inputs and externally supplied labels, not state-conditioned action pairs with a non-human oracle. SAC is widely used for locomotion-scale continuous control (Haarnoja, Zhou, Abbeel, & Levine, 2018). In the standard SAC algorithm, critics learn using TD targets on scalar rewards, which is not the source of supervision used here.

Therefore, the combination studied here is not covered by any of these literatures on its own. This thesis integrates batched action-pair acquisition, log-density preference supervision, and SAC-style policy learning in one online loop. Chapter II contrasts that setting with prior work in detail.

1.4 Research Questions

The thesis is guided by three research questions that provide the structure used to connect the framework, the evaluation protocol, and the evidence reported in Chapter V. The first question defines the primary within-family evaluation, while the remaining questions organize descriptive and mechanism-support analyses.

RQ1. Under the matched headline protocol on Hopper-v4, does each acquisition family's active rule outperform its matched random baseline under the same family-specific candidate exposure?

RQ2. Within each acquisition family, which framework primitives, including candidate-pool composition, filtering or shortlist choices, diversity mechanisms, anchor strategy, and schedules, appear to affect outcomes under the shared scaffold?

RQ3. Under the shared scaffold at the headline budget, what is the descriptive ordering of BADGE-inspired and DBAL-style acquisition?

RQ1 is treated as the primary inferential question and is evaluated with paired within-family active-versus-matched-random contrasts over the shared seed set. RQ2 is treated as a mechanism-support question, and RQ3 is descriptive only because the two acquisition families use family-specific matched random baselines rather than a paired cross-family experimental design. Section 1.7 states these evidence tiers more explicitly.

1.5 Contributions

(1) Methodological. The thesis develops an online preference-learning framework for continuous control. The framework fixes a query cadence inside the training loop, selects state-conditioned action pairs at each round through an active

batched acquisition rule, supplies labels from a fixed expert log-density oracle, and updates a SAC-style actor mediated by preference-supervised critics. The framework is instantiated with two parallel acquisition families, a BADGE-inspired family that adapts gradient-embedding batch selection and a DBAL-style family that adapts uncertainty-then-diversity shortlisting, under one shared training and query interface. Chapter IV specifies the oracle, the preference-supervised training loop, and the two acquisition pipelines.

(2) Empirical. The thesis reports multi-seed Hopper-v4 evidence under designated within-family primary contrasts at the headline protocol. The BADGE-inspired active method outperforms its matched random baseline, with a paired bootstrap confidence interval that excludes zero. The DBAL-style active method shows no detectable difference from its matched random baseline, with a paired bootstrap confidence interval that spans zero by a wide margin, so the DBAL-style result is treated as inconclusive rather than as evidence of a null effect. The cross-family ordering of the two active methods is descriptive only and is not supported by a paired test. Chapter V reports the headline-protocol parameters, the full numerical results, the learning curves, and the mechanism-support contrasts.

(3) Engineering. The thesis treats candidate-pool composition, anchor strategy, and a BADGE-specific disagreement gate as framework primitives inside the shared online training loop, rather than as auxiliary implementation details. Pool and anchor settings are shared across both acquisition families; the disagreement gate is specific to the BADGE-inspired path. Chapter IV specifies how each primitive enters pool construction and query issuance.

1.6 Scope and Caveats

The preference-supervised critic path is non-Bellman. The critics are trained on pairwise preference labels rather than on temporal-difference targets on scalar rewards, so the learned values are preference-trained quantities and not reward-identified action values. The actor retains the SAC entropy-regularized form and remains Q-mediated through the minimum of the two preference-supervised critics. Chapter IV specifies this deviation in full.

Primary inferential claims are restricted to Hopper-v4 under the designated headline protocol. The thesis is not a new SAC algorithm, not a supervised batch active-learning method in the classical pool-based sense, and not a human-preference or reinforcement-learning-from-human-feedback pipeline (Ouyang et al., 2022). Generalization beyond the reported Hopper-v4 configurations is treated as future work unless separately validated.

1.7 Evaluation Stance

The evaluation stance distinguishes three tiers of evidence, corresponding to RQ1–RQ3 in Section 1.4. The full contract (primary row IDs, metric definition, bootstrap settings, and the claim-to-evidence map) is recorded in Appendix A (Section A.1 and Table 9). Within-family active-versus-matched-random contrasts under the headline protocol support RQ1 and are interpreted through paired bootstrap confidence intervals over the ten shared seeds (procedure in Appendix A.4, Section A.4). Mechanism analyses for RQ2 are descriptive: each contrast holds the surrounding configuration fixed, and some rows use an alternative query schedule to check sensitivity away from the headline protocol. Cross-family comparisons for RQ3 remain at the descriptive tier. Mean episodic return over each seed's final 100 completed training episodes serves as the external evaluation metric, with the

interquartile mean across seeds reported as a robustness check (Agarwal, Schwarzer, Castro, Courville, & Bellemare, 2021). Each contribution in this chapter is tied to a downstream artifact in Chapter V (matching aggregate-table rows and bootstrap output) and to the corresponding rows in Appendix A, Table 9. Implementation and run provenance appear in Appendices B and A.3. Contributions are stated only at the tier supported by the corresponding evidence.

1.8 Thesis Map

The remainder of the thesis develops the framework and reports the Hopper-v4 evaluation in five chapters. Chapter II reviews preference-based reinforcement learning, batch active learning, and SAC for continuous control, and positions the thesis setting against that work. Chapter III defines the notation for the MDP, SAC-style actor–critic framework, preference labels, and batch acquisition vocabulary. Chapter IV specifies the log-density oracle, the shared preference-supervised training loop, the two acquisition families, and the pool-and-anchor primitives they share. Chapter V reports the headline within-family results, descriptive cross-family comparisons, and mechanism-support ablations on Hopper-v4. Chapter VI interprets the evidence, states limitations, and outlines future work. Appendix 6.6 records the evaluation contract, reproducibility audit trail, and supporting tables for Chapters IV–V.

CHAPTER II

RELATED WORK

2.1 Chapter Overview

This chapter explains where the thesis fits in the existing literature. The closest points of contact are preference-based reinforcement learning, batch active learning, and entropy-regularized actor–critic methods for continuous control. Each addresses part of learning under limited supervision, but none of them by itself specifies the full setting studied here. Section 2.2 reviews preference-based RL and feedback; Section 2.3 reviews batch active learning and selective querying; and Section 2.4 reviews soft actor–critic as the algorithmic basis of both the preference-trained learner and the frozen expert policy that supplies labels. Section 2.5 then contrasts the thesis setting with prior work along axes that matter for interpreting the methods and experiments that follow. Finally, Section 2.6 states the contribution as a specific combination of acquisition, supervision, and SAC-style learning under a shared training and query interface.

2.2 Preference-Based Reinforcement Learning and Feedback

Reinforcement learning traditionally assumes a hand-specified reward function, but reward design is difficult when objectives are multi-dimensional or hard to encode numerically. Preference-based reinforcement learning (PbRL) instead learns from comparative feedback about which behaviors are preferred (Wirth et al., 2017). A common PbRL protocol presents two alternatives, records which one the teacher ranks higher, and uses those labels to train a policy or an intermediate reward model. A survey of preference-based methods (Wirth et al., 2017) organizes design axes, including the type of preference feedback, the representation used to capture preferences, and how exploration interacts with query collection. Those axes are useful

background because the present thesis also depends on pairwise comparisons, but it fixes a different label source and query object than most PbRL systems discussed there.

Deep RL from human preferences (Christiano et al., 2017) showed how preference feedback could be incorporated into deep RL by eliciting human preferences over short trajectory segments, fitting a reward model to the observed comparisons, and optimizing a policy against the learned reward. The teacher provides feedback on only a small fraction of agent interactions, which makes the approach practical for complex control tasks, but the supervision is still organized around trajectory-level clips and routed through an explicit reward model before policy improvement. Later work in continuous control extended the same broad pattern while improving sample and feedback efficiency. PEBBLE (Lee, Smith, & Abbeel, 2021) trains an off-policy agent from actively queried preferences, relabels past experience when the reward model changes, and uses unsupervised pre-training so that queries cover a broader state distribution. That line of work demonstrates that preference supervision can support locomotion-scale tasks, yet the supervision is still mediated through a learned reward model and the primary query unit remains behavior segments rather than state-conditioned action pairs issued on a fixed schedule during online training.

Benchmarking work has made query design an explicit object of study within PbRL rather than an implementation detail. B-Pref (Lee, Smith, Dragan, & Abbeel, 2021) introduces simulated teachers with varied irrationalities and evaluates algorithms partly on how well they choose informative preference queries under those teachers. The benchmark therefore treats selective querying as part of the PbRL problem, but it still evaluates reward-model-based pipelines on locomotion and manipulation tasks with teacher semantics tied to underlying task reward or

structured irrationalities, not to a fixed expert log-density oracle on state-conditioned action pairs.

Active preference learning studies which comparisons to ask, not only how to learn from a fixed query log. Active preference-based reward learning (Sadigh et al., 2017) synthesizes trajectory comparisons in shared scenarios and updates a distribution over reward-function parameters from the human response. The queries are actively constructed, but they remain trajectory-level and aimed at learning a reward function that explains human choices in a dynamical system. Batch active preference-based reward learning (Biyik & Sadigh, 2018) extends that line by batching preference queries over reward functions so that each round selects a set of informative comparisons while keeping query-generation cost manageable. This batch active preference-learning setting is one of the closest PbRL neighbors to the present thesis in its emphasis on query budgets, yet it still targets reward-function learning from human preferences in robotics tasks rather than periodic batches of state-conditioned action pairs labeled by a non-human oracle inside a shared training loop.

Large-scale alignment pipelines also use human preference data, but they address a different supervision contract. Instruction-following language-model training (Ouyang et al., 2022) applies human preference labels over model outputs within an RLHF-style pipeline. That work underscores the practical importance of preference supervision, yet it is not a continuous-control study, does not use batched action-pair queries at a fixed environment-step cadence, and does not provide the evaluation setup used here. The oracle in Chapter IV is a fixed expert log-density rule, so labels reflect behavioral alignment with a pretrained policy rather than human judgments or reward-model fitting.

2.3 Batch Active Learning and Selective Querying

A separate literature studies which unlabeled items to label when annotation budget is limited. Pool-based active learning maintains a candidate set, scores items for informativeness, and queries labels for a subset (Settles, 2009). Batch mode is important in modern deep learning because labels are requested in groups rather than one point at a time (Hoi, Jin, Zhu, & Lyu, 2006). The batch setting introduces a second design question beyond informativeness, namely how to avoid redundant queries that all probe the same region of the input space.

Diverse mini-batch active learning (DBAL) (Zhdanov, 2019) pre-filters examples near the classifier decision boundary and then selects a diverse batch by clustering examples within the shortlist. The abbreviation *DBAL* in this thesis refers throughout to Zhdanov (2019); in parts of the deep active-learning literature the same abbreviation is used for the deep Bayesian active learning approach of Gal, Islam, and Ghahramani (2017), which is a separate method. The method is defined for supervised classification over a pool of feature vectors, and the original presentation does not place it inside an online reinforcement-learning loop with state-conditioned action pairs. Chapter IV implements a DBAL-*style* pipeline in that RL setting, so the relevant prior work is the uncertainty-then-diversity pattern rather than a direct transfer of the original task definition.

Batch active learning by diverse gradient embeddings (BADGE) (Ash et al., 2020) forms gradient-based embeddings from provisional class labels and selects a batch with k-means++ seeding in the embedding space. The construction is designed to combine uncertainty with diversity within a single batch. Like DBAL, BADGE was developed for pool-based classification with vector inputs and human- or ground-truth labels supplied after selection. The present thesis uses a BADGE-*inspired* acquisition

path in which embeddings are built from preference-trained critics, oracle labels are applied only after selection, and candidates are state-conditioned action pairs rather than static examples.

Other batch scorers pursue a similar goal with different machinery. BatchBALD (Kirsch, van Amersfoort, & Gal, 2019) approximates mutual information between a batch of points and model parameters for deep Bayesian active learning. The details differ from both DBAL and BADGE, but the shared motivation is that greedy one-at-a-time acquisition can send redundant queries when labels are requested in parallel.

These batch active-learning methods supply selection principles for the acquisition modules in Chapter IV, yet they do not define the full training and query design evaluated in this thesis. Section 2.5 summarizes how the adjacent literatures in Sections 2.2 and 2.3 relate to that setup; Section 2.4 below reviews the continuous-control optimizer context.

2.4 Actor–Critic Methods for Continuous Control

The training procedure shared by both acquisition families in this thesis is based on soft actor–critic (SAC) (Haarnoja et al., 2018). SAC is an off-policy entropy-regularized actor–critic algorithm for continuous action spaces, with two critics updated against TD targets on scalar rewards and a stochastic actor that maximizes expected return regularized by policy entropy. The maximum-entropy objective encourages exploration and is commonly used for stable learning in continuous control without requiring on-policy rollouts, which makes SAC a standard reference point for locomotion tasks of the kind studied in Chapter V.

This thesis adopts the SAC family as a substrate but replaces Bellman critic supervision with pairwise preference labels from the oracle; the actor retains

the entropy-regularized SAC form evaluated against those critics (Section 3.3; Chapter IV).

Pretrained SAC policies also play a second role in the framework. The oracle is a fixed SAC-trained expert policy whose log-density supplies the preference labels formalized in Chapter IV. SAC is therefore neither the source of supervision nor the preference-learning algorithm being studied, but the actor–critic substrate shared by the preference-trained learner and the frozen expert policy that labels its queries.

2.5 Synthesis

Preference-based reinforcement learning, batch active learning, and entropy-regularized actor–critic methods address overlapping parts of learning under limited supervision, but they are usually developed with different query objects, label sources, and evaluation conventions. Table 1 summarizes how the present problem differs from prior settings along axes that matter for interpreting the methods and experiments that follow.

Taken together, these literatures motivate the present setting but do not fully determine the combination studied in this thesis. Prior preference-based and active preference-learning work provides the foundations for preference supervision and query design, while batch active learning provides uncertainty–diversity and gradient-embedding selection principles from supervised pool-based settings. The remaining chapters use this distinction to define the thesis setup more precisely. Training uses periodic batched queries over state-conditioned action pairs in continuous control, labels each query with a fixed expert log-density oracle, updates preference-supervised critics rather than reward-identified Bellman targets, and evaluates acquisition through within-family active versus matched-random comparisons. These comparisons are conducted under a shared pool-and-anchor scaffold that serves as the

common training and query interface developed in later chapters. Chapter IV specifies that scaffold and the two adapted acquisition pipelines; Chapter V reports designated within-family tests and mechanism-support contrasts on Hopper-v4 without treating cross-family ordering as a paired confirmatory claim.

2.6 Positioning and Contribution

Therefore, this thesis contributes an online preference-learning framework for continuous control that combines periodic batched active pair acquisition, preference labels from a fixed non-human expert log-density oracle, and SAC-style actor updates mediated by preference-supervised critics. The framework instantiates this interface through two parallel acquisition families, BADGE-inspired and DBAL-style, and evaluates each family against a matched random baseline drawn from the same family-specific exposed pool or shortlist.

The contribution lies in the combined interface and in the explicit treatment of candidate-pool composition, filtering, anchor strategy, and policy-mix design as framework primitives rather than incidental implementation details. It also lies in the designated within-family active-versus-matched-random evaluation on Hopper-v4 under the headline protocol, with fixed query budget, query interval, training horizon, and seed set. This evaluation contract supports interpretable within-family inference for each acquisition family while leaving cross-family comparisons descriptive.

The contribution is not a new soft actor–critic algorithm, a new batch active-learning algorithm in the supervised pool-based sense, or a claim of standard human-feedback RLHF semantics. Density labels encourage behavioral alignment with the expert's local action likelihood rather than identifying a return-maximizing reward function, and broader generalization beyond the designated Hopper-v4 protocol is left to future work.

Chapter IV specifies the oracle, the preference-supervised training loop, and the two adapted acquisition pipelines. Chapter V reports designated within-family comparisons and mechanism-support contrasts under the headline protocol on Hopper-v4, and Chapter VI interprets those results within the scope established here.

Table 1. Adjacent related-work settings versus the setting studied in this thesis. Rows highlight design axes that are especially important for interpreting the methods and experiments that follow.

Axis	Adjacent related-work settings	This thesis
Label source	Human preference labels, simulated preference teachers, or reward-derived supervision used to train reward models	Fixed expert log-density oracle (behavioral alignment, not human RLHF semantics)
Query object	Trajectory segments, behavior clips, synthesized rollouts, or reward-function queries	State-conditioned *action pairs* queried on a fixed schedule during training
Acquisition	Reward-model uncertainty, active reward-function learning, or batch query selection in task-specific preference settings	BADGE-inspired and DBAL-style batch rules under one shared training procedure
Pipeline coupling	Query selection, labeling, and policy update typically organized as largely separate stages; policy improvement mediated by a fitted reward model or preference distribution	One online training loop integrating query selection, labeling, and policy update; preference-supervised critics updated directly without an intermediate reward model
Random baseline	Varies by study; random comparators are not always defined around the same exposed candidate pool	Matched random baseline within each family (same pool and anchor exposure, family-specific shortlist rules)
Inferential focus	Single algorithm, benchmark sweep, or cross-method leaderboard	Designated within-family active versus matched-random contrasts at a fixed K, query interval, and horizon
Pool and anchors	Often implicit in the query distribution	Explicit framework primitives (mixed versus uniform-only pools, single-anchor versus multi-anchor querying, family-specific shortlists and gates)

CHAPTER III

BACKGROUND AND NOTATION

3.1 Chapter Overview

This chapter defines the notation for the methods and experiments chapters. Section 3.2 defines the Markov decision process for the continuous-control environment and the external evaluation metric. Section 3.3 presents the SAC-style actor–critic substrate together with the non-Bellman caveat on the critic supervision path. Section 3.4 defines pairwise preference labels and the preference buffer that stores them. Section 3.5 states the batch active-learning vocabulary for the acquisition methods, including anchor and pool-composition terms. Section 3.6 consolidates the symbols defined in the preceding sections as a single reference table.

The chapter restricts itself to symbols and conventions that appear in the subsequent chapters. Identification claims about the supervision signal, full update rules, and the family-specific acquisition algorithms are specified in Chapter IV. Headline-protocol parameter values are specified in Chapter V.

3.2 Markov Decision Processes and Continuous Control

The environment is modeled as a Markov decision process $(\mathcal{S}, \mathcal{A}, P, r, \gamma, \rho_0)$, where $\mathcal{S} \subseteq \mathbb{R}^{d_s}$ is a continuous state space, $\mathcal{A} \subseteq \mathbb{R}^{d_a}$ is a bounded continuous action space, $P(s' \mid s, a)$ is the transition kernel, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the environment reward function, $\gamma \in [0, 1)$ is a discount factor, and ρ_0 is the initial-state distribution. At each step the agent observes a state $s_t \in \mathcal{S}$, selects an action $a_t \in \mathcal{A}$ according to a stochastic policy $\pi(a \mid s)$, transitions to $s_{t+1} \sim P(\cdot \mid s_t, a_t)$, and the environment emits $r(s_t, a_t)$.

The training horizon H is the total number of environment steps in a single training run. Within H the agent generates a sequence of episodes $\tau =$

$(s_0, a_0, s_1, a_1, \ldots, s_{T-1}, a_{T-1})$ that terminate either when the environment signals episode completion or when H is reached. The undiscounted episodic return of an episode τ of length T is

$$G_\tau \;=\; \sum_{t=0}^{T-1} r(s_t, a_t). \tag{3.1}$$

The environment reward r defines the task's external evaluation criterion only. The supervision signal for policy training in this thesis is not r; it is the pairwise preference label introduced in Section 3.4. The evaluation metric reported in Chapter V is the mean of per-seed episodic returns over each seed's final 100 completed training episodes, with the interquartile mean across seeds reported as a robustness check (Agarwal et al., 2021). The continuous-control environment throughout the thesis is Hopper-v4 from the MuJoCo locomotion suite (Todorov, Erez, & Tassa, 2012), accessed through Gymnasium (Towers et al., 2023).

3.3 SAC-Style Actor–Critic Substrate as Implemented

The training procedure in this thesis follows the soft actor–critic (SAC) family of entropy-regularized actor–critic algorithms for continuous control (Haarnoja et al., 2018). Following SAC, the framework instantiates a stochastic actor π, two critic networks Q_1, Q_2, and an entropy coefficient α that weights the policy entropy term in the actor objective. The actor is a squashed-Gaussian policy whose log-density is defined on the bounded continuous action space and includes the change-of-variables correction for the squashing transform. The twin-critic construction is retained to supply a conservative score in the actor's policy-improvement step.

In a standard SAC implementation, both critics are supervised by temporal-difference (TD) targets formed from the environment reward r and a target-network evaluation of the next state, and the actor maximizes an entropy-regularized objective evaluated against the minimum of the two critics.

Non-Bellman caveat on the critic supervision path. The preference-loss critic path is non-Bellman; the actor remains Q-mediated SAC-style. The critics are supervised by pairwise preference labels rather than by TD targets on r, so the learned critic values are preference-trained quantities and not reward-identified action values. The actor objective retains the SAC entropy-regularized form evaluated against the minimum of the two critics, so policy improvement remains Q-mediated even though the critics themselves are not Bellman-targeted. The exact preference loss, the actor objective, and the optimization schedule are specified in Chapter IV.

3.4 Pairwise Preference Labels

The unit of supervision is a state-conditioned action pair. A query has the form (s, a_1, a_2), where $s \in \mathcal{S}$ is an anchor state and $a_1, a_2 \in \mathcal{A}$ are two candidate actions evaluated at that state. The query is answered by a binary preference label

$$y \in \{0, 1\}, \tag{3.2}$$

with the convention that $y = 1$ indicates the labeler prefers a_1 to a_2 at s and $y = 0$ indicates the reverse. The label source is denoted π^E and is fixed throughout training; its form and the rule it uses to assign labels are specified in Chapter IV.

Each labeled tuple (s, a_1, a_2, y) is stored together with its mirror $(s, a_2, a_1, 1-y)$, which enforces antisymmetry of the label under exchange of the two actions. The collection of labeled tuples accumulated during training is the preference buffer

$$\mathcal{D}_{\text{pref}} = \left\{ \left(s^{(k)}, a_1^{(k)}, a_2^{(k)}, y^{(k)}\right) \right\}_{k=1}^{N_{\text{pref}}}, \tag{3.3}$$

where N_{pref} is the number of labeled tuples in the buffer. The buffer serves as the data source for both critic and actor optimization in Chapter IV.

The state-conditioned action-pair convention used here differs from the trajectory-segment convention common in preference-based reinforcement learning

(Christiano et al., 2017; Lee, Smith, & Abbeel, 2021), in which the query unit is a pair of short trajectory segments and the label is a comparison of cumulative behavior over those segments. Both conventions produce binary labels of the form in Equation (3.2); the choice between them is a modeling decision about what an individual label identifies.

3.5 Batch Active Learning Vocabulary

The acquisition methods studied here are instances of batch active learning. At each active-learning round, the framework constructs a candidate pool of possible queries, selects a batch of K queries from that pool according to an acquisition rule, and submits the selected batch for labeling. The candidate pool is the set of candidate query objects (s, a_1, a_2) exposed to the acquisition rule at a given round. The query budget K fixes the number of queries selected per round. The query interval is the number of environment steps between consecutive active-learning rounds. Specific values for K, the query interval, and the training horizon H are specified in Chapter V.

Within each round, queries are anchored at one or more states drawn from the agent's interaction with the environment. An anchor state is a state s at which the candidate pairs of that round are constructed. The framework supports two anchor strategies. In single-anchor mode, every candidate pair shares the agent's current environment observation, called the frontier anchor. In multi-anchor mode, the frontier anchor is supplemented with additional anchors drawn from a short window of recent observations. Their exact mechanics, including the split of K across anchors, are specified in Chapter IV.

The candidate pool at each anchor state is constructed from source-consistent action pairs, meaning that both actions in a pair are drawn from the same source. Two pool-composition variants appear in the experimental setup. A uniform-only

pool draws both actions uniformly from the bounded action space. A mixed policy-and-uniform pool, or mixed pool, draws some pairs from the current policy and the remainder from the uniform sampler. The mixture rule is specified in Chapter IV.

Batch active learning rules in the supervised setting typically combine two principles (Ash et al., 2020; Zhdanov, 2019). Uncertainty selects queries on which the current model is least confident, on the grounds that such queries are most informative about the model's parameters or predictions. Diversity selects queries that are non-redundant under a representation of the query objects, since a batch of mutually similar queries provides less information than a batch that covers different regions of the representation space. The two principles are combined in either a sequential arrangement, in which one principle filters the pool before the other selects from the remainder, or in a joint embedding-space arrangement, in which both principles are expressed through a single batch-selection rule. The family-specific realizations are an uncertainty-then-diversity rule and a gradient-embedding rule, both specified in Chapter IV.

The canonical control for any active acquisition rule in the thesis is the matched random baseline, defined as uniform random selection of K queries from the same family-specific candidate exposure that the active rule sees. The two acquisition families instantiate that exposure differently: the DBAL-style random baseline samples uniformly from the per-anchor candidate pool, while the BADGE-inspired random baseline samples uniformly from the post-prefilter shortlist that would otherwise be passed to embedding-space selection. The full specification of these per-family exposures is given in Chapter IV. The combination of K, query interval, H, anchor strategy, pool composition, and seed count that defines the primary within-

family contrasts is referred to as the headline protocol, whose parameter values are specified in Chapter V.

3.6 Notation Summary

Table 2 consolidates the symbols introduced in the preceding sections. The third column indicates where each symbol is defined.

Table 2. Notation for Chapters IV and V, defined in this chapter.

Symbol	Description	Defined in
$\mathcal{S}, \mathcal{A}$	State and action spaces	§3.2
s, a	A state and a single action	§3.2
P, r, γ, ρ_0	Transition kernel, reward, discount, initial-state distribution	§3.2
H	Training horizon, in environment steps	§3.2, §3.5
τ, T	A single episode and its length	§3.2
G_τ	Undiscounted episodic return of episode τ	§3.2
$\pi(a \mid s)$	Stochastic policy (learner)	§3.3
Q_1, Q_2	Twin preference-supervised critics	§3.3
α	SAC entropy coefficient	§3.3
(s, a_1, a_2)	State-conditioned action-pair query	§3.4
y	Binary preference label, $y \in \{0, 1\}$	§3.4
π^E	Label source (frozen expert policy)	§3.4
$\mathcal{D}_{\text{pref}}$	Preference buffer of labeled tuples	§3.4
N_{pref}	Number of labeled tuples in $\mathcal{D}_{\text{pref}}$	§3.4
K	Number of queries selected per active-learning round	§3.5
Query interval	Environment steps between consecutive rounds	§3.5
Candidate pool	Set of (s, a_1, a_2) exposed to the acquisition rule	§3.5
Anchor state	State at which a round's candidate pairs are constructed	§3.5
Frontier anchor	Agent's current observation at the round trigger	§3.5
Mixed pool	Candidate pool with policy and uniform action pairs	§3.5
Uniform-only pool	Candidate pool with uniform action pairs only	§3.5
Matched random baseline	Uniform selection from the same family-specific candidate exposure	§3.5
Headline protocol	Designated K, query interval, H, anchor, pool, seeds	§3.5

CHAPTER IV

METHODOLOGY

4.1 Overview

This chapter presents the online active-preference learning framework developed for the continuous-control setting and describes the design choices evaluated in the subsequent experiments. The framework combines a fixed expert oracle, a SAC-style training procedure, and a periodic active query mechanism into a unified system. At each query step, the method selects a batch of action pairs according to its acquisition criterion, obtains pairwise preference labels from the oracle, stores these labels in a preference buffer, and updates the policy using the accumulated preference data. Because the oracle, training procedure, and candidate pool are held constant across experimental conditions, the within-family comparisons in Chapter 5 remain interpretable: observed differences in performance can be attributed to the active learning strategy rather than to variations in the surrounding experimental setup.

Two methodological choices are central to how results throughout this work should be read, and are stated here before the technical specification begins. First, the oracle generates preference labels from the log-density of a pretrained expert policy rather than from human annotations. Consequently, the policy is trained toward behavioral alignment with the expert's action likelihood rather than toward reward maximization, a distinction defined precisely in Section 4.2. Second, although the training loop is a SAC-variant, it does not rely on temporal-difference bootstrapping to supervise the critics. The nature of this deviation from standard SAC, and its implications for what the critics represent, are clarified in Section 4.3.

Within this fixed scaffold, the work studies two acquisition strategies: one based on uncertainty scoring and action-feature diversity, and the other on gradient-

embedding clustering. Each strategy is evaluated against a matched random baseline drawn from the same family-specific candidate exposure. That exposure is shaped by design decisions about how candidate pairs are constructed, filtered, and associated with states in the replay buffer; these choices proved consequential enough in practice to treat as substantive elements of the methodology rather than as implementation details. Section 4.6 specifies them in full.

The chapter proceeds from the oracle through the training loop to the two acquisition strategies and their surrounding components, in the order that reflects how each piece depends on the one before it. Figure 1 summarizes the framework before the technical specification begins.

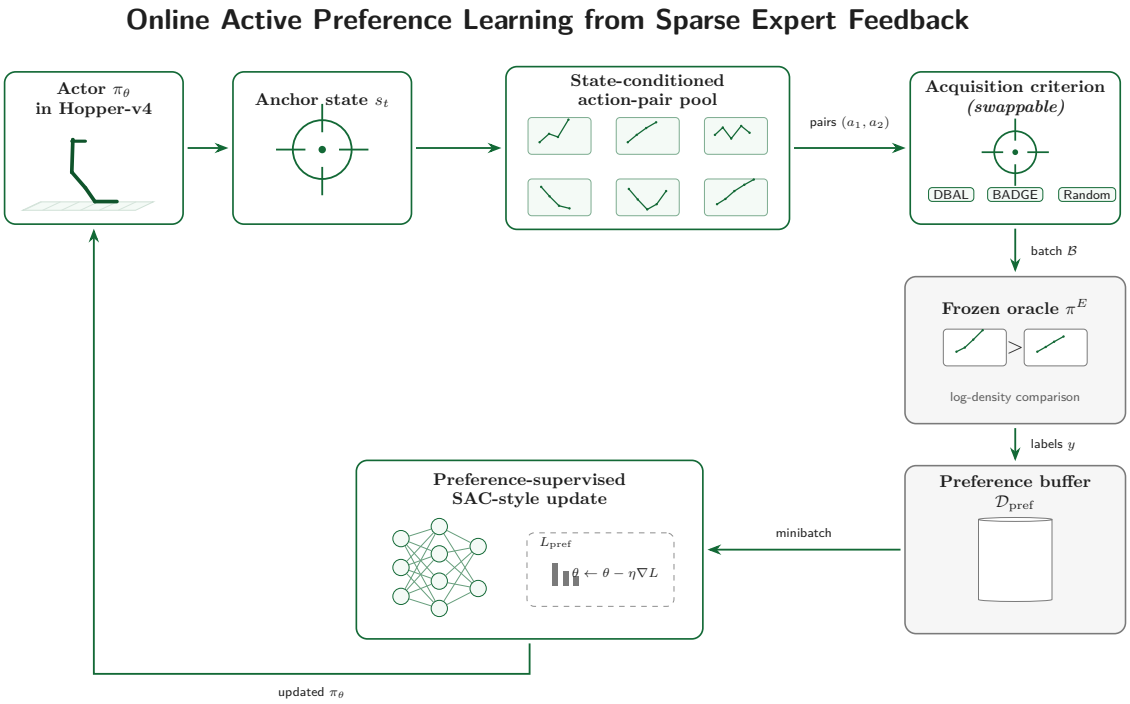


Figure 1. Online active-preference learning framework as a cycle. The learner's current policy π_θ observes a state s_t; a candidate pool of state-conditioned action pairs is constructed at that anchor; the acquisition criterion selects a batch $\mathcal{B}$ of K pairs; the frozen expert oracle π^E (gray fill) returns binary preference labels by log-density comparison; labels accumulate in the preference buffer $\mathcal{D}_{\text{pref}}$; preference-supervised SAC-style updates use minibatches from the buffer to update π_θ. Rounded rectangles are processes; gray fill marks external or stored components.

4.2 The Expert Oracle and Identification

The oracle defines the source of preference supervision used throughout the framework. It is a frozen expert policy, π^E, pretrained with the standard soft actor–critic (SAC) algorithm (Haarnoja et al., 2018) on the same task as the learner, and its role is limited to assigning pairwise labels to candidate actions queried during training. In this framework, the oracle is not an external reward model or human annotator, but a fixed policy trained beforehand in the given environment. This section specifies how those labels are generated and, just as importantly, what kind of alignment they identify.

Given a state s and two candidate actions a_1 and a_2, the oracle returns the binary label

$$y = \mathbb{1}\left[\log \pi^E(a_1 \mid s) > \log \pi^E(a_2 \mid s)\right] \tag{4.1}$$

The expert policy is stochastic, taking the form of an entropy-regularized squashed-Gaussian policy that deliberately maintains spread over the action space. The oracle, however, does not sample from this distribution when answering a query. It applies a deterministic comparison to the expert's action density, so the same pair of actions at the same state always receives the same label. The label therefore records which proposed action is more likely under π^E at that state, rather than expressing a probabilistic draw from the expert.

The density used in this comparison includes the squashed-Gaussian change-of-variables correction, which allows densities to be compared in the bounded continuous action space of the environment rather than in the unsquashed Gaussian parameterization. This correction matters especially near action bounds, where comparisons based only on the pre-squash Gaussian density would not correspond to the density of the executed action. The full computation is provided in Appendix B.5. To avoid making the oracle depend on an idiosyncratic expert run, the oracle is a fixed checkpoint deliberately selected to minimize its distance to the interquartile mean of 10 independent expert training runs (see Appendix B.6).

Each oracle query is state-conditioned and pointwise. The oracle compares two candidate actions at a single observed state, so the resulting label identifies a local density ranking rather than a trajectory-level preference. As training proceeds, the preference buffer accumulates many such local comparisons, each indicating which action the expert policy assigns higher likelihood at the queried state. The buffer

should therefore not be read as a set of judgments about cumulative return, task success over a rollout, or the long-horizon consequences of selecting either action.

This interpretation also bounds the form of alignment that training can produce. Because queries are issued at states encountered by the learner, the supervision signal is defined over the learner's state marginal, d^{π_L}, rather than over the expert's state distribution. The oracle will provide a deterministic density-based label at any state the learner visits, but it does not itself move the learner toward states visited by the expert. Matching the oracle's local action-density preferences therefore need not recover the expert's trajectory distribution or return-maximizing behavior. The signal should instead be understood as deterministic, state-conditioned preference supervision that encourages alignment with the expert's action likelihood conditional on the learner's own visited states. Having established that these deterministic, state-conditioned labels reflect local expert density rather than global return, Section 4.3 describes how the continuous-control learner accumulates these labels and trains against them.

4.3 The Shared Training Loop

The training loop specifies how labeled action comparisons are collected, stored, and used for policy learning. It is identical across all experimental conditions — each acquisition strategy and baseline runs a separate training instance under the same query budget, update schedule, preference-buffer mechanics, and optimization objectives. The acquisition sections define how each method selects queries from the constructed candidate pool, and Section 4.6 specifies the candidate-pool construction used by the experimental setup.

Each run begins with an initial warm-up phase that seeds the preference buffer. The agent steps through the environment to obtain states, and candidate actions

are sampled uniformly from the action space to form the initial action pairs. For each queried pair, the oracle returns a binary preference label and yields tuples (s, a_1, a_2, y) that are appended to the preference buffer, together with the mirrored tuples $(s, a_2, a_1, 1-y)$ that enforce label antisymmetry. Warm-up concludes once the prescribed number of oracle-labeled pairwise comparisons has been collected.

After the initial collection phase, the loop interleaves environment stepping, preference acquisition, and scheduled parameter updates. At active acquisition intervals, candidate pairs are constructed according to the standard experimental setup, and the acquisition method selects which pairs to query according to its own criterion. The oracle labels the selected pairs, and the resulting tuples are added to the same preference buffer. The loop also includes periodic background pairs in which one action is drawn from the current policy and the other is sampled uniformly from the action space. These background labels consume part of the total oracle budget and are included to counterbalance the concentration in $\mathcal{D}_{\text{pref}}$ that can arise from selective acquisition alone. Together, the active-query batch size, active acquisition frequency, and background-query schedule define the oracle budget used in each run.

The preference buffer is the data source for both critic and actor optimization. Critic updates use labeled tuples (s, a_1, a_2, y), while actor updates use states sampled from those tuples and draw fresh actions from the current policy. The departure from standard SAC is therefore not in the use of twin critics or an entropy-regularized actor, but in the source of supervision. The critics are trained from preference labels rather than scalar rewards or next-state targets, so their learned scores are preference-trained quantities rather than Bellman action values.

Each critic is trained to assign higher scores to preferred actions. Given a labeled tuple (s, a_1, a_2, y), the per-critic preference loss is the binary cross-entropy

between the oracle label and the temperature-normalized score margin, following the Bradley–Terry-style pairwise classification form adopted in deep preference learning (Christiano et al., 2017):

$$\mathcal{L}_{\text{pref},i} = \text{BCE}\left(\frac{Q_i(s, a_1) - Q_i(s, a_2)}{T},\ y\right) \tag{4.2}$$

The final critic objective averages the two per-critic losses and adds small alignment and centering regularizers that penalize divergence between the two critics and prevent unconstrained score drift. The regularizer definitions, the temperature update rule, and all coefficient values are given in Appendix B.1.

The actor uses the learned critic scores as the objective signal for policy improvement. For a state sampled from $\mathcal{D}_{\text{pref}}$, the current policy draws a reparameterized action a and evaluates its log-density $\log \pi(a \mid s)$. The actor minimizes

$$\mathcal{L}_\pi = \mathbb{E}_{s \sim \mathcal{D}_{\text{pref}},\, a \sim \pi(\cdot|s)}\big[\alpha \log \pi(a \mid s) - \min_{i \in \{1,2\}} Q_i(s, a)\big], \tag{4.3}$$

which preserves the SAC-style entropy-regularized policy objective while replacing return critics with preference-trained critics. The minimum over the two critics gives the actor the more conservative of the two learned scores. The entropy coefficient α follows the standard SAC dual update once policy updates begin; before that point, it remains fixed at its initialization. The update ratios, hyperparameter settings, and full algorithmic pseudocode are provided in Appendix B.1.

4.4 DBAL-Style Acquisition

The first acquisition strategy adapts the uncertainty-then-diversity structure of diverse mini-batch active learning (DBAL) (Zhdanov, 2019) to state-conditioned action-pair queries in the shared training loop. Algorithm 1 summarizes the outer loop; the acquisition step is detailed below.

In this framework, the candidate objects are not unlabeled observations, but state-conditioned action pairs (s, a_1, a_2) drawn from the candidate pool defined by the experimental condition. Depending on that condition, the pool may be constructed from uniformly sampled actions or from a mixed policy–uniform construction, as specified in Section 4.6. A pair is treated as informative when the current critics assign similar scores to the two actions, since such near-indifference indicates that the learned preference model cannot confidently rank the pair. For each candidate pair, the acquisition score is

$$\phi(s, a_1, a_2) = 1 - 2 \left| \sigma\left(\min_{i \in \{1,2\}} Q_i(s, a_1) - \min_{i \in \{1,2\}} Q_i(s, a_2) \right) - 0.5 \right|. \tag{4.4}$$

This score is highest when the conservative critic-score margin between the two actions is close to zero and decreases as the margin grows in absolute value. The sigmoid-transformed margin is used here as an acquisition heuristic, not as the normalized and clipped logit used for critic optimization in Section 4.3. Thus, the score should be read as a measure of local critic indifference rather than as a calibrated preference probability. All candidate pairs in the pool are scored before the query batch is selected.

After scoring, the most informative pairs are retained as the uncertainty subset on which the diversity step is applied. This mirrors the DBAL mechanism: diversity is imposed only after restricting attention to candidates near the model's decision boundary. In this implementation, the same restriction also keeps the k-means step tractable for large candidate pools. Within this shortlist, the concatenated action-pair vectors (a_1, a_2) are clustered using k-means, and the most informative pair in each cluster is selected for oracle labeling. State information therefore enters the acquisition score through the critics, while diversity is imposed over the action-pair representation used by the query object.

Algorithm 1 Soft Actor-Critic Variant with DBAL Active Preference Learning

Input: policy params θ; critic params ϕ_1, ϕ_2; buffer $\mathcal{D}_{\text{pref}} \leftarrow \emptyset$; frozen expert π^E.
Warm-up: collect uniform action pairs along the agent's trajectory, label by π^E, and append (with mirrors) to $\mathcal{D}_{\text{pref}}$ until the warm-up budget is reached.
repeat
    Observe s; if terminal, reset.
    if query step **then**
        Build candidate pool $\mathcal{P}$ at anchor s (Section 4.6).
        $\mathcal{U}_M \leftarrow$ top-M of $\mathcal{P}$ by critic-margin uncertainty ϕ (Eq. 4.4).
        $\mathcal{B} \leftarrow$ argmax-ϕ representative of each cluster under K-means on action-pair vectors in $\mathcal{U}_M$.
        Label $\mathcal{B}$ by oracle π^E; append (with mirrors) to $\mathcal{D}_{\text{pref}}$.
    end if
    if update step **then**
        For $j = 1, \ldots, N_{\text{update}}$: preference-supervised SAC update on a minibatch from $\mathcal{D}_{\text{pref}}$ (Section 4.3).
    end if
    Execute $a \sim \pi_\theta(\cdot \mid s)$.
until convergence

This procedure preserves DBAL's uncertainty-then-diversity structure, but its interpretation differs from the original supervised-classification setting. The uncertainty score is a critic-margin heuristic for pairwise action ranking, not classifier confidence over labels, and the diversity step operates on action-pair vectors rather than learned observation representations. The method should therefore be read as a DBAL-style acquisition rule adapted to state-conditioned preference queries, with the candidate pool itself supplied by the experimental construction.

4.5 BADGE-Inspired Acquisition

The second acquisition strategy adapts batch active learning with diverse gradient embeddings (BADGE) (Ash et al., 2020) to the same state-conditioned preference-query setting. Algorithm 2 summarizes the outer loop at the same level of abstraction as Algorithm 1; the multi-stage selection routine BADGE-SELECT is detailed below and its hyperparameter settings are listed in Appendix B.3.

In this framework, the queried objects are not unlabeled observations with class labels, but state-conditioned action pairs (s, a_1, a_2) whose labels are supplied only after oracle query. To construct a BADGE-style embedding before the oracle is queried, the method first assigns each candidate pair a diagnostic provisional label from the current critics:

$$\hat{y}(s, a_1, a_2) = \mathbb{1}\left[\min_{i\in\{1,2\}} Q_i(s, a_1) > \min_{i\in\{1,2\}} Q_i(s, a_2)\right]. \tag{4.5}$$

This provisional label is used only to form the acquisition embedding. It is not written to $\mathcal{D}_{\text{pref}}$ and does not replace the oracle label used for critic training.

For each critic $j \in \{1, 2\}$, let $h_j(s, a)$ denote the penultimate feature representation used by that critic, and let $d^{(j)}(s, a_1, a_2) = Q_j(s, a_1) - Q_j(s, a_2)$ denote its score margin for the candidate pair. The BADGE-inspired embedding uses the closed-form last-layer contribution associated with a Bernoulli preference surrogate:

$$g^{(j)} = \left(\sigma\left(d^{(j)}(s, a_1, a_2)\right) - \hat{y}(s, a_1, a_2)\right)\left(h_j(s, a_1) - h_j(s, a_2)\right), \qquad g = [g^{(1)}; g^{(2)}]. \tag{4.6}$$

The concatenation preserves both critic-specific representation views rather than reducing the twin critics to a single scalar channel. The vector g is used only for acquisition; it is not an additional optimization objective and does not define a calibrated belief about the expert oracle's preference.

Before these embeddings are passed to the diversity step, the BADGE-inspired path narrows the candidate pool using a two-stage heuristic. First, each candidate pair receives a critic-disagreement score, measured by the average of $|Q_1(s, a) - Q_2(s, a)|$ over $a \in \{a_1, a_2\}$, and only the highest-disagreement fraction of the pool is retained. This screen favors pairs for which the twin critics are least reconciled at the level of the individual query actions. Second, within the retained set, candidates

are ranked by preference ambiguity under the conservative comparison. Writing

$$\Delta_{\text{cons}}(s, a_1, a_2) = \min_{i \in \{1,2\}} Q_i(s, a_1) - \min_{i \in \{1,2\}} Q_i(s, a_2),$$

the method favors pairs with the smallest scaled value of $|\Delta_{\text{cons}}|$, where the scale is given by a slow exponential moving average of typical conservative-margin magnitudes tracked for the anchor state. This normalization keeps the notion of a near tie comparable as critic score scales drift during training. Candidates are retained under this ambiguity ranking until a fixed maximum shortlist size is reached. If the disagreement screen produces no candidates, the same scaled-margin ranking and shortlist cap are applied to the full candidate pool for that anchor. The retained fraction, scaling rule, fallback behavior, and shortlist size are specified in Appendix B.3.

The resulting shortlist is embedded using g, normalized to unit ℓ_2 norm, and selected by k-means++ seeding (Arthur & Vassilvitskii, 2007) in the normalized embedding space. The seed indices define the query batch directly, without a subsequent Lloyd refinement phase. The resulting batch therefore favors candidates whose gradient-like embeddings point in diverse directions, while the preceding filter concentrates the embedding computation on pairs where the twin critics disagree or remain close to conservative indifference. The corresponding random baseline for this acquisition family uses the same surviving set immediately before the k-means++ step, but draws query indices uniformly rather than by embedding-space diversity.

This procedure should therefore be read as BADGE-inspired rather than as a direct application of BADGE. The original method constructs gradient embeddings for classification examples using provisional class labels; here, the embedding is a closed-form critic-head surrogate for state-conditioned action-pair preferences. Oracle

Algorithm 2 Soft Actor-Critic Variant with BADGE Active Preference Learning

Input: policy params θ; critic params ϕ_1, ϕ_2 with penultimate features h_1, h_2; buffer $\mathcal{D}_{\text{pref}} \leftarrow \emptyset$; frozen expert π^E.
Warm-up: collect uniform action pairs along the agent's trajectory, label by π^E, and append (with mirrors) to $\mathcal{D}_{\text{pref}}$ until the warm-up budget is reached.
repeat
    Observe s; if terminal, reset.
    if query step **then**
        Build candidate pool $\mathcal{P}$ at anchor s (Section 4.6).
        $\mathcal{B} \leftarrow$ BADGE-SELECT($\mathcal{P}$, K; h_1, h_2, Q_1, Q_2): disagreement screen, scaled-margin ambiguity ranking, closed-form per-critic embedding $\tilde{g}$, k-means++ seeding on $\tilde{g}$ (Arthur & Vassilvitskii, 2007) (full procedure in Appendix B.3).
        Label $\mathcal{B}$ by oracle π^E; append (with mirrors) to $\mathcal{D}_{\text{pref}}$.
    end if
    if update step **then**
        For $j = 1, \ldots, N_{\text{update}}$: preference-supervised SAC update on a minibatch from $\mathcal{D}_{\text{pref}}$ (Section 4.3).
    end if
    Execute $a \sim \pi_\theta(\cdot \mid s)$.
until convergence

labels remain external to the embedding construction and enter the learning process only after selected pairs are queried.

4.6 Candidate-Pool Construction and Multi-Anchor Querying

Candidate-pool construction determines which state-conditioned action pairs are exposed to the acquisition rule. A candidate query has the form (s, a_1, a_2), where the oracle is asked to compare two proposed actions at a particular anchor state. The oracle interface and parameter-update procedure are defined in Section 4.3; the present section specifies how anchor states are chosen, how candidate queries are generated at those states, and how the random comparators match to the corresponding acquisition family.

Acquisition rounds may run in either single-anchor or multi-anchor mode. In single-anchor mode, all candidate queries are constructed at the current frontier state, i.e., the agent's current environment observation when the acquisition round

is triggered. In multi-anchor mode, the frontier state is always included and is supplemented with additional states sampled from a short window of recent agent observations. The total per-round budget K is split across anchors using fixed relative weights for the frontier versus each additional anchor, with integer rounding so that the sum stays at K. The intent is to reduce the extent to which an acquisition round is concentrated on a single transient observation, while still preserving the same total oracle budget.

At each anchor state s, candidate queries are generated from source-consistent action pairs. For each pair (s, a_1, a_2), both actions are drawn from the same source: either both are sampled from the current policy at s, or both are sampled uniformly from the bounded action space. Depending on the experimental setup, the candidate pool contains either uniform pairs only or a mix of uniform and policy pairs.

The acquisition methods then select queries from the anchor-level pools according to the procedures defined in the preceding sections. DBAL-style acquisition applies its uncertainty-then-diversity rule to the exposed pool, whereas BADGE-inspired acquisition applies its disagreement and ambiguity preselection before embedding-space selection. These steps determine which candidate queries are sent to the oracle, but the selected queries are labeled by the same oracle and stored in the same preference buffer.

The random baselines are matched within acquisition family rather than defined by a single global random rule. For DBAL-style acquisition, the random comparator samples uniformly from the same per-anchor candidate pool under the same anchor, budget, and mixture settings, bypassing the DBAL uncertainty and diversity steps. For BADGE-inspired acquisition, the random comparator samples uniformly from the same post-prefilter shortlist that would otherwise be passed to embedding-space

selection; it does not construct gradient embeddings or run k-means++ selection. This preserves the intended active-versus-random comparison for each family without changing the oracle, candidate exposure, or training loop.

After selection, all queried pairs follow the shared path described in Section 4.3. The oracle supplies preference labels, the resulting tuples are added to the preference buffer, and critic and actor updates use the same scheduled objectives. Because the oracle, preference buffer, and update procedure are fixed across conditions, the subsequent experiments can interpret differences between active and random variants as effects of acquisition and candidate-exposure design rather than differences in the training objective itself. Optional schedule controls, including variants that alter screening aggressiveness or per-round query allocation, are reported in Appendix B.4.

CHAPTER V

EXPERIMENTS AND RESULTS

5.1 Chapter Overview

This chapter evaluates the online preference-learning framework proposed in Chapter IV, studying it on the Hopper-v4 continuous-control task under two acquisition families: BADGE-inspired selection and DBAL-style selection. For each family, the central question is whether the active acquisition method outperforms uniform random sampling from the same candidate pool when the oracle interface, query budget, and training horizon are otherwise matched.

The two families are analyzed separately rather than as a cross-family factorial design. Within each family, an active acquisition condition is paired with a matched random baseline under the same training configuration, making the acquisition method the only systematic procedural difference between conditions. This design supports interpreting within-family performance differences as effects of query selection rather than consequences of different budgets or training infrastructure. Beyond these primary comparisons, the chapter examines how selected design primitives shift outcomes within each acquisition family and places the two families in a descriptive comparison under the headline protocol. These primary, mechanism-support, and descriptive analyses address RQ1, RQ2, and RQ3 respectively, as defined in Section 1.4.

All primary within-family comparisons in this chapter are specific to Hopper-v4 under the headline protocol defined in Section 5.2.

5.2 Evaluation Protocol and Claim Tiers

5.2.1 Environment and Training Conditions. All experiments were conducted on Hopper-v4 (Todorov et al., 2012), a planar locomotion task from the

MuJoCo continuous-control suite accessed through Gymnasium (Towers et al., 2023). Each configuration was replicated across ten seeds, with each replication training for 1,000,000 environment steps.

The headline protocol queried $K = 8$ preference pairs per active-learning round, triggered every 100 environment steps. Each run issued approximately 82,800 preference-pair queries to the oracle in total: 80,000 from active-learning rounds ($K = 8$ per round over 10,000 triggered rounds), 2,000 from the warmup phase, and 800 from buffer-diversity random queries issued every 2,000 environment steps, as established in Chapter IV. Configurations with $K = 16$ and a 200-step query interval appear as mechanism-support evidence in Section 5.6.

5.2.2 Performance Metric. Performance was measured as the mean episodic return over each seed's final 100 completed training episodes, averaged across the ten seeds. Episodic return served as an external evaluation metric only; training supervision was provided by pairwise preference labels from the oracle, not by reward signals. The interquartile mean (IQM) of the ten seed-level values (Agarwal et al., 2021) serves as a robustness check; any substantive divergence between the two is noted in the relevant subsection.

5.2.3 Primary Comparisons. Two within-family active-versus-random pairs were designated as the chapter's primary comparisons, one per acquisition family. For the BADGE-inspired pair, the active and random-pool conditions shared the same candidate filtering and shortlist construction, with the selection criterion as the only difference between them. For the DBAL-style pair, the random-pool baseline was constructed under the same per-anchor candidate-pool and oracle protocol, with the random baseline drawing uniformly from that pool while bypassing the uncertainty scoring and diversity steps that together constitute the DBAL acquisition

criterion; full provenance is documented in Appendix A.3. Table 3 identifies the designated pairs.

Table 3. Designated primary within-family comparisons. Both pairs used $K = 8$ preference pairs per round, 100-step query interval, a 1,000,000-step training horizon, and seeds 1–10.

Family	Acquisition criterion	Random baseline
BADGE-inspired	Gradient-embedding selection	Uniform random from same pool
DBAL-style	Uncertainty-diversity selection	Uniform random from same pool

Each pair was assessed using paired bootstrap over the ten shared seeds (50,000 resamples, 95% nominal coverage), with runs matched by seed assignment. The two within-family tests were conducted and interpreted independently.

5.2.4 Evidence Tiers. Beyond the primary active-versus-random comparisons, Sections 5.6 and 5.5 examine two further aspects of the framework's behavior, each supported by a distinct type of evidence. Mechanism-support analyses in Section 5.6 vary one design primitive at a time within a fixed family, comparing configurations that differ only in that primitive. The three primitives examined are pool composition, anchor strategy, and the disagreement gate. Pool composition and anchor strategy are tested within both acquisition families; the disagreement gate is specific to the BADGE-inspired path. The cross-family comparison in Section 5.5 cannot be treated as a controlled test because the two families' random baselines were constructed under family-specific protocols, so observed differences reflect both the acquisition methods and baseline construction simultaneously. Table 4 summarizes the evidential role of each class.

Table 4. Evidence classes and their inferential role.

Evidence class	Inferential role	Section
Primary within-family comparisons	Within-family paired inference	5.4
Mechanism-support analyses	One primitive varied, all others matched; both families	5.6
Cross-family comparison	Descriptive; non-shared baselines	5.5

5.3 Implementation and Reproducibility

Experiments were implemented in Python 3.10 using custom training scripts built on the CleanRL library (Huang et al., 2022). The core numerical stack comprised PyTorch 2.5.1 (CUDA 12.1), MuJoCo 2.3.3, and Gymnasium 0.28.1; post-training aggregate and reliability statistics used NumPy 1.24.1, SciPy 1.11, and rliable 1.2 (Agarwal et al., 2021). All runs were executed on Talapas2, the University of Oregon HPC cluster running Red Hat Enterprise Linux 8; each of the ten seeds per configuration was submitted as an independent SLURM job allocated one NVIDIA A100 GPU, four CPU cores, and 16 GB of RAM.

Episodic returns were recorded at each episode boundary during training, captured from the environment's terminal information and written to per-seed training logs. Configuration-level aggregates were computed by taking each seed's mean of its last 100 episodic returns and then averaging those per-seed means across seeds, consistent with the metric definition in Section 5.2.2. Bootstrap statistics were derived from these per-seed means using a dedicated analysis script. Learning curves were constructed from each seed's complete episodic return sequence, parsed from its training log and interpolated onto a fixed step grid using last-observation carry-forward; the mean and standard deviation across seeds at each grid point produced

the reported curves. Complete software version details and experimental artifact paths are provided in Appendix A.3.

5.4 Primary Within-Family Results

This section reports the central evidence for RQ1: whether the active acquisition method produces a different mean return than its matched random baseline within each acquisition family. Each family is evaluated separately under the headline protocol (K=8, query interval 100 steps, H=1,000,000, ten seeds). Table 5 summarises the per-condition means with 95% bootstrap confidence intervals and the paired bootstrap difference for each within-family comparison.

Table 5. Primary within-family results under the headline protocol (K=8, query interval 100 steps, H=1,000,000, n=10 seeds). Means are the mean of per-seed last-100-episode means; CIs are 95% seed-resampled bootstrap intervals. Paired differences use matched seeds (50,000 resamples). †Oracle is the reward-trained SAC policy used as preference supervisor; reported as a scale reference only.

	Active method		Matched random		Active − Random
Family	Mean (95% CI)	IQM	Mean (95% CI)	IQM	Mean diff. (95% CI)
Oracle†	2645	2808	—	—	—
BADGE-inspired	1897 [1787, 1984]	1917	1593 [1478, 1722]	1557	**+304 [86, 476]**
DBAL-style	1708 [1614, 1805]	1714	1693 [1479, 1903]	1691	+15 [−281, 309]

5.4.1 BADGE-Inspired Active Method vs. Matched Random.

Under the headline protocol, the BADGE-inspired active method yielded a mean return of 1897 (95% CI: 1787–1984), corresponding to approximately 72 % of the oracle reference return. Its matched random baseline reached 1593 (95% CI: 1478–1722). The two marginal confidence intervals do not overlap, and the IQM values follow the same ordering (1917 vs. 1557), indicating the ordering is robust to the top and bottom quartiles of seeds.

The paired bootstrap analysis, which matched seeds across the two conditions to control for seed-level stochasticity, produced an estimated difference of +304 return units (95% CI: 86–476). Because the entire confidence interval lies above zero, the paired comparison provides evidence that the active method produced higher final-period returns than the random baseline under this protocol. The lower bound of 86 return units is non-trivial relative to the observed within-condition spread.

5.4.2 DBAL-Style Active Method vs. Matched Random. The DBAL-style multi-anchor active method averaged 1708 across ten seeds (95% CI: 1614–1805), approximately 65 % of the oracle reference return. Its matched random baseline draws uniformly from the same per-anchor candidate pool under the same anchor, budget, and mixture settings, bypassing the uncertainty scoring and diversity steps that together constitute the DBAL acquisition criterion. The random baseline produced a mean of 1693 (95% CI: 1479–1903). The random condition exhibits a notably wider bootstrap CI than its active counterpart (width 424 vs. 191), reflecting elevated seed-level variability in that configuration. The IQM values (1714 active vs. 1691 random) are directionally consistent with the mean-of-means, indicating the positive point estimate is not driven by outlier seeds in either condition.

The paired bootstrap analysis yielded an estimated difference of +15 return units (95% CI: −281–309). The point estimate is positive, but the confidence interval spans zero by a wide margin, and the available evidence at ten seeds does not permit a strong inferential claim about the direction of the effect. The DBAL-family result is therefore treated as inconclusive under the headline protocol, independent of the BADGE-family result in Section 5.4.1.

5.4.3 Primary Learning Curves. Figures 2 and 3 show the mean episodic return across seeds over the full training horizon for each primary pair. Curves

display the mean and one-standard-deviation band across ten seeds, constructed by interpolating each seed's episodic return sequence onto a fixed 10,000-step grid using last-observation carry-forward.

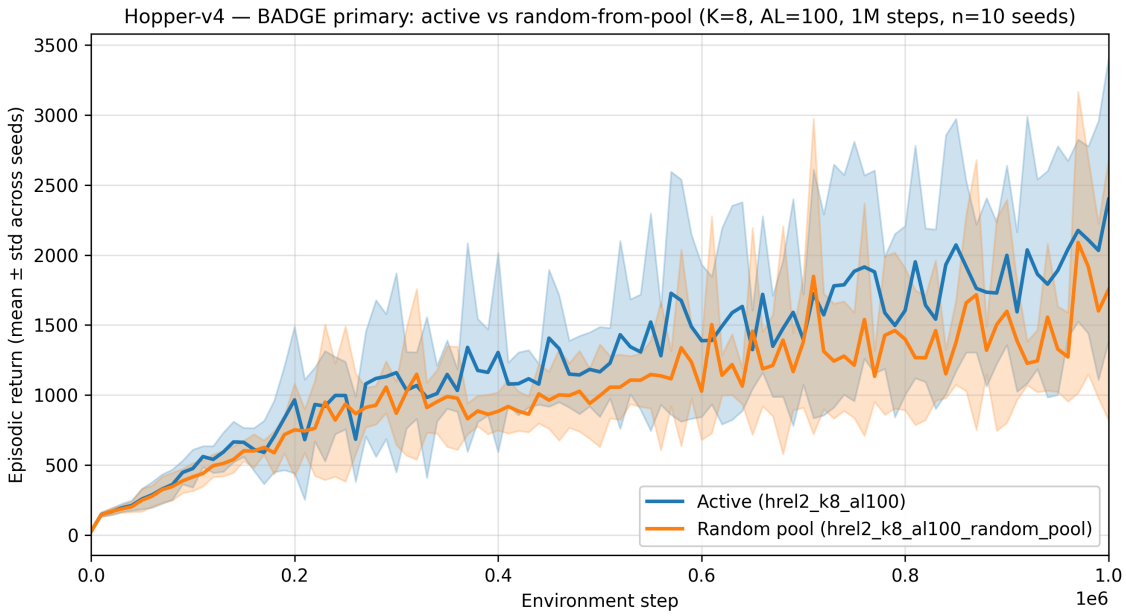


Figure 2. Hopper-v4 learning curves for the BADGE-inspired primary pair under the headline protocol (K=8, query interval 100 steps, H=1,000,000, n=10 seeds). Solid lines show the mean across seeds; shaded bands show ±1 standard deviation. Curves are interpolated onto a fixed 10,000-step grid using last-observation carry-forward.

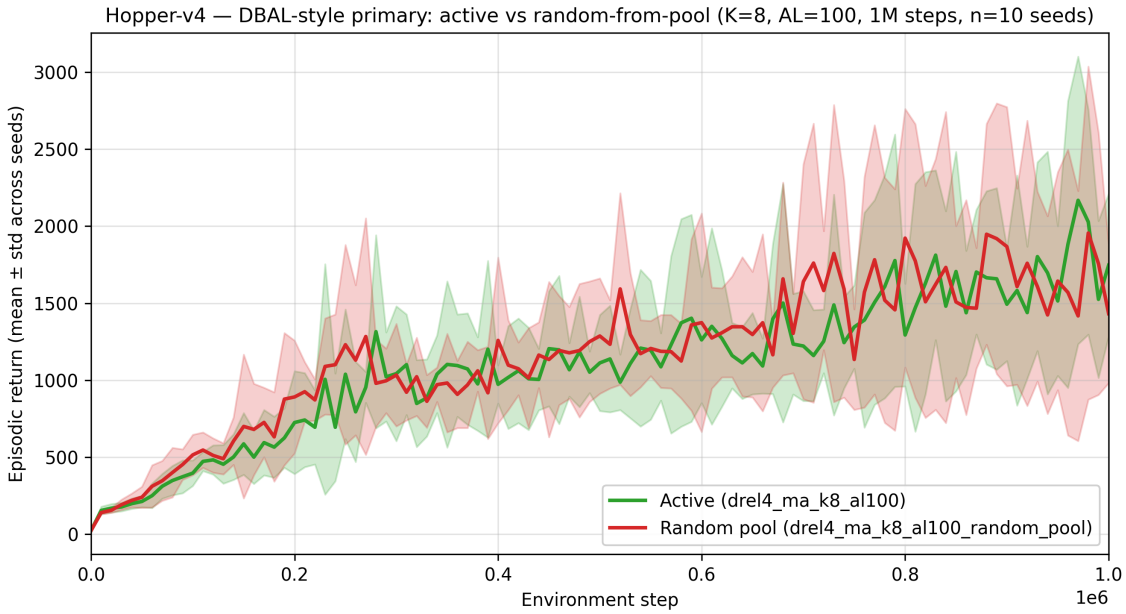


Figure 3. Hopper-v4 learning curves for the DBAL-style primary pair under the headline protocol (K=8, query interval 100 steps, H=1,000,000, n=10 seeds). Solid lines show the mean across seeds; shaded bands show ±1 standard deviation. Curves are interpolated onto a fixed 10,000-step grid using last-observation carry-forward.

For the BADGE-inspired pair (Figure 2), the active condition maintains a positive and progressively widening margin over the random baseline from early in training, with the gap growing steadily from roughly 200,000 steps onward and reaching approximately 450 return units at the terminal period. The mean difference reported in Table 5 is consistent with this trajectory.

For the DBAL-style pair (Figure 3), the random condition's mean return tracks above the active condition's through most of the training horizon, with the two trajectories converging around 900,000 steps. The active condition's mean modestly exceeds the random's only near the terminal period, and the uncertainty bands overlap substantially throughout. This trajectory is consistent with the near-zero mean difference and the wide paired confidence interval reported in Section 5.4.2.

5.5 Descriptive Cross-Family Comparison

This section reports the descriptive evidence for RQ3: the cross-family ordering of the two active acquisition methods under the shared scaffold at the headline budget. Table 5 includes all four headline conditions and allows a descriptive cross-family reading. Across the two active methods, the BADGE-inspired active acquisition method produced the higher mean return (1897 vs. 1708), and the IQM values follow the same ordering (1917 vs. 1714). Because pool construction, oracle, query budget, and training loop are matched across the two families, the gap between the active methods reflects their complete acquisition strategies, including family-specific candidate-exposure and selection steps, rather than any difference in training infrastructure. No cross-family paired comparison is reported.

The cross-family ordering of the random baselines is not interpretable on the same basis. As established in Section 5.4.2, the DBAL matched random draws from the full per-anchor candidate pool while the BADGE matched random draws from the same filtered shortlist as its active condition. The two baselines therefore differ in candidate-pool exposure at selection time (1693 vs. 1593), and the within-family active-versus-random margins are not directly comparable across families. The primary within-family comparisons in Section 5.4 provide the primary evidence for RQ1.

5.6 Mechanism Results

This section reports the mechanism-support evidence for RQ2: which framework primitives appear to affect outcomes within each acquisition family under the shared scaffold. The primary within-family comparisons in Section 5.4 evaluate each active acquisition method as a complete unit. This section isolates specific design primitives whose variation shifts outcomes within each family. Two subsections address three

design primitives. Section 5.6.1 examines candidate-pool composition and anchor exposure, both of which appear in both acquisition families, while Section 5.6.2 evaluates the disagreement gate, which is specific to the BADGE-inspired method. All comparisons are evaluated via bootstrap confidence intervals and interpreted descriptively relative to the primary results; they do not constitute additional within-family hypothesis tests.

5.6.1 Candidate-Pool Composition and Anchor Exposure. Table 6 organizes the configurations examined in this analysis. Headline active conditions from Table 5 serve as reference rows. The remaining rows vary pool composition or anchor count relative to those references; within each targeted contrast, all parameters other than the one being varied are held fixed.

Table 6. Mechanism-support configurations for pool composition and anchor exposure. Headline conditions (marked †) are reproduced from Table 5. All runs use 10 seeds over a 1,000,000-step horizon. *Anchor*: single = frontier anchor only; multi = frontier + rolling recent-policy anchors. *Pool*: mixed = policy_frac = 0.5; uniform = policy_frac = 0.0.

Family	Configuration	K	AL	Anchor	Mean
BADGE	Multi-anchor, mixed pool†	8	100	multi	1897
BADGE	Single-anchor, mixed pool	8	100	single	1785
BADGE	Single-anchor, mixed pool	16	200	single	1590
BADGE	Single-anchor, uniform pool	16	200	single	1430
DBAL	Multi-anchor active†	8	100	multi	1708
DBAL	Single-anchor active	8	100	single	1617
DBAL	Single-anchor, mixed pool	16	200	single	1640
DBAL	Single-anchor, uniform pool	16	200	single	1261

Pool composition. Within the BADGE single-anchor conditions at $K = 16$, AL = 200, supplying a mixed policy-and-uniform pool raises the mean return by 160 points relative to a uniform-only pool (1590 versus 1430). The same direction appears at $K = 8$, where the single-anchor mixed-pool condition reaches 1785, well above its $K = 16$ uniform-only counterpart (1430). Because K and AL also differ between

those two, the comparison is not a clean pool-only test, but the directional signal is consistent across both schedules. For DBAL, the matched comparison at $K = 16$, AL $= 200$ yields a substantially larger gap. The mixed-pool condition reaches 1640 while the uniform-only condition reaches 1261, a difference of 379 points. In both families, mixed pools are associated with higher mean returns than uniform-only pools at the mechanism schedule, with a substantially larger gap for DBAL.

Anchor exposure. Multi-anchor querying supplements the frontier anchor with additional anchors drawn from recent policy observations, broadening the regions of trajectory space from which candidate pairs are sourced. With $K = 8$ and AL $= 100$ held fixed, moving from a single frontier anchor to multi-anchor querying raises the BADGE-inspired active method's mean return from 1785 to 1897, a difference of 112 points. The same comparison at matched K and AL in DBAL yields 1617 (single-anchor) versus 1708 (multi-anchor), a difference of 91 points. Both families show the same directional effect.

5.6.2 Disagreement Gate.

The BADGE-inspired acquisition path applies a disagreement gate prior to gradient-embedding selection. Candidate pairs on which the twin critics disagree least at the query actions are screened out, concentrating the pool on higher-disagreement pairs. In the headline protocol, both the active method and the matched random baseline draw from the same post-gate pool; the gate is therefore a shared structural component of both primary conditions. Table 7 crosses gate status with selection rule using the reliability-grid row IDs in Appendix A.5 (Table 13).

Table 7. Mean returns for the disagreement-gate analysis (Hopper-v4, ten seeds, 10^6 steps). All four cells use the headline query schedule (K=8, AL=100). The gate-enabled row reproduces the headline primary pair: both arms share the same post-gate shortlist (active: gradient-embedding k-means++; random: uniform draw). The gate-disabled row turns disagreement screening off while holding K=8 and AL=100: active uses gradient-embedding selection on the ungated pool; random uses uniform draw from the same ungated pool. Row IDs appear in Appendix A.5, Table 13.

	Active	Random
Gate enabled	1897	1593
Gate disabled	1797	1872

With the gate enabled at the headline budget, the active method leads the matched random baseline by 304 points, the result reported in Section 5.4.1. At the same K=8 and AL=100 schedule, disabling disagreement screening reverses the ordering on the gate-disabled row: the random arm (1872) exceeds the active arm (1797) by 75 points. Relative to the gate-enabled row, the active mean falls by 100 points while the random mean rises by 279 points. The active decrease is consistent with gradient-embedding selection relying on a disagreement-concentrated pool; the random increase is consistent with uniform sampling benefiting when the disagreement screen is removed. The two shifts are asymmetric in magnitude. Because the gate-disabled row holds the headline schedule but removes screening for both arms, it provides a matched mechanism contrast at K=8 and AL=100; it is descriptive mechanism support and does not replace the gate-enabled primary comparison as a confirmatory within-family test.

The headline comparison in Section 5.4 evaluates the complete BADGE-inspired acquisition pipeline against the random baseline under the same post-gate pool at $K = 8$ and AL $= 100$. Both conditions share pool construction and differ only in the selection rule (Section 5.2.3). The gate is a constituent component of that pipeline, not an external modification of the baseline. Table 7 shows that disabling the gate

is associated with an ordering reversal even at the headline schedule, which supports treating the disagreement screen as a design primitive that materially shapes the active-versus-random comparison.

5.7 Summary

This chapter reported Hopper-v4 outcomes under three evidence tiers (Section 5.2.4). The primary within-family comparisons in Section 5.4 support a BADGE-inspired advantage over matched random sampling from the same post-gate exposure and treat the DBAL-style contrast as inconclusive at ten seeds. Mechanism-support analyses in Section 5.6 describe how pool composition, anchor exposure, and the BADGE disagreement gate shift mean return in targeted contrasts. The cross-family comparison in Section 5.5 is descriptive only because family-specific matched random baselines differ in candidate exposure. Chapter VI interprets what these patterns imply under the density oracle, why the primary contrasts split across families, and the scope of the combined evidence.

CHAPTER VI

DISCUSSION, LIMITATIONS, AND CONCLUSION

6.1 Reading the Evidence

Chapter V evaluated the framework from Chapter IV on Hopper-v4 under the evidence tiers stated in the Introduction and in Section 5.2.4. This chapter interprets those results rather than restating the full numerical record. All claims below stay within the configurations and tiers reported in Chapter V and do not add new experiments.

The primary within-family comparisons in Section 5.4 address whether each active method outperforms its matched random baseline under the headline protocol. The BADGE-inspired contrast supports a within-family advantage; the DBAL-style contrast is inconclusive rather than evidence of a null effect. Those two outcomes should not be merged into a family-general claim about active acquisition.

The mechanism-support analyses in Section 5.6 address which framework primitives co-vary with mean return when varied in targeted contrasts. In some mechanism contrasts, the associated mean shifts were comparable to the headline paired estimates; for DBAL-style pool composition at the mechanism schedule, the shift was much larger than the headline active-versus-random paired estimate. They are descriptive sensitivity evidence, not additional paired hypothesis tests. The cross-family reading in Section 5.5 reports a descriptive ordering of the two active methods only; it is not a paired cross-family test because the matched random baselines differ in family-specific candidate exposure. The mechanism-support rows also speak to the framework-primitive question posed in the Introduction by showing which pool, anchor, and filtering choices co-vary with mean return within each family.

Chapter V supports those tier labels with several complementary summaries rather than a single statistic. *Primary inference* comes from paired bootstrap intervals on seed-matched active-versus-random contrasts (Section 5.2.3). *Learning curves* in Section 5.4.3 show whether headline differences appear early and persist, or only near the terminal period. *Interquartile means* (IQM) across seed-level aggregates provide a secondary check on whether a mean ordering depends on the top or bottom quartile of seeds (Agarwal et al., 2021); they do not replace the paired intervals. *Mechanism rows* test whether design primitives shift means in consistent directions under targeted contrasts. The subsections below use all four layers where relevant; bootstrap remains the authoritative summary for the primary within-family claims.

The sections below first fix what the supervision signal identifies, then explain why the two primary contrasts diverged, and finally state what the combined evidence does and does not support.

6.2 What the Supervision Signal Identifies

Pairwise labels came from a fixed expert log-density oracle (Section 4.2). At each queried state, the label favors the action with higher density under a frozen expert policy. The learner is trained on these pairs with a preference loss, so updates favor actions the oracle would rank higher at the query states. That is behavioral alignment with expert action likelihood, not human preference data (Christiano et al., 2017; Lee, Smith, & Abbeel, 2021), reward identification, or guaranteed return maximization.

Chapter V reported episodic return even though training never directly optimizes it. The oracle reference in Table 5 provides a scale benchmark for reward-trained expert behavior on Hopper-v4; it is not a supervised target for the learner. Headline active policies reached roughly 65–72 % of that reference under density supervision,

which describes return induced by the implemented framework rather than shortfall on the expert's own objective.

Because queries are issued at states the learner visits, supervision is defined over the learner's state marginal rather than over the expert's state distribution. Matching local density preferences therefore need not recover expert trajectories or return-maximizing behavior (Section 4.2). A within-family return gain under this oracle means the active pipeline used density labels more effectively than matched random sampling from the same family-specific exposure at the same query budget. It does not show return optimality, nor does it predict the same ordering under human labels or a reward-based oracle, since simulated-teacher choice is itself known to shift preference-based RL outcomes (Lee, Smith, Dragan, & Abbeel, 2021). How large such a gain can be depends on which candidate actions the framework exposes and how the acquisition criterion places queries within that exposure.

6.3 Why the Primary Contrasts Split Across Families

Within each family, the headline primary contrasts change only the acquisition criterion while holding the oracle interface, K, query interval, horizon, and seeds fixed (Section 5.2.3). They are therefore the right unit for asking whether active selection beats matched random sampling, but they do not randomize every pipeline primitive across families. The BADGE-inspired family showed a clear within-family separation at ten seeds; the DBAL-style family did not. The subsections below interpret that split using mechanism patterns from Chapter V, statistical scope, and the match between each acquisition heuristic and density-based supervision.

6.3.1 BADGE-Inspired Pipeline. The BADGE primary pair compared the full BADGE-inspired pipeline, including post-gate pool construction, to a matched random baseline that sampled uniformly from the same post-gate shortlist

(Section 5.4.1). The active lead is therefore not explained by a narrower pool available only to the active arm. Under the density oracle, embedding-based selection within that shared shortlist produced a stronger final policy than uniform draws from the same exposure.

Learning curves in Figure 2 show the active condition ahead of random from early training, with a widening margin after roughly 200,000 steps. That temporal pattern aligns with the paired bootstrap conclusion and with the IQM ordering in Table 5, so the BADGE advantage is not plausibly a single-seed terminal artifact.

A plausible reading, consistent with the mechanism rows, is that the disagreement gate concentrates the shortlist on pairs where twin critics disagree at the query actions, and gradient-like embeddings then select diverse directions within that structured set. Under density supervision, such pairs may be more likely to move the conservative critic margin that drives policy improvement than uniformly sampled pairs from the same post-gate pool. This remains a hypothesis tied to the reported aggregates; the thesis did not measure per-query label utility or held-out preference accuracy.

The gate-off 2×2 in Table 7 and Section 5.6.2 shows that disabling disagreement screening materially changes the BADGE comparison, including an ordering reversal on the gate-disabled row at the headline schedule (1872 vs. 1797). That table should be read as descriptive mechanism support at $K = 8$ and a 100-step query interval, not as a second confirmatory within-family test. The headline primary comparison kept the gate on for both arms at $K = 8$ and a 100-step query interval, so the gate is part of the method under test rather than a one-sided baseline modification.

6.3.2 DBAL-Style Pipeline. The DBAL headline contrast is inconclusive at ten seeds (Section 5.4.2). Wide dispersion in the random arm is one contributor;

its bootstrap interval spans 424 points on the random arm versus 191 on the active arm (Section 5.4.2), a pattern that is common when deep RL configurations exhibit high seed-level variance (Agarwal et al., 2021). A real but modest active advantage could therefore remain unresolved under the present seed budget rather than ruled out.

Figure 3 tells a different temporal story. Random tracking above active through most of training, substantial overlap in the seed bands, and convergence only near the terminal period are consistent with a near-zero headline mean and a wide paired interval. IQM follows the same direction as the mean but does not narrow that uncertainty.

DBAL ranks candidates by critic-margin uncertainty and action-feature diversity before querying. Under a density oracle, those scores need not mark the action pairs whose labels most improve Hopper return. High-uncertainty pairs at learner-visited states may still lie in action regions that are correct under π^E yet weakly connected to return improvement under the learner's current state distribution.

Mechanism rows suggest that candidate-pool design was a larger lever than the headline DBAL active-versus-random contrast. At $K = 16$ with a 200-step query interval, mixed pools outperformed uniform-only pools by 379 points in the DBAL family versus 160 points in the BADGE family (Table 6). Headline DBAL active and random runs already used multi-anchor mixed pools, so uncertainty ranking may have added little beyond an already strong scaffold. The pool-only gap at the mechanism schedule exceeds the headline paired estimate for DBAL active versus random, which is consistent with DBAL depending more heavily on the quality of the exposed candidate set before uncertainty scoring.

Fair random-pool provenance is documented in Appendix A.3. The DBAL inconclusive result does not weaken the matched BADGE primary finding; it limits how far the combined Hopper evidence should be generalized to active acquisition as a whole.

6.3.3 Mechanism Primitives and the Shared Scaffold. Section 5.6 varied pool composition, anchor exposure, and the BADGE disagreement gate in targeted contrasts. Those shifts support treating pool and anchor design as framework primitives rather than neutral implementation details (Section 4.6), consistent with the engineering contribution stated in Chapter I.

Mixed pools expose both policy-sampled and uniformly sampled action pairs at the anchor states, whereas uniform-only pools restrict the acquisition rule to uniformly sampled pairs. Under the density oracle, mixed pools place more comparisons in action regions connected to the current policy, where expert density contrasts may be more likely to shape subsequent updates. Uniform-only pools can still receive oracle-correct labels, but pairs drawn far from the learner's behavior may contribute less to return improvement at the mechanism schedule in Table 6. The larger pool-composition gap for DBAL-style acquisition is consistent with that family depending more heavily on the exposed candidate set before uncertainty scoring.

Multi-anchor querying supplements the frontier anchor with states from recent policy observations. At the headline schedule ($K = 8$, query interval 100 steps), moving from single-anchor to multi-anchor *active* configurations raised mean return by 112 points in the BADGE family and 91 points in the DBAL family (Table 6). Both families moved in the same direction, which points to anchor layout as a property of the shared training loop rather than of one acquisition module alone. More anchors increase the chance that queries cover states the policy actually visits, which matches

the density-oracle account in Section 6.2. These rows did not form a full factorial design on pools and anchors, so they establish design sensitivity rather than isolated causal estimates.

For BADGE-inspired acquisition, the disagreement gate is a family-specific primitive that concentrates the shortlist before embedding-based selection. Section 5.6.2 reports that disabling the gate materially changes the comparison at the headline schedule, including an ordering reversal on the gate-disabled row (random 1872 vs. active 1797). The headline primary pair therefore evaluates the implemented pipeline with the gate enabled, not selection in isolation from that filter.

6.3.4 Synthesis. Taken together, the primary and mechanism evidence suggest different roles for the two acquisition pipelines under density supervision.

The BADGE-inspired headline test evaluates selection *within* a post-gate shortlist that already restricts the candidate set. When that scaffold includes mixed pools and multi-anchor exposure, the reported active advantage is consistent with embedding-based batching adding value on top of the shared pool design.

The DBAL-style headline test scores uncertainty and diversity over the full per-anchor pool exposed before selection. If density labels are most useful for policy-connected action contrasts, margin-based uncertainty over uniformly sampled pairs may not rank queries by return relevance even when the labels are oracle-correct. That misalignment is one plausible explanation for an unresolved primary contrast alongside a much larger pool-composition effect at the mechanism schedule.

Batch active learning in the supervised setting supplies selection principles, not the full continuous-control preference scaffold studied here (Chapter II). The split primary outcomes therefore support a narrower claim. Under the headline protocol on Hopper-v4, the BADGE-inspired pipeline improved density-supervised return over

matched random sampling from the same post-gate exposure when embedded in a scaffold that structures candidate exposure through mixed pools, multi-anchor querying, and disagreement screening. Uncertainty-then-diversity scoring over the full DBAL per-anchor pool did not show a resolved advantage at ten seeds under the same oracle and query budget.

6.4 Scope, Limitations, and Validity

Statistical scope. The primary comparisons use ten seeds. That count supports the BADGE-inspired within-family finding, where the paired bootstrap interval lies entirely above zero, but it leaves limited power for smaller or more variable effects. The DBAL-style result should not be read as evidence of no effect; its confidence interval spans both signs.

Environmental and oracle scope. Confirmatory evidence is limited to Hopper-v4 under the headline protocol. The oracle supplies deterministic log-density labels, not human preference or trajectory-level feedback. Episodic return is an external evaluation metric, not the quantity optimized by the preference loss. Within-family claims therefore describe the implemented framework under this density proxy on one locomotion task.

Design and mechanism scope. Mechanism rows varied pool composition, anchor exposure, and the BADGE disagreement gate in targeted contrasts. They show design sensitivity but not a factorial decomposition; pool, filtering, anchor layout, and selection rules interact. Several rows use $K = 16$ with a 200-step query interval and are mechanism-support evidence only. Cross-family readings of the two active methods are descriptive because family-specific filtering, shortlisting, and matched random baselines differ; active-minus-random margins are not aligned

across families (Section 5.5). No paired bootstrap contrast was defined across families. Several mechanism rows use schedules outside the headline protocol and are not primary confirmatory evidence; the thesis also does not apply a family-wise multiple-comparison correction across the full experimental grid.

Method adaptation and evaluation metrics. Both acquisition families are adapted to state-conditioned action-pair preferences rather than pool-based classification (Chapter IV). Findings attach to the implemented BADGE-inspired and DBAL-style pipelines, not to unchanged algorithms from the batch active-learning literature. Critics are preference-trained scoring functions rather than Bellman action values (Section 3.3). The primary aggregate is final-period episodic return; paired bootstrap, learning curves, and IQM provide complementary summaries as described in Section 6.1. Held-out preference accuracy, critic calibration, and per-query label utility were not primary metrics and would help separate preference-learning gains from downstream control gains.

Provenance. Primary claims depend on the designated aggregate rows, matched seed sets, and family-specific random-pool comparisons documented in Chapter V and Appendix A.3. Pool exposure and baseline construction affect the interpretation of active-versus-random comparisons, so older exploratory runs and script variants are excluded from primary evidence.

6.5 Future Work

Final-period return supported the primary within-family active-versus-random comparisons, with a significant advantage for the BADGE-inspired pair and an inconclusive contrast for the DBAL-style pair, but it could not distinguish whether active selection used preference labels more efficiently or simply yielded better

downstream control from the same oracle budget. Held-out preference accuracy, critic-margin calibration, return against cumulative oracle calls, and per-query label utility would address that gap directly. They would also help interpret the DBAL inconclusive contrast by distinguishing weak pair ranking from weak exploitation of otherwise informative labels during training.

Generalization beyond Hopper-v4 and the expert log-density oracle remains an open question. Density labels are a deliberate proxy for limited expert feedback, not a stand-in for human preference, reward-model supervision, or trajectory-level comparisons (Christiano et al., 2017; Lee, Smith, Dragan, & Abbeel, 2021; Ouyang et al., 2022). Replicating the headline within-family grid on additional continuous-control tasks, with matched random baselines drawn from the same family-specific exposure in each environment, would test whether the BADGE–DBAL split persists under different dynamics. Varying the label source in parallel would test how much of the observed query behavior depends on density supervision rather than on batch selection itself.

Sharper cross-method claims will also require tighter experimental design than the present grid affords. Larger seed panels would narrow bootstrap intervals for high-variance configurations such as DBAL-style random sampling (Agarwal et al., 2021), and a pre-specified evaluation contract would keep confirmatory headline contrasts distinct from exploratory mechanism rows as the sweep grows. For cross-family comparisons, a shared exposed pool, common filtering stage, and matched random baseline across pipelines would replace the current descriptive ordering with a paired test of selection rules under identical candidate exposure.

6.6 Conclusion

This thesis developed an online preference-learning framework for continuous control. The framework fixes a query cadence inside the training loop, selects state-conditioned action pairs through active batched acquisition, labels them with a fixed expert log-density oracle, and updates a SAC-style actor mediated by preference-trained critics. Within that shared interface, the thesis instantiated BADGE-inspired gradient-embedding batch selection and DBAL-style uncertainty-then-diversity selection as parallel acquisition pipelines (Chapter IV).

On Hopper-v4 under the headline protocol ($K = 8$, 100-step query interval, ten seeds, $H = 1{,}000{,}000$), the empirical evidence answers the research questions as follows. For the primary within-family contrasts, the BADGE-inspired active method reached mean final-period return 1897 against 1593 for its matched random baseline, with paired bootstrap difference +304 (95% CI: 86–476); the interval excludes zero, and the IQM ordering (1917 vs. 1557) matches the mean. The DBAL-style active method averaged 1708 against 1693 for its matched random baseline, with paired difference +15 (95% CI: −281–309); the interval spans zero, so that contrast is inconclusive rather than evidence of no effect. These two outcomes should not be combined into a family-general claim that active acquisition improves over matched random sampling. For the descriptive cross-family reading, the BADGE-inspired active method ranked above the DBAL-style active method (1897 vs. 1708; IQM 1917 vs. 1714), but no paired cross-family test was defined because the matched random baselines differ in family-specific candidate exposure.

Mechanism-support analyses showed that candidate-pool composition, anchor exposure, and the BADGE disagreement gate are consequential framework components, not incidental tuning. At $K = 16$ with a 200-step query interval,

mixed pools exceeded uniform-only pools by 379 points for DBAL-style acquisition and by 160 points for the BADGE family; at the headline schedule, multi-anchor *active* configurations exceeded their single-anchor counterparts by 112 and 91 points in the two families. For DBAL-style acquisition, the pool-composition gap at the mechanism schedule (379 points) exceeded the headline active-versus-random paired estimate (+15), which helps explain why an inconclusive primary contrast can coexist with strong sensitivity to scaffold design.

The central inference is that acquisition criteria cannot be evaluated apart from the pools, filters, and anchors that define family-specific candidate exposure. The BADGE-inspired headline test measures embedding-based selection *within* a post-gate shortlist under mixed pools and multi-anchor querying; under the density oracle, that pipeline improved return over matched random draws from the same exposure. The DBAL-style headline test scores uncertainty and diversity over the full per-anchor pool; under the same oracle and a scaffold that already included mixed pools and multiple anchors, that scoring did not yield a resolved advantage at ten seeds. Batch active-learning principles from the supervised literature therefore transfer only as part of this larger preference-learning scaffold, not as standalone guarantees on Hopper-v4.

The thesis supports online active preference acquisition as a viable approach to continuous control under limited expert feedback when the full pipeline is specified honestly. It does not establish human-preference semantics, return optimality, cross-family superiority of one active method over the other, or generalization beyond Hopper-v4 under the reported configurations. Stronger cross-environment evidence, richer label sources, and diagnostics beyond final-period return remain the main paths for testing how broadly these findings hold (Section 6.5).

APPENDIX OVERVIEW

The appendices support the main chapters with the evaluation contract, reproducibility pointers, and implementation reference material cited from Chapters I–VI. Appendix A states designated primary pairs, the headline metric, bootstrap policy, claim-to-evidence map, reproducibility notes, and mechanism or gate reference tables (A.3, A.4, A.5). Appendix B gives training-loop and acquisition hyperparameters, candidate-pool settings, and oracle implementation detail. This appendix is self-contained and records the configurations and aggregate outcomes used in the thesis; exact scripts and raw logs are not included in this document.

APPENDIX A

EVALUATION, REPRODUCIBILITY, AND STATISTICS

A.1 Primary-Pair Definitions, Metric, and Curve Policy

This appendix records the evaluation contract used throughout the thesis. *Designated primary* comparisons were chosen before inspecting outcomes as headline within-family tests for RQ1; they are not formal preregistration.

All primary rows train on Hopper-v4 for $H = 1{,}000{,}000$ steps with seeds $S = \{1, \ldots, 10\}$. Within each family, active and matched-random rows share $K = 8$ pairs per acquisition round every 100 environment steps, plus the shared warm-up and background schedule described in the SAC-variant training-loop section of Appendix B (approximately 82,800 oracle queries per run; Chapter V, Section 5.2). Rows with other K, AL cadences, or anchor schedules are mechanism-support only.

Table 8 lists the two within-family pairs for RQ1. Only the selection rule differs between active and matched random (see the candidate-pool and acquisition-method sections in Appendix B).

Table 8. Designated primary within-family comparisons (Hopper-v4, headline protocol). Row IDs are the `ABLATION_SLUG` identifiers used in the experiment configuration.

Family / role	Row ID
BADGE-inspired, active	`hrel2_k8_al100`
BADGE-inspired, matched random	`hrel2_k8_al100_random_pool`
DBAL-style, active	`drel4_ma_k8_al100`
DBAL-style, matched random	`drel4_ma_k8_al100_random_pool`

Headline metric. The headline scalar is the **mean of per-seed means** of episodic return over each seed's last 100 training episodes (`mean_of_per_seed_means` in each family's aggregate table). Training uses preference labels only; return is an

external evaluation metric. IQM across seed-level means is reported as a robustness check (Agarwal et al., 2021) via `rliable`.

Inferential procedure (RQ1). Per-condition and paired within-family intervals use percentile bootstrap over seeds (B = 50,000, 95% coverage). Family-level tests are interpreted independently; cross-family ordering (RQ3) is descriptive only.

Learning curves. Primary curves (Figures 2, 3) plot the across-seed mean at fixed step grids with ± 1 standard deviation bands (Chapter V, Section 5.4.3). Mechanism and exploratory slugs are summarized in Appendix A.5 and are excluded from RQ1 confirmatory wording unless promoted in Chapter V.

A.2 Claim-to-Evidence Map

Table 9 links Chapter I contributions to row IDs, inferential tier, and thesis artifacts.

Table 9. Claim-to-evidence map. † = designated primary row.

RQ	Claim (summary)	Row ID(s)	Test / tier	Artifact
—	Online preference-learning framework	(implementation)	N/A	Ch. IV; App. B
RQ1	BADGE-inspired active > matched random	`hrel2_k8_al100`†; `hrel2_k8_al100_random_pool`†	Paired bootstrap; CI excludes 0	Tab. 5; Fig. 2
RQ1	DBAL-style active vs. random: inconclusive	`drel4_ma_k8_al100`†; `drel4_ma_k8_al100_random_pool`†	Paired bootstrap; CI spans 0	Tab. 5; Fig. 3
RQ3	Descriptive ordering of active methods	Same four † rows	Descriptive	Tab. 5; §5.5
RQ2	Mixed pool > uniform-only pool	`hrel3_frontier_k16_al200_mix50` vs. `uniform`; `drel3_*_v2`	Descriptive; K=16, AL=200	Tab. 6
RQ2	Multi-anchor > single frontier anchor	Primary † vs. `hrel3_frontier_k8_al100_mix50` / `drel2_k8_al100`	Descriptive	Tab. 6
RQ2	BADGE disagreement gate (2×2)	`hrel2_k8_al100`†; `hrel2_k8_al100_random_pool`†; `hrel2_k8_al100_no_disagree`; `hrel2_k8_al100_random_no_disagree`	Descriptive mechanism contrast at headline schedule	Tab. 7; Tab. 13

Audit trail. To reproduce a headline number, one needs the per-family aggregate table for the relevant row ID and the per-seed last-100 means used by the bootstrap; those artifacts are not embedded in this document. This appendix therefore records the row IDs, protocols, and summary statistics used in the thesis, and it states the seed-pairing convention required for paired inference (Section A.3.2).

A.3 Reproducibility and Artifact Provenance

A.3.1 Software Stack and Launchers. Training uses Python $\geq$ 3.10 with pinned dependencies (notably PyTorch 2.5.1+cu121, Gymnasium 0.28.1, and scikit-learn for BADGE k-means++ seeding). Each row ID is replicated across ten independent runs with seeds 1–10 for 1,000,000 environment steps under the same shared training loop; acquisition-specific settings differ only by row ID and method family.

Primary matched-random rows. The designated matched-random comparators are **not** legacy `random_pool` slugs, which omit the headline K=8/AL=100 settings (mean return $\approx$ 1437 for BADGE under such a slug). Table 10 records the row IDs and acquisition flags that define the headline within-family random arms; each row was submitted as a separate ten-seed batch from its active primary counterpart (Section A.3.3).

Table 10. Designated matched-random primary configurations (headline protocol).

Family	Row ID	Distinguishing acquisition settings
BADGE-inspired	`hrel2_k8_al100_random_pool`	`pair-selection random`; `active-learning-steps` = 100; `num-query-pairs` = 8; same post-gate shortlist as active
DBAL-style	`drel4_ma_k8_al100_random_pool`	`pair-selection random`; `active-learning-steps` = 100; `num-query-pairs` = 8; `recent-anchor-selection-size` = 3; uniform draw from per-anchor pool

A.3.2 Log Paths and Primary Aggregates.

Each row ID corresponds to ten seed-level runs (seeds 1–10), and each family produces a single aggregate table reporting the mean of per-seed means and the across-seed standard deviation for each row ID.

Table 11. Primary row aggregates (mean of per-seed means).

Row ID	Mean	σ_{seeds}	n_{seeds}
`hrel2_k8_al100`	1897.02	160.45	10
`hrel2_k8_al100_random_pool`	1592.92	198.18	10
`drel4_ma_k8_al100`	1707.94	154.29	10
`drel4_ma_k8_al100_random_pool`	1693.01	344.27	10
`chosen_oracle` (seed 2)	2713.12	934.66	1

A.3.3 Fair Random-Pool Provenance.

Matched-random rows share the active row's training loop, oracle, anchor schedule, and candidate-pool construction;

only the final query-selection rule differs. For BADGE-inspired acquisition, both arms sample uniformly from the same post-prefilter shortlist $\mathcal{S}$ (after disagreement screening and margin ranking), not from the raw pool $\mathcal{P}$. For DBAL-style acquisition, the matched-random arm samples uniformly from the same per-anchor candidate pool while bypassing the uncertainty shortlist and diversity steps used by the active rule. Both sides use the same ten seeds for paired bootstrap (Appendix A.4). No primary seed was dropped after submission.

A.4 Bootstrap Procedure

Chapter V intervals come from a percentile bootstrap with $B = 50{,}000$ and $\alpha = 0.05$ applied to per-seed last-100 means. Unpaired intervals bootstrap the mean across seeds; paired intervals bootstrap seed-matched differences (active minus random). IQM uses deterministic trimming, not bootstrap.

A.5 Mechanism and Gate Reference Tables

Descriptive mechanism rows for RQ2 (Chapter V, Section 5.6); not designated primary comparisons.

Table 12. Mechanism-support rows (10^6 steps, ten seeds). † = headline primary at K=8, AL=100.

Family	Row ID	K	AL	Mean	σ_{seeds}
BADGE	`hrel2_k8_al100`†	8	100	1897.02	160.45
DBAL	`drel4_ma_k8_al100`†	8	100	1707.94	154.29
BADGE	`hrel3_frontier_k8_al100_mix50`	8	100	1784.78	180.26
DBAL	`drel2_k8_al100`	8	100	1616.56	129.28
BADGE	`hrel3_frontier_k16_al200_mix50`	16	200	1589.89	162.33
BADGE	`hrel3_frontier_k16_al200_uniform`	16	200	1430.15	232.01
DBAL	`drel3_frontier_k16_al200_mix50_v2`	16	200	1640.37	239.14
DBAL	`drel3_frontier_k16_al200_uniform_v2`	16	200	1261.84	182.95

Table 13 records the row IDs and full means for the disagreement-gate 2×2 summarized in Chapter V, Table 7.

Table 13. Disagreement-gate 2×2 (BADGE). All cells at K=8, AL=100.

	Active	Matched random
Gate enabled	1897.02 (`hrel2_k8_al100`)	1592.92 (`hrel2_k8_al100_random_pool`)
Gate disabled	1796.53 (`hrel2_k8_al100_no_disagree`)	1871.56 (`hrel2_k8_al100_random_no_disagree`)

APPENDIX B

IMPLEMENTATION REFERENCE

B.1 SAC-Variant Training Loop

Shared loop across both method families; critics train from preference BCE only (no Bellman targets). Each query appends (s, a_1, a_2, y) and mirror $(s, a_2, a_1, 1 - y)$. Critic loss combines temperature-normalized pairwise BCE, twin-critic alignment ($\lambda_{\text{align}} = 10^{-4}$), and centering ($\lambda_{\text{center}} = 10^{-4}$) regularizers; T follows an EMA with $\alpha_T = 0.01$. Actor and α updates begin after $N_{\text{pref,min}} = 2000$ labeled tuples. Table 14 lists schedule constants; acquisition flags are defined by the row ID (Section A.3.1).

Table 14. Shared training-loop hyperparameters (primary rows).

Symbol / flag	Description	Value
`total_timesteps`	Training horizon	10^6
`num_warmup_pairs`	Warm-up oracle queries	2000
$N_{\text{pref,min}}$	Gate before actor/α updates	2000
`active_learning_steps` / K	Primary acquisition cadence	100 / 8
N_{post}	Extra critic steps after acquisition	3
F_{rand} / B_{rand}	Background pair cadence / batch	2500 / 2
`batch_size`	Minibatch size	256
`policy_lr` / `q_lr`	Adam rates (actor / critics)	$3{\times}10^{-4}$ / 10^{-3}
α (init) / `autotune`	Entropy coefficient	0.2 / on
Hidden width	Actor and critic MLPs	256

Algorithm 3 Preference-supervised SAC-variant loop (shared).

Warm-up: uniform pairs, oracle labels, mirrored buffer writes.
repeat
  if acquisition round **then**
    $\mathcal{B} \leftarrow$ ACQUIRE($\mathcal{P}$); label; append mirrors; N_{post} critic steps.
  end if
  if background cadence **then**
    policy–uniform pairs; label; append mirrors.
  end if
  Critic step on $\mathcal{D}_{\text{pref}}$; if $|\mathcal{D}_{\text{pref}}| \geq N_{\text{pref,min}}$, actor and α steps.
  Execute $a \sim \pi_\theta(\cdot \mid s)$.
until horizon reached

B.2 DBAL Acquisition Hyperparameters

Primary row `drel4_ma_k8_al100`: `--recent-anchor-selection-size 3 --num-query-pairs 8 --active-learning-steps 100`. Matched random adds `--pair-selection random` on the same per-anchor pool (Section A.3.3).

Table 15. DBAL-style acquisition hyperparameters (primary row).

Symbol / flag	Description	Value
K / U / M	Query batch / pool size / shortlist cap	8 / 1024 / 256
`policy_frac`	On-policy fraction of pool	0.5
`recent_anchor_selection_size`	Extra recent anchors	3
`diversity`	Shortlist diversity primitive	`kmeans`
Active-learning period	Steps between rounds	100

B.3 BADGE-Select Procedure and Hyperparameters

Four-stage routine (BADGE-SELECT) in Algorithm 4. Matched random uses Stages 1–2 only, then uniform K draws from $\mathcal{S}$.

Algorithm 4 BADGE-SELECT: multi-stage query selection.

1: **Stage 1.** Retain top-$\lceil \rho_d |\mathcal{P}| \rceil$ pairs by twin-critic disagreement D; empty fallback $\mathcal{P}' \leftarrow \mathcal{P}$.
2: **Stage 2.** Rank by scaled conservative-margin ambiguity; keep top-M as shortlist $\mathcal{S}$; update EMA scale $\bar{\Delta}$.
3: **Stage 3.** Closed-form per-critic embeddings $g^{(j)} = (\sigma(d^{(j)}) - \hat{y})(h_j(a_1) - h_j(a_2))$; normalize $\tilde{g}$.
4: **Stage 4.** k-means++ seeding on $\{\tilde{g}\}$; return K seed pairs (no Lloyd refinement).

B.3.1 Hyperparameter Values for the Primary BADGE Row.

Primary active `hrel2_k8_al100`; matched random `hrel2_k8_al100_random_pool` (Section A.3.1).

Table 16. BADGE-Select hyperparameters (primary row).

Symbol	Description	Value
K	Pairs per acquisition round	8
ρ_d	Disagreement-screen fraction	0.30
α_{EMA}	$\bar{\Delta}$ smoothing	0.01
M / U	Shortlist cap / pool size	256 / 1024
`policy_frac`	On-policy pool fraction	0.5
Active-learning period	Steps between rounds	100

B.4 Candidate-Pool Construction and Schedule Variants

Primary rows use multi-anchor mode: frontier state plus `recent_anchor_selection_size` = 3 recent states; $K = 8$ split by `badge_frontier_min_pairs` = 10 and `badge_rest_per_anchor` = 2. At each anchor, $U = 1024$ pairs with `policy_frac` = 0.5. Exploratory rows (single-anchor frontier, `policy_frac` $\in \{0, 0.3, 0.7\}$, schedule decays) appear in the mechanism and gate reference tables in Appendix A but not in RQ1 primary claims.

B.5 Oracle Log-Density Computation

Expert π^E is a squashed-Gaussian SAC policy. For latent u, executed action $a = c\tanh(u) + b$, and Hopper bounds ($c = 1$, $b = 0$),

$$\log \pi^E(a \mid s) = \sum_{i=1}^{|\mathcal{A}|} \Big[\log \mathcal{N}\big(u_i;\, \mu_i(s),\, \sigma_i(s)^2\big) - \log\big(c_i(1 - \tanh^2(u_i)) + \epsilon\big) \Big], \qquad \text{(B.1)}$$

with $u = \operatorname{atanh}(a)$ (clamped), $\log \sigma \in [-5, 2]$, and $\epsilon = 10^{-6}$. Preference labels compare this density at queried (a_1, a_2).

B.6 Oracle Checkpoint Selection

Ten independent expert runs (seeds 1–10, 10^6 steps) yield last-100 means μ_s. Oracle seed $s^\star = \arg\min_s |\mu_s - \text{IQM}(\mu_1, \ldots, \mu_{10})|$ (tie-break: lower σ_s). Selected: seed 2, $\mu_2 = 2713.12$, IQM = 2808.18.

Table 17. Oracle selection (Hopper-v4 expert runs, last-100 episode means).

Seed	μ_s	σ_s	$\lvert\mu_s - \text{IQM}\rvert$	Note
1	3019.60	799.59	211.43	
2	2713.12	934.66	95.06	**SELECTED**
3	1745.02	785.92	1063.16	
4	2981.52	593.28	173.34	
5	1531.91	213.97	1276.27	
6	3125.52	958.07	317.34	
7	3188.83	636.18	380.65	
8	3125.65	718.18	317.48	
9	1883.65	614.98	924.53	
10	3133.00	854.55	324.82	
IQM of ten seed means:				2808.18

Frozen checkpoint: the seed-2 expert model checkpoint selected by the criterion above.

REFERENCES CITED

Agarwal, R., Schwarzer, M., Castro, P. S., Courville, A., & Bellemare, M. G. (2021). Deep reinforcement learning at the edge of the statistical precipice. In *Advances in neural information processing systems* (Vol. 34, pp. 29304–29320). Retrieved from https://proceedings.neurips.cc/paper/2021/hash/f514cec81cb148559cf475e7426eed5e-Abstract.html

Arthur, D., & Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual acm-siam symposium on discrete algorithms* (pp. 1027–1035). Philadelphia, PA, USA: Society for Industrial and Applied Mathematics. Retrieved from https://dl.acm.org/doi/10.5555/1283383.1283494

Ash, J. T., Zhang, C., Krishnamurthy, A., Langford, J., & Agarwal, A. (2020). Deep batch active learning by diverse, uncertain gradient lower bounds. In *International conference on learning representations.*

Biyik, E., & Sadigh, D. (2018). Batch active preference-based learning of reward functions. In A. Billard, A. Dragan, J. Peters, & J. Morimoto (Eds.), *Proceedings of the 2nd conference on robot learning* (Vol. 87, pp. 519–528). PMLR. Retrieved from https://proceedings.mlr.press/v87/biyik18a.html

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. In *Advances in neural information processing systems* (Vol. 30, pp. 4299–4307). Retrieved from https://proceedings.neurips.cc/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html

Gal, Y., Islam, R., & Ghahramani, Z. (2017). Deep Bayesian active learning with image data. In D. Precup & Y. W. Teh (Eds.), *Proceedings of the 34th international conference on machine learning* (Vol. 70, pp. 1183–1192). PMLR. Retrieved from https://proceedings.mlr.press/v70/gal17a.html

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In J. Dy & A. Krause (Eds.), *Proceedings of the 35th international conference on machine learning* (Vol. 80, pp. 1861–1870). PMLR. Retrieved from https://proceedings.mlr.press/v80/haarnoja18b.html

Hoi, S. C. H., Jin, R., Zhu, J., & Lyu, M. R. (2006). Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd international conference on machine learning* (pp. 417–424). ACM.

Huang, S., Dossa, R. F. J., Ye, C., Braga, J., Chakraborty, D., Mehta, K., & Araújo, J. G. M. (2022). Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, *23*(274), 1–18. Retrieved from http://jmlr.org/papers/v23/21-1342.html

Kirsch, A., van Amersfoort, J., & Gal, Y. (2019). BatchBALD: Efficient and diverse batch acquisition for deep bayesian active learning. In *Advances in neural information processing systems* (Vol. 32, pp. 7026–7037). Retrieved from https://proceedings.neurips.cc/paper/2019/hash/95323660ed2124450caaac2c46b5ed90-Abstract.html

Lee, K., Smith, L., Dragan, A., & Abbeel, P. (2021). B-Pref: Benchmarking preference-based reinforcement learning. In *Advances in neural information processing systems* (Vol. 34). Retrieved from https://openreview.net/forum?id=ps95-mkHF_

Lee, K., Smith, L. M., & Abbeel, P. (2021). PEBBLE: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th international conference on machine learning* (Vol. 139, pp. 6152–6163). PMLR. Retrieved from https://proceedings.mlr.press/v139/lee21i.html

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., ... Lowe, R. (2022). Training language models to follow instructions with human feedback. In *Advances in neural information processing systems* (Vol. 35). Retrieved from https://proceedings.neurips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html

Sadigh, D., Dragan, A. D., Sastry, S. S., & Seshia, S. A. (2017, July). Active preference-based learning of reward functions. In *Proceedings of robotics: Science and systems.* Cambridge, Massachusetts. Retrieved from http://roboticsproceedings.org/rss13/p53.html doi: 10.15607/RSS.2017.XIII.053

Settles, B. (2009). *Active learning literature survey* (Computer Sciences Technical Report No. 1648). University of Wisconsin–Madison.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). Cambridge, MA: MIT Press.

Todorov, E., Erez, T., & Tassa, Y. (2012). MuJoCo: A physics engine for model-based control. In *2012 ieee/rsj international conference on intelligent robots and systems* (pp. 5026–5033).

Towers, M., Terry, J. K., Kwiatkowski, A., Balis, J. U., de Cola, G., Deleu, T., ... Younis, O. G. (2023, March). *Gymnasium.* Zenodo. Retrieved from `https://zenodo.org/records/8127026` doi: 10.5281/zenodo.8127026

Wirth, C., Akrour, R., Neumann, G., & Fürnkranz, J. (2017). A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, *18*(136), 1–46. Retrieved from `https://jmlr.org/papers/v18/16-634.html`

Zhdanov, F. (2019). Diverse mini-batch active learning. *arXiv preprint arXiv:1901.05954*.