INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2026IPPAS007

Thèse de doctorat

A Systematic Study of Offline Reinforcement Learning: Methodological Evolution and Algorithmic Design

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom SudParis

École doctorale n°626 Ecole doctorale de l'Institut Polytechnique de Paris (ED IP Paris)
Spécialité de doctorat: Informatique

Thèse présentée et soutenue à Palaiseau, le 26/06/2026, par

XINCHEN HAN

Composition du Jury :

Salima Benbernou Professeur, Université Paris Cité	Présidente
Lyes Khoukhi Professeur, Sorbonne Université	Rapporteur
Fabrice Mourlin Maître de conférences, Université Paris-Est Créteil	Rapporteur
Pascal Lorenz Professeur, Université de Haute-Alsace	Examinateur
Nadjib Aitsaadi Professeur, Université Paris-Saclay	Examinateur
Hossam Afifi Professeur, Institut Polytechnique de Paris, Télécom SudParis	Directeur de thèse
Michel Marot Professeur, Institut Polytechnique de Paris, Télécom SudParis	Co-directeur de thèse

Declaration

I hereby declare that this thesis represents my own work which has been done after registration for the degree of PhD at Institut Polytechnique de Paris, and has not been previously included in a thesis or dissertation submitted to this or any other institution for a degree, diploma or other qualifications.

I have read the University's current research ethics guidelines, and accept responsibility for the conduct of the procedures in accordance with the University's Committee on the Use of Human and Animal Subjects in Teaching and Research (HASC). I have attempted to identify all the risks related to this research that may arise in conducting this research, obtained the relevant ethical and/or safety approval (where applicable), and acknowledged my obligations and the rights of the participants.

Xinchen Han

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my motherland, China, whose support has given me the opportunity to pursue academic research in France with stability, confidence, and a deep sense of responsibility.

I am profoundly grateful to my parents and to my fiancée. Throughout this journey, especially during difficult moments, you have given me warmth, courage, and unconditional love in every possible way. Your trust and support have been a constant source of strength, and this thesis would not have been possible without you.

I would also like to sincerely thank my supervisors for their guidance, encouragement, and patience throughout my PhD. Their insightful guidance, constructive feedback, and constant encouragement have shaped the way I think about research.

I am equally grateful to my friends, colleagues, and all the people who supported, accompanied, and encouraged me during these years. Every discussion, act of kindness, and word of encouragement has meant a great deal to me.

Finally, I would like to extend my warmest wishes to everyone who has accompanied me on this journey, and also to myself. May we all remain healthy, happy, fortunate, and surrounded by love and peace.

Abstract

Offline Reinforcement Learning (offline RL) seeks to learn decision-making policies from fixed datasets without further interaction with the environment. It is especially appealing in domains where online collection is costly, risky, or infeasible. However, offline RL faces fundamental challenges, including distribution shift, out-of-distribution (OOD) actions, and extrapolation error, which make reliable policy evaluation and improvement difficult. This thesis studies model-free offline RL with two goals: to understand its methodological evolution and to develop more effective offline RL algorithms.

First, this thesis revisits offline RL through the lens of the deadly triad and analyzes its central challenges. Building on this perspective, it develops an algorithmic taxonomy of offline RL and traces the evolution of offline RL design principles. Rather than treating existing methods as isolated categories, the thesis highlights how major algorithmic families emerge and evolve in response to challenges such as value overestimation and distribution shift.

Second, to address the tension between in-sample learning and multi-step learning, Projective Implicit Q-Learning (PIQL) is proposed. PIQL replaces the fixed expectile in Implicit Q-Learning with a projection-based adaptive parameter, enabling a multi-step interpretation while preserving the in-sample learning property. In addition, PIQL adopts a support-constrained policy-improvement objective that is better aligned with the policy-evaluation stage. Theoretical analysis establishes monotonic policy improvement and a progressively stricter criterion for advantageous actions, while experiments on standard benchmarks demonstrate strong performance.

Third, the thesis introduces the Continuous Constraint Interpolation (CCI) framework and the Automatic Constraint Policy Optimization (ACPO) algorithm to unify and adapt policy constraints. CCI provides a unified optimization view in which weighted behavior cloning (wBC), KL-based density regularization, and support constraints arise as special cases along a continuous constraint spectrum. Building on CCI, ACPO formulates adaptive constraint learning as a primal-dual optimization problem and automatically tunes the interpolation parameter. The thesis also derives a maximum-entropy performance difference lemma and performance lower bounds for both the optimal policy and its parametric projection. Experiments show that ACPO achieves strong and consistent results across diverse offline RL benchmarks.

Fourth, this thesis studies failure modes of CVAE-based offline RL and identifies latent action projection space collapse, a phenomenon whereby posterior collapse weakens downstream policy optimization. Based on this analysis, the thesis proposes Expand Latent Action Projection SpacE (ELAPSE), a simple yet effective method that enlarges the projection space. ELAPSE improves the cooperation between generative behavior modeling and policy optimization, and empirical results show that it substantially enhances the performance of representative CVAE-based offline RL methods.

Overall, this thesis studies model-free offline RL from both methodological and algorithmic perspectives. At the analytical level, it clarifies the key challenges and the methodological evolution of offline RL. At the algorithmic level, it develops new methods for policy evaluation, policy improvement, and generative behavior modeling. Together, these contributions advance offline RL toward more reliable, adaptive, and theoretically grounded decision-making from static datasets.

Résumé

L'apprentissage par renforcement hors ligne (offline RL) vise à apprendre des politiques de décision à partir de données fixes, sans interaction avec l'environnement. Il est pertinent lorsque la collecte en ligne est coûteuse, risquée ou impossible. Toutefois, l'offline RL se heurte à des défis fondamentaux, notamment le décalage de distribution, les actions hors distribution (out-of-distribution, OOD) et l'erreur d'extrapolation, qui compliquent l'évaluation et l'amélioration fiables des politiques. Cette thèse étudie l'offline RL sans modèle avec deux objectifs : comprendre son évolution méthodologique et développer des algorithmes plus efficaces.

Premièrement, cette thèse réexamine l'offline RL à la lumière de la deadly triad et en analyse les défis. Sur cette base, elle développe une taxonomie algorithmique de l'offline RL et retrace l'évolution de ses principes de conception. Plutôt que de considérer les méthodes existantes comme des catégories isolées, la thèse montre comment les familles algorithmiques émergent et évoluent face à des défis tels que la surestimation des valeurs et le décalage de distribution.

Deuxièmement, afin de répondre à la tension entre l'apprentissage in-sample et l'apprentissage multi-pas, Projective Implicit Q-Learning (PIQL) est proposé. PIQL remplace l'expectile fixe d'Implicit Q-Learning par un paramètre adaptatif fondé sur la projection, autorisant une interprétation multi-pas tout en préservant la propriété d'apprentissage in-sample. En outre, PIQL adopte un objectif d'amélioration de politique sous contrainte de support, mieux aligné sur l'étape d'évaluation. L'analyse théorique établit une amélioration monotone de la politique ainsi qu'un critère plus strict pour les actions avantageuses, tandis que les expériences sur des benchmarks montrent de bonnes performances.

Troisièmement, la thèse introduit le cadre Continuous Constraint Interpolation (CCI) et l'algorithme Automatic Constraint Policy Optimization (ACPO) pour unifier et adapter les contraintes de politique. CCI fournit une vision unifiée de l'optimisation où le clonage comportemental pondéré (weighted behavior cloning, wBC), la régularisation de densité fondée sur la divergence de Kullback-Leibler et les contraintes de support apparaissent comme des cas particuliers d'un spectre continu. Sur cette base, ACPO formalise l'apprentissage adaptatif des contraintes comme un problème primal-dual et ajuste automatiquement le paramètre d'interpolation. La thèse établit aussi un lemme de différence

de performance à entropie maximale ainsi que des bornes inférieures pour la politique optimale et sa projection paramétrique. Les expériences montrent qu'ACPO obtient des résultats cohérents sur divers benchmarks d'offline RL.

Quatrièmement, cette thèse étudie les modes de défaillance des méthodes d'offline RL fondées sur les CVAE et identifie le phénomène de latent action projection space collapse, par lequel l'effondrement du posterior affaiblit l'optimisation de politique en aval. Sur cette base, la thèse propose Expand Latent Action Projection SpacE (ELAPSE), une méthode qui élargit l'espace de projection. ELAPSE améliore l'articulation entre la modélisation générative du comportement et l'optimisation de politique, et les résultats empiriques montrent qu'elle améliore substantiellement les performances de méthodes représentatives d'offline RL fondées sur les CVAE.

Dans l'ensemble, cette thèse étudie l'offline RL sans modèle sous des angles méthodologique et algorithmique. Sur le plan analytique, elle clarifie les défis et l'évolution méthodologique de l'offline RL. Sur le plan algorithmique, elle développe de nouvelles méthodes pour l'évaluation et l'amélioration de politique ainsi que pour la modélisation générative du comportement. Ensemble, ces contributions font progresser l'offline RL vers une prise de décision sur données statiques plus fiable, plus adaptative et mieux fondée théoriquement.

Contents

Acknowledgements **3**

Abstract **4**

Résumé **6**

1 General Introduction **17**
1.1 Motivation . . . 17
1.2 Thesis Scope and Goal . . . 20
1.3 Contributions . . . 21
1.4 Thesis Organization . . . 22

2 Offline RL: Fundamentals, Core Challenges and Methodology **24**
2.1 Offline RL Fundamentals . . . 24
2.2 Core Challenges in Offline RL . . . 28
2.3 Taxonomy of Offline RL Algorithms . . . 32
2.4 Evolution of Offline RL Design Principles . . . 39
2.5 Summary and Connections to the Subsequent Chapters . . . 47

3 Projective Implicit Q-Learning with Support Constraint **48**
3.1 Introduction . . . 48
3.2 Related Work . . . 49
3.3 Preliminaries . . . 50
3.4 Proposed Algorithm – PIQL . . . 52
3.5 Experiments . . . 59
3.6 Conclusion . . . 64

4 Automatic Constraint Policy Optimization based on the Continuous Constraint Interpolation Framework **65**
4.1 Introduction . . . 66
4.2 Related Work . . . 67

4.3 Preliminaries . . . 68
4.4 Proposed Algorithm – ACPO . . . 71
4.5 Experiments . . . 76
4.6 Conclusion . . . 81

5 Expand Latent Action Projection Space for Policy Optimization 82
5.1 Introduction . . . 82
5.2 Related Work . . . 84
5.3 Preliminaries . . . 84
5.4 Problem Statement . . . 85
5.5 Proposed Algorithm – ELAPSE . . . 91
5.6 Experiments . . . 94
5.7 Conclusion . . . 101

6 General Conclusion and Future Work 103
6.1 General Conclusion . . . 103
6.2 Future Work . . . 105

7 Appendix 107
7.1 Proofs – PIQL . . . 107
7.2 Experimental Details – PIQL . . . 114
7.3 Proofs – ACPO . . . 115
7.4 Experimental Details – ACPO . . . 125
7.5 Proofs – ELAPSE . . . 130
7.6 Experimental Details – ELAPSE . . . 133

List of Tables

3.1 Averaged normalized scores of PIQL on Gym-MuJoCo-v2 datasets. Scores are reported as mean ± standard deviation. **Bold** indicates the top-2 performance in each task. . . . 60
3.2 Averaged normalized scores of PIQL on AntMaze-v0 and Kitchen-v0 datasets. Scores are reported as mean ± standard deviation. **Bold** indicates the top-2 performance in each task. . . . 60
3.3 Averaged normalized scores of PIQL on NeoRL2 datasets. Scores are reported as mean ± standard deviation. **Bold** indicates the top-2 performance in each task. . . . 63

4.1 Constraint spectrum induced by the interpolation parameter λ in the CCI framework. . . . 72
4.2 Averaged normalized scores of ACPO on Gym-MuJoCo-v2, AntMaze-v2, and Kitchen-v0 datasets over five seeds. Scores are reported as mean ± standard deviation. **Bold** indicates the best performance in each task. . . . 77
4.3 Averaged normalized scores of ACPO on NeoRL2 benchmark over three seeds. Scores are reported as mean ± standard deviation. **Bold** indicates the best performance in each task. . . . 78
4.4 Sensitivity Analysis of ϵ. . . . 81

5.1 Normalized scores of PLAS, LAPO, SPOT, and their ELAPSE-enhanced versions on the Gym-MuJoCo and AntMaze D4RL datasets. **Bold** indicates that the ELAPSE-enhanced version outperforms the corresponding vanilla method. . . . 98
5.2 The normalized score of ELAPSE-LAPO and other strong baselines. We indicate in **Bold** that the scores of ELAPSE-LAPO are the top two among all algorithms. . . . 99
5.3 Numbers of samples in the D4RL Hopper datasets. . . . 100

7.1 Hyperparameters in PIQL . . . 114
7.2 D4RL hyperparameters in ACPO, where s_{dim} and a_{dim} are the state and action dimensions, respectively. . . . 125

7.3 Dataset abbreviations for D4RL. 126
7.4 NeoRL2 hyperparameters. Architectures are denoted by MLP layer sizes, where s_{dim} and a_{dim} are the state and action dimensions, respectively. 127

List of Figures

1.1 The Thesis Organization. 22

2.1 Illustration of different RL paradigms: (a) on-policy RL, (b) off-policy RL, and (c) offline RL. This figure is adapted from [1]. 28
2.2 The Deadly Triad. 29
2.3 The Taxonomy Tree of Offline RL. 33
2.4 From behavior cloning to support-constraint optimization: the evolution of policy improvement paradigms in offline RL. 39
2.5 From Pessimism to Selective Optimism: the Evolution of Value Estimation Paradigms in Offline RL 41
2.6 From Bellman Backups to Trajectory-level Planning: Bellman backups, Transformer-based method, and Diffusion Planning. 42
2.7 From Unimodal Gaussian Policies to Expressive Multimodal Policies. 43
2.8 From Fixed Penalties to Adaptive Constraints in Offline RL. 44
2.9 From Action-level Behavior Constraints to State Correction in Offline RL. . 46
2.10 From Shallow Approximators to Scalable Offline RL. 46

3.1 Sensitivity of the Parameter $\tau \in \{0.3, 0.6, 0.9\}$ across 10 D4RL Tasks. 53
3.2 The training curves of normalized score and $\tau_{\text{proj}}(a|s)$ on AntMaze-v0 and Kitchen-v0 datasets. 62
3.3 The training curves of normalized score and $\tau_{\text{proj}}(a \mid s)$ on Kitchen-v0 datasets under different batch sizes. 63

4.1 Special-case comparisons and the evolution of the interpolation parameter λ during ACPO training. 78
4.2 Normalized score differences on Gym-MuJoCo datasets. 79
4.3 Distributions of $\log \pi_\beta(a|s)$ under Gaussian and CVAE behavior estimators on in-dataset actions and OOD actions. 80

5.1 Training flow and latent action projection space collapse in PLAS, LAPO, and SPOT. 86

5.2 Visualization of latent action projection space collapse in PLAS. By default, PLAS defines $z = \pi_\theta(s) \in [-2, 2]$. Accordingly, we also restrict each latent coordinate to $z_i \in [-2, 2]$, $i = 1, 2, 3$. 87
5.3 The $\mathcal{L}_R^\times$ and $\mathcal{L}_D^\times$ curves of PLAS. 88
5.4 The $\mathcal{L}_R^\times$ and $\mathcal{L}_D^\times$ curves of LAPO. 90
5.5 The $\mathcal{L}_R^\times$ and $\mathcal{L}_D^\times$ curves of SPOT. 91
5.6 ELAPSE-PLAS with different BN weights w. 93
5.7 Normalized scores of the random policy and the optimized policy in PLAS without perturbation and in ELAPSE-PLAS. 96
5.8 Sample-complexity results of ELAPSE-LAPO on the Hopper datasets. . . . 100
5.9 Comparison results with freeze w=0.2, freeze b=0.2, KL cost annealing, $\beta = 0.1, 1$ in β-VAE, and cyclical annealing. 101

7.1 Gaussian and CVAE behavior models under $\xi \in \{0.1, 0.3, 0.5, 0.8\}$. 128
7.2 NeoRL2 Learning Curves. 128
7.3 D4RL Learning Curves. 129
7.4 The training curves of ELAPSE-PLAS, ELAPSE-LAPO, and ELAPSE-SPOT algorithms on D4RL datasets. 136
7.5 The training curves of ELAPSE-LAPO on Antmaze datasets. 136

List of Algorithms

1 Projective Implicit Q-Learning (PIQL) 59
2 Automatic Constraint Policy Optimization (ACPO) 74
3 Expand Latent Action Projection SpacE (ELAPSE) 95
4 ELAPSE-LN 131
5 ELAPSE-TR 133

List of Acronyms

ACPO Automatic Constraint Policy Optimization

AE Autoencoder

Batch RL Batch Reinforcement Learning

BC Behavior Cloning

BN Batch Normalization

CCI Continuous Constraint Interpolation

CVAE Conditional Variational Autoencoder

D4RL Datasets for Deep Data-Driven Reinforcement Learning

DICE Distribution Correction Estimation

DP Dynamic Programming

DRL Deep Reinforcement Learning

ELAPSE Expand Latent Action Projection SpacE

ELBO Evidence Lower Bound

GAN Generative Adversarial Network

IPM Integral Probability Metric

IS Importance Sampling

IWAE Importance Weighted Autoencoder

KL Kullback-Leibler divergence

LLM Large Language Model

MC Monte Carlo

MDP Markov Decision Process

MMD Maximum Mean Discrepancy

MSE Mean Squared Error

offline RL Offline Reinforcement Learning

OOD Out-of-Distribution

OPE Off-Policy Evaluation

OT Optimal Transport

PIQL Projective Implicit Q-Learning

RL Reinforcement Learning

SNIS Self-Normalized Importance Sampling

TD Temporal Difference

TV Total Variation

VAE Variational Autoencoder

wBC weighted Behavior Cloning

Chapter 1

General Introduction

1.1 Motivation

Reinforcement Learning (RL) studies how to learn a mapping from observations to actions so as to maximize a cumulative reward signal [2]. The field has evolved rapidly since the 1950s, when Richard Bellman's work on Dynamic Programming (DP) established core principles underlying numerous modern RL algorithms [3]. Over the past decades, the combination of deep representation learning and RL has driven remarkable progress across a wide range of domains, including games [4], Large Language Models (LLMs) [5], recommender systems [6], and many others [7, 8]. These developments highlight RL as a central paradigm for sequential decision making.

Despite this broad success, most RL algorithms are fundamentally designed for the online learning paradigm, in which an agent iteratively interacts with the environment to collect experience and improve its policy. Conceptually, online RL methods are often categorized into on-policy algorithms, which learn from data generated by the current policy, and off-policy algorithms, which can in principle reuse data collected under different policies. However, online interaction is often impractical in real-world settings where data collection is expensive (e.g., large-scale industrial systems), safety-critical or hazardous (e.g., autonomous driving and healthcare), constrained by ethical or regulatory requirements (e.g., medical treatment and education), or where high-fidelity simulators are unavailable or unreliable.

Offline RL has emerged as a compelling alternative to the online interaction paradigm, aiming to learn a policy solely from a fixed dataset collected by possibly unknown or ill-defined behavior policies. Early literature often referred to this setting as *batch RL* [9]. Lange *et al.* [10] later provided a systematic definition: *"Batch RL refers to RL from a fixed batch of previously collected experience, without further interaction with the environment."* The term *offline RL* has since become prevalent, and the tutorial by Levine *et al.* [1] further clarified the standard protocol: *"The dataset is collected once, and is not altered during training,*

which makes it feasible to use large previous collected datasets. The training process does not interact with the MDP at all, and the policy is only deployed after being fully trained."

While this data-driven paradigm enables learning from large-scale logged experience, it also introduces substantial challenges. *Distribution shift* [1, 11, 12] is almost inevitable in offline RL. The root cause is that offline RL typically aims to learn a policy that outperforms the behavior policy used to collect the dataset. Once the learned policy deviates from the support of the behavior policy, the offline data distribution no longer matches the visitation distribution induced by the learned policy, making policy evaluation systematically biased and unreliable.

From another perspective, this difficulty reflects a trade-off between mode-seeking and mode-covering. Mode-seeking optimization emphasizes moving toward high-value regions, which can exacerbate distribution shift by "sharpening" the learned policy [13, 14]. In contrast, mode-covering approaches prioritize matching or covering the data distribution and can mitigate distribution shift, but imitation- or supervision-based objectives may still yield suboptimal performance because the behavior policy is unknown and the data quality is often heterogeneous [15, 16].

More concretely, distribution shift encourages the learned policy to select out-of-distribution (OOD) actions, leading to *extrapolation error*, i.e., the overestimation of Q-values for OOD actions [17]. Moreover, policy improvement commonly selects actions by maximizing the learned Q-function, while policy evaluation relies on bootstrapped updates involving the $\max Q$ operator. When Q-values are overestimated in OOD regions, such errors can be amplified and propagated backward through Temporal-Difference (TD) backups, resulting in systematic overestimation and potentially unstable value learning [18] [1].

To address these issues, a broad range of offline RL algorithms have been developed. In brief, from a DP perspective, offline RL methods can be organized into policy iteration and value iteration,

- Policy Iteration.
  - *Policy Improvement* step updates the policy by optimizing it with respect to the estimated value function or Q-function. In offline RL, this step is typically combined with policy constraints or regularization to prevent the learned policy from deviating excessively from the behavior policy [19–22].
  - *Policy Evaluation* step estimates the value function or Q-function of the learned policy. In offline RL, this step is often augmented with conservative value learning to alleviate extrapolation-induced overestimation [23–26].
- Value Iteration.

[1] A more detailed discussion of the core challenges in offline RL is provided in Chapter 2.2.

- *One-Step* approaches optimize the learned policy based on the behavior value function [27–29], i.e., policy evaluation is performed solely from offline data.
- *Multi-Step* approaches leverage more flexible backups but often require additional policy constraints or conservatism mechanisms to control OOD risks [14, 30] [2].

Despite the rapid progress in offline RL, several critical issues are not yet fully understood and remain insufficiently addressed. Motivated by the above discussion, this thesis is organized around the following four research questions:

1. Offline RL has witnessed rapid methodological diversification, and several influential survey papers have provided broad overviews and taxonomies of existing approaches [1, 31–33]. However, these surveys are relatively dated and primarily emphasize horizontal categorization and comparison across methods. Less attention has been paid to a longitudinal perspective that examines how algorithm design in offline RL has evolved over time, including its emerging principles, shifting emphases, and underlying developmental trends. As a result, the field still lacks a coherent macro-level account of how recent offline RL methods relate to one another in terms of design evolution. Therefore, the first critical research question emerges:

   > *RQ*1. *How to develop a finer-grained taxonomy of recent offline RL methods and, on that basis, distill the macro-level evolution of offline RL algorithm design?*

2. In-sample learning [23, 27, 29, 34] mitigates OOD risks by grounding both policy evaluation and improvement on the dataset, but it can overly restrict policy improvement beyond the dataset. In contrast, multi-step learning [14] leverages multi-step backups or rollouts that are more tightly coupled to the learned policy, potentially enlarging the improvement space while introducing additional OOD risks and extrapolation-induced instability. Therefore, the second critical research question emerges:

   > *RQ*2. *How to reconcile in-sample learning and multi-step learning to achieve safe and effective policy evaluation and improvement?*

3. Mainstream policy-constrained offline RL methods can be broadly organized along a common constraint spectrum: weighted Behavior Cloning (wBC) [35–38], density regularization (often KL-based) [19–22], and support constraints [13, 14, 18, 39]. Overly conservative constraints may stabilize value learning but undermine the core promise of offline RL – policy improvement beyond the dataset; overly weak

[2] A more comprehensive taxonomy and comparison of offline RL methods is provided in Chapter 2.3.

constraints fail to prevent OOD actions and can induce severe value misestimation that misguides optimization [25, 26]. Therefore, the third critical research question emerges:

> *RQ3. How to develop a unified framework that bridges different constraint families and automatically tunes constraint type and strength?*

4. Generative models such as Conditional Variational AutoEncoders (CVAE) [40] are widely used in offline RL for behavior modeling and candidate-action generation. However, CVAE can suffer from posterior collapse, where the latent variables become uninformative and the decoder effectively ignores the stochastic latent code, behaving similarly to a deterministic autoencoder. Therefore, the fourth critical research question emerges:

> *RQ4. How to identify and mitigate the failure modes that arise when integrating CVAE into offline RL to maximize policy improvement?*

Collectively, these four research questions $RQ1 - 4$ articulate the central motivation and scope of this thesis.

1.2 Thesis Scope and Goal

Building on the four research questions outlined above, this thesis focuses on model-free offline RL and clarifies its scope and overall goal accordingly. Broadly, existing offline RL methods can be divided into model-based and model-free approaches. Model-based offline RL first fits an explicit or implicit dynamics model from the offline dataset and then optimizes a policy within the learned model. While appealing, model-based methods can be highly sensitive to distribution shift, since model errors may accumulate when the learned policy induces trajectories that deviate from the data distribution [1]. In contrast, model-free offline RL avoids explicit dynamics modeling and performs policy evaluation and improvement directly from the dataset. This thesis therefore focuses on model-free offline RL.[3]

Our primary goal is to study the methodological evolution of offline RL and, based on this understanding, develop more efficient and adaptive offline RL algorithms.

[3]For simplicity, we use the term offline RL throughout to refer specifically to model-free offline RL.

In particular, this thesis focuses on four closely related directions: (i) developing a finer-grained taxonomy of recent offline RL methods and uncovering the macro-level evolution of offline RL algorithm design; (ii) reconciling in-sample learning with multi-step learning so as to balance richer learning signals with reduced OOD risks; (iii) jointly optimizing the constraint types and the constraint strength in policy improvement, enabling flexible yet adaptive control of conservatism across datasets and tasks; and (iv) identifying and mitigating key failure modes that arise when integrating CVAE-based generative models into offline RL, with a particular focus on posterior collapse and its impact on policy improvement.

Together, these directions define the conceptual scope and methodological objectives of this thesis.

1.3 Contributions

Guided by the research questions and thesis goal, the main contributions of this thesis are summarized as follows. Specifically, $C1$ provides a unified problem perspective that underpins the entire thesis, and addresses $RQ1$ by offering a refined taxonomy and evolutionary analysis of offline RL methods, and $C2$–$C4$ address $RQ2$–$RQ4$, respectively, through corresponding algorithmic design.

- $C1$. To address $RQ1$, we provide a unified analytical framework for understanding the methodological evolution of offline RL. Through the lens of the classical deadly triad in RL – function approximation, bootstrapping, and off-policy learning – we analyze how distribution shift and extrapolation error emerge as central sources of value misestimation and unstable policy optimization in the offline setting. Building on this perspective, we develop an algorithmic-perspective taxonomy of offline RL methods and distill a set of macro-level evolutionary insights and design principles that clarify how major method families arise and evolve in response to these fundamental challenges.

- $C2$. To address $RQ2$, we propose Projective Implicit Q-Learning (PIQL), which reconciles in-sample learning with multi-step learning in the policy evaluation stage. In the policy improvement stage, PIQL incorporates a support constraint to keep policy updates close to the behavior support, thereby enabling safer and more effective policy improvement.

- $C3$. To address $RQ3$, we develop a maximum-entropy-based unified optimization framework, Continuous Constraint Interpolation (CCI), which exposes a single interpolation parameter that enables smooth transitions and principled combinations among wBC, KL-based density regularization, and support constraints for policy improvement. Building on CCI, we further propose Automatic Constraint Policy

Optimization (ACPO), which automatically optimizes the interpolation parameter via a Lagrangian dual update, thereby eliminating the need to manually select fixed constraint forms and strengths.

- *C*4. To address *RQ*4, we analyze the consequences of latent action projection space collapse in CVAE-based offline RL algorithms, and characterize how such degeneracy affects policy improvement. We then propose an easy-to-implement algorithm, Expand Latent Action Projection SpacE (ELAPSE), which not only encourages the CVAE to approximate the behavior policy but also enhances the generative coverage of dataset-supported actions, thereby mitigating collapse and improving policy optimization.

1.4 Thesis Organization

The thesis is organized in a progressive manner from problem formulation and methodological analysis to algorithmic development, as illustrated in Fig. 1.1.

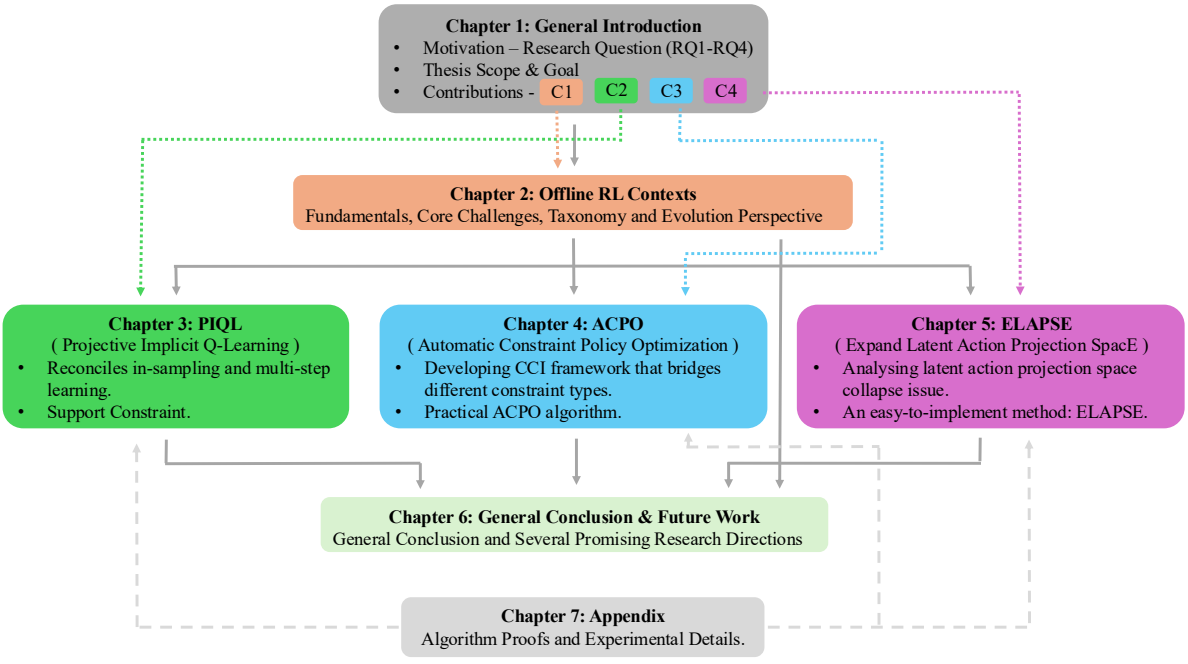


Figure 1.1: The Thesis Organization.

Chapter 1 presents the thesis motivation, formulates the four research questions, and defines the scope, goals, and main contributions.

Chapter 2 mainly corresponds to contributions *C*1. It first introduces the fundamentals of offline RL, including the definition of MDPs, Bellman operators, and the offline RL setting. It then discusses the core challenges in offline RL, including distribution shift and extrapolation error, and analyzes their underlying causes and consequences from the perspective of the deadly triad. Based on this understanding, it further reviews the methodological landscape of offline RL, including a taxonomy of existing methods and a macro-level evolutionary perspective on algorithm design. This chapter serves as the conceptual foundation for the later technical studies.

Chapter 3 mainly focuses on contribution *C*2. It proposes Projective Implicit Q-Learning (PIQL), which reconciles in-sample learning and multi-step learning. In the policy evaluation stage, PIQL incorporates a projection step that constrains the learned Q-function to remain consistent with the dataset distribution. In the policy improvement stage, PIQL incorporates a support constraint to keep policy updates close to the behavior support. This design is more aligned with the policy evaluation stage and can better leverage the learned Q-function, thereby improving the effectiveness of offline policy improvement.

Chapter 4 mainly focuses on contribution *C*3. It first develops Continuous Constraint Interpolation (CCI), a unified optimization framework in which the three major constraint families arise as special cases along a common constraint spectrum. The CCI framework introduces a single interpolation parameter that enables smooth transitions and principled combinations across constraint types. Building on CCI, this chapter then develops Automatic Constraint Policy Optimization (ACPO), a practical primal–dual algorithm that adapts the interpolation parameter via a Lagrangian dual update.

Chapter 5 mainly focuses on contribution *C*4. It analyzes the failure modes arising from the use of CVAE in offline RL, with a particular focus on a phenomenon termed latent action projection space collapse. It then proposes ELAPSE, an easy-to-implement algorithm that balances the trade-off between CVAE regularization and policy optimization while avoiding unnecessary perturbations introduced in previous works.

Chapter 6 concludes the thesis by summarizing the main findings, contributions, and outlining several promising directions for future research.

Chapter 7 provides technical proofs, additional experimental details and results, and discussions that complement the main chapters.

Chapter 2

Offline RL: Fundamentals, Core Challenges and Methodology

This chapter focuses on the first research question ($RQ1$ in Section 1.1) of the thesis: how to develop a finer-grained taxonomy of recent offline RL methods and, on that basis, distill the macro-level evolution of offline RL algorithm design. Although a large number of offline RL methods have been proposed, they are often studied as isolated algorithmic variants, making it difficult to form a coherent view of how different design paradigms emerge and evolve in response to the fundamental challenges of the offline setting. To address this issue, this chapter develops a unified analytical perspective on offline RL through the lens of the deadly triad, and on this basis presents both a refined taxonomy of offline RL methods and an evolutionary view of their design principles.

The remainder of this chapter is organized as follows. Section 2.1 introduces the fundamentals of offline RL. Section 2.2 discusses the core challenges in offline RL. Section 2.3 presents a taxonomy of existing offline RL algorithms. Section 2.4 analyzes the macro-level evolution of offline RL design principles.

2.1 Offline RL Fundamentals

2.1.1 RL Preliminaries

Markov Decision Process. Conventionally, an RL problem is formulated as a Markov Decision Process (MDP) [2], specified by a tuple

$$\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, r, \gamma \rangle,$$

where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $\mathcal{T}$ represents the transition function, r is the reward function, and $\gamma \in [0, 1)$ is the discount factor.

The return G is defined as the discounted sum of rewards up to horizon T, where T can be finite (episodic-MDP) or infinite (continuous-MDP):

$$G \triangleq \sum_{t=0}^{T} \gamma^t r(s_t, a_t).$$

The goal of RL is to optimize a policy $\pi(a \mid s)$ so as to maximize the expected discounted return. Formally,

$$\pi^\star \in \arg\max_{\pi} \; J(\pi) \triangleq \mathbb{E}_{\tau \sim p_\pi(\tau)} \left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t) \right], \tag{2.1}$$

where $\tau = (s_0, a_0, s_1, a_1, \ldots)$ denotes a trajectory generated by following π in the MDP, and

$$p_\pi(\tau) = \rho_0(s_0) \prod_{t=0}^{T-1} \pi(a_t \mid s_t)\, \mathcal{T}(s_{t+1} \mid s_t, a_t)$$

is the trajectory distribution induced by π, with ρ_0 denoting the initial-state distribution.

Value Function. To facilitate policy evaluation and policy improvement, the action-value function (or Q-function) is defined as the expected cumulative discounted reward starting from a state-action pair (s, a):

$$Q^\pi(s, a) \triangleq \mathbb{E}_{\tau \sim p_\pi(\tau)} \Big[\sum_{t=0}^{T} \gamma^t r(s_t, a_t) \,\Big|\, s_0 = s,\; a_0 = a \Big]. \tag{2.2}$$

For a given state s, the value function (or V-function) is defined as

$$V^\pi(s) \triangleq \mathbb{E}_{\tau \sim p_\pi(\tau)} \Big[\sum_{t=0}^{T} \gamma^t r(s_t, a_t) \,\Big|\, s_0 = s \Big]. \tag{2.3}$$

Based on the definition of Q-function and V-function, we can derive recursive definitions for these value functions,

$$V^\pi(s) = \mathbb{E}_{a \sim \pi(\cdot|s)} \big[Q^\pi(s, a) \big], \tag{2.4}$$

$$Q^\pi(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim \mathcal{T}(\cdot|s,a)} \big[V^\pi(s') \big]. \tag{2.5}$$

Using the definitions of the return and the value functions, the V- and Q-functions satisfy the Bellman expectation equations. For any policy π, we have

$$V^\pi(s) = \mathbb{E}_{a \sim \pi(\cdot|s)} \Big[r(s, a) + \gamma \mathbb{E}_{s' \sim \mathcal{T}(\cdot|s,a)} \big[V^\pi(s') \big] \Big], \tag{2.6}$$

$$Q^\pi(s,a) = r(s,a) + \gamma \mathbb{E}_{s'\sim\mathcal{T}(\cdot|s,a)} \mathbb{E}_{a'\sim\pi(\cdot|s')}\big[Q^\pi(s',a')\big]. \tag{2.7}$$

Equivalently, Eq. (2.7) can be written using the Bellman expectation operator associated with policy π. For any function $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, define

$$\big(\mathcal{B}^\pi Q\big)(s,a) \triangleq r(s,a) + \gamma \mathbb{E}_{s'\sim\mathcal{T}(\cdot|s,a)} \mathbb{E}_{a'\sim\pi(\cdot|s')}\big[Q(s',a')\big]. \tag{2.8}$$

Then Q^π is a fixed point of $\mathcal{B}^\pi$, i.e., $Q^\pi = \mathcal{B}^\pi Q^\pi$.

Similarly, the Bellman optimality operator is defined as

$$\big(\mathcal{B}^\star Q\big)(s,a) \triangleq r(s,a) + \gamma \mathbb{E}_{s'\sim\mathcal{T}(\cdot|s,a)}\Big[\max_{a'\in\mathcal{A}} Q(s',a')\Big], \tag{2.9}$$

and the optimal action-value function satisfies the fixed-point equation $Q^\star = \mathcal{B}^\star Q^\star$.

Under the standard discounted setting ($\gamma \in [0,1)$), the Bellman expectation operator $\mathcal{B}^\pi$ is a γ-contraction in the sup-norm, implying that repeated application of $\mathcal{B}^\pi$ converges to its unique fixed point Q^π. Likewise, the Bellman optimality operator $\mathcal{B}^\star$ is also a γ-contraction, and iterating $\mathcal{B}^\star$ converges to $Q^\star$.

Moreover, the advantage function $A^\pi(s,a)$ measures the relative benefit of taking action a over the average action under π at state s:

$$A^\pi(s,a) \triangleq Q^\pi(s,a) - V^\pi(s). \tag{2.10}$$

Value Function Approximation. When the state-action space is large or continuous, explicitly computing and storing values for all state-action pairs is infeasible. A standard remedy is to employ parametric function approximation – most notably deep neural networks – to represent value functions and policies. In particular, we approximate the action-value function by $Q_\phi(s,a)$ with parameters ϕ, and learn it by minimizing the TD regression loss derived from the Bellman expectation equation. Given a fixed policy π, the TD loss is

$$\mathcal{L}_{TD}(\phi) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{B}}\Big[\big(r + \gamma \mathbb{E}_{a'\sim\pi(\cdot|s')}\big[Q_{\hat{\phi}}(s',a')\big] - Q_\phi(s,a)\big)^2\Big], \tag{2.11}$$

where $\mathcal{B}$ denotes the replay buffer in online settings, and $Q_{\hat{\phi}}$ is a target network used to stabilize learning.

Policy Extraction. Given an estimated action-value function Q_ϕ, a canonical way to obtain a policy in value-based RL is to act greedily (or softly greedily) with respect to Q_ϕ:

$$\pi_{\text{greedy}}(a \mid s) \in \arg\max_{a\in\mathcal{A}} Q_\phi(s,a), \qquad \text{or} \qquad \pi_{\text{soft}}(a \mid s) \propto \exp\Big(\tfrac{1}{\alpha} Q_\phi(s,a)\Big), \tag{2.12}$$

where $\alpha > 0$ is a temperature parameter. However, when $\mathcal{A}$ is continuous or high-dimensional, solving the arg max operator (or sampling from π_{soft}) can be nontrivial. A

common remedy is to introduce an explicit parametric policy (actor) $\pi_\theta(a \mid s)$ and optimize it directly using gradient-based methods, leading to actor-critic algorithms.

Concretely, we maximize the expected return in Eq. (2.1) under π_θ. Based on the policy gradient theorem [41],

$$\nabla_\theta J(\theta) = \mathbb{E}_{s\sim d^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)}\Big[\nabla_\theta \log \pi_\theta(a \mid s)\, Q^{\pi_\theta}(s,a)\Big], \tag{2.13}$$

where d^{π_θ} denotes the discounted state visitation distribution induced by π_θ. In practice, Q^{π_θ} is replaced by the learned critic Q_ϕ, yielding the actor update

$$\nabla_\theta J(\theta) \approx \mathbb{E}_{s\sim d^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)}\Big[\nabla_\theta \log \pi_\theta(a \mid s)\, Q_\phi(s,a)\Big]. \tag{2.14}$$

Classic RL algorithms. For discrete action spaces, value-based methods such as DQN [42] and its variants (e.g., Double DQN [43] and Rainbow [44]) are widely used. These methods typically derive a policy by acting greedily with respect to the learned Q-values, often combined with exploration mechanisms such as ϵ-greedy or softmax action selection.

For continuous action spaces, actor–critic methods are often more suitable, as they parameterize both the policy (actor) and the value function (critic). The actor can be optimized via stochastic policy gradients [41] (e.g., REINFORCE [45] and A3C [46]) or deterministic policy gradients [47] (e.g., DDPG [48] and TD3 [49]).

2.1.2 Offline RL Setting

Offline RL shares the same high-level goal as online RL: to learn a policy that maximizes the expected discounted return. The key distinction lies in the training protocol. In offline RL, the agent has no access to environment interaction during training; instead, learning is performed exclusively from a fixed dataset of previously collected transitions. Concretely, the offline dataset consists of a set of trajectories

$$\mathcal{D} = \{\tau_i\}_{i=1}^{N},$$

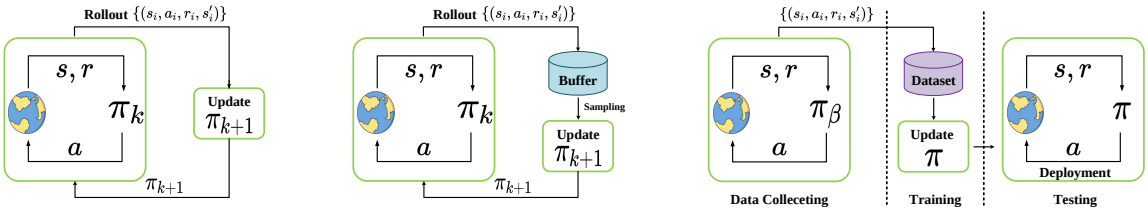
where each trajectory is given by

$$\tau_i = (s_0^i, a_0^i, r_0^i, s_1^i, a_1^i, r_1^i, \ldots, s_T^i),$$

and the dataset is collected by one or more behavior policies, denoted by π_β. In most practical scenarios, the exact form of π_β is unknown and can only be inferred implicitly from the logged data in $\mathcal{D}$.

Fig. 2.1 contrasts three learning paradigms by highlighting how data are generated and how policies are updated. (Online) on-policy RL alternates between collecting rollouts with the current policy and updating the policy using the newly collected trajectories (Fig. 2.1(a)). (Online) off-policy RL continuously gathers experience to populate a replay buffer, from which mini-batches are sampled for learning, enabling experience reuse

across different data-collecting policies (Fig. 2.1(b)). In offline RL, the dataset is collected once and then kept fixed: training proceeds only by sampling from $\mathcal{D}$, and the learned policy is deployed after training is complete, without further interaction (Fig. 2.1(c)).



(a) On-policy RL **(b) Off-policy RL** **(c) Offline RL**

Figure 2.1: Illustration of different RL paradigms: (a) on-policy RL, (b) off-policy RL, and (c) offline RL. This figure is adapted from [1].

A seemingly feasible idea is to directly apply standard off-policy RL algorithms in this offline setting. However, off-policy evaluation (OPE) typically relies on an "absolute continuity" assumption [50–52], which requires the behavior policy to assign non-zero probability to all state-action pairs that are relevant under the learned policy. In online off-policy RL, this assumption can, at least in principle, be encouraged through continued exploration and interaction with the environment. In offline RL, by contrast, the fixed dataset may provide insufficient coverage, inducing pronounced distribution shift between the behavior policy and the learned policy. As a result, naively reusing off-policy RL methods can lead to severe extrapolation errors and poor performance [17].

From a methodological perspective, this limited coverage makes policy evaluation intrinsically challenging: function approximation may produce spuriously high value estimates in regions that are weakly supported or entirely unsupported by the data. Meanwhile, policy improvement must also be performed with care: aggressively optimizing against imperfect value estimates tends to push the learned policy further outside the dataset support, thereby exacerbating distribution shift and compounding estimation errors. Since policy evaluation and policy improvement are typically interleaved and tightly coupled, errors in either component can propagate through bootstrapping and be repeatedly amplified in the absence of corrective feedback from the environment, resulting in unstable learning dynamics and even algorithmic collapse. These offline-specific challenges are discussed in the next Chapter 2.2.

2.2 Core Challenges in Offline RL

This section presents a general overview of the core challenges in offline RL. Rather than focusing on any single algorithmic family, we first analyze these challenges from the perspective of the deadly triad and then complement this view with a data-centric discussion.

2.2.1 Challenges from the Deadly Triad Perspective

The instability of off-policy RL with function approximation is commonly attributed to the so-called *deadly triad* [2, 53] – the combination of function approximation, bootstrapping, and off-policy learning, as illustrated in Fig. 2.2. When these three ingredients co-occur, bootstrapped value learning can become unstable and may even diverge. Offline RL can be viewed as an extreme regime of off-policy RL: learning is performed solely from a fixed dataset without any corrective interaction, and thus the mismatch between the learned policy and the behavior policy can be severe. As a result, all three components of the deadly triad are typically present and often more pronounced in offline RL, especially off-policy learning and bootstrapping.

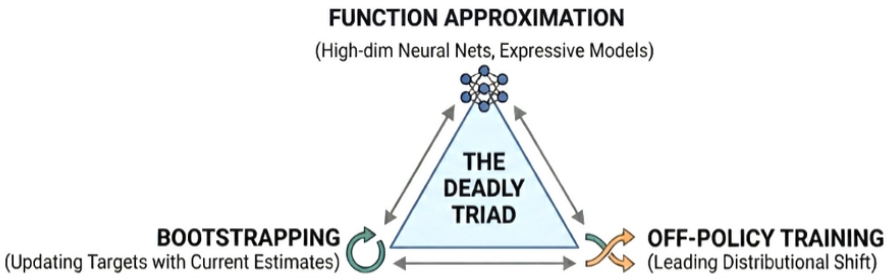


Figure 2.2: The Deadly Triad.

Function Approximation. Function approximation is typically implemented using deep neural networks or other representative parametric models, and has become a standard and almost indispensable component of modern RL. In most applications, the state and action spaces are high-dimensional or continuous, rendering tabular representations of value functions or policies infeasible. As a result, offline RL algorithms routinely rely on function approximators to represent value functions and policies. From the deadly-triad perspective, function approximation is therefore essentially unavoidable and can be regarded as part of the basic problem setting rather than a design choice. In the remainder of this subsection, we focus on bootstrapping and off-policy learning, and examine how they give rise to the characteristic instabilities of offline RL.

Bootstrapping. Bootstrapping refers to updating value functions using targets that depend on current value estimates – as in DP and TD learning – rather than relying exclusively on realized rewards and Monte Carlo returns [2]. For instance, in Eq. (2.11), the TD target takes the form

$$y_{\text{target}} = r + \gamma \mathbb{E}_{a' \sim \pi(\cdot|s')} \left[Q_{\hat{\phi}}(s', a') \right],$$

where $Q_{\hat{\phi}}$ denotes a target network (or equivalently, a stop-gradient copy of Q_ϕ). In offline RL, the expectation over a' may assign non-negligible probability to actions such

that (s', a') is OOD with respect to the dataset. In such OOD regions, the critic estimate $Q_{\hat{\phi}}(s', a')$ is weakly constrained by data and can be arbitrarily inaccurate, giving rise to *extrapolation error* [17].

Moreover, policy improvement commonly involves maximizing the learned Q-function. When Q-functions are inaccurate in OOD regions, the max operator tends to select actions with spuriously high estimates, inducing the well-known *overestimation* bias. Bootstrapping can then propagate these errors backward through TD updates, leading to systematic value inflation and potentially unstable learning dynamics. The following lemma provides a simple illustration of the overestimation effect.

Lemma 2.1 (Overestimation)**.** *Fix a state s and consider a finite action set $\mathcal{A}$. Let $Q^\pi(s,a)$ be the true action-value function under policy π, and let $\hat{Q}^\pi(s,a) = Q^\pi(s,a) + \epsilon(s,a)$ be an estimator with estimation noise $\epsilon(s,a)$. If the noise is unbiased for each action, i.e., $\mathbb{E}[\epsilon(s,a)] = 0$ for all $a \in \mathcal{A}$, then*

$$\mathbb{E}\left[\max_{a\in\mathcal{A}} \hat{Q}^\pi(s,a)\right] \;\geq\; \max_{a\in\mathcal{A}} Q^\pi(s,a). \tag{2.15}$$

Proof. The function $f(x) = \max_{a\in\mathcal{A}} x_a$ is convex. By Jensen's inequality,

$$\mathbb{E}\Big[\max_a \hat{Q}^\pi(s,a)\Big] = \mathbb{E}\big[f\big(Q^\pi(s,\cdot) + \epsilon(s,\cdot)\big)\big] \geq f\big(\mathbb{E}\big[Q^\pi(s,\cdot) + \epsilon(s,\cdot)\big]\big) = \max_a Q^\pi(s,a).$$

□

Lemma 2.1 provides a simple but important insight: unbiased estimation at the action level does not imply unbiased maximization at the policy level. Even if each $\hat{Q}^\pi(s,a)$ is correct on average, the max operator systematically favors actions with positive estimation noise, thereby inducing overestimation. In offline RL, where OOD actions are often evaluated with high uncertainty due to limited data support, this phenomenon becomes particularly severe. As a result, policy improvement may be misled toward spuriously high-valued OOD actions, and bootstrapped TD learning can further amplify such errors over time.

Off-policy Learning. Off-policy learning refers to the setting where the data-generating (behavior) policy differs from the target policy whose value we seek to evaluate or improve. Concretely, value learning is driven by samples drawn from the state–action visitation distribution induced by the behavior policy, denoted by $d^{\pi_\beta}(s,a)$, while the bootstrapped targets in the Q-update are defined with respect to the learned policy π. This mismatch naturally gives rise to *distribution shift* in offline RL:

$$d^{\pi_\beta}(s,a) \neq d^{\pi}(s,a), \qquad \text{where } d^{\mu}(s,a) \triangleq d^{\mu}(s)\,\mu(a \mid s).$$

In practice, distribution shift is hard to avoid whenever we aim to learn a policy π that improves upon π_β. In addition, distributional shift can lead to a worst-case bound on

the expected number of mistakes that scales as $\ell(\pi) \leq C + H^2\epsilon$, where H is the horizon and ϵ is the supervised learning error under $d^{\pi_\beta}(s)$, implying that the worst-case mistake bound grows as $\mathcal{O}(H^2)$ rather than $\mathcal{O}(H)$ [1,54].

During the training stage, distribution shift primarily stems from policy discrepancy, i.e., $\pi(a \mid s) \neq \pi_\beta(a \mid s)$. While state samples are drawn from the fixed dataset and thus follow $d^{\pi_\beta}(s)$, the actions appearing in bootstrapped targets are typically selected according to the learned policy π, e.g., $a' \sim \pi(\cdot \mid s')$ for stochastic policies or $a' = \pi(s')$ for deterministic policies. When π deviates from π_β, the induced (s', a') pairs may fall outside the dataset support, leading to severe extrapolation error in $Q_{\hat{\phi}}(s', a')$. Together, distribution shift and extrapolation error can form a detrimental feedback loop: distribution shift increases OOD action selection; OOD evaluation yields extrapolation error; bootstrapped TD updates amplify this error into systematic overestimation; and the resulting distorted critic further drives the policy toward OOD actions, thereby exacerbating distribution shift.

During the evaluation stage, distribution shift manifests not only in the action space but also in the state space. The state visitation distribution $d^{\pi}(s)$ induced by the learned policy can differ significantly from the dataset state distribution $d^{\pi_\beta}(s)$, giving rise to *OOD states* for which both the critic and the policy are weakly constrained by data. OOD states can arise from several sources:

1. The learned policy selects unreliable (potentially OOD) actions, which transition the system into OOD states.

2. The initial state distribution in the real environment differs from that of the logged dataset; moreover, stochastic dynamics may transition into OOD states even when the policy takes in-distribution actions.

3. Non-stationary or perturbed dynamics in real-world environments can shift the state distribution over time.

As a consequence, the learned policy may encounter states where value predictions are highly unreliable, increasing the risk of suboptimal performance and, in safety-critical domains, unsafe behaviors.

The algorithmic challenges discussed above, including distribution shift and extrapolation error, have directly motivated the rapid emergence of offline RL methods in recent years. Meanwhile, offline RL is inherently a data-driven learning paradigm, where dataset coverage, quality, and heterogeneity can fundamentally shape learnability and generalization. We therefore complement the above algorithmic perspective with a data-centric discussion.

2.2.2 Challenges from the Data Perspective

Beyond the challenges induced by the deadly triad, the *quality*, *coverage*, and *heterogeneity* of offline datasets pose fundamental difficulties for the design and deployment of offline RL algorithms:

- **Quality.** Offline datasets often mix trajectories of widely varying quality, ranging from expert demonstrations to clearly suboptimal or even random behavior. Prior work has repeatedly observed that the same offline RL algorithm can behave very differently across expert, medium-quality, and random datasets [24, 38, 55]. Consequently, a central challenge for offline RL is how to "separate the wheat from the chaff": algorithms should preferentially weight high-return or high-confidence data, and ideally be able to "stitch" together good behavior segments from different trajectories to synthesize a policy that surpasses the behavior policy.

- **Coverage.** Even with high-quality data, offline RL can be severely constrained when the dataset is small or provides highly non-uniform coverage of the state-action space. In such cases, the induced behavior distribution $d^{\pi_\beta}(s, a)$ offers only partial support over regions where the learned policy may place probability mass. This lack of support fundamentally limits what can be reliably evaluated or improved without additional interaction: performance gains beyond the dataset are inherently fragile when the relevant state–action pairs are rarely or never observed.

- **Heterogeneity.** In many applications, offline datasets aggregate experience from multiple behaviors, tasks, or operating conditions (e.g., diverse manipulation skills such as grasping and pushing, or data collected under different system dynamics). Learning an effective policy for a specific target task from such heterogeneous data requires identifying task-relevant subsets, extracting shared representations or reusable skills, and mitigating interference or negative transfer across conflicting behaviors.

2.3 Taxonomy of Offline RL Algorithms

Accordingly, the taxonomy of offline RL in this section is centered on model-free methods. Nevertheless, for completeness, we also briefly review several representative model-based offline RL approaches at the end of this section.

In this section, we aim to develop a taxonomy that can accommodate the broad landscape of offline RL methods and, at the same time, help clarify the key design choices underlying these algorithms, namely, what is learned and how it is learned. This is non-trivial in offline RL, because many methods modify different components of the learning pipeline in ways that are not mutually exclusive and can often be combined into new algorithmic variants. Traditional taxonomies are typically based on *is-a* relationships (e.g.,

a dog is a mammal), whereas mereology focuses on *has-a* relationships (e.g., a dog has a tail). Motivated by this distinction, we propose a high-level taxonomy to organize offline RL algorithms at the family level, while still relying on a mereological perspective to distinguish the constituent design elements within each algorithm [31].

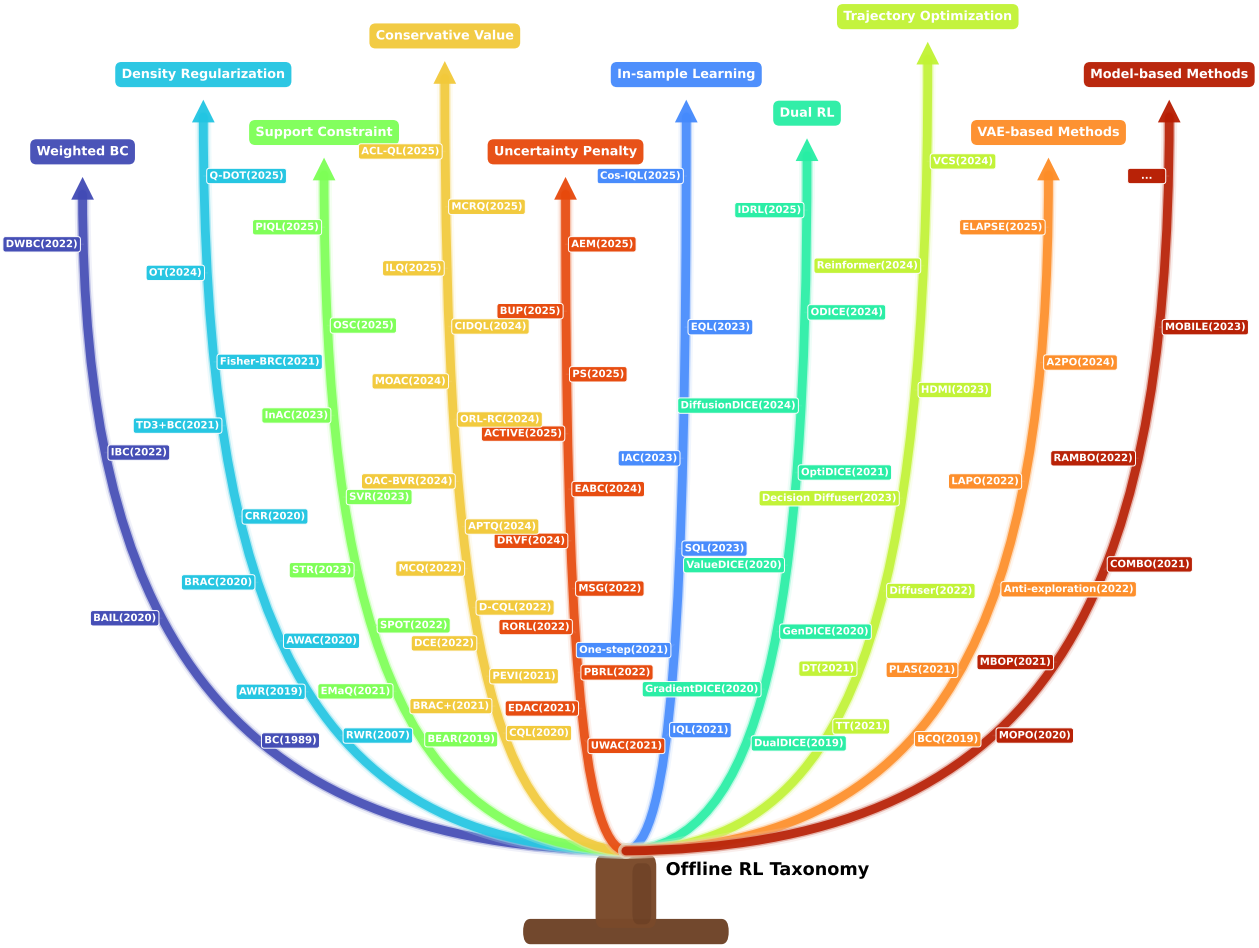


Figure 2.3: The Taxonomy Tree of Offline RL.

Fig. 2.3 provides an intuitive illustration of the taxonomy of offline RL algorithms. In the following subsections, we discuss these categories in detail.

2.3.1 Weighted Behavior Cloning

Weighted Behavior Cloning (wBC) methods either cast policy learning as (weighted) supervised imitation on the dataset, or incorporate an explicit BC regularizer into the policy improvement objective [56].

Representative methods differ mainly in how the weights are defined. BAIL [35] learns a value function and imitates only actions identified as advantageous under the learned critic. IBC [36] improves policy expressiveness through implicit energy-based modeling, alleviating some limitations of explicit regression-based policy classes. DWBC [37] trains a discriminator to estimate expert-likeness and uses this signal to reweight the cloning objective.

Overall, wBC methods represent an intuitive strategy for safe offline policy improvement: they remain close to the dataset while attempting to favor better-than-average behavior. However, because policy improvement is still largely tied to imitating observed actions, these methods may be overly conservative and can be limited in their ability to exploit the full improvement space beyond the dataset support.

2.3.2 Density Regularization

Density-regularization methods are a central line of research in offline RL, as they explicitly address distribution shift by discouraging the learned policy from selecting OOD actions. At a high level, many policy-improvement objectives in this family can be summarized in the following unified form:

$$\max_{\pi} \mathcal{J}_{\mathrm{DR}}(\pi) = \underbrace{\mathbb{E}_{s\sim\mathcal{D}}\Big[\mathbb{E}_{a\sim\pi(\cdot|s)}\big[Q(s,a)\big]\Big]}_{\text{policy improvement}} - \lambda \underbrace{\mathbb{E}_{s\sim\mathcal{D}}\big[\mathcal{F}(\pi(\cdot\mid s), \pi_\beta(\cdot\mid s))\big]}_{\text{policy constraint}}, \tag{2.16}$$

where $Q(s,a)$ (or the advantage function $A(s,a)$) is obtained from policy evaluation, $\mathcal{F}(\cdot,\cdot)$ measures the deviation between π and π_β, and $\lambda \geq 0$ controls the constraint strength.

1. **KL-based density regularization.** When $\mathcal{F}$ is chosen as a KL divergence [57] (or a closely related f-divergence), Eq. (2.16) recovers a broad class of density-regularized policy-improvement objectives that limit distributional shift while allowing controlled improvement, e.g., RWR [19], AWR [20], AWAC [21], and CRR [22].

2. **Action-space regression.** When $\mathcal{F}$ is instantiated as an ℓ_2 regularizer that encourages actions close to the dataset actions, one recovers BC-style regularization as in TD3+BC [38, 58].

3. **Integral probability metrics (IPMs).** When $\mathcal{F}$ is designed as an IPM that matches action distributions induced by π and π_β, policy improvement can be constrained

via distances such as Maximum Mean Discrepancy (MMD), as used in BEAR [18] and BRAC [59], or via Wasserstein / Optimal Transport (OT)-type discrepancies [60, 61].

4. **Fisher divergence regularization.** Fisher-divergence-based constraints also fit naturally into divergence-based regularization, but measure discrepancy through score functions (gradients of log-densities) [62].

2.3.3 Support Constraint

Compared with density regularization, support-constraint methods are often viewed as imposing a looser form of behavioral restriction: instead of matching the full behavior density, they aim to restrict the learned policy to select actions that lie within the support of the behavior policy, i.e.,

$$\pi_\beta(a \mid s) = 0 \;\Rightarrow\; \pi(a \mid s) = 0.$$

Representative methods in this category include EMaQ [63], which draws multiple candidate actions from the behavior policy and performs backups over these in-support candidates; STR [13], which integrates trust-region policy optimization with an explicit support constraint through importance sampling; SPOT [39], which learns a VAE-based estimator of the behavior support and employs a plug-in regularization term to confine policy improvement to the estimated support; and InAC [64], which adopts an in-sample softmax operator and realizes support-constrained policy improvement by sampling actions directly from the dataset.

2.3.4 Conservative Value

The above categories mainly intervene in policy improvement to control distribution shift, i.e., they constrain how far the learned policy may deviate from the behavior policy. An alternative and largely complementary line of work operates on policy evaluation: it mitigates extrapolation-induced value errors, often referred to as overestimation (as illustrated in Lemma 2.1), by learning a conservative or pessimistic critic, thereby discouraging the subsequent policy from exploiting OOD actions.

A broad class of conservative-value methods can be summarized as augmenting Bellman regression with an explicit conservatism regularizer:

$$\min_Q \mathcal{L}_{\mathrm{CV}}(Q) = \underbrace{\mathbb{E}_{(s,a,s',r)\sim\mathcal{D}}\Big[\big(Q(s,a) - (r(s,a) + \gamma\, \mathbb{E}_{a'\sim\pi(\cdot|s')}\bar{Q}(s',a'))\big)^2\Big]}_{\text{Bellman regression}} + \lambda \underbrace{\mathcal{R}_{\mathrm{cons}}(Q;\mathcal{D},\mu)}_{\text{conservative regularizer}}, \tag{2.17}$$

where $\bar{Q}$ denotes a target network (or a slowly-updated copy of Q), and $\mu(\cdot|s)$ is an action proposal distribution used to characterize and penalize weak-support (potentially OOD) actions (e.g., μ can be instantiated as π, π_β, or their mixtures/perturbations).

1. **Penalizing values on OOD actions.** A representative strategy is to anchor values on dataset actions while penalizing overly large Q-values on actions sampled outside the data support. Conservative Q-Learning (CQL) explicitly pushes down $Q(s, a)$ for actions drawn from the learned policy π or a broad proposal distribution μ while keeping relatively higher values on in-dataset actions, yielding a conservative value function that lower-bounds the relevant Q-values [24]. Recent variants further adjust the strength of conservatism to avoid over-pessimism, e.g., MCQ [65], APTQ [66] and ACL-QL [26].

2. **Conservatism via value regularization.** Instead of aggressively suppressing all unseen actions, another family enforces mild conservatism by explicitly regularizing the discrepancy between value estimates of actions induced by the current policy and those present in the dataset, e.g. [67–69].

3. **Uncertainty-aware pessimism.** A further complementary approach is uncertainty-aware pessimism, where an ensemble disagreement signal is converted into a penalty. Representative methods in this category are discussed in more detail in a dedicated subsection below.

2.3.5 Ensemble-based Uncertainty Penalty

In offline RL, uncertainty characterizes the limited reliability of value estimates inferred from a fixed dataset $\mathcal{D}$, and thus indicates the potential discrepancy between the learned value function and the true value induced by the environment. It arises primarily from (i) distribution shift, where the learned policy queries actions that are rare or absent in $\mathcal{D}$, and (ii) bootstrapping error propagation, where TD targets reuse learned values and thereby amplify early misestimates. These factors mainly correspond to epistemic uncertainty, which stems from limited data and knowledge, rather than aleatoric uncertainty caused by intrinsic stochasticity in the environment [70].

Ensemble-based offline RL typically instantiates multiple independently trained critics. The disagreement among ensemble members, e.g., the variance or standard deviation of $\{Q_i(s, a)\}_{i=1}^K$, serves as a proxy for epistemic uncertainty [71–75]. This uncertainty can then be converted into a pessimistic penalty to calibrate value estimation (like conservative regularizer in Eq. (2.17)) and guide policy improvement, without relying on explicit density models or hard policy constraints. Moreover, naive ensembles may collapse to similar critics [76]; methods such as EDAC [55] explicitly promote critic diversity via gradient-based regularization.

2.3.6 In-sample Learning

In-sample learning methods address the OOD challenge by restricting both policy improvement and bootstrapping targets to in-dataset actions, thereby reducing extrapolation error. The key idea is to perform learning using only samples drawn from the offline dataset, so that both policy improvement and policy evaluation remain within the empirical support of $\mathcal{D}$.

A canonical instantiation is Implicit Q-Learning (IQL) [23], which decouples value learning and policy extraction: it first learns a conservative implicit value via expectile regression, effectively yielding an in-sample policy-evaluation step. This perspective has inspired a family of methods that replace hard maximization over the continuous action space with in-sample operators, such as softmax / log-sum-exp backups or dataset-restricted greedy steps, leading to variants including SQL and EQL [29], $\mathcal{X}$-QL [34], CosIQL [25], and in-sample actor-critic formulations such as IAC [77]. A closely related viewpoint is one-step offline RL, which performs only a single policy-improvement step (or avoids iterative off-policy evaluation) to reduce error compounding [27, 28].

2.3.7 Dual RL

Dual-RL, also known as DIstribution Correction Estimation (DICE), was first used to ensure unbiased estimation of the on-policy policy gradient using off-policy data [78]. Rather than directly performing policy improvement under distribution shift, DICE-style methods estimate a distribution correction between the target occupancy and the data distribution, enabling stable Off-Policy Evaluation (OPE).

DICE-style dual RL methods estimate an occupancy correction ratio through a dual (saddle-point) formulation, enabling stable OPE. DualDICE [52] and GenDICE [79] provide core ratio-estimation objectives, with GradientDICE improving optimization and convergence [80]. Building on ratio estimation, OptiDICE performs offline policy optimization via stationary distribution correction [81]. Subsequent work improves learning dynamics (e.g., O-DICE [82] via orthogonalized gradients) and extends the framework with more expressive models (e.g., Diffusion-DICE [83]). Finally, IDRL connects dual RL to discriminator-weighted imitation in an iterative framework [78].

2.3.8 Trajectory Optimization

Motivated by the strong sequence modeling and generative capabilities of Transformers [84] and diffusion models [85], recent work revisits offline RL from a trajectory-level viewpoint. Instead of explicitly learning value functions through Bellman backups, these methods treat trajectories as the primary modeling object and formulate decision making as conditional generation or trajectory optimization.

Transformer-based approaches cast a trajectory as an ordered token sequence and train an autoregressive model to maximize $\log p(\tau)$ with causal masking [86–90]. By condi-

tioning on returns, goals, or other trajectory descriptors, they generate action sequences directly from historical data.

Diffusion-based approaches model trajectories through iterative denoising, learning a denoiser (or score model) that maps noisy trajectory samples to clean ones, and enabling planning by sampling trajectories with guidance toward high return [91–94]. Compared with autoregressive generation, diffusion models often provide greater flexibility in modeling complex multimodal trajectory distributions.

Overall, trajectory-optimization methods represent a shift from step-wise Bellman-based learning to direct trajectory modeling and planning. Their main appeal lies in leveraging powerful generative models to capture long-horizon dependencies from offline datasets. However, because they rely less explicitly on value estimation, they may sacrifice some of the interpretability and policy-evaluation guarantees offered by critic-based methods.

2.3.9 VAE-based Methods

Typical VAE-based offline RL methods employ CVAE models as a behavior prior or a support estimator, and can be broadly categorized into the following types:

1. **Behavior modeling.** Training the CVAE model to approximate the behavior-induced action distribution [17, 63].

2. **Latent-space policy optimization.** Parameterizing the policy in the CVAE latent space and optimizing the latent distribution, which is then decoded into actions, thereby restricting policy search to a data-supported manifold and mitigating OOD actions [95, 96].

3. **Explicit policy constraints.** Incorporating the VAE prior into the policy objective as a policy constraint [39, 97].

Importantly, VAE-based (and more broadly, generative-model-based) approaches are not fully orthogonal to the above taxonomy: depending on how the learned generative prior is used, they can be viewed as instances of policy constraints, in-sample learning, or conservative value learning. We nevertheless list VAE-based methods separately, since Chapter 5 specifically analyzes the collapse phenomenon in VAE-based offline RL and studies its implications for stable policy learning.

2.3.10 Model-based Methods

Model-based offline RL methods differ fundamentally from the model-free approaches discussed above in that they explicitly learn a dynamics model (or reward model) from the offline dataset and perform policy optimization via model-based planning, synthetic rollouts, or model-augmented value learning [98–100]. Since this thesis primarily focuses

on the design principles of model-free offline RL, we do not further elaborate on model-based methods.

2.4 Evolution of Offline RL Design Principles

Beyond categorizing existing methods, it is also useful to examine how offline RL algorithm design has evolved over time. From this perspective, several broad developmental trends can be identified.

2.4.1 From Behavior Cloning to Support-Constraint Optimization

Early policy-improvement approaches largely centered on policy regularization: since the learned policy π may query actions outside the dataset support, algorithms constrain distribution shift by adding an explicit penalty that keeps π close to the behavior policy π_β or to dataset actions. This yields a unified family of objectives of the form "maximize Q while penalizing divergence," instantiated by various metrics such as weighted behavior cloning and density regularization mentioned in Chapter 2.3.1 and 2.3.2.

A clear shift in later work is toward support-constrained optimization, which incorporates the offline constraint more directly by restricting action selection to in-support actions. Compared with wBC and density regularization algorithms, support constraints are typically less restrictive and thus preserve a larger feasible space for policy optimization. In particular, instead of matching the full behavior density, support-constrained methods only require the learned policy to select actions that lie within the empirical support of the dataset. This looser constraint allows the policy to be guided more effectively by the reward signal or policy evaluation function, while still mitigating OOD action selection.

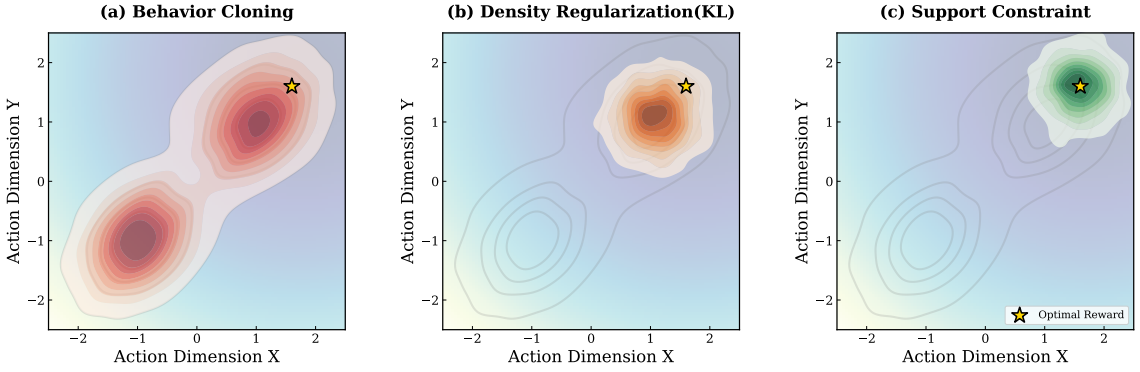


Figure 2.4: From behavior cloning to support-constraint optimization: the evolution of policy improvement paradigms in offline RL.

Figure 2.4 illustrates the fundamental trade-off in offline RL: the tension between maximizing reward and minimizing distributional shift. In this scenario, we design a bimodal action distribution (gray contours) and a reward signal where the optimal reward (gold star) is located at the peripheral edge of the high-density data region.

As illustrated in Fig. 2.4, pure behavior cloning (Fig. 2.4(a)) mainly matches the dataset density and therefore reproduces the multi-modal structure of the behavior distribution, without leveraging the reward signal to concentrate probability mass on high-return regions. In contrast, density regularization (e.g., KL-based density in Fig. 2.4(b)) enables reward-driven improvement by shifting the policy toward the optimal-reward area. However, since it must trade off return maximization against staying close to the behavior density, the learned policy typically converges to a compromise: it moves toward the optimum but remains confined to relatively high-density regions, especially when the optimal actions lie in low-density tails. By comparison, support constraints (Fig. 2.4(c)) only require the policy to stay within the empirical support of the dataset, without matching the full behavior density. This looser constraint preserves a larger feasible space for policy optimization, allowing the policy to be guided more effectively by the reward signal or the learned critic and to concentrate on high-reward subsets within the support.

2.4.2 From Pessimism to Selective Optimism

A key difficulty in offline RL is that value learning is performed under limited coverage, where bootstrapping and function approximation can yield severe extrapolation errors on weak-support actions. To prevent the learned policy from exploiting such spurious values, early successful approaches advocated pessimistic or conservative value estimation, i.e., systematically underestimating values in regions that are poorly supported by the dataset. While pessimism improves stability and safety, it also introduces an inherent trade-off: overly conservative critics can suppress legitimate improvement and collapse performance toward behavior cloning.

Recent progress therefore moves from uniform pessimism toward selective optimism: value estimation should remain conservative where epistemic uncertainty is high (e.g., under weak data support), but should allow more optimistic updates within well-supported regions where value estimates are reliable. This shift is operationalized by mild or relaxed conservatism, as well as adaptive constraints that modulate pessimism strength according to data support.

Fig. 2.5 (a) – (d) visualize this evolution of policy evaluation. The purple area depicts the empirical data support, with red dots indicating samples, the blue curve denotes the ground-truth optimal value function, and the orange curve denotes the estimated value used for policy improvement.

In panel (a), pessimistic value estimation [24] aggressively suppresses values outside the support, which effectively avoids selecting OOD actions but can also under-estimate values within the support, thereby limiting performance.

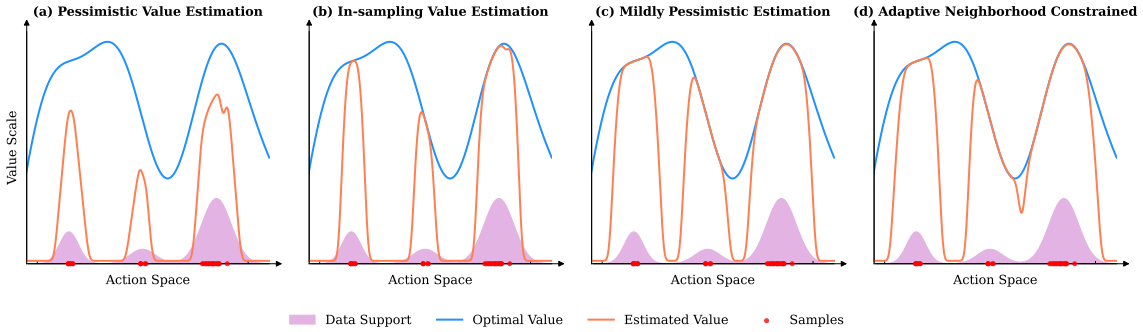


Figure 2.5: From Pessimism to Selective Optimism: the Evolution of Value Estimation Paradigms in Offline RL

Panel (b) illustrates in-sampling value estimation [23, 25, 29], where backups are restricted to in-dataset actions; this removes the need to assign values to unsupported actions, but may yield a fragmented estimate that tightly follows where samples exist and offers limited generalization across nearby regions.

Panel (c) shows mildly pessimistic estimation [65, 67, 101]: instead of uniformly pushing down all weak-support actions, the critic is biased to be conservative yet still tracks high-value structure within supported regions, enabling improvement without excessive underestimation.

Finally, panel (d) demonstrates an adaptive neighborhood-constrained strategy [102], where the effective constraint strength is action dependent: the estimator remains conservative in low-support neighborhoods while allowing selective optimism in well-supported neighborhoods, producing a value landscape that is both safe and sufficiently sharp to guide policy optimization toward genuinely high-return actions.

2.4.3 From Bellman Backups to Trajectory-level Planning

Classical offline RL is predominantly built upon Bellman backups and bootstrapped value fitting. As illustrated in Fig. 2.6(a), learning proceeds by repeatedly regressing a value function (or Q-function) to a one-step TD target, and then improving the policy by maximizing the learned critic. This paradigm is attractive for its simplicity, yet it is particularly fragile in the offline setting: once the learned policy deviates from the behavior policy, the critic must extrapolate to weak-support state-action pairs, and small value errors can be amplified through bootstrapping and greedy policy improvement. Much of the offline RL literature reviewed above can be interpreted as designing increasingly explicit constraints to stabilize this Bellman-based loop, e.g., density regularization, support constraint, and conservative value learning.

Recent progress introduces a complementary viewpoint that bypasses Bellman backups by shifting the learning objective from local TD consistency to trajectory-level planning. The central idea is to learn a conditional generative model over trajectories from the offline dataset and to perform decision making by sampling trajectories consistent with a desired condition (e.g., return or goals), optionally guided or filtered by learned scoring functions. This reframing is motivated by the observation that modern generative models–notably Transformers and diffusion models–provide strong inductive biases for modeling long-range dependencies and multi-modal distributions, both of which are common in offline datasets.

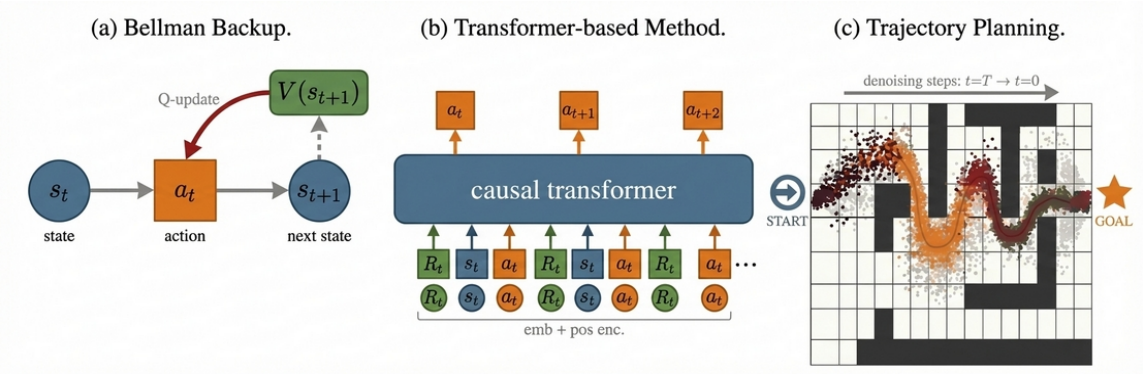


Figure 2.6: From Bellman Backups to Trajectory-level Planning: Bellman backups, Transformer-based method, and Diffusion Planning.

Fig. 2.6(a) shows the conventional Bellman-backup-based RL paradigm. Fig. 2.6(b) depicts the Transformer-based paradigm, where a trajectory is treated as a token sequence and a causal Transformer is trained to maximize conditional log-likelihood [86–90]. Fig. 2.6(c) illustrates diffusion-based trajectory planning, where a trajectory is generated by iteratively denoising an initial noisy sample [91–94].

Overall, the shift from Bellman backups to trajectory-level planning should be viewed as a methodological expansion rather than a wholesale replacement. Generative approaches offer (i) trajectory-level coherence and expressive multi-modal modeling, and (ii) new avenues for dataset enhancement via synthesis and trajectory stitching. At the same time, they introduce failure modes that admit the same constraint-centric interpretation as classical offline RL: reliability still depends on remaining within the dataset-supported manifold and preserving long-horizon consistency under compounding errors.

Consequently, many recent methods implicitly combine generative modeling with conservative evaluation or support-aware filtering [89]. This points to a convergent theme: whether the control signal is produced by Bellman backups or by trajectory generation, effective offline RL ultimately requires mechanisms that control distribution shift while retaining sufficient optimization freedom to realize high return.

2.4.4 From Unimodal Gaussian Policies to Expressive Multimodal Policies

In offline RL, the choice of policy model is not merely an implementation detail, but a central factor in determining whether policy improvement can remain aligned with dataset support. Recent analysis has emphasized that many actor-critic offline RL methods [21, 23] implicitly rely on a unimodal action-distribution assumption, while practical offline datasets are often inherently multimodal due to the aggregation of data from multiple behaviour policies, human demonstrations, or diverse exploration strategies. Under such conditions, a unimodal policy may collapse distinct valid behaviours into an averaged action that is poorly supported by the dataset, or even invalid for decision making.

This limitation has motivated a growing line of work on more expressive policy classes, including latent-variable models [17, 63], flow-based generators [103], diffusion-based policies [104, 105], energy-based models [36, 106], and more recently mode-selective methods [107], all of which aim to better capture the structure of multimodal behaviour. At the same time, recent work suggests that fully modelling the entire multimodal action distribution may be unnecessarily complex for policy improvement. Instead, it may suffice to identify and exploit the subset of behaviour modes that are most relevant to high-return decision making.

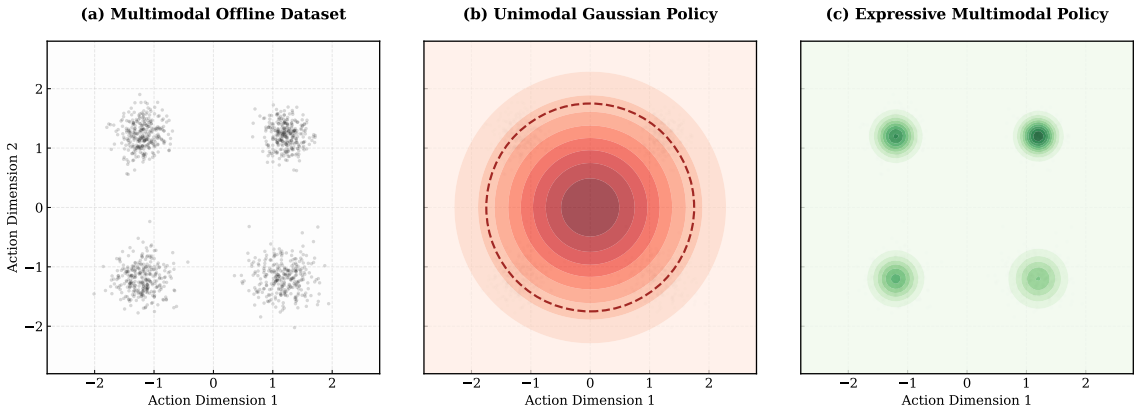


Figure 2.7: From Unimodal Gaussian Policies to Expressive Multimodal Policies.

Fig. 2.7 (a) illustrates a representative offline dataset containing multiple distinct action modes. Fig. 2.7 (b) demonstrates the limitation of a unimodal Gaussian policy, which suffers from mode-averaging artifacts; by attempting to minimize the distance to all data points simultaneously, it places high probability mass in low-density regions between modes, often resulting in physically non-viable actions. In contrast, Fig. 2.7 (c) displays an expressive multimodal policy, which accurately recovers the underlying behavior manifold.

2.4.5 From Fixed Penalties to Adaptive Constraints

A recurring theme in offline RL is that many practical algorithms rely on an explicit constraint or regularizer to control distribution shift and extrapolation error, e.g., a KL/MMD penalty toward the behavior policy, an action regression term toward dataset actions, or a pessimism coefficient in conservative value learning. Historically, these constraints were implemented with a fixed global weight (hyperparameter) λ, tuned for each task and dataset. While conceptually simple, a single scalar coefficient is inherently brittle in offline settings: the appropriate constraint strength is highly non-uniform across the state-action space.

Regions with dense coverage admit more reliable value estimates and can tolerate more aggressive improvement, whereas weak-support regions exhibit higher epistemic uncertainty and require stronger conservatism to prevent OOD exploitation [21, 23, 29, 38, 67]. Consequently, fixed penalties often force a trade-off between safety and performance: setting λ too small risks instability (Fig. 2.8(a)), while setting λ too large degenerates toward BC and under-utilizes value information (Fig. 2.8(b)).

This motivates the evolution toward adaptive constraints (Fig. 2.8(c)), where the effective regularization strength becomes state- or action-dependent, i.e., $\lambda \rightarrow \lambda(s, a)$, and is modulated by signals such as dataset density, uncertainty, or Monte Carlo return [14, 25, 26, 89]. By increasing constraint strength in low-support neighborhoods and relaxing it in well-supported regions, adaptive designs aim to achieve selective conservatism: safe where necessary, but sufficiently permissive to pursue high return where the data provide reliable guidance.

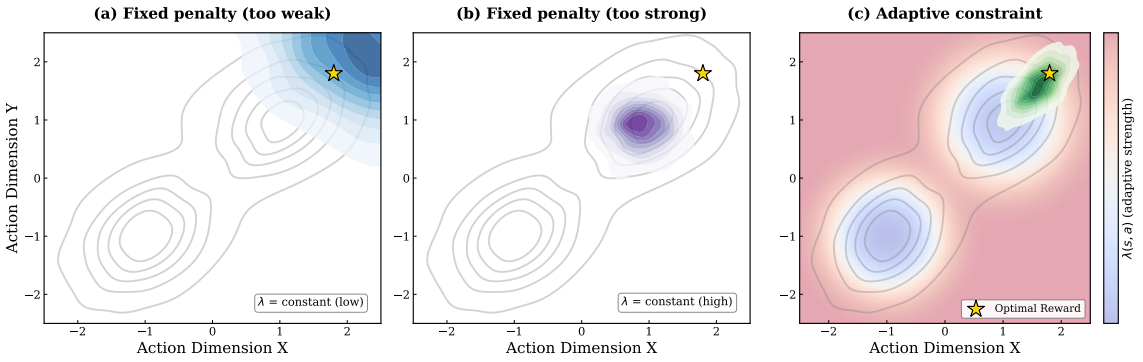


Figure 2.8: From Fixed Penalties to Adaptive Constraints in Offline RL.

Fig. 2.8 illustrates why a single global penalty coefficient is often inadequate in offline RL. The gray contours denote the dataset-supported action distribution, and the star marks the high-reward target, which lies near the boundary of a supported mode. In Fig. 2.8(a), when the fixed penalty is too weak, policy optimization can be driven too aggressively toward high-value regions, causing the learned policy to drift into poorly

supported or OOD areas. In Fig. 2.8(b), when the fixed penalty is too strong, the policy is overly anchored to the behavior distribution and fails to exploit reward-relevant directions, effectively collapsing toward imitation. By contrast, Fig. 2.8(c) depicts the desired adaptive regime: the effective constraint is relaxed within well-supported regions but strengthened outside the dataset manifold, allowing policy improvement to move toward high reward while suppressing unreliable OOD deviations.

2.4.6 From Action-level Behavior Constraints to State Correction

In offline RL, OOD actions refer to actions not selected by the behavior policy in a given state, rather than simply actions absent from the offline dataset. In other words, OOD actions are inherently state-dependent, often expressed as OOD state-action pairs, which constitute a form of batch constraint. During training, evaluating OOD actions leads to extrapolation error, which is amplified through bootstrapping, causing severe value overestimation and consequently significantly degrading the performance of the learned policy during testing. Indeed, besides the OOD action problem during training affecting test-time performance, the OOD state issue (also known as state distribution shift) can also impair performance.

OOD states refer to states encountered by the agent during testing that lie outside the offline dataset distribution. The occurrence of OOD states is primarily attributed to the agent executing OOD actions, the initial state of random environment falling outside the dataset distribution, and perturbations in the real environment [108]. However, the aforementioned offline RL methods do not involve policy training under OOD states. Therefore, once the agent enters an OOD state during testing, it often exhibits uncontrolled behavior, continuously exacerbating the state deviation between the test trajectory and the offline dataset $\mathcal{D}$, even leading to catastrophic failure.

Existing state correction approaches can be broadly categorized into action stitching and state stitching. Action stitching aims to train a policy that can recover from near-OOD states back to in-distribution regions by selecting corrective actions [109–112]. In contrast, state stitching performs correction at the transition level by decoupling decision making into two stages: first predicting a desirable next state s' using a state transition model constrained by dataset support and value information, and then inferring the action needed to realize this transition via an inverse dynamics model $I(a \mid s, s')$. By extending behavior constraints from the action space to the state-transition space, state stitching more directly prevents state distribution drift and can bypass overly conservative action-level policy constraints [113–116].

Overall, the progression from action-level behavior constraints to state correction reflects a broader recognition: robust offline RL requires not only avoiding unsupported actions during training, but also anticipating and recovering from state distribution shift at deployment. By explicitly correcting state deviation—either through recovery actions,

conservative state-aware critics, or transition-level planning—modern offline RL seeks to achieve both high return and stability under real-world distributional mismatch.

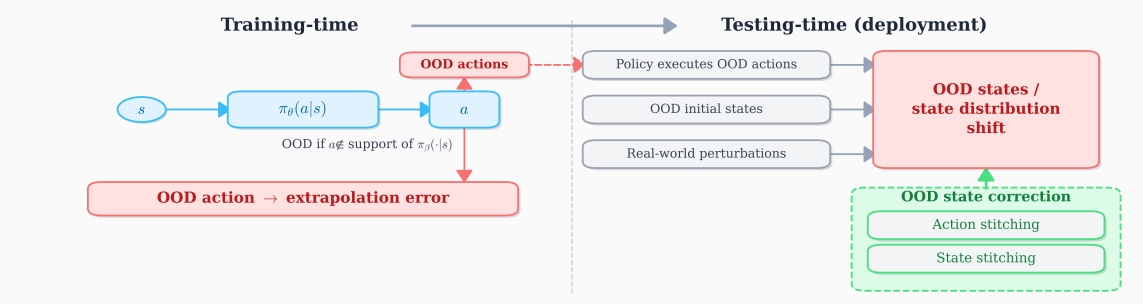


Figure 2.9: From Action-level Behavior Constraints to State Correction in Offline RL.

2.4.7 From Shallow Approximators to Scalable Offline RL

The community's increasing focus on scalability is closely connected to the success of large foundation models (e.g., LLMs): these successes suggest that, with sufficient data, compute, and model capacity, learning systems can acquire broad and transferable capabilities. Offline RL offers an appealing setting to pursue a similar scaling trajectory. However, recent evidence indicates that achieving scalable performance requires the co-design of model capacity, learning objectives, and structural mechanisms that address horizon-induced compounding errors, rather than relying on scale (data and parameters) alone [117, 118].

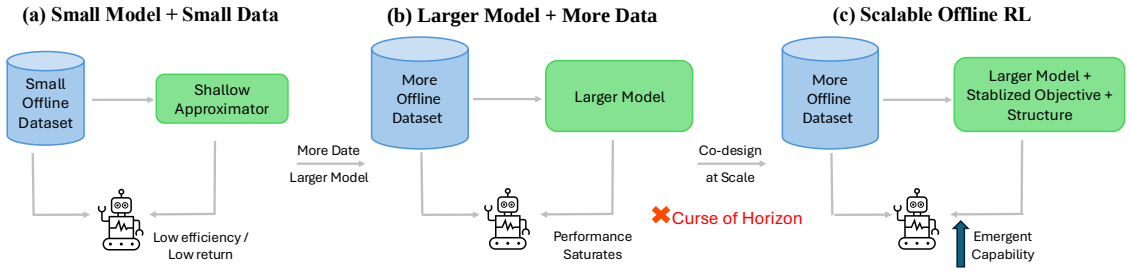


Figure 2.10: From Shallow Approximators to Scalable Offline RL.

Concretely, [117] provides large-scale empirical evidence that simply enlarging offline datasets does not necessarily yield proportional gains, and attributes the observed saturation primarily to the "curse of horizon" in TD-style learning and policy extraction (Fig. 2.10(a)–(b)); it demonstrates that explicitly reducing the effective horizon by simple tricks can substantially improve scaling behavior.

Complementing this structural view, [118] shows that replacing purely TD-centric training with self-supervised, goal-conditioned objectives can enable stable scaling to extremely deep networks and unlock qualitatively stronger goal-reaching capabilities. In addition, many prior works on scaling offline RL focus on scalability to more tasks by training a large, multitask agent that is capable of solving more diverse tasks [119–122].

2.5 Summary and Connections to the Subsequent Chapters

In summary, this chapter has reviewed the fundamentals, core challenges, taxonomy, and macro-level design evolution of model-free offline RL. More importantly, the above analysis highlights several recurring tensions that remain central to the field:

- the tension between in-sample safety and multi-step improvement,
- the need for flexible and adaptive constraint types and strengths rather than fixed forms of conservatism control,
- the trade-off between generative behavior modeling and downstream policy optimization.

These observations motivate the subsequent technical chapters: Chapter 3 studies how to reconcile in-sample and multi-step learning, Chapter 4 develops a unified and adaptive framework for policy constraints, and Chapter 5 analyzes and mitigates failure modes in CVAE-based offline RL.

Chapter 3

Projective Implicit Q-Learning with Support Constraint

This chapter focuses on the second research question ($RQ2$ in Section 1.1) of the thesis: how to reconcile in-sample learning with multi-step learning to achieve safe and effective policy improvement in offline RL. Existing in-sample learning methods are attractive because they reduce extrapolation error by grounding learning on the offline dataset, but they often sacrifice the flexibility and potential gains of multi-step learning. To address this tension, we propose Projective Implicit Q-Learning (PIQL), which extends the in-sample learning paradigm to a multi-step formulation through a projection-based mechanism, while further incorporating a support constraint to stabilize policy improvement.

The remainder of this chapter is organized as follows. Section 3.1 introduces the problem background and motivation for PIQL. Section 3.2 reviews the related work, and Section 3.3 presents the necessary preliminaries. Section 3.4 details the proposed PIQL algorithm, including its policy evaluation and policy improvement components. Section 3.5 reports the experimental and comparison results on standard offline RL benchmarks.

3.1 Introduction

Offline RL aims to learn policies solely from static datasets collected by an unknown behavior policy, thereby avoiding costly or unsafe online interaction with the environment [1, 31]. While this paradigm is highly appealing for real-world applications, it also gives rise to severe extrapolation error caused by OOD actions [17]. Such errors can be amplified through bootstrapped Bellman updates and may significantly distort value estimation and policy optimization. Existing offline RL methods typically address this challenge from two complementary perspectives: policy improvement, by constraining the learned policy toward the behavior distribution or support, and policy evaluation, by learning conservative critics.

Among recent methods, Implicit Q-Learning (IQL) [23] has emerged as a representative in-sample offline RL algorithm. By replacing the hard maximum in Q-learning with expectile regression, IQL avoids explicit evaluation of unsupported actions and achieves strong empirical performance. However, IQL still suffers from two notable limitations. First, it relies on a fixed expectile hyperparameter, which often requires dataset-specific tuning. Second, it essentially follows a one-step learning paradigm [27], which limits its extension to more expressive multi-step learning frameworks.

To address these limitations, we propose Projective Implicit Q-Learning (PIQL), a projective variant of IQL equipped with a support-constrained policy improvement mechanism. PIQL introduces a projection-based adaptive parameter, $\tau_{\text{proj}}(a \mid s)$, which is derived by projecting the behavior policy onto the learned policy and thus removes the need for manually tuned dataset-specific expectiles. This design preserves the in-sample learning and expectile-regression structure of IQL while naturally extending one-step value estimation to a multi-step formulation. In addition, PIQL incorporates a support constraint that is aligned with the policy evaluation stage, enabling more effective policy improvement while mitigating extrapolation error.

Theoretically, we demonstrate that PIQL can be interpreted as a multi-step offline RL method while retaining the expectile-regression objective and the in-sample learning paradigm. Moreover, PIQL is provably guaranteed to yield monotonic policy improvement and to enforce a progressively more rigorous criterion for identifying and selecting advantageous actions.

Experimentally, PIQL achieves state-of-the-art performance on D4RL [123], with particularly strong results on challenging "stitching" navigation tasks that require composing sub-trajectories over long horizons. We further evaluate PIQL on the recent NeoRL2 [124] benchmarks, where it consistently surpasses the strongest baselines.

3.2 Related Work

The most relevant prior work to PIQL lies along two closely related directions in offline RL: conservative value methods that mitigate extrapolation error during policy evaluation, and policy-constraint methods that restrict distribution shift during policy improvement.

Methods for achieving conservative value estimation in offline RL broadly include classical conservative-value methods and in-sample value-learning methods. Classical approaches aim to reduce overestimation caused by OOD actions by penalizing overly optimistic value estimates. For example, CQL [24] introduces an explicit penalty on the Q-function for OOD actions, while extensions such as ACL-QL [125] dynamically adjust the penalty strength to alleviate over-conservatism, and CSVE [126] regularizes state-value estimates across the dataset to improve generalization. A more closely related line is in-

sample learning, which avoids evaluating unsupported actions altogether. Representative methods include Onestep [27], IQL [23], and SQL [29].

Among these methods, IQL is particularly relevant, as it replaces the hard maximization in Q-learning with expectile regression and achieves strong empirical performance. However, IQL still relies on a fixed expectile hyperparameter that typically requires careful tuning, and it essentially follows a one-step learning paradigm, which limits its extension to more expressive multi-step formulations.

The second relevant direction concerns policy-constraint methods. These methods control distribution shift by encouraging the learned policy to remain close to the behavior policy, using mechanisms such as KL-style regularization [21, 127], regression-based penalties [38], or generative behavior priors such as CVAE [17, 95, 96, 128]. While such density-based constraints are effective under moderate distribution shift, they can be overly restrictive and may struggle when the dataset is highly suboptimal. Support-constraint methods instead impose a looser restriction by requiring the learned policy to stay within the support of the behavior policy. Representative examples include BEAR [18], which constrains the policy using Maximum Mean Discrepancy (MMD), SPOT [39], which estimates behavior support using a VAE-based model, and DASCO [129], which leverages discriminator scores in a GAN-based framework.

In contrast to prior conservative or in-sample value-learning methods, PIQL preserves the in-sample learning advantages of IQL while extending them to a multi-step formulation through a projection-based adaptive parameter. In contrast to density-based policy-constraint methods, PIQL belongs to the family of support-constrained approaches and therefore permits greater flexibility in policy optimization. Moreover, although PIQL adopts a policy-improvement scheme related to STR [13], it further introduces a progressively stricter criterion for identifying advantageous actions from the dataset, thereby promoting more stable and effective policy improvement.

3.3 Preliminaries

3.3.1 Support Constraint

The support-constraint policy class Π is defined as

$$\Pi = \{\pi \mid \pi(a \mid s) = 0 \ \text{ whenever } \ \pi_\beta(a \mid s) = 0\}, \tag{3.1}$$

where π_β denotes the behavior policy that generated the offline dataset.

Following [13], a general policy optimization objective for weighted behavior cloning (wBC) methods can be written as

$$\max_\theta J_\pi^{\text{wBC}}(\theta) = \mathbb{E}_{s\sim\mathcal{D}, a\sim\pi_{\text{base}}}\left[f(s,a;\pi_{\text{pe}}) \log \pi_\theta(a \mid s)\right], \tag{3.2}$$

where π_{base} denotes the sampling policy and π_{pe} denotes the policy used for policy evaluation. The weighting function is defined as

$$f(s,a;\pi_{\text{pe}}) = \frac{1}{Z(s)} \exp\left(\frac{A^{\pi_{\text{pe}}}(s,a)}{\lambda}\right),$$

where λ is a temperature parameter and

$$Z(s) = \int_a \pi_{\text{base}}(a \mid s) \exp\left(\frac{A^{\pi_{\text{pe}}}(s,a)}{\lambda}\right) da.$$

In practice, the normalization term $Z(s)$ is often omitted during training, since estimating it can introduce additional error without commensurate practical benefit. Accordingly, one typically uses the unnormalized weight

$$f(s,a;\pi_{\text{pe}}) = \exp\left(\frac{A^{\pi_{\text{pe}}}(s,a)}{\lambda}\right).$$

Since $f(s,a;\pi_{\text{pe}}) > 0$, the wBC update preserves the support of the sampling policy. In particular, if the initial policy lies within the support of π_β, then repeated policy-improvement steps keep the learned policy within the support of π_β. Based on this observation, STR [13] proposes a support-constrained policy-improvement objective by setting $\pi_{\text{base}} = \pi_\theta$ and $\pi_{\text{pe}} = \pi_\theta$, and further applying the Importance Sampling (IS) technique:

$$\max_\theta J_\pi^{\text{STR}}(\theta) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\frac{\bar{\pi}_\theta(a \mid s)}{\pi_\beta(a \mid s)} f(s,a;\bar{\pi}_\theta) \log \pi_\theta(a \mid s)\right], \tag{3.3}$$

where $\bar{\pi}_\theta(a \mid s)$ denotes the current policy with stop-gradient applied.

3.3.2 IQL

IQL [23] modifies the policy-evaluation objective in Eq. (2.11) by replacing the hard maximum used in Q-learning with an upper expectile. Moreover, to mitigate the influence of transition stochasticity arising from $s' \sim \mathcal{T}(\cdot \mid s,a)$, IQL introduces an additional value function to estimate an expectile under the action distribution. The objectives for the value function and Q-function in IQL are defined as

$$L_V^{\text{IQL}}(\psi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[L_2^\tau\big(Q_{\hat{\phi}}(s,a) - V_\psi(s)\big)\right], \tag{3.4}$$

$$L_Q^{\text{IQL}}(\phi) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\big(r(s,a) + \gamma V_\psi(s') - Q_\phi(s,a)\big)^2\right], \tag{3.5}$$

where

$$L_2^\tau(u) = |\tau - \mathbb{I}(u<0)|u^2, \qquad \tau \in (0,1),$$

and the τ-expectile of a random variable is defined as the solution to the corresponding asymmetric least-squares problem.

IQL extracts a policy using a wBC-style objective. Its policy-improvement objective is defined as

$$\max_{\theta} J_{\pi}^{\mathrm{IQL}}(\theta) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[f(s,a;\pi_\beta)\log \pi_\theta(a \mid s)\right]. \tag{3.6}$$

Following the definition of one-step algorithms [27], IQL can be classified as a one-step algorithm since the actions used for policy evaluation are drawn directly from the behavior policy, i.e., $a \sim \pi_\beta$ in $f(s,a;\pi_\beta)$. Furthermore, the actions for policy improvement are also restricted to the dataset $a \sim \mathcal{D}$, which allows IQL to benefit from in-sample learning and thus avoid OOD risk. However, this property simultaneously limits its potential for policy improvement.

3.4 Proposed Algorithm – PIQL

In this section, we present the proposed Projective Implicit Q-Learning (PIQL) algorithm in detail, including policy evaluation, policy improvement, and practical implementation.

3.4.1 Policy Evaluation in PIQL

For clarity, we begin from the expectile-regression view adopted in IQL [23]. For a random variable X, let $\mathbb{E}^{\tau}_{x\sim X}[x]$ be a τ-th expectile of X, i.e., the minimizer of the corresponding asymmetric least-squares objective. In particular, $\mathbb{E}^{0.5}[X]$ reduces to the standard expectation. Under this notation, the recursive relations between the value function $V_\tau(s)$ and the action-value function $Q_\tau(s,a)$ can be written as

$$V_\tau(s) = \mathbb{E}^{\tau}_{a\sim\pi_\beta(\cdot|s)}\left[Q_\tau(s,a)\right], \tag{3.7}$$

$$Q_\tau(s,a) = r(s,a) + \gamma \mathbb{E}_{s'\sim\mathcal{T}(\cdot|s,a)}\left[V_\tau(s')\right]. \tag{3.8}$$

Lemma 3.1. *For any state s and any τ_1, τ_2 satisfying $\tau_1 < \tau_2$, we have*

$$V_{\tau_1}(s) \le V_{\tau_2}(s).$$

By Lemma 3.1, a larger τ yields a more optimistic policy evaluation, and as $\tau \to 1$, the value update becomes increasingly close to the hard-max behavior in Q-learning. Conversely, a smaller τ induces a more conservative evaluation. Therefore, τ plays the role of a trade-off parameter between optimism and conservatism, and has a direct influence on the overall performance of IQL.

To empirically illustrate the sensitivity to τ, we evaluate IQL under three settings, $\tau \in \{0.3, 0.6, 0.9\}$, across 10 D4RL datasets. The results in Fig. 3.1 show that the performance of IQL varies substantially with different values of τ on the same dataset. This observation indicates that a fixed, manually chosen τ can lead to significantly suboptimal performance. Moreover, the optimal τ varies across datasets, which suggests that dataset-specific tuning is necessary to achieve good performance with IQL.

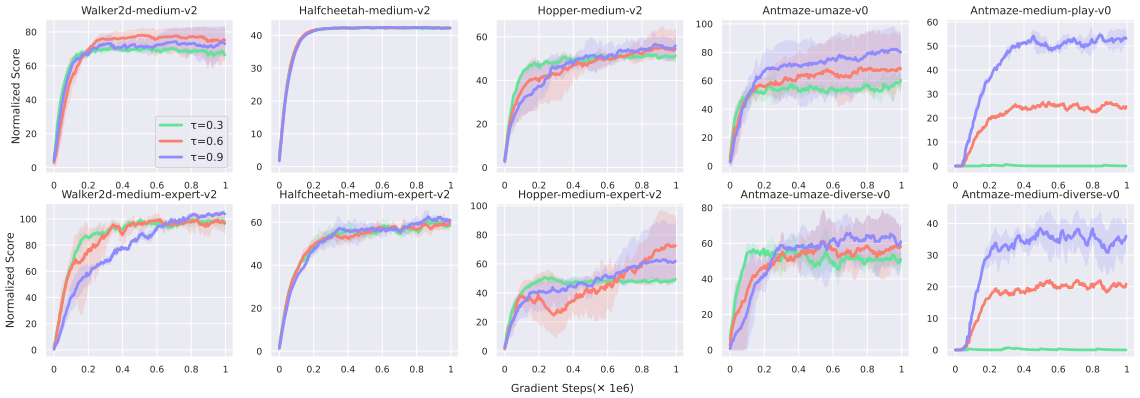


Figure 3.1: Sensitivity of the Parameter $\tau \in \{0.3, 0.6, 0.9\}$ across 10 D4RL Tasks.

This naturally raises the question of how τ should be chosen. To answer this question, we revisit the wBC-style policy-improvement objective employed in IQL, shown in Eq. (3.6). Formally, this objective can be interpreted through the following two-stage optimization problem.

First stage:

$$\begin{aligned} \pi^*(a \mid s) = \arg\max_{\pi} \mathbb{E}_{a \sim \pi(\cdot|s)} \left[A^{\pi_\beta}(s, a) \right], \\ \text{s.t.} \quad D_{\mathrm{KL}} \left[\pi(\cdot \mid s) \,\|\, \pi_\beta(\cdot \mid s) \right] \leq \epsilon, \\ \int_a \pi(a \mid s)\, da = 1. \end{aligned} \tag{3.9}$$

Second stage:

$$\pi_\theta = \arg\min_{\theta} \mathbb{E}_{d^{\pi_\beta}(s)} \left[D_{\mathrm{KL}} \left[\pi^*(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid s) \right] \right], \tag{3.10}$$

where π^* denotes the closed-form optimal solution, π_θ is the parametric policy used to approximate π^*, and $d^{\pi_\beta}(s)$ denotes the state distribution induced by the behavior policy.

According to the constraints in optimization problem (3.9) and (3.10), the parameter τ should first be designed to reflect the distance $D(\pi_\theta || \pi_\beta)$ between the learned policy π_θ and the behavior policy π_β. Intuitively, when π_θ is close to π_β, the policy evaluation

can afford to be more optimistic; when the two policies diverge, policy evaluation should become more conservative in order to control extrapolation error.

To quantify this relation, we introduce a projection-based adaptive parameter. Recall that the projection of a vector $\vec{u}$ onto a vector $\vec{v}$ is defined by

$$\text{Proj}_{\vec{v}}\vec{u} = \frac{\vec{u}\cdot\vec{v}}{\|\vec{v}\|}\frac{\vec{v}}{\|\vec{v}\|} = \frac{\vec{u}\cdot\vec{v}}{\|\vec{v}\|^2}\vec{v},$$

where $\|\vec{x}\| = \sqrt{\sum_{i=1}^{m} x_i^2}$ denotes the ℓ_2-norm of the vector $\vec{x} \in \mathbb{R}^m$.

In practice, the learned policy π_θ is approximated by neural networks and modeled as a K-dimensional diagonal Gaussian distribution, where K denotes the dimension of the action space. In most implementations, the policy network is trained and updated using mini-batch samples. Given n input samples $[(s_1, a_1), (s_2, a_2), \ldots, (s_n, a_n)]$, the policy network outputs

$$\vec{\pi}_\theta(a|s) = [\pi_\theta(a_1|s_1), \pi_\theta(a_2|s_2), ..., \pi_\theta(a_n|s_n)]\,.$$

Similarly, the behavior policy π_β can also be approximated by a neural network that output

$$\vec{\pi}_\beta(a|s) = \left[\pi_\beta(a_1|s_1), \pi_\beta(a_2|s_2), ..., \pi_\beta(a_n|s_n)\right].$$

Therefore, we propose a parameter $\vec{\tau}_{\text{proj}}(a|s)$ based on the vector projection $\text{Proj}_{\vec{\pi}_\theta}\vec{\pi}_\beta$ of the behavior policy vector $\vec{\pi}_\beta(a|s)$ onto the learned policy vector $\vec{\pi}_\theta(a|s)$,

$$\vec{\tau}_{\text{proj}}(a|s) = \frac{\vec{\pi}_\beta(a|s)\cdot\vec{\pi}_\theta(a|s)}{\|\vec{\pi}_\theta(a|s)\|^2}\vec{\pi}_\theta(a|s). \tag{3.11}$$

For a single state-action pair in the mini-batch, the corresponding component $\tau_{\text{proj}}(a \mid s)$ is given by

$$\tau_{\text{proj}}(a \mid s) = \frac{\vec{\pi}_\beta(a \mid s)\cdot\vec{\pi}_\theta(a \mid s)}{\|\vec{\pi}_\theta(a \mid s)\|^2}\pi_\theta(a \mid s). \tag{3.12}$$

Based on the $\tau_{\text{proj}}(a|s)$, the value function and Q-function for policy evaluation in PIQL are defined as

$$L_V(\psi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[L_2^{\tau_{\text{proj}}(a|s)}\big(Q_{\hat{\phi}}(s,a) - V_\psi(s)\big)\right], \tag{3.13}$$

$$L_Q(\phi) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\big(r(s,a) + \gamma V_\psi(s') - Q_\phi(s,a)\big)^2\right]. \tag{3.14}$$

The key advantages of $\tau_{\text{proj}}(a \mid s)$ can be summarized as follows.

1. Reflecting Policy Distance and Adapting Automatically. $\tau_{\text{proj}}(a|s)$ characterizes the length of the projection of $\vec{\pi}_\beta(a|s)$ onto $\vec{\pi}_\theta(a|s)$ at the state-action pair (s,a). As $\vec{\pi}_\theta(a|s)$ moves closer to $\vec{\pi}_\beta(a|s)$, the projection becomes larger, leading to a larger $\tau_{\text{proj}}(a|s)$ and thus permitting a more optimistic evaluation of the policy $\pi_\theta(a|s)$. In contrast, a smaller $\tau_{\text{proj}}(a|s)$ corresponds to a more conservative evaluation. The behavior of $\tau_{\text{proj}}(a|s)$ is consistent with the optimization problems in (3.9) and (3.10). Furthermore, $\tau_{\text{proj}}(a|s)$ is computed from batch samples under the current policy rather than specified as a fixed hyperparameter in advance. Therefore, it can be adjusted adaptively without requiring time-consuming manual tuning.

2. Multi-step yet In-sample Learning Paradigm. In IQL, both policy evaluation and policy improvement rely on in-dataset actions, which is why IQL is essentially a one-step method. By contrast, PIQL incorporates information from the current policy π_θ into the policy-evaluation stage through $\tau_{\text{proj}}(a \mid s)$, and thus admits a multi-step interpretation. At the same time, PIQL retains the expectile-regression structure and the in-sample learning paradigm, thereby continuing to mitigate the risk of OOD actions. This property is formalized in Theorem 3.2.

Theorem 3.2. *If the ϵ in the constraint of Eq. (3.9) is sufficiently small for the set*

$$\{a \mid \pi_\beta(a \mid s) > 0,\ \pi_\theta(a \mid s) > 0,\ Q_{\hat{\phi}}(s,a) - V_\psi(s) < 0,\ \forall s\},$$

then Eq. (3.13) can be transformed into

$$L_V(\psi) = \mathbb{E}_{s\sim\mathcal{D},\, a\sim\pi_\theta(\cdot|s)}\left[L_2^{\bar{\tau}_{\text{proj}}(a|s)}\left(Q_{\hat{\phi}}(s,a) - V_\psi(s)\right)\right],$$

where

$$\bar{\tau}_{\text{proj}}(a \mid s) = \frac{\vec{\pi}_\beta(a \mid s)\cdot\vec{\pi}_\theta(a \mid s)}{\|\vec{\pi}_\theta(a \mid s)\|^2}\pi_\beta(a \mid s).$$

Theorem 3.2 shows that the loss function $L_V(\psi)$ can be equivalently rewritten in a form where actions are sampled from the current policy π_θ. This equivalence provides a multi-step interpretation of PIQL, even though the practical objective in Eq. (3.13) is computed from dataset actions. Moreover, Theorem 3.2 reformulates the objective within a complete expectile-regression framework by transforming $\tau_{\text{proj}}(a|s)$ into $\bar{\tau}_{\text{proj}}(a|s)$.

It should be emphasized that Theorem 3.2 is derived under the assumption that ϵ is sufficiently small for actions satisfying

$$\{a \mid Q_{\hat{\phi}}(s,a) - V_\psi(s) < 0\}.$$

This assumption is reasonable because such actions are generally regarded as suboptimal. For these actions, the desirable behavior is to stay close to imitation so as to avoid OOD overestimation, rather than to encourage unnecessary exploration.

3.4.2 Policy Improvement in PIQL

The policy improvement step in IQL adopts a wBC-style objective, as shown in Eq. (3.6). However, directly applying the same objective to PIQL does not yield satisfactory performance. Intuitively, when the KL constraint in Eq. (3.9) is too restrictive, the learned policy cannot move sufficiently far from the behavior policy to realize meaningful policy improvement. This issue is formalized in Lemma 3.3.

Lemma 3.3. *If* $D_{\mathrm{KL}}\left[\pi_\theta(\cdot \mid s) \,\|\, \pi_\beta(\cdot \mid s)\right] \leq \epsilon, \forall s$, *then the performance* $\eta(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$ *satisfies*

$$\eta(\pi_\theta) \leq \eta(\pi_\beta) + \frac{V_{\max}}{\sqrt{2}(1-\gamma)}\sqrt{\epsilon},$$

where

$$0 \leq Q^\pi, V^\pi \leq \frac{R_{\max}}{1-\gamma} =: V_{\max}.$$

According to Lemma 3.3, achieving strong policy improvement may require a relatively large ϵ. However, such a requirement conflicts with the small-ϵ assumption used in Theorem 3.2 for policy evaluation. Therefore, the density-constrained wBC policy-improvement objective is not well aligned with the policy-evaluation design of PIQL. By contrast, support-constrained policy improvement relaxes density matching while still preventing the learned policy from selecting unsupported actions. For this reason, PIQL adopts the support-constrained objective:

$$\begin{aligned}
\max_\theta J_\pi(\theta) &= \mathbb{E}_{s\sim\mathcal{D},\, a\sim\pi_\theta(\cdot|s)}\left[f(s,a;\bar{\pi}_\theta,\bar{\tau}_{\mathrm{proj}})\log\pi_\theta(a \mid s)\right] \\
&= \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\frac{\bar{\pi}_\theta(a \mid s)}{\pi_\beta(a \mid s)} f(s,a;\bar{\pi}_\theta,\tau_{\mathrm{proj}})\log\pi_\theta(a \mid s)\right] \\
&= \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\frac{\bar{\pi}_\theta(a \mid s)}{\pi_\beta(a \mid s)} \exp\left(\frac{A^{\bar{\pi}_\theta}_{\tau_{\mathrm{proj}}}(s,a)}{\lambda}\right)\log\pi_\theta(a \mid s)\right],
\end{aligned} \tag{3.15}$$

where $A^{\bar{\pi}_\theta}_{\tau_{\mathrm{proj}}}(s,a) = Q^{\bar{\pi}_\theta}_{\tau_{\mathrm{proj}}}(s,a) - V^{\bar{\pi}_\theta}_{\tau_{\mathrm{proj}}}(s)$.

With an exact Q-function, PIQL enjoys monotonic policy improvement, as stated in Theorem 3.4.

Theorem 3.4. *Assume that the Q-function is exact and that*

$$\tau_{k+1}(a \mid s) \geq \tau_k(a \mid s),$$

where, for simplicity, $\tau_k(a \mid s)$ *denotes the value of* $\tau_{\mathrm{proj}}(a \mid s)$ *at iteration k. Then the sequence of policies* $\{\pi_k\}$ *generated by PIQL satisfies*

$$Q^{\pi_{k+1}}_{\tau_{k+1}}(s,a) \geq Q^{\pi_k}_{\tau_k}(s,a), \qquad \forall s,a.$$

Theorem 3.4 demonstrates that PIQL guarantees monotonic policy improvement under support-constrained policy optimization.

Compared with STR [13], which also adopts a support-constraint approach, PIQL exhibits a clear advantage by enforcing a more rigorous criterion for evaluating *advantageous actions*, where advantageous actions are defined as $\{a \mid Q(s,a) - V(s) \geq 0\}$. Intuitively, progressively tightening the advantageous actions evaluation criterion leads to a more "concentrated" policy, i.e., one that selects advantageous actions with higher probability.

To formalize this idea, we consider the Monte Carlo view of estimating the probability of advantageous actions under a sampling policy. Specifically, we sample a large number of actions according to a certain policy $a \sim \pi(\cdot|s)$ and compare the probabilities associated with advantageous actions. If

$$P\left\{Q_{\tau_{k+1}}^{\pi_{k+1}}(s,a) - V_{\tau_{k+1}}^{\pi_{k+1}}(s) \geq 0\right\} \leq P\left\{Q_{\tau_k}^{\pi_k}(s,a) - V_{\tau_k}^{\pi_k}(s) \geq 0\right\}, \tag{3.16}$$

then we say that the advantageous-action evaluation criterion at iteration $k+1$ is more rigorous than that at iteration k, where $P\{\mathcal{M}\}$ denotes the probability of event $\mathcal{M}$.

We first establish the expectation form of this increasingly rigorous criterion in Theorem 3.5.

Theorem 3.5 (Expectation Rigorous Criterion for Advantageous Actions)**.** *For any $s \in \mathcal{D}$ and $a \sim \pi_{k+1}(\cdot \mid s)$, if*

$$0.5 \leq \tau_k(a \mid s),\ \tau_{k+1}(a \mid s) \leq 1,$$

then

$$\mathbb{E}_{a\sim\pi_{k+1}(\cdot|s)}\left[Q_{\tau_{k+1}}^{\pi_{k+1}}(s,a) - V_{\tau_{k+1}}^{\pi_{k+1}}(s)\right] \leq \mathbb{E}_{a\sim\pi_{k+1}(\cdot|s)}\left[Q_{\tau_k}^{\pi_k}(s,a) - V_{\tau_k}^{\pi_k}(s)\right].$$

Theorem 3.5 establishes the expectation form of the advantageous action evaluation criterion. Specifically, it states that the expectation of advantage function $A_{\tau_{k+1}}^{\pi_{k+1}}(s,a)$ under the policy π_{k+1} is lower than the expectation of $A_{\tau_k}^{\pi_k}(s,a)$. Intuitively, for the same sampling policy π_{k+1}, if $A_{\tau_{k+1}}^{\pi_{k+1}}(s,a)$ is smaller than $A_{\tau_k}^{\pi_k}(s,a)$, then the corresponding expectation is also smaller. This result indicates that the evaluation criterion for advantageous actions at the $(k+1)$-th iteration, $A_{\tau_{k+1}}^{\pi_{k+1}}(s,a)$, is more rigorous than the previous criterion $A_{\tau_k}^{\pi_k}(s,a)$, as formalized in Eq. (3.16).

Furthermore, Theorem 3.7 extends the result in Theorem 3.5 to a maximum-case formulation of the increasingly rigorous evaluation criterion under two arbitrary policies. Before presenting Theorem 3.7, we first introduce Lemma 3.6.

Lemma 3.6. *For any random variable X, if $0.5 \leq \tau_1 \leq \tau_2 \leq 1$, then*

$$\mathrm{Var}^{\tau_1}(X) \leq \mathrm{Var}^{\tau_2}(X),$$

where

$$\mathrm{Var}^{\tau}(X) = \mathbb{E}\left[\left(X - \mathbb{E}^{\tau}[X]\right)^2\right].$$

Theorem 3.7 (Maximum Rigorous Criterion for Advantageous Actions). *For any $s \in \mathcal{D}$, if*

$$0.5 \leq \tau_k(a \mid s) \leq \tau_{k+1}(a \mid s) \leq 1,$$

then

$$\max_{a \sim \pi_1(\cdot|s)} P\left\{Q_{\tau_{k+1}}^{\pi_{k+1}}(s,a) - V_{\tau_{k+1}}^{\pi_{k+1}}(s) \geq 0\right\} \leq \max_{a \sim \pi_2(\cdot|s)} P\left\{Q_{\tau_k}^{\pi_k}(s,a) - V_{\tau_k}^{\pi_k}(s) \geq 0\right\},$$

where $\pi_1(\cdot \mid s)$ and $\pi_2(\cdot \mid s)$ are arbitrary policies.

Theorem 3.7 generalizes Theorem 3.5 by extending the sampling policy π_{k+1} to two arbitrary policies. Intuitively, Theorem 3.7 shows that the maximum probability of selecting advantageous actions becomes progressively more rigorous across iterations, regardless of which policies are utilized for sampling. This result indicates that even under the most favorable conditions, the evaluation criterion tightens over time, thereby ensuring that the policy improvement process consistently focuses on truly advantageous actions. In addition, Theorem 3.7 is established under the assumption that $\tau_k(a|s) \leq \tau_{k+1}(a|s)$. In subsequent experiments, $\tau_{\text{proj}}(a|s)$ exhibits an increasing trend, which supports the validity of this assumption.

The detailed proof is provided in Appendix 7.1.

3.4.3 Practical Implementation in PIQL

Behavior Policy. The behavior policy $\pi_\beta(a \mid s)$ is approximated by a neural network trained via behavior cloning:

$$\pi_\beta(a \mid s) = \arg\max_{\pi_\beta} \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[\log \pi_\beta(a \mid s)\right]. \tag{3.17}$$

Projection-parameter Processing. In Theorems 3.5 and 3.7, we assume that the projection parameter satisfies

$$0.5 \leq \tau_{\text{proj}}(a \mid s) \leq 1.$$

In practice, this constraint is enforced by clipping:

$$\tau_{\text{proj}}(a \mid s) = \text{clip}\big(\tau_{\text{proj}}(a \mid s),\, 0.5,\, 1\big),$$

where $\text{clip}(x, l, u) = \min(\max(x, l), u)$.

In implementation, we found that computing $\tau_{\text{proj}}(a \mid s)$ independently for each state-action pair can lead to overly fine-grained and unstable policy evaluation. To improve stability, we therefore use the mean value of $\tau_{\text{proj}}(a \mid s)$ over the mini-batch in each update step.

Importance Sampling. To mitigate high variance as well as training instability, we employ the Self-Normalized Importance Sampling (SNIS) method [13], which normalizes the IS within each mini-batch sample.

Overall Architecture. Combining the policy evaluation objective, the policy improvement objective, and the practical implementation described above, the complete PIQL algorithm is summarized in Algorithm 1.

Algorithm 1 Projective Implicit Q-Learning (PIQL)

Input: Offline dataset $\mathcal{D} = \{(s, a, r, s')\}$.
Initialize: Q-network parameters ϕ, target Q-network parameters $\hat{\phi}$, value-network parameters ψ, and policy-network parameters θ.

// Pre-train the behavior policy for N_β steps.
for $t = 1$ to N_β **do**
    Update the behavior policy π_β using Eq. (3.17).
end for
// Policy evaluation and policy improvement.
for each gradient step **do**
    Sample a mini-batch $(s, a, r, s') \sim \mathcal{D}$.
    Compute $\tau_{\text{proj}}(a \mid s)$ using Eq. (3.12).
    Compute the mini-batch mean of $\tau_{\text{proj}}(a \mid s)$.
    Update the value network V_ψ using Eq. (3.13).
    Update the Q-network Q_ϕ using Eq. (3.14).
    Update the policy network π_θ using Eq. (3.15).
    Soft-update the target Q-network: $\hat{\phi} \leftarrow (1 - \lambda_{\text{tar}})\hat{\phi} + \lambda_{\text{tar}}\phi$.
end for
return policy π_θ.

3.5 Experiments

3.5.1 Comparisons on D4RL Benchmarks

Gym-MuJoCo Locomotion Tasks are popular benchmarks used in prior offline RL works. Accordingly, we compare PIQL with several competitive and commonly adopted baselines, including Decision Transformer (DT) [87], RvS [130], POR [113], CQL [24], LAPO [96], TD3+BC [38], Reinformer [90], CRR [22] and IQL [23]. The results are reported in Tab. 3.1, where we average the mean return over 10 evaluation trajectories and 5 random seeds, and additionally report the standard deviation.

To facilitate comparison across tasks, the normalized score in D4RL is defined as,

$$\text{Normalized Score} = \frac{\textit{Score} - \text{Random Score}}{\text{Expert Score} - \text{Random Score}} \times 100\%. \tag{3.18}$$

Table 3.1: Averaged normalized scores of PIQL on Gym-MuJoCo-v2 datasets. Scores are reported as mean ± standard deviation. **Bold** indicates the top-2 performance in each task.

Dataset	DT	RvS	POR	CQL	LAPO	TD3+BC	Reinformer	CRR	IQL	PIQL
halfcheetah-med	42.6	41.6	**48.8**	44.4	46.0	42.8	42.9	47.1	47.4	**48.7**±0.45
hopper-med	67.6	60.2	78.6	**86.6**	51.6	**99.5**	81.6	38.1	66.3	68.2±13.90
walker2d-med	74.0	71.7	**81.1**	74.5	80.8	79.7	80.5	59.7	78.3	**83.4**±0.63
halfcheetah-med-rep	36.6	38.0	43.5	**46.2**	43.9	43.3	39.0	44.4	44.2	**45.5**±0.84
hopper-med-rep	82.7	73.5	**98.9**	48.6	37.6	31.4	83.3	25.5	94.7	**95.7**±10.63
walker2d-med-rep	66.6	60.6	**76.6**	32.6	52.3	25.2	72.8	27.0	73.9	**89.5**±2.68
halfcheetah-med-exp	86.8	92.2	**94.7**	62.4	86.1	**97.9**	92.0	85.2	86.7	94.2±1.04
hopper-med-exp	107.6	101.7	90.0	**111.0**	100.7	**112.2**	107.8	53.0	91.5	101.5±19.82
walker2d-med-exp	108.1	106.0	109.1	98.7	109.4	101.1	109.3	91.3	**109.6**	**112.2**±0.93
Gym-MuJoCo total	672.6	645.5	**721.3**	605.0	608.4	633.1	709.2	471.3	692.4	**738.9**±50.92

Although the performance of many offline RL algorithms on the Gym-MuJoCo-v2 benchmarks has become largely saturated, PIQL still delivers strong results, ranking within the top two on 6 of the 9 tasks and achieving the highest aggregate score across all 9 datasets. These results further demonstrate the effectiveness of PIQL.

Table 3.2: Averaged normalized scores of PIQL on AntMaze-v0 and Kitchen-v0 datasets. Scores are reported as mean ± standard deviation. **Bold** indicates the top-2 performance in each task.

Dataset	BC	BCQ	BEAR	CQL	Onestep	TD3+BC	Diffusion-QL	$\mathcal{X}$-QL	IQL	PIQL
antmaze-u	54.6	78.9	73.0	74.0	62.4	78.6	93.4	**93.8**	87.5	**98.0**±6.0
antmaze-u-d	45.6	55.0	61.0	**84.0**	44.7	71.4	66.2	82.0	62.2	**98.0**±6.0
antmaze-m-p	0.0	0.0	0.0	61.2	5.4	10.6	**76.6**	76.0	71.2	**94.0**±14.0
antmaze-m-d	0.0	0.0	8.0	53.7	1.8	3.0	**78.6**	73.6	70.0	**88.0**±8.0
antmaze-l-p	0.0	6.7	0.0	15.8	0.1	0.2	46.4	**46.5**	39.6	**72.0**±9.8
antmaze-l-d	0.0	2.2	0.0	14.9	0.9	0.0	**56.6**	49.0	47.5	**50.0**±10.0
AntMaze total	100.2	142.8	142.0	303.6	115.2	163.8	417.8	**420.9**	378.0	**500.0**±53.8
kitchen-complete	65.0	8.1	0.0	43.8	66.0	61.5	**84.0**	82.4	62.5	**87.0**±4.6
kitchen-partial	38.0	18.9	13.1	49.8	59.3	52.8	60.5	**73.7**	46.3	**67.0**±6.0
kitchen-mixed	51.5	8.1	47.2	51.0	56.5	60.8	**62.6**	**62.5**	51.0	55.5±3.2
Kitchen total	154.5	35.1	60.3	144.6	181.7	175.1	207.1	**218.6**	159.8	**209.5**±13.8

The AntMaze and Kitchen tasks are considerably more challenging than the Gym-MuJoCo locomotion tasks. The primary difficulty lies in learning policies capable of long-horizon planning from datasets that lack optimal trajectories. To assess the performance of PIQL, we compare it with several strong baselines, including BC [56], BCQ [17],

BEAR [18], CQL [24], Onestep [27], TD3+BC [38], Diffusion-QL [104], $\mathcal{X}$-QL [34], and IQL [23]. For fairness, we use the same dataset versions as those adopted in IQL. The corresponding results are reported in Tab. 3.2.

As reported in Tab. 3.2, imitation learning methods such as BC and BCQ fail to achieve competitive results. A main reason is that these tasks are characterized by sparse rewards and datasets with a large number of suboptimal trajectories, which place higher demands on accurate and stable estimation of both the Q-function and the value function. PIQL addresses these challenges by dynamically and adaptively tuning the τ_{proj} parameter, so that the Q-function and value function can be estimated according to the current policy. As a result, PIQL achieves excellent performance on the AntMaze-v0 and Kitchen-v0 datasets, and shows particularly strong results on AntMaze, where it obtains the best scores on 5 of the 6 datasets.

3.5.2 The Training Curves of τ_{proj}

Unlike the fixed hyperparameter τ used in IQL, $\tau_{\text{proj}}(a \mid s)$ in PIQL can be adjusted in a flexible and adaptive manner. To illustrate this property, we visualize the curves of $\tau_{\text{proj}}(a \mid s)$ together with the normalized scores on the AntMaze-v0 and Kitchen-v0 datasets. The corresponding results are presented in Fig. 3.2.

As illustrated in Fig. 3.2, the curve of $\tau_{\text{proj}}(a \mid s)$ generally increases over training, which indicates that the projected component of $\pi_\theta(a \mid s)$ within $\pi_\beta(a \mid s)$ becomes larger. One possible explanation is that, in PIQL, the policy update can be interpreted as a wBC procedure, as shown in Eq. (3.15), where the corresponding weight is

$$\frac{\bar{\pi}_\theta(a \mid s)}{\pi_\beta(a \mid s)} \exp\left(\frac{A^{\bar{\pi}_\theta}_{\tau_{\text{proj}}}(s,a)}{\lambda}\right) > 0.$$

Nevertheless, this weight differs fundamentally from that used in the aforementioned wBC method, because the support constraint in PIQL is established via importance sampling. In addition, the observed increasing tendency is consistent with our assumption $0.5 \leq \tau_k(a \mid s) \leq \tau_{k+1}(a \mid s) \leq 1$ in Theorem 3.4 and Theorem 3.7.

A further interesting finding is that the improvement in normalized score appears later than the increase in $\tau_{\text{proj}}(a \mid s)$. This suggests that adjusting $\tau_{\text{proj}}(a \mid s)$ first helps produce more accurate estimates of the Q-function and value function, and these improved estimates subsequently promote continual policy improvement, leading to higher normalized scores. This tendency is less obvious on the `Kitchen-complete-v0` dataset, likely because this dataset is much smaller and contains only $3,680$ samples. By comparison, `AntMaze-v0` datasets usually include around 10^6 samples, while the other `Kitchen-v0` datasets contain $136,950$ samples.

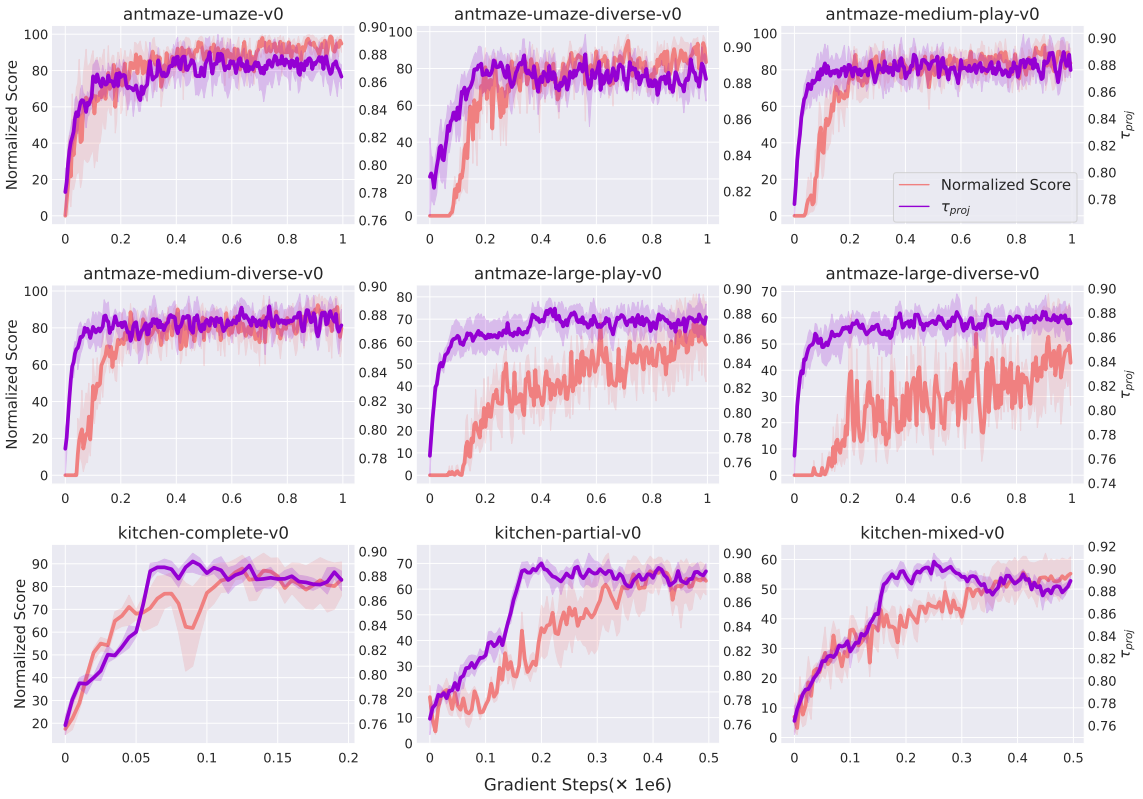


Figure 3.2: The training curves of normalized score and $\tau_{\text{proj}}(a|s)$ on AntMaze-v0 and Kitchen-v0 datasets.

3.5.3 Ablation Experiment

In the proposed PIQL algorithm, every sample within a batch is regarded as one component of the vector for computing $\tau_{\text{proj}}(a \mid s)$; therefore, the *batch size* is a crucial hyperparameter. To examine its influence, we tested the behavior of $\tau_{\text{proj}}(a \mid s)$ and the corresponding normalized scores with batch sizes of 16, 64, 128, and 256 on three `Kitchen-v0` datasets. The experimental results are shown in Fig. 3.3.

As illustrated in Fig. 3.3, larger batch sizes generally lead to higher normalized scores, which is consistent with intuition. This gain can be attributed to the fact that increasing the batch size enables more effective optimization of both the policy network and the value network. However, such an observation alone does not completely explain how batch size specifically affects $\tau_{\text{proj}}(a \mid s)$. For this reason, we further visualize the relation between batch size and $\tau_{\text{proj}}(a \mid s)$. The results show that, as the batch size becomes larger, $\tau_{\text{proj}}(a \mid s)$ behaves more stably and exhibits smaller fluctuations. Such increased stability



Figure 3.3: The training curves of normalized score and $\tau_{\text{proj}}(a \mid s)$ on Kitchen-v0 datasets under different batch sizes.

of $\tau_{\text{proj}}(a \mid s)$ is especially helpful for improving performance, since it leads to more stable policy evaluation.

3.5.4 Comparisons on NeoRL2 Benchmarks

NeoRL2 [124] is a recently introduced benchmark intended to better capture the practical challenges of offline RL. It contains domains inspired by real-world scenarios, including industrial control, nuclear fusion, and healthcare. In addition, practical limitations in data collection restrict both the size of the datasets and their coverage.

Table 3.3: Averaged normalized scores of PIQL on NeoRL2 datasets. Scores are reported as mean ± standard deviation. **Bold** indicates the top-2 performance in each task.

Dataset	Data	BC	CQL	EDAC	MCQ	TD3+BC	MOPO	COMBO	RAMBO	MOBILE	PIQL
Pipeline	69.3	68.6	81.1	72.9	49.7	**82.0**	-26.3	55.5	24.1	65.5	**89.3**±14.8
Simglucose	73.9	**75.1**	11.0	8.1	29.6	74.2	34.6	23.2	10.8	9.3	**82.1**±3.6
RocketRecovery	75.3	72.8	74.3	65.7	76.5	**79.7**	-27.7	74.7	-44.2	43.7	**81.6**±7.5
RandomFrictionHopper	28.7	28.0	33.0	**34.7**	31.7	29.5	32.5	34.1	29.6	**35.1**	34.2±0.7
DMSD	56.6	65.1	70.2	**78.7**	**77.8**	60.0	68.2	68.3	76.2	64.4	54.2±0.1
Fusion	48.8	55.2	55.9	58.0	49.7	54.6	-11.6	55.5	**59.6**	5.0	**59.8**±2.1
SafetyHalfCheetah	**73.6**	70.2	**71.2**	53.1	54.7	68.6	23.7	57.8	-422.4	8.7	70.7±1.2
NeoRL2 total	426.2	435.0	396.7	371.2	369.7	**448.6**	93.4	369.1	-266.3	231.7	**471.9** ± 30.0

We evaluate PIQL against the baselines reported on NeoRL2, and the corresponding results are listed in Tab. 3.3. As can be seen from Tab. 3.3, PIQL achieves the best results

on 4 of the 7 datasets and obtains the highest overall score, significantly surpassing the strongest competing baseline.

3.6 Conclusion

In this chapter, we proposed Projective Implicit Q-Learning (PIQL), a support-constrained offline RL algorithm that reconciles in-sample learning with multi-step learning on the basis of IQL. In the policy-evaluation stage, PIQL replaces the fixed expectile hyperparameter in IQL with a projection-based adaptive parameter, thereby extending one-step learning to a multi-step formulation while retaining the expectile-regression structure. In the policy-improvement stage, PIQL adopts a support-constrained objective that is better aligned with the policy-evaluation design and enables more effective policy optimization.

We further provided a theoretical foundation for PIQL, showing that it admits a multi-step interpretation while preserving the in-sample learning paradigm, guarantees monotonic policy improvement under support-constrained updates, and induces a progressively stricter criterion for identifying advantageous actions across iterations. Extensive experiments on D4RL and NeoRL2 demonstrated that PIQL consistently achieves strong empirical performance, with particularly notable gains on long-horizon stitching-style tasks.

Chapter 4

Automatic Constraint Policy Optimization based on the Continuous Constraint Interpolation Framework

This chapter addresses the third research question ($RQ3$ in Section 1.1) of the thesis: how to develop a unified framework that bridges different policy-constraint families and automatically tunes both constraint type and strength in offline RL. Existing policy-constrained methods are effective at mitigating extrapolation error by restricting policy updates toward the behavior distribution or its support, but they typically operate within a single constraint paradigm and require manual selection of both the constraint form and its strength. To address this limitation, we propose the Continuous Constraint Interpolation (CCI) framework, which unifies wBC, KL-based density regularization, and support constraints under a single interpolation mechanism, and further develop Automatic Constraint Policy Optimization (ACPO), a practical primal-dual algorithm that automatically adapts the interpolation parameter during training.

The remainder of this chapter is organized as follows. Section 4.1 introduces the problem background and motivation for ACPO. Section 4.2 reviews the related work, and Section 4.3 presents the necessary preliminaries. Section 4.4 presents the proposed CCI framework and the ACPO algorithm, followed by the corresponding theoretical analysis. Finally, Section 4.5 reports the experimental results and ablation studies on standard offline RL benchmarks.

4.1 Introduction

Policy constraints constitute a dominant paradigm for mitigating extrapolation error in offline RL. By restricting the learned policy to remain close to the behavior policy or its support, these methods aim to prevent policy optimization from selecting OOD actions that are weakly supported by the dataset. However, in practical offline RL settings, the behavior policy is often unknown or highly heterogeneous, since the data may be collected from random exploration, mixtures of multiple tasks, multiple behavior sources, or logged trajectories produced during online RL training [65,96]. Under such ambiguity, the choice of policy-constraint form and strength becomes a critical design issue.

From a high-level perspective, mainstream policy-constrained offline RL approaches can be organized into three categories along a common constraint spectrum: wBC, density regularization, and support constraints. Methods in the wBC family remain essentially imitation-based and usually reduce to weighted supervised learning over the offline dataset [35–38]. Density-regularization methods permit controlled deviation from the behavior policy at the distribution level. Among them, KL-based density regularization is particularly common for restricting the extent of distributional shift away from the behavior distribution [19–22]. By contrast, support-constraint methods do not aim to match the behavior distribution directly, but instead encourage the learned policy to choose actions that lie within the support of the behavior policy [13,14,18,39].

A central challenge is that neither the appropriate constraint family nor the appropriate constraint strength is known in advance. Overly conservative constraints may improve stability but undermine the core promise of offline RL – policy improvement beyond the dataset. Overly weak constraints, on the other hand, fail to meaningfully restrict OOD actions and may induce severe value misestimation that misguides optimization [25,26].

Several recent works [14,131,132] have explored adaptive constraint mechanisms to cope with this trade-off. However, existing approaches typically adapt only within a single constraint paradigm. A unified view that bridges wBC, KL-based density regularization, and support constraints, while also enabling automatic adaptation across them, remains missing. As a result, practitioners still need to manually choose the constraint form and then tune its strength, which can be nontrivial across datasets and tasks.

In this chapter, we develop the *Continuous Constraint Interpolation (CCI)* framework, a unified optimization framework in which the above three policy-constraint families arise as special cases along a common interpolation spectrum. Specifically, the CCI framework exposes a single interpolation parameter that enables smooth transitions and principled combinations among wBC, KL-based density regularization, and support constraints for policy improvement. Building on CCI, we further propose *Automatic Constraint Policy Optimization (ACPO)*, a practical primal-dual offline RL algorithm that automatically tunes the interpolation parameter via a Lagrangian dual update, thereby adapting both the effective constraint type and its strength during training. Intuitively, the interpolation pa-

rameter acts as a constraint "dial" that automatically adapts the constraint regime during training.

Our key contributions in this chapter are summarized as follows:

- **Unified CCI Framework.** We propose CCI, a unified optimization framework that bridges three major policy-constraint families: wBC, KL-based density regularization, and support constraints, and enables smooth transitions as well as principled combinations across constraint regimes.
- **Practical ACPO Algorithm.** Building on CCI, we develop ACPO, a practical primal-dual algorithm that automatically adapts the interpolation parameter through a Lagrangian dual update. We further establish a maximum-entropy performance difference lemma and derive performance lower bounds for both the optimal policy and its practical parametric projection.
- **Empirical Evaluation.** We conduct extensive comparisons and ablation studies on the standard D4RL and the recent NeoRL2 benchmarks, showing that ACPO achieves consistent improvements across diverse domains and tasks.

4.2 Related Work

In offline RL, policy constraints are not the sole approach for reducing extrapolation error caused by OOD actions. In addition to policy-constrained methods, previous studies have investigated several alternative directions, such as conservative value estimation [24, 66, 133], uncertainty-aware approaches based on ensembles [55, 72, 134], and DIstribution Correction Estimation (DICE) methods [52, 81, 83]. To better emphasize the strengths of our proposed framework and algorithm, the remainder of this section concentrates on policy-constrained offline RL. Specifically, we organize representative methods into three classes: wBC, density regularization, and support constraint.

Methods in the wBC category either formulate policy learning as a form of (weighted) supervised imitation over the offline dataset, or introduce an additional BC regularization term into the policy improvement objective. For example, BAIL [35] first learns a value function and then imitates only those actions judged to be high-quality according to that value estimate. IBC [36] enhances policy expressiveness through implicit energy-based policies, thereby avoiding the restrictions of explicit regression-style models. DWBC [37] employs a discriminator to measure expert-likeness and uses the resulting scores to weight the BC objective. By comparison, TD3+BC [38] combines value-based policy improvement with an explicit BC penalty, which helps reduce distributional drift during the improvement process.

Density-regularization approaches permit the learned policy to deviate from the behavior policy in a controlled way by penalizing policy-level distribution shift. Among them, KL-based regularization [19–23] is the most widely adopted choice because it

yields an elegant and tractable policy improvement objective while keeping updates stable through in-sample learning. For completeness, we also consider density regularization forms beyond KL divergence, including MMD-based constraints for sample-level distribution matching [59], Fisher divergence regularization [62], and Wasserstein-distance-based regularization [61, 135].

Support-constraint methods carry out policy improvement while restricting the policy to remain within the support of the behavior policy. EMaQ [63] generates multiple candidate actions from the behavior policy and performs backups over these in-support action proposals. STR [13] integrates trust-region policy optimization with an explicit support constraint implemented via Importance Sampling (IS), so that each policy update stays inside the behavior support. SPOT [39] learns a VAE-based estimator for the support of the behavior policy and uses a simple plug-in regularizer to ensure that policy improvement is confined to the estimated support.

In summary, most existing methods are developed within only one family of policy constraints. What is still lacking is a unified optimization perspective that connects all three families in a single framework, allowing smooth interpolation, flexible integration, and automatic adaptation across different constraint regimes. This limitation motivates the proposed CCI framework, which offers such a unified perspective, as well as ACPO, which further enables practical automatic adaptation over the induced spectrum of constraints.

4.3 Preliminaries

This section introduces the main formulations used in the development of the CCI framework and ACPO, including the maximum-entropy RL setting and the generic policy-optimization objectives associated with the three policy-constraint families considered in this chapter.

4.3.1 Maximum Entropy RL

Since the CCI framework and the theoretical analysis in this chapter are developed under the maximum-entropy RL setting, we first briefly recall the corresponding objective and value definitions. Let $\eta(\pi)$ denote the standard discounted return:

$$\eta(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t)\right], \tag{4.1}$$

where $\mathbb{E}_{\tau\sim\pi}[...]$ denotes the expectation under the trajectory τ induced by the interconnection of π and the environment.

The performance difference lemma [136] in the standard RL setting states that

$$\eta(\pi') - \eta(\pi) = \frac{1}{1-\gamma} \sum_s d_{\pi'}(s) \sum_a \pi'(a|s)\, A^{\pi}(s,a), \tag{4.2}$$

where $d_{\pi'}(s)$ denotes the discounted state visitation distribution under π', and $A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s)$ is the advantage function.

In the maximum-entropy setting [137–140], the objective augments the reward with an entropy regularizer:

$$\mathcal{J}(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t \Big(r(s_t,a_t) - \alpha \log \pi(a_t \mid s_t) \Big)\right], \tag{4.3}$$

where $\alpha > 0$ is the temperature parameter. The entropy of policy π at state s is

$$\mathcal{H}(s,\pi) = \mathbb{E}_{a\sim\pi(\cdot|s)}\big[-\log \pi(a \mid s)\big].$$

The corresponding maximum-entropy value function is defined as

$$V^{\pi}(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}\left[Q^{\pi}(s,a) - \alpha \log \pi(a \mid s)\right], \tag{4.4}$$

and the soft Q-function satisfies

$$Q^{\pi}(s,a) = r(s,a) + \gamma \mathbb{E}_{s'\sim\mathcal{T}(\cdot|s,a)}\left[V^{\pi}(s')\right]. \tag{4.5}$$

As $\alpha \to 0$, the maximum-entropy RL recovers the standard RL.

4.3.2 wBC Methods

The generalized wBC objective can be defined as follows,

$$\max_{\pi}\ \mathcal{J}^{\mathrm{wBC}}(\pi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\Big[w^{\mathrm{bc}}(s,a) \log \pi(a \mid s)\Big], \tag{4.6}$$

where $w^{\mathrm{bc}} : \mathcal{S}\times\mathcal{A} \to \mathbb{R}_+$ denotes a weight function. For example, $w^{\mathrm{bc}}(s,a) = 1$ corresponds to the vanilla BC method, while more sophisticated choices are used in, e.g., [35, 37].

4.3.3 KL-based Density Regularization Methods

KL-based density-regularization methods can be summarized by the following constrained policy-optimization problem [13]:

$$\begin{aligned} \pi^*(a \mid s) &= \arg\max_{\pi} \mathbb{E}_{a\sim\pi(\cdot|s)}\left[A^{\pi_{\mathrm{pe}}}(s,a)\right], \\ \text{s.t.}\quad & D_{\mathrm{KL}}\left[\pi(\cdot \mid s) \,\|\, \pi_{\mathrm{base}}(\cdot \mid s)\right] \le \epsilon, \\ & \int_a \pi(a \mid s)\, da = 1, \end{aligned} \tag{4.7}$$

where π_{base} denotes a reference (or sampling) policy and π_{pe} denotes the evaluation policy. After solving the constrained optimization problem and projecting the resulting optimal policy into a parametric policy class, one obtains the following weighted log-likelihood objective:

$$\max_{\pi} \; \mathcal{J}^{\text{KL}}(\pi) = \mathbb{E}_{s\sim\mathcal{D},\, a\sim\pi_{\text{base}}} \left[w^{\text{kl}}(s,a) \log \pi(a \mid s) \right], \tag{4.8}$$

where

$$w^{\text{kl}}(s,a) = \exp\left(\frac{A^{\pi_{\text{pe}}}(s,a)}{\alpha} \right).$$

Different choices of π_{base} and π_{pe} give rise to different algorithmic instantiations [21, 22, 141], while sharing the same KL-regularized principle.

4.3.4 Support Constraint Methods

The support constraint requires the learned policy to assign positive density only on actions supported by the behavior policy, e.g.,

$$\pi_\beta(a \mid s) = 0 \quad \Rightarrow \quad \pi(a \mid s) = 0.$$

InAC [64] introduces an in-sample softmax operator and implements support-constrained policy improvement by sampling actions directly from the dataset:

$$\max_{\pi} \; \mathcal{J}^{\text{Supp}}(\pi) = \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[w^{\text{supp}}(s,a) \log \pi(a \mid s) \right], \tag{4.9}$$

where

$$w^{\text{supp}}(s,a) = \exp\left(\frac{A^{\pi}(s,a)}{\alpha} - \log \pi_\beta(a \mid s) \right).$$

Because $e^{\left(-\log \pi_\beta(a|s)\right)} = \pi_\beta(a \mid s)^{-1}$, the objective Eq. (4.9) is equivalent to requiring $\pi(a \mid s) = 0$ whenever $\pi_\beta(a \mid s) = 0$ [1], and otherwise

$$\pi(a \mid s) \propto \exp\left(\frac{A^{\pi}(s,a)}{\alpha} \right).$$

Thus, the learned policy is constrained only by the support of π_β rather than by its full action-probability profile [64].

[1] InAC defines $0 \cdot \infty = 0$.

4.4 Proposed Algorithm – ACPO

4.4.1 Continuous Constraint Interpolation Framework

The CCI framework provides a unified optimization view of three policy-constraint families: wBC, KL-based density regularization, and support constraints. It treats them as instances along a common constraint spectrum, and enables smooth switching and principled combinations via a single continuous interpolation parameter.

Consider the following constrained maximum-entropy policy optimization problem,

$$\begin{aligned} \max_{\pi} \ & \mathbb{E}_{s\sim\mathcal{D},\, a\sim\pi(\cdot|s)}\big[Q(s,a) - \alpha \log \pi(a \mid s)\big], \\ \text{s.t.} \ & \mathbb{E}_{s\sim\mathcal{D},\, a\sim\pi(\cdot|s)}\big[\log \pi_\beta(a \mid s)\big] \geq \epsilon, \\ & \int \pi(a \mid s)\, da = 1, \qquad \forall s. \end{aligned} \tag{4.10}$$

Here the only per-state constraint is the standard normalization condition; the behavior-related constraint itself is imposed only in expectation over dataset states. We do not impose a separate hard constraint for each state, since that would introduce an impractical – potentially infinite – number of constraints [39]. Instead, we enforce the constraint on average over dataset states, yielding a single scalar constraint.

Solving the optimization problem in Eq. (4.10) via the Lagrangian,

$$\begin{aligned} \mathcal{L}(\pi, \lambda, \nu) = \ & \mathbb{E}_{s\sim\mathcal{D},\, a\sim\pi(\cdot|s)}\left[Q(s,a) - \alpha \log \pi(a \mid s)\right] \\ & + \lambda\Big(\mathbb{E}_{s\sim\mathcal{D},\, a\sim\pi(\cdot|s)}\left[\log \pi_\beta(a \mid s)\right] - \epsilon\Big) \\ & + \mathbb{E}_{s\sim\mathcal{D}}\left[\nu(s)\Big(\int \pi(a \mid s)\, da - 1\Big)\right], \end{aligned} \tag{4.11}$$

we obtain the following closed-form optimal policy:

$$\pi^*_\lambda(a \mid s) = \exp\Big(\frac{Q(s,a) - Z_\lambda(s)}{\alpha}\Big)\pi_\beta(a \mid s)^{\lambda/\alpha}, \tag{4.12}$$

where $\lambda \geq 0$ denotes the dual variable, and

$$Z_\lambda(s) = \alpha \log \int \exp(Q(s,a)/\alpha)\, \pi_\beta(a|s)^{\lambda/\alpha}\, da$$

is the log-normalizer.

We then project the optimal nonparametric policy π^*_λ into a parametric policy class π_θ by minimizing the reverse KL divergence:

$$\pi_\theta = \arg\min_{\pi_\theta} \mathbb{E}_{s\sim\mathcal{D}}\Big[D_{\mathrm{KL}}\big[\pi^*_\lambda(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid s)\big]\Big]. \tag{4.13}$$

λ	Constraint	$w_\lambda(s,a)$
$\lambda = 0$	Support Constraint	$\exp\left(\frac{A(s,a)}{\alpha} - \log \pi_\beta(a \mid s)\right)$.
$0 < \lambda < \alpha$	Support-to-Density Constraint	$\exp\left(\frac{A(s,a)}{\alpha} - \frac{\alpha-\lambda}{\alpha} \log \pi_\beta(a \mid s)\right)$.
$\lambda = \alpha$	KL-based Density Constraint	$\exp\left(\frac{A(s,a)}{\alpha}\right)$.
$\lambda > \alpha$	Density-to-wBC Constraint	$\exp\left(\frac{A(s,a)}{\alpha} + \frac{\lambda-\alpha}{\alpha} \log \pi_\beta(a \mid s)\right)$.
$\lambda \gg \alpha + A(s,a)(\log \pi_\beta(a \mid s))^{-1}$	Practical wBC Constraint	$\approx \exp\left(\frac{\lambda-\alpha}{\alpha} \log \pi_\beta(a \mid s)\right)$.

Table 4.1: Constraint spectrum induced by the interpolation parameter λ in the CCI framework.

This yields the following policy-improvement objective under the CCI framework:

$$\max_\theta \ \mathcal{J}^{\mathrm{CCI}}(\theta;\lambda) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[w_\lambda(s,a) \log \pi_\theta(a \mid s)\right], \tag{4.14}$$

where

$$w_\lambda(s,a) = \exp\left(\frac{A(s,a)}{\alpha} + \frac{\lambda-\alpha}{\alpha} \log \pi_\beta(a \mid s)\right).$$

Eq. (4.14) shows that varying λ continuously interpolates among different policy-constraint regimes. As summarized in Table 4.1, the CCI objective recovers support constraints, KL-based density regularization, and wBC into a continuous constraint spectrum. Therefore, CCI does not merely unify the three families conceptually, but also enables smooth transitions and flexible combinations among them.

We note that the practical wBC regime in Tab. 4.1 corresponds to choosing

$$\lambda \gg \alpha + A(s,a)\big(\log \pi_\beta(a \mid s)\big)^{-1}.$$

Under the maximum-entropy setting, assume rewards are bounded by $|r(s,a)| \le R_{\max}$ and we stabilize training by clipping log-probabilities for any stochastic policy as $\log \pi(a|s) \in [\ell_{\min}, \ell_{\max}]$. Let

$$C_{\max} \triangleq \max\{|\ell_{\min}|, |\ell_{\max}|\} \Rightarrow |-\alpha \log \pi(a \mid s)| \le \alpha C_{\max}.$$

Then one can obtain a uniform advantage bound $|A^\pi(s,a)| \le \frac{2\left(R_{\max}+\alpha C_{\max}\right)}{1-\gamma}$ and consequently, with $C^\beta_{\min} \triangleq \min |\log \pi_\beta(a \mid s)|$ and a small constant $\delta > 0$ to avoid division by zero,

$$\left|\frac{A(s,a)}{\log \pi_\beta(a \mid s)}\right| \le \frac{2\left(R_{\max}+\alpha C_{\max}\right)}{(1-\gamma)(C^\beta_{\min}+\delta)}. \tag{4.15}$$

Therefore, a sufficient condition for the practical wBC constraint is

$$\lambda \gg \alpha + \frac{2\left(R_{\max}+\alpha C_{\max}\right)}{(1-\gamma)(C^\beta_{\min}+\delta)}. \tag{4.16}$$

The derivations for this subsection are provided in Appendix 7.3.1.

4.4.2 Automatic Constraint Policy Optimization

Building on the CCI framework, we propose Automatic Constraint Policy Optimization (ACPO), a practical primal-dual method that automatically adapts both the effective constraint type and its strength through the interpolation parameter λ.

For a fixed Q-function, the objective in Eq. (4.10) is maximizing a concave objective in π, and the inequality constraint and normalization constraints are affine. Therefore, under standard regularity assumptions (ensuring $\mathcal{L}$ is well-defined) and Slater's condition, strong duality holds, and the primal problem is equivalent to the dual form,

$$\begin{aligned} &\min_{\lambda} \max_{\pi_\theta \in \Delta} \mathcal{L}(\pi_\theta, \lambda), \\ &\text{s.t. } \lambda \geq 0, \end{aligned} \tag{4.17}$$

where Δ denotes the set of valid policies.

ACPO alternates between two coupled updates:

1. updating π_θ to maximize the CCI objective in Eq. (4.14) for a fixed λ;
2. updating λ by gradient descent so as to enforce the constraint.

The dual update takes the form

$$\lambda \leftarrow \Big[\lambda - \eta_\lambda \big(\mathbb{E}_{s\sim\mathcal{D}, a\sim\pi_\theta}[\log \pi_\beta(a \mid s)] - \epsilon\big)\Big]_+, \tag{4.18}$$

where η_λ is the dual learning rate and $[\cdot]_+$ denotes projection onto the nonnegative reals.

This update increases λ when the constraint is violated, and decreases it when the constraint is overly conservative. By updating λ online via Eq. (4.18), ACPO automatically adapts the effective constraint regime to the dataset.

For policy evaluation, we adopt standard maximum-entropy critic updates together with Clipped Double Q-Learning [142]:

$$\begin{aligned} \mathcal{L}_V(\psi) &= \mathbb{E}_{s\sim\mathcal{D}, a\sim\pi_\theta(\cdot|s)}\left[\frac{1}{2}\big(V_\psi(s) - Y_V(s,a)\big)^2\right], \\ Y_V(s,a) &\triangleq \min_{i\in\{1,2\}} Q_{\phi_i}(s,a) - \alpha \log \pi_\theta(a \mid s), \end{aligned} \tag{4.19}$$

$$\mathcal{L}_Q(\phi_i) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\frac{1}{2}\left(r + \gamma V_\psi(s') - Q_{\phi_i}(s,a)\right)^2\right]. \tag{4.20}$$

Following common practice in offline RL [14, 64], we fit the behavior policy π_β on the dataset by maximum likelihood:

$$\mathcal{L}(\pi_\beta) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[-\log \pi_\beta(a \mid s)\right]. \tag{4.21}$$

Integrating policy improvement, policy evaluation, and dual optimization, the complete ACPO algorithm is summarized in Algorithm 2.

Algorithm 2 Automatic Constraint Policy Optimization (ACPO)

Input: Dataset $\mathcal{D} = \{(s, a, r, s')\}$.
Parameters: Q-networks $Q_{\phi_{1,2}}$, target Q-networks $Q_{\phi'_{1,2}}$, value network V_ψ, policy network π_θ, behavior policy network π_β.

// Pre-train behavior policy for N_β steps.
for $t = 1$ to N_β **do**
  Optimize π_β by Eq. (4.21).
end for
// Policy evaluation, improvement, and dual update.
for each gradient step **do**
  Sample $(s, a, r, s') \sim \mathcal{D}$.
  Optimize V_ψ by Eq. (4.19).
  Optimize Q_{ϕ_i} by Eq. (4.20).
  Update λ by Eq. (4.18).
  Optimize π_θ by Eq. (4.14).
  Soft-update target networks $\phi' \leftarrow (1-\tau)\phi' + \tau\phi$.
end for
return Policy π_θ.

4.4.3 Theoretical Analysis

The CCI framework admits a closed-form optimizer π^*_λ parameterized by the dual variable λ. We first formalize how λ controls conservatism by studying the constraint $\mathbb{E}_{a\sim\pi^*_\lambda(\cdot|s)}[\log \pi_\beta(a \mid s)]$. We then derive a maximum-entropy performance difference lemma and establish performance lower bounds for both the optimal policy π^*_λ and its parametric projection π_θ.

Proposition 4.1. *Fix any state s and temperature $\alpha > 0$. Let $\pi^*_\lambda(\cdot \mid s)$ denote the closed-form optimizer in Eq. (4.12) for a given $\lambda \geq 0$. Define*

$$g_s(\lambda) := \mathbb{E}_{a\sim\pi^*_\lambda(\cdot|s)}\big[\log \pi_\beta(a \mid s)\big]. \tag{4.22}$$

Then $g_s(\lambda)$ is monotone non-decreasing in λ, and

$$\frac{d}{d\lambda} g_s(\lambda) = \frac{1}{\alpha}\mathrm{Var}_{a\sim\pi^*_\lambda(\cdot|s)}\big(\log \pi_\beta(a \mid s)\big) \geq 0. \tag{4.23}$$

Consequently, the dataset-averaged quantity

$$g(\lambda) := \mathbb{E}_{s\sim\mathcal{D}}[g_s(\lambda)]$$

also satisfies $g'(\lambda) \geq 0$.

Proposition 4.1 formalizes the following intuition. As λ increases, the closed-form optimal policy π^*_λ assigns more probability mass to actions with larger $\pi_\beta(a \mid s)$, thereby producing a more conservative update. Conversely, a smaller λ allows π^*_λ to retain greater flexibility within the support of π_β, so that the update is guided more strongly by the advantage function. This monotonicity is fully consistent with the role of λ in the CCI spectrum summarized in Table 4.1.

The detailed proof is provided in Appendix 7.3.2.

Lemma 4.2 (Maximum Entropy Performance Difference)**.** *Let π and π_β be two stationary policies. Consider the maximum-entropy performance* [2]

$$\mathcal{J}(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t\big(r(s_t,a_t) - \alpha\log\pi(a_t \mid s_t)\big)\right],$$

then the maximum entropy performance difference satisfies

$$\mathcal{J}(\pi) - \mathcal{J}(\pi_\beta) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\pi}\Big[\mathbb{E}_{a\sim\pi(\cdot|s)}\big[\tilde{A}^{\pi_\beta}(s,a)\big] - \alpha\mathrm{KL}\big(\pi(\cdot \mid s)\|\pi_\beta(\cdot \mid s)\big)\Big], \tag{4.24}$$

where $\tilde{A}^{\pi_\beta}(s,a)$ is the advantage function under the shaped reward $\tilde{r}(s,a) = r(s,a) - \alpha\log\pi_\beta(a \mid s)$.

Theorem 4.3 (Performance Lower Bound for the Optimal Policy)**.** *Define $\epsilon_\beta(\pi) := \max_s\left|\mathbb{E}_{a\sim\pi(\cdot|s)}\big[\tilde{A}^{\pi_\beta}(s,a)\big]\right|$, $\kappa_\beta(\pi) := \max_s \mathrm{KL}\big(\pi(\cdot \mid s)\|\pi_\beta(\cdot \mid s)\big)$ and the average total variation distance under ν by $\Delta_{\mathrm{TV}}(\pi,\pi_\beta;\nu) := \mathbb{E}_{s\sim\nu}\big[D_{\mathrm{TV}}(\pi(\cdot \mid s),\pi_\beta(\cdot \mid s))\big]$. Assume $|\log\pi_\beta(a \mid s)| \le B$ for all (s,a). For a given $\lambda \ge 0$ and $\alpha > 0$, π^*_λ satisfies the following maximum-entropy performance lower bound:*

$$\begin{aligned}\mathcal{J}(\pi^*_\lambda) \ge \mathcal{J}(\pi_\beta) &- \frac{2B|\lambda-\alpha|}{1-\gamma}\Delta_{\mathrm{TV}}(\pi^*_\lambda,\pi_\beta;d_{\pi_\beta}) \\ &- \frac{4\alpha B + 2\gamma\Big(\epsilon_\beta(\pi^*_\lambda) + \alpha\kappa_\beta(\pi^*_\lambda)\Big)}{(1-\gamma)^2}\Delta_{\mathrm{TV}}(\pi^*_\lambda,\pi_\beta;d_{\pi_\beta}).\end{aligned} \tag{4.25}$$

Theorem 4.4 (Performance Lower Bound for a Parametric Policy)**.** *Define the suboptimality gap of a parametric policy π_θ by*

$$\delta^{\mathrm{sub}}_\theta(\lambda) := F_\lambda(\pi^*_\lambda) - F_\lambda(\pi_\theta) \;\ge\; 0, \tag{4.26}$$

where

$$F_\lambda(\pi) = \mathbb{E}_{s\sim d_{\pi_\beta}}\Big[\mathbb{E}_{a\sim\pi(\cdot|s)}\tilde{A}^{\pi_\beta}(s,a) - \alpha\mathrm{KL}\big(\pi(\cdot|s)\|\pi_\beta(\cdot|s)\big) + (\lambda-\alpha)\mathbb{E}_{a\sim\pi(\cdot|s)}\log\pi_\beta(a|s)\Big].$$

[2] We use $\eta(\pi)$ to denote the standard discounted return and $\mathcal{J}(\pi)$ for the maximum-entropy objective.

Then π_θ satisfies

$$\begin{aligned}\mathcal{J}(\pi_\theta) \geq\ & \mathcal{J}(\pi_\beta) - \frac{2B|\lambda-\alpha|}{1-\gamma}\Delta_{\mathrm{TV}}\big(\pi_\theta, \pi_\beta; d_{\pi_\beta}\big) - \frac{\delta_\theta^{\mathrm{sub}}(\lambda)}{1-\gamma} \\ & - \frac{4\alpha B + 2\gamma\Big(\epsilon_\beta(\pi_\theta) + \alpha\kappa_\beta(\pi_\theta)\Big)}{(1-\gamma)^2}\Delta_{\mathrm{TV}}(\pi_\theta, \pi_\beta; d_{\pi_\beta}).\end{aligned} \tag{4.27}$$

The proofs of Theorems 4.3 and 4.4 suggest a decomposition of the lower bound into two components. The first-order term $\mathcal{O}\Big(\frac{1}{1-\gamma}\Big)$ evaluates policy improvement under the fixed state distribution d_{π_β}. The second-order term $\mathcal{O}\Big(\frac{1}{(1-\gamma)^2}\Big)$ captures distribution shift, arising from the mismatch between d_π and d_{π_β}.

Compared to the KL-density constraint in standard RL, whose performance lower bound is typically $\mathcal{J}^{\mathrm{KL}}(\pi^*) \geq \mathcal{J}(\pi_\beta) - \mathcal{O}(\frac{1}{(1-\gamma)^2})$ [65], our bound additionally exposes an explicit first-order term and makes the role of λ transparent: the factor $|\lambda - \alpha|$ quantifies how the λ modulates the worst-case performance.

The detailed proof is provided in Appendix 7.3.4 and 7.3.5.

4.5 Experiments

4.5.1 Comparisons on D4RL Benchmarks

Tasks. We conduct experiments in Gym-MuJoCo locomotion domains and more challenging AntMaze and Kitchen tasks [123]. The main challenge of the latter is to learn policies for long-horizon planning from datasets that do not contain optimal trajectories, which require "stitching" ability.

Baselines. We select competitive offline RL baselines, including BCQ [17], IQL [23], CQL [24], TD3+BC [38], SPOT [39], ACTIVE [134], EQL [143], OTR [144].

Results. The whole results are shown in Tab. 4.2. Overall, ACPO achieves the best performance across all three D4RL domains, obtaining the highest total scores on Gym-MuJoCo, AntMaze, and Kitchen. These results indicate that ACPO provides a robust policy-constraint method across diverse offline RL domains rather than relying on a single fixed constraint regime.

4.5.2 Comparisons on NeoRL2 Benchmarks

Tasks. NeoRL2 [124] is a recently introduced benchmark that aims to capture practical challenges in offline RL. It covers domains inspired by industrial control, nuclear fusion, and healthcare, and is characterized by significant time-delay effects and external disturbances. In addition, real-world limitations on data collection restrict both dataset size and

Table 4.2: Averaged normalized scores of ACPO on Gym-MuJoCo-v2, AntMaze-v2, and Kitchen-v0 datasets over five seeds. Scores are reported as mean ± standard deviation. **Bold** indicates the best performance in each task.

Dataset	BCQ	IQL	CQL	TD3+BC	SPOT	ACTIVE	EQL	OTR	ACPO
halfcheetah-med	47.0	47.4	44.0	48.3	**58.4**	50.6	47.2	43.3	52.5 ± 0.76
hopper-med	56.7	66.3	58.5	59.3	86.0	80.1	74.6	78.7	**97.9** ± 2.14
walker2d-med	72.6	78.3	72.5	83.7	86.4	84.8	83.2	79.4	**88.0** ± 1.1
halfcheetah-med-rep	40.4	44.2	45.5	44.6	**52.2**	51.1	44.5	41.3	47.5 ± 0.54
hopper-med-rep	53.3	94.7	95.0	60.9	100.2	**101.1**	98.1	84.8	100.6 ± 0.54
walker2d-med-rep	52.1	73.9	77.2	81.8	91.6	85.4	76.6	66.0	**93.6** ± 1.86
halfcheetah-med-exp	89.1	86.7	90.7	90.7	86.9	93.2	90.6	89.6	**97.2** ± 1.24
hopper-med-exp	81.8	91.5	105.4	98.0	99.3	89.2	105.5	93.2	**112.3** ± 0.72
walker2d-med-exp	109.0	109.6	109.6	110.1	112.0	112.0	110.2	109.3	**113.1** ± 0.22
Locomotion Total	602	692.6	698.4	677.4	773	747.5	730.5	685.6	**802.7** ± 9.12
antmaze-u	78.9	85.5	84.8	78.6	**93.5**	91.6	93.2	83.4	90.0 ± 3.0
antmaze-u-d	55.0	66.7	43.4	71.4	40.7	78.4	65.4	68.9	**86.0** ± 3.4
antmaze-m-p	0	72.2	65.2	10.6	74.7	74.6	**77.5**	70.5	76.0 ± 4.24
antmaze-m-d	0	71.0	54.0	3.0	**79.1**	75.2	70.0	70.4	76.0 ± 4.24
antmaze-l-p	6.7	39.6	38.4	0.2	35.3	51.2	45.6	45.3	**56.0** ± 4.94
antmaze-l-d	2.2	47.5	31.6	0.0	36.3	**48.2**	42.5	45.5	48.0 ± 4.98
Antmaze Total	142.8	382.5	317.4	163.8	359.6	419.2	394.2	384.0	**432.0** ± 24.8
kitchen-m	10.6	52.8	51.0	0.8	0.0	52.1	55.6	-	**72.5** ± 5.5
kitchen-p	18.9	46.1	49.8	0.0	0.0	69.6	**74.5**	-	64.0 ± 15.52
Kitchen Total	29.5	98.9	100.8	0.8	0.0	121.7	130.1	-	**136.5** ± 21.02

coverage, thereby imposing stricter requirements on the robustness and adaptability of offline RL methods.

Baselines. We compare ACPO with representative baselines that are commonly reported on NeoRL2, including BC [56], CQL [24], EDAC [55], MCQ [65], TD3+BC [38], RAMBO [145], and MOBILE [100] [3].

Results. As reported in Tab. 4.3, ACPO shows clear gains on NeoRL2. It achieves the best results on 4 of the 7 datasets and obtains the highest total score across tasks, significantly surpassing the strongest baseline, TD3+BC.

4.5.3 Special-Case Comparisons and λ Dynamics

Within the CCI framework, several representative forms of policy constraints can be recovered by fixing the interpolation parameter λ. To examine the role of λ alone, we conduct special-case comparisons by turning off the dual update in ACPO and training each variant with a fixed λ over the entire process. At the level of policy improvement, these special cases correspond to wBC, AWAC [21], and InAC [64]. By contrast, the policy eval-

[3] We report the strongest baselines used in NeoRL2 and omit MOPO [98] and COMBO [99] for conciseness.

Table 4.3: Averaged normalized scores of ACPO on NeoRL2 benchmark over three seeds. Scores are reported as mean ± standard deviation. **Bold** indicates the best performance in each task.

Dataset	Data	BC	CQL	EDAC	MCQ	TD3+BC	RAMBO	MOBILE	ACPO
Pipeline	69.3	68.6	81.1	72.9	49.7	82.0	24.1	65.5	**95.2** ± 4.62
Simglucose	73.9	75.1	11.0	8.1	29.6	74.2	10.8	9.3	**94.9** ± 12.73
RocketRecovery	75.3	72.8	74.3	65.7	76.5	79.7	-44.2	43.7	**84.6** ± 10.23
RandomFrictionHopper	28.7	28.0	33.0	34.7	31.7	29.5	29.6	**35.1**	33.7 ± 0.37
DMSD	56.6	65.1	70.2	**78.7**	77.8	60.0	76.2	64.4	74.9 ± 5.67
Fusion	48.8	55.2	55.9	58.0	49.7	54.6	59.6	5.0	**61.8** ± 2.07
SafetyHalfCheetah	**73.6**	70.2	71.2	53.1	54.7	68.6	-422.4	8.7	67.1 ± 6.90
NeoRL2 Total	426.2	435.0	396.7	371.2	369.7	448.6	-266.3	231.7	**512.2** ± 42.59

uation procedures may vary among these methods. To ensure a controlled comparison, we fix the policy evaluation component and adopt the same maximum-entropy critic updates in Eq. (4.19) and Eq. (4.20) as those used in ACPO for all variants.

Settings. For the Gym-MuJoCo tasks, we choose $\alpha = 0.1$, while for the NeoRL2 tasks, we use $\alpha = 0.5$. Based on these settings, we construct three special cases: (i) practical wBC with $\lambda = 100$, which approximates the wBC regime, (ii) AWAC with $\lambda = \alpha$, and (iii) InAC with $\lambda = 0$.

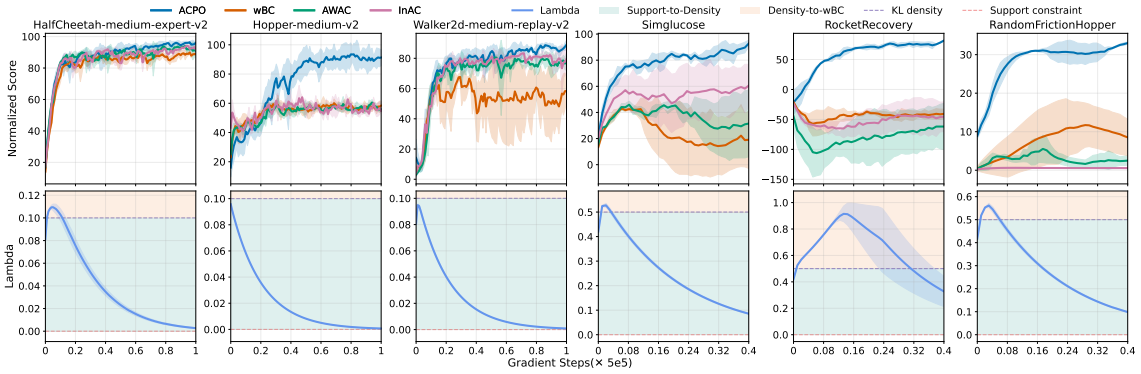


Figure 4.1: Special-case comparisons and the evolution of the interpolation parameter λ during ACPO training.

Fig. 4.1 presents the training curves of ACPO and the three corresponding baselines on 6 datasets, together with the adaptive trajectory of λ during training. From Fig. 4.1, we make the following observations:

Overall Superiority. ACPO consistently delivers the best performance across all 6 tasks, with especially notable improvements on the more challenging NeoRL2 datasets,

where data coverage is limited. This result demonstrates the advantage of automatically adjusting the effective constraint regime according to the characteristics of the dataset.

λ **Dynamics.** We observe that λ is often non-monotonic: it usually increases during the early stage of training and then decreases gradually afterwards. At the beginning of training, critic estimates are not yet well calibrated, and actor updates are therefore more likely to exploit OOD actions. In response, the dual update raises λ, which strengthens the policy constraint and steers updates toward higher-density behavior actions, thereby improving training stability. As training continues and the value estimates become more reliable, the risk from OOD actions is reduced. Consequently, λ decreases, loosening the constraint and allowing more effective policy improvement.

4.5.4 Behavior Density Estimator Ablation Study

Because ACPO explicitly involves $\log \pi_\beta(a|s)$, the selection of the behavior density estimator can have a direct influence on both the constraint signal and the weights used in policy improvement.

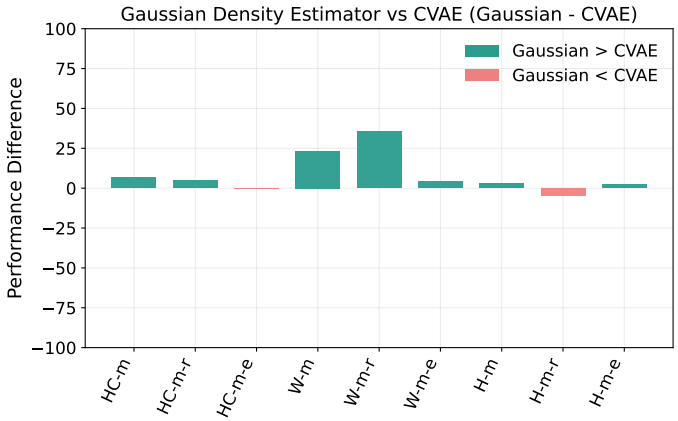


Figure 4.2: Normalized score differences on Gym-MuJoCo datasets.

In addition to modeling π_β with a Gaussian density estimator, we further investigate a conditional variational autoencoder (CVAE) [17, 128] as an alternative approximation of the behavior policy. Furthermore, inspired by prior work [39], we also incorporate the IS-corrected variant to alleviate bias in expectation estimates involving π_β. We compare Gaussian and CVAE behavior models on 9 Gym-MuJoCo datasets, and the corresponding results are presented in Fig. 4.2.

As reported in Fig. 4.2, the Gaussian and CVAE estimators achieve similar performance on most datasets. However, a clear difference appears on `walker2d-medium-v2` and `walker2d-medium-replay-v2`, where the performance gap is substantial. To better understand this phenomenon, we visualize the distributions of $\log \pi_\beta(a \mid s)$ produced by

the Gaussian and CVAE estimators for (i) in-dataset actions a_{in} and (ii) perturbed OOD actions a_{ood}, where the latter are constructed as $a_{\text{ood}} = a_{\text{in}} + \xi z$, with ξ controlling the perturbation magnitude and $z \sim \mathcal{N}(0, I)$. The resulting distributions are shown in Fig. 4.3.

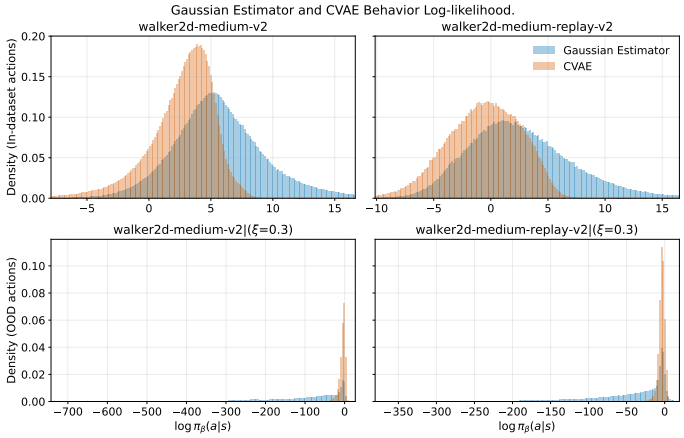


Figure 4.3: Distributions of $\log \pi_\beta(a|s)$ under Gaussian and CVAE behavior estimators on in-dataset actions and OOD actions.

Fig. 4.3 provides an explanation for the inferior performance of the CVAE estimator on `walker2d-medium-v2` and `walker2d-medium-replay-v2`. For in-dataset actions, the CVAE produces a noticeably sharper distribution of $\log \pi_\beta(a \mid s)$ than the Gaussian estimator, indicating a more concentrated density model. More importantly, for perturbed OOD actions, the CVAE does not always assign sufficiently low likelihood relative to the Gaussian estimator: the distribution of $\log \pi_\beta(a_{\text{ood}} \mid s)$ does not move leftward as much as expected under clearly OOD perturbations. Such overestimation of behavior likelihood on OOD actions can weaken the constraint signal in ACPO, make the method more vulnerable to extrapolation error, and eventually lead to degraded performance.

4.5.5 Sensitivity Analysis of ϵ

The hyperparameter ϵ in Eq. 4.10 controls the effective conservatism of the policy update and therefore directly affects how strongly ACPO avoids potentially unreliable actions. Since one of the main goals of our method is to reduce task-specific manual tuning, it is important to verify that ACPO does not rely on a narrowly tuned choice of ϵ to perform well.

To evaluate this, we conduct a sensitivity study over a broad range of ϵ values on Gym-MuJoCo-v2 tasks. The results are shown in Tab. 4.4.

Overall, the results suggest that ACPO is reasonably stable across a fairly wide range of ϵ values.

Complete training details for all experiments are provided in Appendix 7.4.

Table 4.4: Sensitivity Analysis of ϵ.

Dataset	$\epsilon = -0.2$	$\epsilon = -0.7$	$\epsilon = -5.0$
HalfCheetah-m	52.5 ± 0.76	48.2 ± 1.02	48.4 ± 0.72
HalfCheetah-m-e	97.2 ± 1.24	95.1 ± 0.40	95.7 ± 0.49
HalfCheetah-m-r	47.5 ± 0.54	47.2 ± 0.23	44.6 ± 0.67
Hopper-m	89.9 ± 10.88	95.9 ± 4.29	97.9 ± 2.14
Hopper-m-e	109.1 ± 8.36	110.3 ± 0.51	112.3 ± 0.72
Hopper-m-r	100.2 ± 1.02	99.6 ± 1.10	100.6 ± 0.54
Walker2d-m	84.3 ± 0.99	88.0 ± 1.1	85.1 ± 1.12
Walker2d-m-e	109.5 ± 0.22	113.1 ± 0.22	109.6 ± 0.34
Walker2d-m-r	82.3 ± 13.01	93.6 ± 1.86	85.8 ± 4.02
Average Score	85.8	87.9	86.7

4.6 Conclusion

In this chapter, we presented CCI, a unified optimization framework that bridges wBC, KL-based density regularization, and support constraints under a common constraint spectrum. By explicitly introducing the interpolation parameter, CCI enables smooth transitions and principled combinations across different constraint regimes. Building on this framework, we further developed ACPO, a practical primal-dual algorithm that automatically adapts the effective constraint type and strength through a Lagrangian dual update.

We additionally established a maximum-entropy performance difference lemma and derived lower performance bounds for both the closed-form optimal policy and its parametric projection. Experimental results on D4RL and NeoRL2 show that ACPO delivers robust improvements across diverse domains and, in many cases, outperforms strong baselines overall. Extensive ablation studies further validate that ACPO achieves strong tuning-free generality and robust stability.

Chapter 5

Expand Latent Action Projection Space for Policy Optimization

This chapter addresses the fourth research question ($RQ4$ in Section 1.1) of the thesis: how to identify and mitigate the failure modes that arise when integrating CVAE into offline RL so as to maximize policy improvement. CVAE-based offline RL methods are attractive because they can model the behavior distribution and provide an implicit or explicit mechanism for constraining policy updates within the dataset support. However, when combined with offline RL, CVAE may suffer from posterior collapse, which in turn induces a more detrimental phenomenon in policy learning: the latent action projection space can become severely compressed, thereby neutralizing policy optimization in the latent space and undermining effective improvement. To address this issue, this chapter analyzes the causes and consequences of latent action projection space collapse in representative CVAE-based offline RL algorithms, and proposes Expand Latent Action Projection SpacE (ELAPSE), a simple yet effective method that enlarges the projection space while preserving the behavior-modeling role of CVAE.

The remainder of this chapter is organized as follows. Section 5.1 introduces the motivation and background of CVAE-based offline RL. Section 5.2 reviews the related work, and Section 5.3 presents the necessary preliminaries. Section 5.4 formulates the latent action projection space collapse phenomenon and analyzes its effects on representative CVAE-based offline RL algorithms. Section 5.5 then presents the proposed ELAPSE method. Finally, Section 5.6 reports the experimental results and ablation studies on standard offline RL benchmarks.

5.1 Introduction

CVAE-based offline RL methods have attracted considerable attention because they can model the behavior distribution and provide an implicit or explicit mechanism for con-

straining policy optimization within the dataset support [17, 39, 95–97]. In this way, CVAE [40, 146] serves as a powerful generative prior for mitigating distribution shift and reducing the risk of selecting OOD actions. However, integrating CVAE into offline RL also introduces a critical failure mode that remains insufficiently understood.

A key issue is that posterior collapse in CVAE can induce a more harmful phenomenon in policy learning: the latent action projection space may become severely compressed, thereby weakening or even neutralizing policy optimization in the latent space. Intuitively, we can view offline RL under CVAE constraint as a two-stage optimization approach: the first stage involves optimizing the policy in the latent action space, and the second stage involves projecting the policy within the support set of the behavior policy using the CVAE model. As mentioned above, there is a critical trade-off between the two stages. If the CVAE model is over-dominant, the policy optimization stage will be neutralized by the projection stage. Specifically, the entire latent action space conditioned by the state s will be projected to a compact original action space close to the action a from the dataset tuple (s, a). We denote this phenomenon as *latent action projection space collapse*, where the latent action projection space is a non-empty subset of the original action space.

This problem is particularly harmful for representative CVAE-based offline RL algorithms such as PLAS [95], LAPO [96], SPOT [39], and Anti-exploration [97]. Existing methods often introduce perturbations after latent policy optimization in an attempt to recover exploration ability [17, 95, 96]. However, uncontrolled perturbations can themselves induce OOD actions, thereby reintroducing extrapolation error and undermining reliable policy improvement.

In this chapter, we analyze the causes and consequences of latent action projection space collapse in representative CVAE-based offline RL algorithms, and further conduct visualization experiments to illustrate the collapse phenomenon. Based on this analysis, we propose Expand Latent Action Projection SpacE (ELAPSE), a simple yet effective method that enlarges the latent action projection space while preserving the behavior-modeling role of CVAE. Empirically, ELAPSE yields substantial improvements over the original CVAE-based methods and achieves strong performance on D4RL benchmark [123].

The main contributions of this chapter are summarized as follows:

- We explain the issue of latent action projection space collapse in offline RL algorithms under CVAE constraint through theoretical analysis and visualization experiments.
- We propose the ELAPSE algorithm, which expands the latent action projection space for policy optimization and balances the trade-off between policy optimization and CVAE constraint.
- We compare ELAPSE with other competitive offline RL algorithms, and ELAPSE achieves state-of-the-art performance on most D4RL datasets.

5.2 Related Work

To position the contribution of this chapter more clearly, we briefly review two lines of work that are most relevant to ELAPSE: policy-constraint offline RL and generative-model-based offline RL.

Policy Constraint Policy-constrained offline RL methods mitigate distribution shift by explicitly encouraging the learned policy to remain close to the behavior policy, using mechanisms such as KL divergence or kernel MMD [20, 21, 38, 59, 147, 148]. BRAC [59] evaluates several policy-constraint metrics and shows that many technical complexities are not necessary for strong empirical performance, while ReBRAC [148] further emphasizes the importance of seemingly minor design choices introduced alongside core algorithmic ideas. AWR [20] and AWAC [21] optimize the policy via weighted maximum likelihood, thereby connecting supervised learning and offline RL in a unified objective. BPPO [147] shows that online on-policy algorithms can also solve offline RL tasks when initialized with behavior cloning.

Generative Model Generative-model-based offline RL methods have demonstrated strong empirical performance. Representative generative paradigms include diffusion models [149, 150], GANs [129, 151], and CVAE-based methods [17, 39, 63, 95–97]. Since the present chapter focuses on the failure modes arising from the integration of CVAE into offline RL, we restrict our attention to CVAE-based approaches.

Among these methods, BCQ [17] learns the behavior policy using a CVAE model and introduces random perturbations on generated actions. EMaQ [63] simplifies BCQ by discarding the perturbation layer. Other representative CVAE-based offline RL algorithms include PLAS [95], LAPO [96], SPOT [39], and Anti-exploration [97], which integrate CVAE into offline RL through latent-space policy optimization, support estimation, or explicit regularization.

However, existing work has mainly treated CVAE as a useful behavior prior or support model, while paying limited attention to the failure mode studied in this chapter, namely latent action projection space collapse. This gap motivates the analysis and method proposed in the remainder of the chapter.

5.3 Preliminaries

5.3.1 Conditional Variational Autoencoders

This section briefly recalls the CVAE formulation used in the remainder of the chapter, since the analysis of posterior collapse and the proposed ELAPSE method both build on its ELBO objective. In offline RL, a CVAE is typically used to model the conditional action

distribution given state by maximizing the conditional log-likelihood $\log p(a \mid s)$ on the dataset through its tractable evidence lower bound (ELBO):

$$\log p(a \mid s) \geq \underbrace{\mathbb{E}_{q_\xi(z|a,s)}\left[\log p_\rho(a \mid z,s)\right] - D_{\mathrm{KL}}\left[q_\xi(z \mid a,s) \,\|\, p(z \mid s)\right]}_{\mathcal{L}_{\mathrm{ELBO}}}. \tag{5.1}$$

where $q_\xi(z \mid a,s)$ is the encoder parameterized by ξ, $p_\rho(a \mid z,s)$ is the decoder parameterized by ρ, and $p(z \mid s)$ is a fixed prior. Here D_{KL} denotes the Kullback–Leibler (KL) divergence [57].

The prior $p(z \mid s)$ is typically chosen as a multivariate Gaussian distribution $\mathcal{N}(0, I)$, in which case the KL term can be written explicitly as

$$D_{\mathrm{KL}}\left[q_\xi(z \mid a,s) \,\|\, p(z \mid s)\right] = \frac{1}{2}\sum_{i=1}^{n}\left(\mu_i^2 + \sigma_i^2 - \log \sigma_i^2 - 1\right), \tag{5.2}$$

where μ_i and σ_i are the mean and standard deviation of the approximate posterior for the i-th latent dimension.

This ELBO formulation will serve as the basis for the later analysis of posterior collapse and latent action projection space collapse in CVAE-based offline RL algorithms.

5.4 Problem Statement

In this section, we introduce the latent action projection space collapse phenomenon and examine its negative effects on PLAS, LAPO, SPOT, and Anti-exploration using both visualization experiments and theoretical analysis.

5.4.1 Collapse in PLAS

PLAS [95] performs policy optimization in the latent action space, thereby implicitly constraining decoded actions to lie within the support of the dataset. It first trains a CVAE model to approximate the behavior policy $\pi_\beta(a \mid s)$. Specifically, the CVAE is learned by minimizing the negative ELBO, which provides a tractable surrogate for $-\log \pi_\beta(a \mid s)$:

$$\mathcal{L}_{\mathrm{CVAE}} = \underbrace{-\mathbb{E}_{q_\xi(z|a,s)}\left[\log p_\rho(a \mid z,s)\right]}_{\text{reconstruction error } (\mathcal{L}_R)} + \underbrace{D_{\mathrm{KL}}\left[q_\xi(z \mid a,s) \,\|\, p(z \mid s)\right]}_{\text{KL regularizer } (\mathcal{L}_D)}. \tag{5.3}$$

As noted earlier, PLAS can be interpreted as a two-stage optimization procedure. In the first stage, the policy optimizes a latent variable $z = \pi_\theta(s)$ in the latent action space. In the second stage, the optimized latent variable is mapped back to the original action space through the pretrained CVAE decoder. However, when the CVAE becomes overly dominant, this projection may collapse a broad range of latent actions into a narrow region of the original action space, as illustrated in Fig. 5.1. In particular, although $z = \pi_\theta(s)$

is iteratively updated through multiple gradient steps in the latent action space (red triangles), distinct optimized latent variables may still be decoded into nearly the same point (green circle) in the latent action projection space.

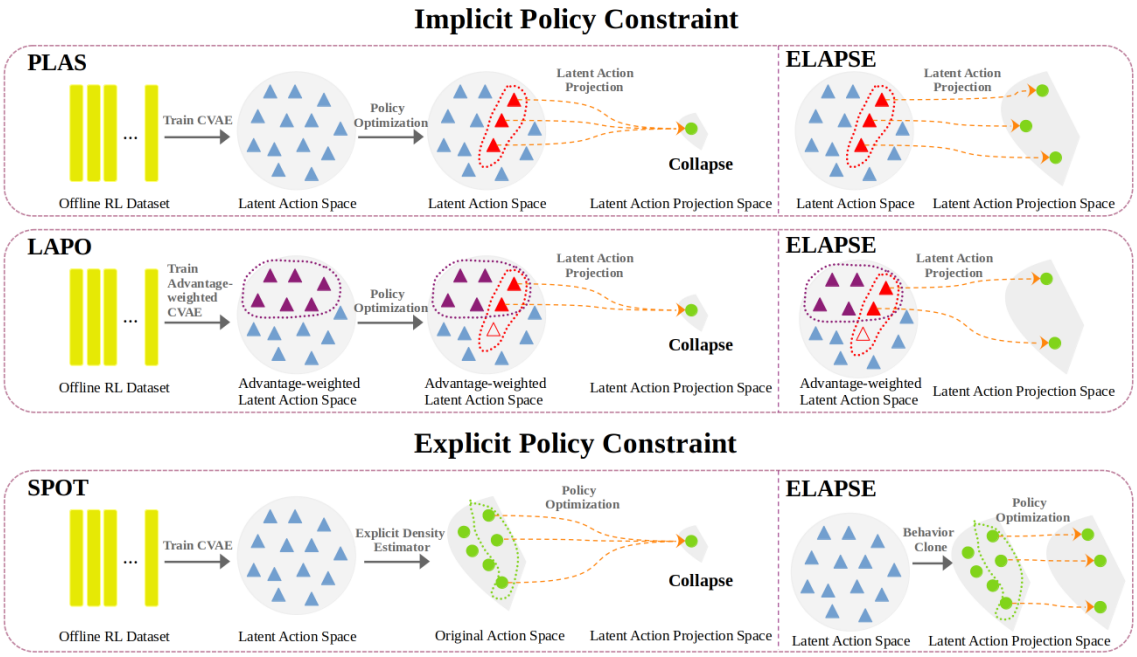


Figure 5.1: Training flow and latent action projection space collapse in PLAS, LAPO, and SPOT.

In Fig. 5.1, the blue triangles denote samples in the latent action space, the purple triangles correspond to high-weight samples, the red triangles mark preferred samples obtained through policy optimization, and the green circles represent samples in the latent action projection space. ELAPSE facilitates policy optimization by enlarging this projection space.

Subsequently, we visualize this collapse phenomenon on three D4RL datasets [123]: `hopper-medium-replay-v2 (hop-m-r)`, `hopper-medium-expert-v2 (hop-m-e-v2)` and `hopper-medium-expert (hop-m-e)` [1]. In these experiments, we first uniformly sample 1000 latent points. We then train three CVAE models, one for each dataset, using the PLAS objective

$$\mathcal{L}_{\text{PLAS}} = \mathcal{L}_R + 0.5\,\mathcal{L}_D.$$

[1] In PLAS, the `hopper-medium-expert` dataset is generated by concatenating the `hopper-medium-v2` and `hopper-medium-expert-v2` datasets.

Finally, we project the 1000 uniformly sampled latent points into the latent action projection space.

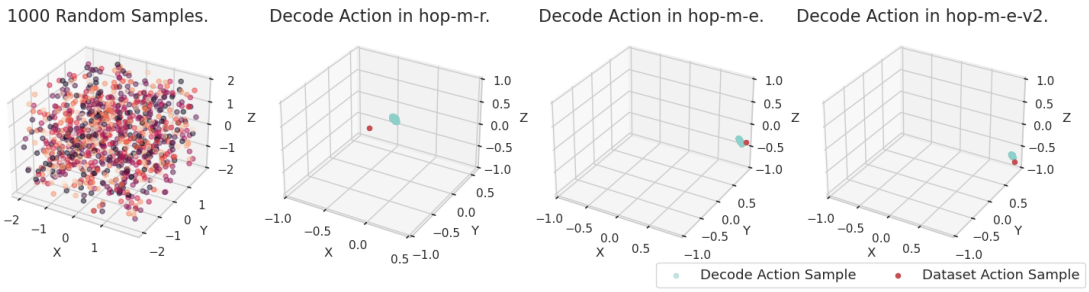


Figure 5.2: Visualization of latent action projection space collapse in PLAS. By default, PLAS defines $z = \pi_\theta(s) \in [-2, 2]$. Accordingly, we also restrict each latent coordinate to $z_i \in [-2, 2]$, $i = 1, 2, 3$.

The visualization results are presented in the right three panels of Fig. 5.2. Although the 1000 latent samples are uniformly distributed in the latent action space, their decoded counterparts collapse into a much smaller region of the original action space. This provides direct evidence of latent action projection space collapse. In actual training, the distribution of z induced by $z = \pi_\theta(s)$ is not uniformly distributed, since it is optimized to maximize $Q(s, a)$. Therefore, the collapse can become even more severe during real policy optimization.

After providing this intuitive visualization, we further analyze the origin of the phenomenon. Under posterior collapse, the decoder no longer meaningfully depends on the latent variable, and the reconstruction and KL terms in Eq. (5.3) degenerate into $\mathcal{L}_R^\times$ and $\mathcal{L}_D^\times$, respectively:

$$\mathcal{L}_R^\times = -\mathbb{E}_{q_\xi(z|s)} \left[\log p_\rho(a \mid s) \right] = -\log p_\rho(a \mid s), \tag{5.4}$$

$$\mathcal{L}_D^\times = D_{\mathrm{KL}} \left[q_\xi(z \mid s) \| p(z \mid s) \right]. \tag{5.5}$$

We now revisit the policy optimization objective of PLAS:

$$\max_\theta \mathcal{J}_{\mathrm{PLAS}} = \mathbb{E}_{a \sim p_\rho(\cdot|z,s),\, z \sim \pi_\theta(s),\, s \sim \mathcal{D}} \left[Q_\phi(s, a) \right]. \tag{5.6}$$

Under collapse, Eq. (5.6) degenerates into

$$\max_\theta \mathcal{J}_{\mathrm{PLAS}}^\times = \mathbb{E}_{a \sim p_\rho(\cdot|s),\, s \sim \mathcal{D}} \left[Q_\phi(s, a) \right]. \tag{5.7}$$

The reason is that, under $\mathcal{L}_R^\times$, the decoder $p_\rho(a \mid s)$ only minimizes the negative log-likelihood of data samples $(s, a) \sim \mathcal{D}$, while under $\mathcal{L}_D^\times$ the encoder $q_\xi(z \mid s)$ is driven to

match the prior $p(z \mid s)$. As a result, the latent policy π_θ disappears from the effective optimization pathway.

Eq. (5.7) is therefore problematic, because π_θ no longer appears in the effective objective. In other words, optimizing the latent policy can no longer improve performance. Under collapse, PLAS effectively degenerates into BC or Easy BCQ [2].

To further support this analysis, we plot the curves of $\mathcal{L}_R^\times$ and $\mathcal{L}_D^\times$ in Fig. 5.3. We observe that $\mathcal{L}_D^\times \to 0$, indicating that the latent action distribution becomes nearly identical to the prior $p(z \mid s)$. Consequently, policy optimization in the latent action space becomes ineffective.

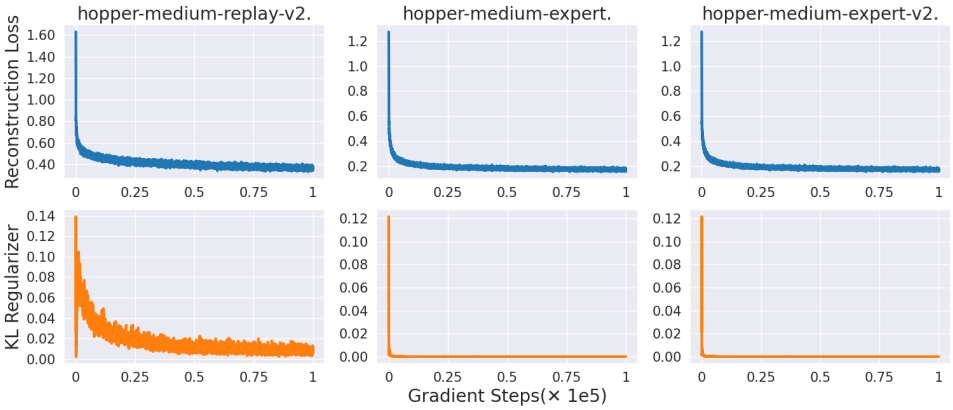


Figure 5.3: The $\mathcal{L}_R^\times$ and $\mathcal{L}_D^\times$ curves of PLAS.

5.4.2 Collapse in LAPO

LAPO [96] is another implicit policy-constrained offline RL method. Similar to PLAS, it performs policy optimization in a latent action space, but adopts an advantage-weighted CVAE. Its weighted ELBO objective is

$$\mathcal{L}_{\text{LAPO}} = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\omega(s,a)\,\mathcal{L}_{\text{CVAE}}\right], \tag{5.8}$$

where the importance weight is

$$\omega(s,a) = \exp\left(A(s,a)/\delta\right),$$

δ is the temperature parameter, $A(s,a)$ is the advantage function.

As illustrated in Fig. 5.1, LAPO first trains an advantage-weighted CVAE so that high-value actions receive greater emphasis in the latent action space (purple triangles). The

[2] Easy BCQ is introduced in [27].

latent variable $z = \pi_\theta(s)$ is then optimized through multiple gradient steps (red triangles). However, because policy optimization is still carried out in the latent action space and decoding still depends on the CVAE, LAPO suffers from the same latent action projection space collapse phenomenon.

The overall objective of LAPO is given by

$$\max_{\xi,\rho,\theta} \mathcal{J}_{\text{LAPO}} = \mathbb{E}_{a\sim p_\rho(\cdot|s,z),\, z\sim\pi_\theta(s),\, s\sim\mathcal{D}} \left[Q_\phi(s,a) \right] - \mathcal{L}_{\text{LAPO}}. \tag{5.9}$$

When collapse occurs, the first term in Eq. (5.9) becomes ineffective for the same reason as in Eq. (5.7). We then expand the second term:

$$\min_{\xi,\rho,\theta} \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[\omega(s,a)\, \mathcal{L}_{\text{CVAE}} \right] \quad \Rightarrow \quad \min_{\xi,\rho,\theta} \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[-e^{A(s,a)/\delta} \log p_\rho(a \mid s) \right]. \tag{5.10}$$

Hence, under collapse, the LAPO objective degenerates into

$$\max_{\rho} \mathcal{J}^{\times}_{\text{LAPO}} = \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[e^{A(s,a)/\delta} \log p_\rho(a \mid s) \right]. \tag{5.11}$$

The objective $\mathcal{J}^{\times}_{\text{LAPO}}$ is consistent with that of AWAC [21], whose optimization objective is

$$\max_{\theta} \mathcal{J}_{\text{AWAC}} = \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[e^{A(s,a)/\delta} \log \pi_\theta(a \mid s) \right], \tag{5.12}$$

which corresponds to a weighted supervised actor update.

The similarity between Eqs. (5.11) and (5.12) reveals two issues for LAPO under collapse. First, the latent policy $z = \pi_\theta(s)$, which is supposed to be optimized in the latent action space, is effectively replaced by the decoder $p_\rho(a \mid s)$, making latent policy optimization invalid. Second, since Eq. (5.12) corresponds to a simple supervised actor update, it implies that the generative capability of the CVAE in LAPO is entirely ignored. As in PLAS, the curves of $\mathcal{L}^{\times}_R$ and $\mathcal{L}^{\times}_D$ for LAPO, shown in Fig. 5.4, exhibit the same qualitative behavior.

5.4.3 Collapse in SPOT

SPOT [39] is an explicit policy-constrained method that relies on a CVAE-based density estimator. Unlike PLAS and LAPO, SPOT optimizes the policy directly in the original action space rather than in the latent action space. The CVAE-estimated behavior density is then incorporated into policy updates as a regularization term.

The objective of SPOT is

$$\max_{\xi,\rho,\theta} \mathcal{J}_{\text{SPOT}} = \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[Q_\phi(s, \pi_\theta(s)) - \lambda \mathcal{L}_{\text{CVAE}} \right]. \tag{5.13}$$

Here we analyze the objective using the ELBO regularization term rather than the importance-weighted autoencoder (IWAE) [152] variant, because the default number of

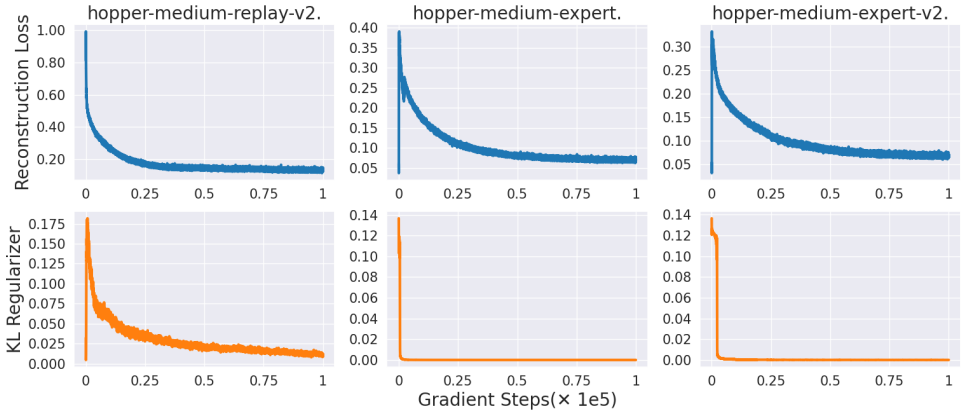


Figure 5.4: The $\mathcal{L}_R^{\times}$ and $\mathcal{L}_D^{\times}$ curves of LAPO.

importance samples in SPOT is 1, in which case the IWAE objective reduces to the ELBO objective.

Under collapse, the SPOT objective becomes

$$\max_{\rho,\theta} \mathcal{J}_{\text{SPOT}}^{\times} = \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[Q_\phi(s, \pi_\theta(s)) + \lambda \log p_\rho(\pi_\theta(s) \mid s) \right]. \tag{5.14}$$

Unlike PLAS and LAPO, policy optimization in SPOT remains active even under collapse. However, the regularization effect degenerates substantially. In this regime, SPOT becomes closely related to TD3+BC [38], whose objective is

$$\max_{\theta} \mathcal{J}_{\text{TD3+BC}} = \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[Q_\phi(s, \pi_\theta(s)) - \lambda (\pi_\theta(s) - a)^2 \right]. \tag{5.15}$$

The simplicity and competitive performance of TD3+BC make it attractive in offline RL. However, its hyperparameter λ is highly sensitive to the dataset, especially on suboptimal datasets [58]. When collapse occurs, SPOT inherits a similar limitation. The curves of $\mathcal{L}_R^{\times}$ and $\mathcal{L}_D^{\times}$ for SPOT are shown in Fig. 5.5.

Anti-exploration [97], another explicit CVAE-based policy-constrained method, introduces an anti-exploration penalty of the form

$$b(s,a) = \mu \|\pi_\theta(s) - p_\rho(a \mid z, s)\|_2^2.$$

Under collapse, it exhibits degeneration behavior similar to that of SPOT, and we therefore do not analyze it separately.

Overall, latent action projection space collapse is especially destructive for implicit CVAE-based policy-constrained methods, because policy optimization can be completely neutralized. By contrast, for explicit constraint methods, the policy optimization step remains active, but the intended regularization effect becomes substantially distorted.

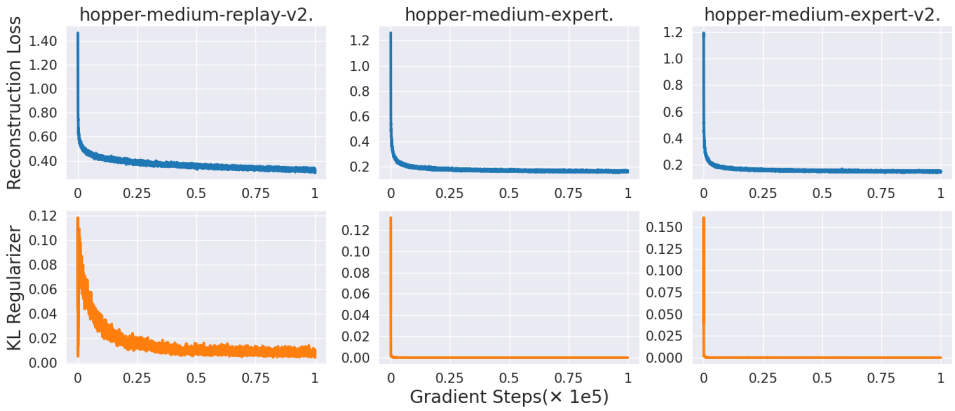


Figure 5.5: The $\mathcal{L}_R^{\times}$ and $\mathcal{L}_D^{\times}$ curves of SPOT.

5.5 Proposed Algorithm – ELAPSE

The analysis in Section 5.4 shows that latent action projection space collapse is closely associated with posterior collapse in the CVAE model. In particular, when the KL regularizer rapidly decays toward zero, the latent posterior becomes nearly indistinguishable from the prior, causing the latent action space to lose diversity and making policy optimization ineffective. Motivated by this observation, we propose **E**xpand **L**atent **A**ction **P**rojection **S**pac**E** (**ELAPSE**), a simple method that enlarges the effective projection space by enforcing a nontrivial lower bound on the KL regularizer of the CVAE model.

From another perspective, the CVAE objective can be viewed as an autoencoder reconstruction loss $\mathcal{L}_R$ together with a conditional KL regularization term $\mathcal{L}_D$, rather than merely as an ELBO objective [153,154]. The empirical results in Figs. 5.3, 5.4, and 5.5 show that the KL regularizer $\mathcal{L}_D$ decreases rapidly during training. When $\mathcal{L}_D \to 0$, the CVAE degenerates toward a conditional autoencoder, which causes latent samples to become insufficiently diverse. Moreover, such a conditional autoencoder is prone to overfitting the dataset [153], and in the present setting this overfitting corresponds to an overly dominant CVAE constraint that undermines policy optimization.

Hence, we propose the ELAPSE method to enhance policy optimization. When the prior $p(z \mid s)$ is a multivariate Gaussian distribution $\mathcal{N}(0, I)$, the KL regularizer

$$\mathcal{L}_D = D_{\mathrm{KL}}\left[q_\xi(z \mid a, s) \,\|\, p(z \mid s)\right]$$

for a single sample (s, a) is computed in Eq. (5.2). In practice, mini-batch samples are processed simultaneously through Monte Carlo estimation:

$$\begin{aligned}\mathcal{L}_D &= \frac{1}{2m}\sum_{j=1}^{m}\sum_{i=1}^{n}\left(\mu_{ij}^2+\sigma_{ij}^2-\log\sigma_{ij}^2-1\right)\\ &= \frac{1}{2}\sum_{i=1}^{n}\left(\frac{1}{m}\sum_{j=1}^{m}\mu_{ij}^2+\frac{1}{m}\sum_{j=1}^{m}\sigma_{ij}^2-\frac{1}{m}\sum_{j=1}^{m}\log\sigma_{ij}^2-1\right),\end{aligned} \tag{5.16}$$

where m is the mini-batch size, n is the latent dimension, and μ_{ij} and σ_{ij} are the mean and standard deviation of the approximate posterior for the i-th latent dimension of the j-th sample.

Inspired by BN-VAE [155], we further assume that, for each latent dimension i, the posterior mean μ_i and variance σ_i^2 follow distributions with fixed dataset-level mean and variance. Under this view, $\mathcal{L}_D$ itself becomes a random quantity induced by the distributions of μ_i and σ_i^2. Its expectation can then be written as

$$\begin{aligned}\mathbb{E}\left[\mathcal{L}_D\right] &= \frac{1}{2}\sum_{i=1}^{n}\left(\mathrm{Var}[\mu_i]+\mathbb{E}^2[\mu_i]+\mathbb{E}[\sigma_i^2]-\mathbb{E}[\log\sigma_i^2]-1\right)\\ &\geq \frac{1}{2}\sum_{i=1}^{n}\left(\mathrm{Var}[\mu_i]+\mathbb{E}^2[\mu_i]\right),\end{aligned} \tag{5.17}$$

where we use $\mathbb{E}[x^2] = \mathrm{Var}[x] + \mathbb{E}^2[x]$, together with the fact that $x - \log x \geq 1$, which implies

$$\mathbb{E}\left[x-\log x-1\right]\geq 0.$$

Eq. (5.17) shows that the lower bound of $\mathbb{E}[\mathcal{L}_D]$ depends directly on the mean and variance of μ_i.

Naturally, Batch Normalization (BN) [156] can be leveraged to transform μ_i into a distribution with a customized mean and variance. The $\hat{\mu}_i$ after BN can be written as

$$\hat{\mu}_i = w\frac{\mu_i-\mathrm{mean}(\mu_i)}{\mathrm{std}(\mu_i)}+b,$$

where $\mathrm{mean}(x)$ and $\mathrm{std}(x)$ denote the mean and standard deviation of x, w and b are the weight and bias parameters in BN layer. Based on $\mathbb{E}\left[\hat{\mu}_i\right] = b$ and $Var\left[\hat{\mu}_i\right] = w^2$, hence we obtain the following lower bound:

$$\mathbb{E}\left[\hat{\mathcal{L}}_D\right]\geq\frac{n}{2}(w^2+b^2). \tag{5.18}$$

Eq. (5.18) reveals the key mechanism of ELAPSE: by controlling the statistics of the posterior means through BN, the KL term is prevented from collapsing to zero, which in turn enlarges the effective latent action projection space. In practice, we freeze the BN scale parameter w as a fixed constant and keep the bias b learnable, which provides a simple and stable way to control the lower bound while maintaining flexibility.

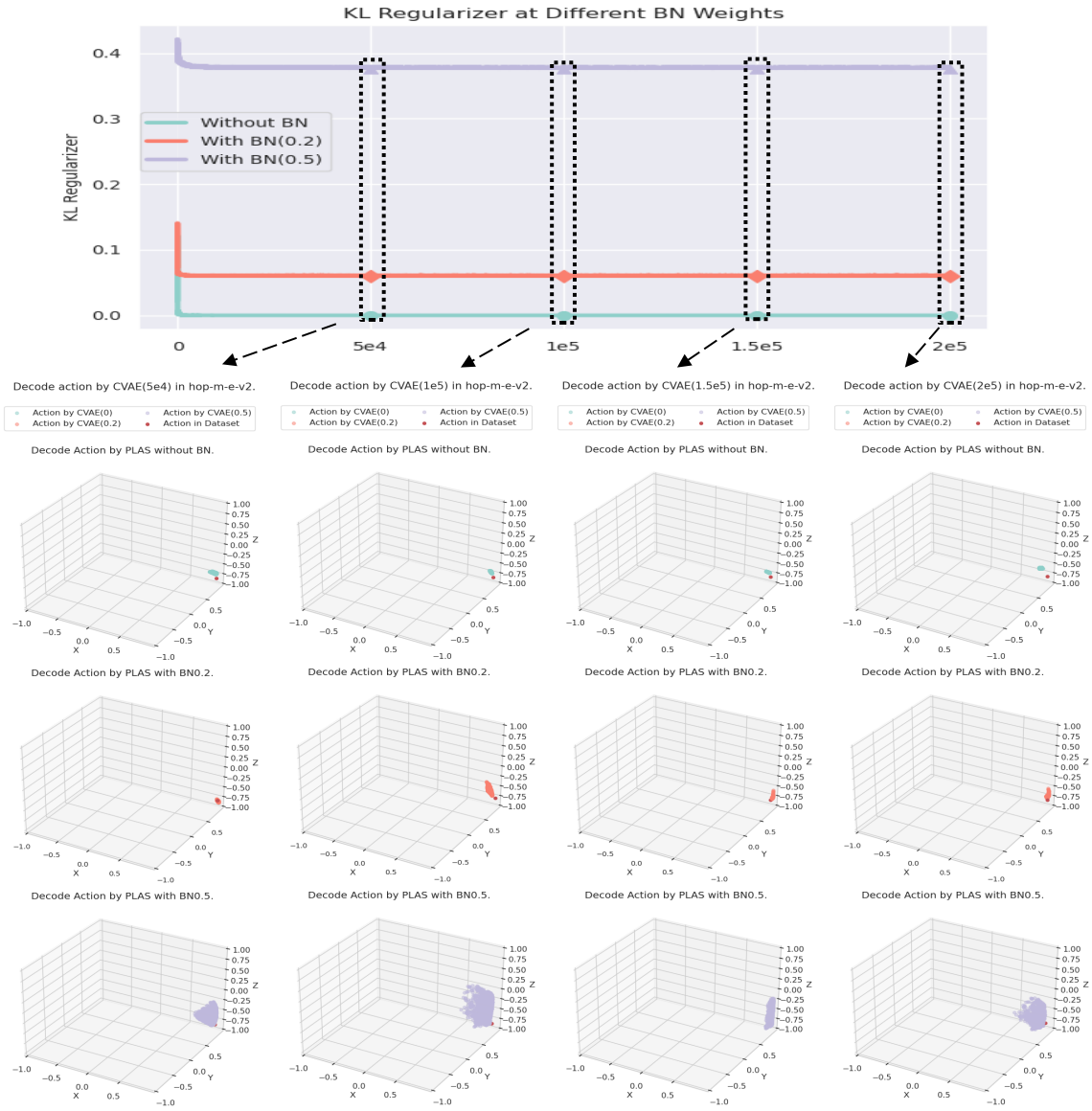


Figure 5.6: ELAPSE-PLAS with different BN weights w.

We then utilize the PLAS algorithm to conduct ELAPSE-PLAS experiments on the `hopper-medium-expert-v2` dataset. The results are shown in Fig. 5.6. In this experiment, we evaluate PLAS without BN, with BN weights of $w = 0.2$ and $w = 0.5$. Firstly, we plot the iterative training figure of the KL regularizer. Since the latent action dimension $n = 3$, the theoretical convergence values of the KL regularizer at different weights $w = 0, 0.2, 0.5$ are:

$$\begin{aligned} &\mathbb{E}[\hat{\mathcal{L}}_D^0] \geq 3 \times (0^2 + b^2)/2 \geq 0, \\ &\mathbb{E}[\hat{\mathcal{L}}_D^{0.2}] \geq 3 \times (0.2^2 + b^2)/2 \geq 0.06, \\ &\mathbb{E}[\hat{\mathcal{L}}_D^{0.5}] \geq 3 \times (0.5^2 + b^2)/2 \geq 0.375. \end{aligned} \tag{5.19}$$

The empirical results are consistent with these theoretical predictions. In addition, we visualize the latent action projection space at different stages of CVAE training using the same protocol as in the collapse visualization experiments (Fig. 5.2). As shown in Fig. 5.6, the latent action projection space becomes substantially expanded when BN weights $w = 0.2$ and $w = 0.5$ are used. This indicates that the CVAE constraint is no longer overly dominant and that policy optimization can be mapped more effectively into the projection space. Moreover, an expanded projection space provides greater room for optimization without requiring additional perturbations.

In practice, ELAPSE is not a separate offline RL backbone, but rather a CVAE enhancement that can be incorporated into representative CVAE-based offline RL algorithms such as PLAS, LAPO, and SPOT. The core idea is simply to replace the original CVAE module with the ELAPSE-CVAE module described above, while leaving the downstream policy optimization procedure unchanged. The generic ELAPSE-enhanced training procedure is summarized in Algorithm 3.

Additional variants are provided in Appendix 7.5.

5.6 Experiments

In the previous sections, we theoretically examined the consequences of latent action projection collapse in PLAS, LAPO, and SPOT, and also visualized this phenomenon. However, two further questions remain: (i) how does this collapse affect the final performance of offline RL algorithms, and in what way does ELAPSE enlarge the projection space to improve performance? (ii) how does ELAPSE compare with state-of-the-art baselines?

To answer these two questions, we next conduct experiments on the standard D4RL benchmark [123], covering a total of 27 datasets.

5.6.1 ELAPSE for Performance Improvement

To address question (i), we first compare the normalized scores of PLAS without perturbation and ELAPSE-PLAS with BN weights $w = 0.2$ and $w = 0.5$ on the `hopper-medium-replay-v2` (`hopper-m-r-v2`) and `hopper-medium-expert-v2`

Algorithm 3 Expand Latent Action Projection SpacE (ELAPSE)

Input: Dataset $\mathcal{D} = \{(s, a, r, s')\}$.
Parameters: Q-networks $Q_{\phi_{1,2}}$, target Q-networks $Q_{\phi'_{1,2}}$, value network V_ψ, policy network π_θ, BN weight w.

// Pre-train CVAE model for N_β steps.
for $t = 1$ to N_β **do**
  Sample mini-batch $(s, a) \sim \mathcal{D}$.
  Apply batch normalization:

$$\hat{\mu} = w\frac{\mu - \text{mean}(\mu)}{\text{std}(\mu)} + b.$$

  Optimize CVAE model using the ELAPSE-modified BN weight w by Eq. (5.3).
end for
// Policy evaluation and Policy improvement.
for each gradient step **do**
  Sample $(s, a, r, s') \sim \mathcal{D}$.
  Decode latent actions using the ELAPSE-CVAE decoder: $\hat{a} = p_\rho(\cdot \mid z = \pi_\theta(s); s)$.
  Update the critic, actor, and other algorithm-specific components according to the chosen base method (e.g., PLAS (5.6), LAPO (5.9), or SPOT (5.13)).
end for
return Policy π_θ.

(`hopper-m-e-v2`) datasets. In PLAS, the latent variable is obtained through policy optimization as $z_\pi = \pi_\theta(s) \in [z_{min}, z_{max}]$. To illustrate the effect of the collapse issue, we additionally use a random policy to sample latent variables in the latent action space, denoted by $z_{ran} = \pi_{\text{random}}[z_{min}, z_{max}]$, and compare them with the optimized latent variables z_π. The corresponding results are shown in Fig. 5.7.

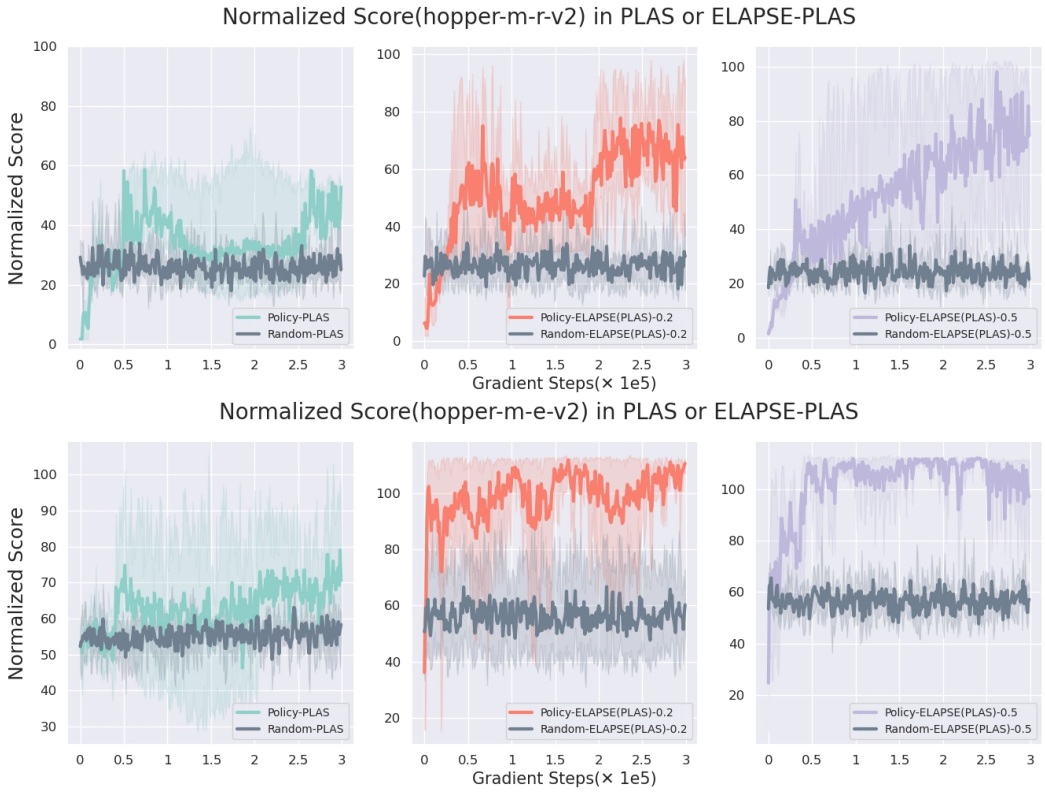


Figure 5.7: Normalized scores of the random policy and the optimized policy in PLAS without perturbation and in ELAPSE-PLAS.

From the two left panels in Fig. 5.7, we observe that on both datasets the gap between the normalized scores produced by z_π (light blue curve) and z_{ran} (grey curve) is small. This indicates that PLAS suffers from the collapse problem, which makes policy optimization ineffective. By contrast, the difference becomes much more evident in ELAPSE-PLAS. Moreover, when comparing the normalized scores of policy-optimized PLAS (light blue curve) with those of ELAPSE-PLAS (orange and purple curves) on both datasets, we find that ELAPSE-PLAS achieves substantially higher scores than PLAS for both $w = 0.2$ and $w = 0.5$. This advantage suggests that ELAPSE-PLAS relaxes the CVAE constraint and thereby improves policy optimization.

Another interesting phenomenon in Fig. 5.7 is that the normalized scores of the random policy (grey curves) show different trends on the two datasets. On `hopper-medium-replay-v2`, the random policy in PLAS attains the highest score compared with its counterparts in ELAPSE-PLAS ($w = 0.2$) and ELAPSE-PLAS ($w = 0.5$). In contrast, on `hopper-medium-expert-v2`, the trend is completely reversed. One possible explanation is that, once the CVAE has been trained on a high-quality dataset and has captured its distribution, it can map random latent policies to high-score action regions with relatively high probability. However, when the CVAE is trained on a low-quality dataset, it is much less likely to project a random latent policy to actions with good performance. In addition, as the BN weight increases from 0 to 0.5, the action projection space becomes more dispersed, as shown in Fig. 5.6. This more dispersed projection space makes the above phenomenon even more pronounced.

Next, we compare PLAS, LAPO, SPOT, and their ELAPSE-enhanced versions on the Gym-MuJoCo and AntMaze D4RL datasets. The results are reported in Tab. 5.1. In the experiments of Tab. 5.1, the BN weight w is set to 0.2 by default, because we found that a large value of w performs poorly on the random, medium, and medium-replay datasets. This is because an over-optimistic expansion of the latent action projection space on low-quality datasets will augment suboptimal samples, thus making policy optimization difficult.

As reported in Tab. 5.1, ELAPSE-PLAS and ELAPSE-LAPO outperform PLAS and LAPO on the majority of datasets, achieving better results on 17 out of 21 datasets for both methods. This clearly demonstrates the effectiveness of ELAPSE in improving the performance of implicit policy-constraint methods.

By contrast, ELAPSE-SPOT does not exhibit the same level of advantage over SPOT, outperforming it on 13 of the 21 datasets. A main reason is that SPOT is sensitive to the hyperparameter λ in Eq. (5.13). When λ is large, SPOT can better suppress extrapolation error, but the resulting constraint may become overly restrictive. When λ is small, the constraint is loosened, but the policy may produce OOD actions. A similar phenomenon has also been observed in the hyperparameter studies reported in [39,58]. Since this issue is not the main focus of the present work, we set $\lambda = 0.1$ by default.

5.6.2 Comparison Experiments

From the previous experiments, we observe that ELAPSE-LAPO performs better than the other ELAPSE variants on most datasets. Therefore, to answer question (ii), we compare ELAPSE-LAPO with prior competitive methods, including TD3+BC [38], Onestep [27], CQL [24], IQL [23], RvS [130], SAC+ML [157], Qsarsa-AC [158], BCQ [17], POR [113], CPI [159], and ODPR-R [160]. The full comparison results are summarized in Tab. 5.2. They show that ELAPSE-LAPO outperforms the baselines on most datasets, achieves top-two performance on 14 out of 21 datasets, and obtains the highest total score.

Please refer to Appendix 7.6 for more experiment details.

Table 5.1: Normalized scores of PLAS, LAPO, SPOT, and their ELAPSE-enhanced versions on the Gym-MuJoCo and AntMaze D4RL datasets. **Bold** indicates that the ELAPSE-enhanced version outperforms the corresponding vanilla method.

Gym-MuJoCo Domains	PLAS	ELAPSE-PLAS	LAPO	ELAPSE-LAPO	SPOT	ELAPSE-SPOT
walker2d-exp-v2	107.1	**110.8**	112.3	110.9	108.7	**109.1**
hopper-exp-v2	110.8	107.7	106.8	**108.5**	103.1	**104.2**
halfcheetah-exp-v2	91.4	**94.1**	95.9	**96.5**	93.1	**94.1**
walker2d-med-exp-v2	97.2	**110.5**	109.4	**110.6**	39.6	**109.7**
hopper-med-exp-v2	73.1	**104.6**	100.7	**109.1**	71.3	**102.8**
halfcheetah-med-exp-v2	54.7	**89.1**	86.1	**94.0**	56.2	**89.7**
walker2d-med-rep-v2	29.9	**64.3**	52.3	**56.1**	91.6	29.7
hopper-med-rep-v2	22.3	**66.7**	37.6	**57.5**	100.2	52.5
halfcheetah-med-rep-v2	35.9	**47.4**	43.9	**44.7**	50.1	40.6
walker2d-med-v2	76.5	62.7	80.8	**81.5**	75.6	**77.7**
hopper-med-v2	51.0	**52.4**	51.6	**52.1**	86.0	56.3
halfcheetah-med-v2	44.5	**50.0**	46.0	**46.5**	55.0	44.2
walker2d-ran-v2	9.2	2.0	1.3	**4.5**	5.1	**5.4**
hopper-ran-v2	6.7	**31.8**	23.5	**32.1**	8.9	**12.9**
halfcheetah-ran-v2	26.5	**29.4**	30.6	24.7	2.3	**3.3**
Total	836.8	**1023.5**	978.8	**1029.3**	946.8	932.2
AntMaze Domains	PLAS	ELAPSE-PLAS	LAPO	ELAPSE-LAPO	SPOT	ELAPSE-SPOT
antmaze-u-v1	68.0	66.0	93.3	**96.0**	62.0	59.3
antmaze-u-d-v1	44.0	**71.3**	91.3	**92.7**	62.0	**67.3**
antmaze-m-p-v1	0.0	**41.3**	93.3	71.3	0.3	**0.7**
antmaze-m-d-v1	0.0	**44.7**	85.7	**87.8**	0.7	**3.3**
antmaze-l-p-v1	0.0	**9.3**	30.0	**38.9**	0.0	0.0
antmaze-l-d-v1	0.0	**21.0**	61.7	54.7	0.0	0.0
Total	112.0	**253.6**	455.3	441.4	125.0	**130.6**

Table 5.2: The normalized score of ELAPSE-LAPO and other strong baselines. We indicate in **Bold** that the scores of ELAPSE-LAPO are the top two among all algorithms.

Gym-MuJoCo Domains	TD3+BC	Onestep	CQL	IQL	RvS	SAC+ML	Qsarsa-AC	ELAPSE-LAPO
walker2d-exp-v2	108.1	107.3	108.5	96.9	-	108.2	107.0	**110.9**
hopper-exp-v2	104.2	105.9	96.5	108.8	-	110.9	95.0	108.5
halfcheetah-exp-v2	94.6	93.8	96.3	95.5	-	93.6	93.8	96.5
walker2d-med-exp-v2	106.4	105.5	107.9	109.6	106.0	108.1	105.2	**110.6**
hopper-med-exp-v2	83.6	45.6	105.6	91.5	101.7	93.1	56.9	**109.1**
halfcheetah-med-exp-v2	91.8	47.9	75.6	86.7	92.2	93.5	91.3	**94.0**
walker2d-med-rep-v2	67.6	28.6	81.8	73.9	60.6	62.2	83.7	56.1
hopper-med-rep-v2	39.3	43.2	88.7	94.7	73.5	48.8	92.6	57.5
halfcheetah-med-rep-v2	42.5	38.1	45.5	44.2	38.0	42.5	42.2	**44.7**
walker2d-med-v2	82.5	70.7	73.3	78.3	71.7	81.2	79.0	**81.5**
hopper-med-v2	46.0	45.5	53.0	66.3	60.2	54.3	78.6	52.1
halfcheetah-med-v2	45.4	42.8	47.0	47.4	41.6	46.3	44.1	**46.5**
walker2d-ran-v2	5.4	1.6	5.1	6.8	-0.2	3.4	5.9	4.5
hopper-ran-v2	12.7	3.6	7.9	7.6	7.7	7.9	7.5	**32.1**
halfcheetah-ran-v2	6.0	2.3	17.5	13.4	3.9	17.3	30.2	**24.7**
Total	936.1	782.4	1010.2	1021.6	-	971.3	1013.0	**1029.3**
AntMaze Domains	BCQ	Onestep	CQL	IQL	POR	CPI	ODPR-R	ELAPSE-LAPO
antmaze-u-v0	78.9	64.3	74.0	87.5	76.8	98.8	87.8	**96.7**
antmaze-u-d-v0	55.0	60.7	84.0	62.2	64.8	88.6	66.0	75.6
antmaze-m-p-v0	0.0	0.3	61.2	71.2	87.2	82.4	72.0	**84.0**
antmaze-m-d-v0	0.0	0.0	53.7	70.0	75.2	80.4	74.2	**82.2**
antmaze-l-p-v0	6.7	0.0	15.8	39.6	24.4	20.6	49.6	**46.7**
antmaze-l-d-v0	2.2	0.0	14.9	47.5	59.2	45.2	43.0	**52.5**
Total	142.8	125.3	303.6	378.0	387.6	416.0	392.6	**437.7**

5.6.3 Sample Complexity

The sample complexity of a machine learning algorithm describes how many training samples are required for it to successfully learn a target function. For offline algorithms, analyzing sample complexity is particularly important. However, for complex RL problems, especially those defined over infinite continuous state-action spaces, this notion is not straightforward to formalize. Therefore, the authors of [161] characterize sample complexity instead in terms of the sensitivity of offline RL algorithms to the size of the training dataset. Based on this definition, they study the sample complexity of several popular offline RL methods on continuous-control benchmark tasks. In what follows, we adopt the same evaluation protocol to analyze ELAPSE-LAPO.

We first report the amount of data contained in the Hopper datasets in Tab. 5.3.

Table 5.3: Numbers of samples in the D4RL Hopper datasets.

Dataset Name	**Samples**
hopper-random-v2	999999
hopper-medium-v2	999998
hopper-medium-replay-v2	401598
hopper-medium-expert-v2	1998966
hopper-expert-v2	999061

Next, following the setup in [161], we evaluate the sample complexity of ELAPSE-LAPO on the 5 Hopper datasets. ELAPSE-LAPO is tested on different fractions of each dataset, namely $0.2, 0.4, 0.6, 0.8$, and 1 times the original number of samples. For instance, the `hopper-random-v2` dataset contains 999,999 samples; accordingly, ELAPSE-LAPO is evaluated on subsets containing $0.2 \times 999{,}999$, $0.4 \times 999{,}999$, $0.6 \times 999{,}999$, $0.8 \times 999{,}999$, and the full dataset. The experimental results are presented in Fig. 5.8.

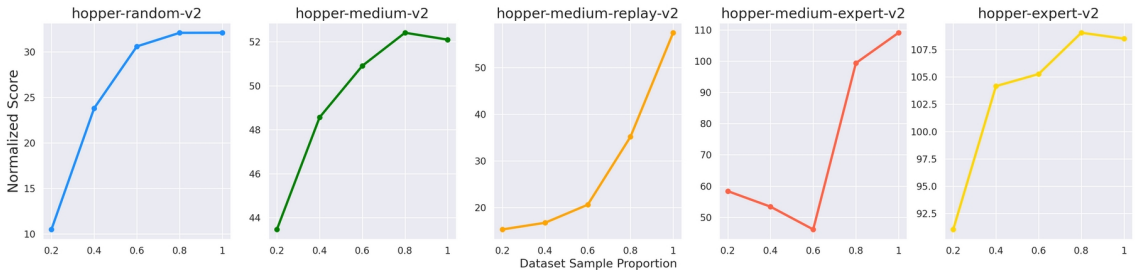


Figure 5.8: Sample-complexity results of ELAPSE-LAPO on the Hopper datasets.

The results show that the performance of ELAPSE-LAPO generally improves as the amount of training data increases. Since a CVAE, as a generative model, is

designed to fit the data distribution, enlarging the dataset can improve the modeling capability of the CVAE and thereby enhance the performance of the overall offline RL method that relies on it. This behavior can be observed on the `hopper-medium-replay-v2` and `hopper-medium-expert-v2` datasets. However, on the `hopper-random-v2`, `hopper-medium-v2`, and `hopper-expert-v2` datasets, performance becomes saturated: increasing the amount of training data does not lead to further improvement and may even reduce performance. This phenomenon is referred to as overfitting in offline RL in [161].

5.6.4 Comparison of BN with Other Methods

In this subsection, we compare the BN-based ELAPSE design with several existing techniques, including KL cost annealing [153], β-VAE [162] with relatively large coefficients $\beta = 0.1$ and 1, and cyclical annealing [154]. These approaches have demonstrated competitive performance in applications such as language modeling and image generation. The comparison results are presented in Fig. 5.9.

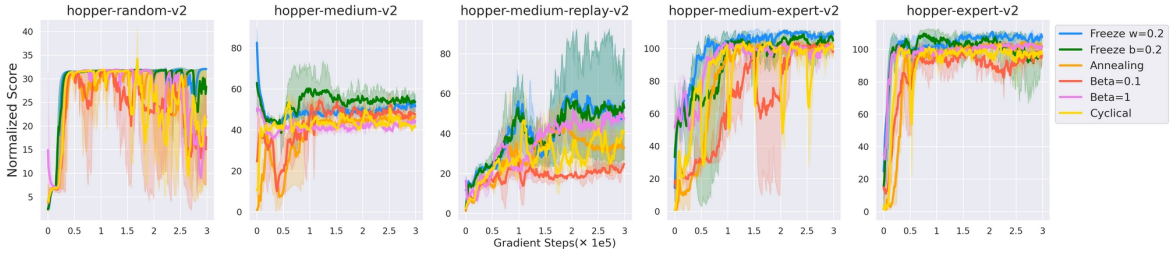


Figure 5.9: Comparison results with freeze w=0.2, freeze b=0.2, KL cost annealing, $\beta = 0.1, 1$ in β-VAE, and cyclical annealing.

As shown in Fig. 5.9, ELAPSE achieves the best performance on the Hopper datasets. In addition, the curve labeled *freeze* $b = 0.2$ denotes the setting in which the bias is fixed at $b = 0.2$, while the weight w is treated as a learnable parameter. This setting is symmetric to ELAPSE, where the weight is fixed at $w = 0.2$ and the bias b is learned. As a result, we find that the two curves, *freeze* $w = 0.2$ and *freeze* $b = 0.2$, are almost identical.

5.7 Conclusion

In this chapter, we identified the latent action projection space collapse problem in several representative CVAE-based offline RL algorithms, including PLAS, LAPO, SPOT, and Anti-exploration, and systematically analyzed how this failure mode undermines their policy optimization mechanisms. Through both theoretical analysis and visualization ex-

periments, we showed that such collapse can severely distort the effective optimization pathway and, in some cases, even neutralize the benefit of policy optimization altogether.

Motivated by this analysis, we proposed ELAPSE, a simple yet effective method that enlarges the latent action projection space and thereby improves policy optimization under CVAE-based constraints. The core idea of ELAPSE is to alleviate the excessive compression induced by the pretrained CVAE model, so that policy updates in the latent space can be more faithfully reflected in the original action space. In this way, ELAPSE helps restore the effectiveness of policy optimization while preserving the advantages of CVAE-based policy constraints.

Extensive experimental results demonstrated that ELAPSE consistently brings substantial performance improvements over the original CVAE-based methods. Moreover, ELAPSE also showed strong empirical performance on the D4RL benchmark when compared with a range of competitive offline RL baselines, further confirming its effectiveness, robustness, and practical value.

Chapter 6

General Conclusion and Future Work

This chapter concludes the thesis by summarizing the main findings and contributions, and by outlining several promising directions for future research. First, it revisits the central questions that motivated this thesis and synthesizes the main conceptual, methodological, and algorithmic contributions developed across Chapters 2–5. It then discusses a set of open problems and future directions, including deeper evolutionary analysis of offline RL, more robust policy evaluation and policy improvement, stronger generative behavior modeling, and scalable offline RL in large-model settings.

6.1 General Conclusion

This thesis investigated model-free offline RL, with a particular focus on how to achieve safe and effective policy improvement under distribution shift. As discussed throughout the thesis, the central difficulty of offline RL lies in the fact that policy improvement must be performed solely from a fixed dataset, without any further interaction with the environment. This setting makes it necessary to improve beyond the behavior policy while simultaneously controlling the risks induced by OOD actions and extrapolation error. Against this background, this thesis was organized around four research questions: the methodological evolution of offline RL, the reconciliation of in-sample and multi-step learning, the unification and adaptation of policy constraints, and the failure modes of CVAE-based behavior modeling.

A first contribution of this thesis is conceptual and analytical. We revisited offline RL from the perspective of the classical deadly triad in RL, and argued that distribution shift and extrapolation error provide a unifying lens through which a large number of offline RL methods can be understood. Building on this perspective, we further developed a refined taxonomy of model-free offline RL methods and analyzed the macro-level evolution of offline RL algorithm design. Rather than viewing existing methods only as a set of horizontally organized categories, this thesis highlighted how different algorithmic families

emerged and evolved in response to recurring challenges. In this sense, the thesis not only proposed new algorithms, but also contributed a broader understanding of how offline RL has developed as a research field.

A second contribution is methodological. To address the tension between in-sample learning and multi-step learning, we proposed PIQL. PIQL extends the in-sample learning paradigm of IQL by replacing the fixed expectile hyperparameter with a projection-based adaptive parameter, thereby enabling a multi-step interpretation while preserving the expectile-regression structure and the in-sample learning property. In addition, PIQL adopts a support-constrained policy-improvement objective that is more consistent with the policy-evaluation stage. Both theoretical and empirical results showed that this design improves the safety and effectiveness of policy improvement, especially in settings where richer learning signals are desirable but OOD risks remain severe.

A third contribution concerns policy constraints in offline RL. Existing policy-constrained methods often rely on a manually chosen constraint family and a manually tuned constraint strength. To address this issue, we developed the CCI framework, which unifies wBC, KL-based density regularization, and support constraints under a common optimization view. Building on CCI, we further proposed ACPO, a practical primal-dual method that automatically adapts the interpolation parameter during training. In doing so, this thesis provided not only a unified perspective on mainstream policy-constraint families, but also a practical mechanism for automatically selecting an appropriate point along the constraint spectrum. This contribution advances offline RL from manually designed conservatism toward more adaptive and principled constraint control.

A fourth contribution focuses on generative behavior modeling. While CVAE-based offline RL methods are attractive because they can model behavior support and constrain policy optimization implicitly or explicitly, this thesis showed that their practical behavior is more fragile than is often assumed. In particular, we identified and analyzed the latent action projection space collapse phenomenon, through which posterior collapse in CVAE can severely weaken or even neutralize policy optimization. Based on this analysis, we proposed ELAPSE, a simple yet effective method that enlarges the projection space by preventing the KL regularizer from collapsing. This contribution demonstrates that, in offline RL, the success of generative-model-based methods depends not only on nominal support modeling, but also on preserving a sufficiently expressive space for policy optimization.

Taken together, the contributions of this thesis form a coherent line of progress from understanding to design. At the analytical level, the thesis clarifies the central challenges and the methodological evolution of offline RL. At the algorithmic level, it develops new methods that address three important dimensions of offline RL design: policy evaluation, policy improvement, and behavior modeling. More broadly, the thesis argues that reliable offline RL requires balancing flexibility and conservatism at multiple levels: balancing in-sample and multi-step learning, balancing different forms and strengths of policy constraints, and balancing support-preserving generative modeling with effective policy

optimization. In this sense, the thesis contributes both to the theoretical understanding of offline RL and to the practical development of more reliable and adaptive offline RL algorithms.

For clarity, the main contributions of the thesis (C1–C4 in Chapter 1) are summarized again below:

- *C1.* We analyzed key challenges in offline RL through the lens of the deadly triad, emphasizing distribution shift and extrapolation error caused by OOD actions, and we provided an algorithmic taxonomy of model-free offline RL and distilled design principles that explain how different method families evolve to address core challenges.
- *C2.* We proposed PIQL to reconcile in-sample learning with multi-step learning, enabling safer and more effective policy improvement beyond the dataset.
- *C3.* We introduced CCI and ACPO to unify policy-constraint families and automatically adapt constraint type and strength.
- *C4.* We characterized latent action projection space collapse in CVAE-based offline RL and proposed ELAPSE to expand the projection space and recover effective policy optimization.

Overall, this thesis advances the understanding and practice of offline RL by addressing both its conceptual foundations and its practical algorithms. It provides a step toward more reliable, adaptive, and theoretically grounded offline RL.

6.2 Future Work

Although the proposed methods improve both the understanding and the performance of offline RL, many important questions remain open. Based on the findings of this thesis, several promising directions for future work can be identified.

1. **Deeper evolutionary analysis.** Building on the analysis in Chapter 2, the evolutionary perspective developed in this thesis can be extended further. The current taxonomy and macro-level perspective mainly focus on model-free offline RL and on the major algorithmic families that have emerged so far. Future work may extend this perspective to include newer paradigms, such as more expressive sequence-modeling methods, diffusion-based decision models, and hybrid approaches that combine model-free, model-based, and generative components. More broadly, a deeper evolutionary analysis may help reveal whether offline RL is converging toward a small number of fundamental design principles or continuing to diversify into task-dependent methodological ecosystems.

2. **Policy Evaluation.** While PIQL reconciles in-sample learning with multi-step learning, it still relies on the coverage and quality of the available dataset. Future work may investigate how to make PIQL more data-efficient, so that it can learn strong policies from substantially smaller offline datasets. Another promising direction is to study stronger projection mechanisms or uncertainty-aware evaluation.

3. **Policy Improvement.** The CCI framework and ACPO can be generalized in multiple ways. In this thesis, the unified constraint spectrum mainly connects wBC, KL-based density regularization, and support constraints. Future work may explore broader unified frameworks that incorporate additional constraint forms, such as neighbor constraints or state-dependent constraints.

4. **Generative-model-based Offline RL.** The analysis in Chapter 5 suggests that there exists a fundamental trade-off between the strength of generative behavior modeling and the effectiveness of downstream policy optimization. A stronger generative model may provide better behavior support estimation and more reliable action generation, but it may also impose an overly dominant constraint that weakens policy optimization; conversely, a weaker generative constraint may enlarge the optimization space but fail to prevent OOD actions. Future work may further analyze and validate this trade-off across a broader family of generative models, including diffusion models, GAN-based models, and flow-matching models.

5. **Scalable Offline RL.** With the rapid development of large-scale models, especially large language models and foundation models, offline RL is also likely to move toward larger model classes and larger datasets. However, as discussed in Chapter 2, simply scaling up model size and dataset size does not automatically resolve the core challenges of offline RL, and may even exacerbate distribution shift and extrapolation error. Future work should therefore investigate how to design offline RL algorithms that remain reliable and adaptive in large-scale settings, for example by integrating stronger constraint mechanisms, scalable uncertainty estimation, and more robust behavior modeling into large architectures. Developing scalable offline RL methods that can effectively leverage large models and large datasets without sacrificing safety or stability is an important long-term research direction.

In summary, this thesis provides a set of foundational steps toward more reliable offline RL, but it also highlights that offline RL remains an evolving field with many unresolved questions. Continued progress will require not only new algorithms, but also deeper understanding, stronger diagnostics, and closer integration between methodological innovation and deployment requirements. Advancing along these directions will further strengthen the safety, adaptability, and applicability of offline reinforcement learning in complex sequential decision-making systems.

Chapter 7

Appendix

This chapter provides supplementary materials for the main results of the thesis, including full proofs, additional theoretical derivations, implementation details, hyperparameter settings, and extra experimental results for PIQL, ACPO, and ELAPSE. In the main chapters, we introduced the corresponding algorithms, theoretical results, and empirical evaluations; here, we present the detailed derivations and supporting evidence that were omitted from the main text for clarity and conciseness.

The remainder of this chapter is organized as follows. Sections 7.1 and 7.2 present the proofs and experimental details for PIQL. Sections 7.3 and 7.4 provide the theoretical derivations and additional implementation details for ACPO, including the CCI framework, performance bounds, and supplementary ablation results. Sections 7.5 and 7.6 present the supplementary analysis and implementation details for ELAPSE, together with additional discussions on its variants and experimental settings.

7.1 Proofs – PIQL

7.1.1 Proof of Lemma 3.1

Lemma 3.1 *For any state s and any τ_1, τ_2 such that $\tau_1 < \tau_2$, we have*

$$V_{\tau_1}(s) \leq V_{\tau_2}(s).$$

Proof. The proof follows that of *Lemma 2* in [23]. □

7.1.2 Proof of Theorem 3.2

Theorem 3.2 *If the ϵ in the constraint of Eq. (3.9) is sufficiently small for the set*

$$\{a \mid \pi_\beta(a \mid s) > 0,\ \pi_\theta(a \mid s) > 0,\ Q_{\hat{\phi}}(s,a) - V_\psi(s) < 0,\ \forall s\},$$

then Eq. (3.13) can be transformed into

$$L_V(\psi) = \mathbb{E}_{s\sim\mathcal{D},\, a\sim\pi_\theta(\cdot|s)}\left[L_2^{\bar{\tau}_{\text{proj}}(a|s)}\left(Q_{\hat{\phi}}(s,a) - V_\psi(s)\right)\right],$$

where

$$\bar{\tau}_{\text{proj}}(a \mid s) = \frac{\vec{\pi}_\beta(a \mid s)\cdot\vec{\pi}_\theta(a \mid s)}{\|\vec{\pi}_\theta(a \mid s)\|^2}\pi_\beta(a \mid s).$$

Proof. For a single state-action pair (s,a), $\tau_{\text{proj}}(a \mid s)$ is defined in Eq. (3.12). Substituting $\tau_{\text{proj}}(a \mid s)$ into Eq. (3.13), we obtain

$$\begin{aligned} L_V(\psi) &= \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[L_2^{\tau_{\text{proj}}(a|s)}\left(Q_{\hat{\phi}}(s,a) - V_\psi(s)\right)\right] \\ &= \begin{cases} \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\tau_{\text{proj}}(a \mid s)\left(Q_{\hat{\phi}}(s,a) - V_\psi(s)\right)^2\right], & \text{if } Q_{\hat{\phi}}(s,a) - V_\psi(s) \geq 0, \\ \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\left(1-\tau_{\text{proj}}(a \mid s)\right)\left(Q_{\hat{\phi}}(s,a) - V_\psi(s)\right)^2\right], & \text{if } Q_{\hat{\phi}}(s,a) - V_\psi(s) < 0. \end{cases} \end{aligned} \tag{7.1}$$

① For $Q_{\hat{\phi}}(s,a) - V_\psi(s) \geq 0$,

$$\begin{aligned} L_V^{Q\geq V}(\psi) &= \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\tau_{\text{proj}}(a \mid s)\left(Q_{\hat{\phi}}(s,a) - V_\psi(s)\right)^2\right] \\ &= \mathbb{E}_{s\sim\mathcal{D},\, a\sim\pi_\beta(\cdot|s)}\left[\frac{\vec{\pi}_\beta(a \mid s)\cdot\vec{\pi}_\theta(a \mid s)}{\|\vec{\pi}_\theta(a \mid s)\|^2}\pi_\theta(a \mid s)\left(Q_{\hat{\phi}}(s,a) - V_\psi(s)\right)^2\right] \\ &= \mathbb{E}_{s\sim\mathcal{D},\, a\sim\pi_\theta(\cdot|s)}\left[\frac{\vec{\pi}_\beta(a \mid s)\cdot\vec{\pi}_\theta(a \mid s)}{\|\vec{\pi}_\theta(a \mid s)\|^2}\pi_\beta(a \mid s)\left(Q_{\hat{\phi}}(s,a) - V_\psi(s)\right)^2\right]. \end{aligned} \tag{7.2}$$

② For $Q_{\hat{\phi}}(s,a) - V_\psi(s) < 0$,

$$\begin{aligned} L_V^{Q<V}(\psi) &= \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\left(1-\tau_{\text{proj}}(a \mid s)\right)\left(Q_{\hat{\phi}}(s,a) - V_\psi(s)\right)^2\right] \\ &= \mathbb{E}_{s\sim\mathcal{D},\, a\sim\pi_\beta(\cdot|s)}\left[\left(1 - \frac{\vec{\pi}_\beta(a \mid s)\cdot\vec{\pi}_\theta(a \mid s)}{\|\vec{\pi}_\theta(a \mid s)\|^2}\pi_\theta(a \mid s)\right)\left(Q_{\hat{\phi}}(s,a) - V_\psi(s)\right)^2\right] \\ &= \mathbb{E}_{s\sim\mathcal{D},\, a\sim\pi_\beta(\cdot|s)}\left[\left(Q_{\hat{\phi}}(s,a) - V_\psi(s)\right)^2\right] - L_V^{Q\geq V}(\psi). \end{aligned} \tag{7.3}$$

Next, consider

$$
\begin{aligned}
&\left|\mathbb{E}_{s\sim\mathcal{D},a\sim\pi_\theta(\cdot|s)}\left[\left(Q_{\hat{\phi}}(s,a)-V_\psi(s)\right)^2\right]-\mathbb{E}_{s\sim\mathcal{D},a\sim\pi_\beta(\cdot|s)}\left[\left(Q_{\hat{\phi}}(s,a)-V_\psi(s)\right)^2\right]\right| \\
&=\left|\mathbb{E}_{s\sim\mathcal{D}}\left[\int\left(\pi_\theta(a\mid s)-\pi_\beta(a\mid s)\right)\left(Q_{\hat{\phi}}(s,a)-V_\psi(s)\right)^2 da\right]\right| \\
&\leq \mathbb{E}_{s\sim\mathcal{D}}\left[\int\left|\pi_\theta(a\mid s)-\pi_\beta(a\mid s)\right|\left(Q_{\hat{\phi}}(s,a)-V_\psi(s)\right)^2 da\right] \\
&\leq \mathbb{E}_{s\sim\mathcal{D}}\left[\int D_{\mathrm{TV}}\left(\pi_\theta(\cdot\mid s)\,\|\,\pi_\beta(\cdot\mid s)\right)\left(Q_{\hat{\phi}}(s,a)-V_\psi(s)\right)^2 da\right] \\
&\leq \frac{1}{\sqrt{2}}\mathbb{E}_{s\sim\mathcal{D}}\left[\int\sqrt{D_{\mathrm{KL}}\left(\pi_\theta(\cdot\mid s)\,\|\,\pi_\beta(\cdot\mid s)\right)}\left(Q_{\hat{\phi}}(s,a)-V_\psi(s)\right)^2 da\right] \qquad \text{(Pinsker's inequality)} \\
&\leq \sqrt{\frac{\epsilon}{2}}\mathbb{E}_{s\sim\mathcal{D}}\left[\int\left(Q_{\hat{\phi}}(s,a)-V_\psi(s)\right)^2 da\right].
\end{aligned}
\tag{7.4}
$$

Under the assumption that the ϵ in Eq. (3.9) is sufficiently small, Eq. (7.4) implies that the difference between

$$\mathbb{E}_{s\sim\mathcal{D},a\sim\pi_\theta(\cdot|s)}\left[\left(Q_{\hat{\phi}}(s,a)-V_\psi(s)\right)^2\right]$$

and

$$\mathbb{E}_{s\sim\mathcal{D},a\sim\pi_\beta(\cdot|s)}\left[\left(Q_{\hat{\phi}}(s,a)-V_\psi(s)\right)^2\right]$$

is sufficiently small.

Therefore, for $Q_{\hat{\phi}}(s,a)-V_\psi(s)<0$, Eq. (7.3) can be rewritten approximately as

$$
\begin{aligned}
L_V^{Q<V}(\psi) &= \mathbb{E}_{s\sim\mathcal{D},a\sim\pi_\beta(\cdot|s)}\left[\left(Q_{\hat{\phi}}(s,a)-V_\psi(s)\right)^2\right]-L_V^{Q\geq V}(\psi) \\
&\approx \mathbb{E}_{s\sim\mathcal{D},a\sim\pi_\theta(\cdot|s)}\left[\left(Q_{\hat{\phi}}(s,a)-V_\psi(s)\right)^2\right]-L_V^{Q\geq V}(\psi) \\
&= \mathbb{E}_{s\sim\mathcal{D},a\sim\pi_\theta(\cdot|s)}\left[\left(1-\frac{\vec{\pi}_\beta(a\mid s)\cdot\vec{\pi}_\theta(a\mid s)}{\|\vec{\pi}_\theta(a\mid s)\|^2}\pi_\beta(a\mid s)\right)\left(Q_{\hat{\phi}}(s,a)-V_\psi(s)\right)^2\right].
\end{aligned}
\tag{7.5}
$$

Combining Eqs. (7.2) and (7.5), we obtain

$$L_V(\psi)=\mathbb{E}_{s\sim\mathcal{D},a\sim\pi_\theta(\cdot|s)}\left[L_2^{\bar{\tau}_{\mathrm{proj}}(a|s)}\left(Q_{\hat{\phi}}(s,a)-V_\psi(s)\right)\right]. \tag{7.6}$$

where

$$\bar{\tau}_{\mathrm{proj}}(a\mid s)=\frac{\vec{\pi}_\beta(a\mid s)\cdot\vec{\pi}_\theta(a\mid s)}{\|\vec{\pi}_\theta(a\mid s)\|^2}\pi_\beta(a\mid s).$$

□

7.1.3 Proof of Lemma 3.3

Lemma 3.3 *If* $D_{\mathrm{KL}}\left[\pi_\theta(\cdot \mid s) \,\|\, \pi_\beta(\cdot \mid s)\right] \leq \epsilon, \forall s$, *then the performance* $\eta(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$ *satisfies*

$$\eta(\pi_\theta) \leq \eta(\pi_\beta) + \frac{V_{\max}}{\sqrt{2}(1-\gamma)}\sqrt{\epsilon},$$

where

$$0 \leq Q^\pi, V^\pi \leq \frac{R_{\max}}{1-\gamma} =: V_{\max}.$$

Proof. The proof follows the *Lemma 3.1* proof in [13]. □

7.1.4 Proof of Theorem 3.4

Theorem 3.4 *Assume that the Q-function is exact and that*

$$\tau_{k+1}(a \mid s) \geq \tau_k(a \mid s),$$

where, for simplicity, $\tau_k(a \mid s)$ *denotes the value of* $\tau_{\mathrm{proj}}(a \mid s)$ *at iteration* k. *Then the sequence of policies* $\{\pi_k\}$ *generated by PIQL satisfies*

$$Q^{\pi_{k+1}}_{\tau_{k+1}}(s,a) \geq Q^{\pi_k}_{\tau_k}(s,a), \qquad \forall s,a.$$

Proof. Under the policy-improvement objective in Eq. (3.15), the policy $\pi_\theta(a \mid s)$ is a parametric approximation of

$$\pi_{k+1}(a \mid s) = \frac{1}{Z_k(s)}\pi_k(a \mid s)\exp\left(\frac{A^{\pi_k}_{\tau_k}(s,a)}{\lambda}\right), \tag{7.7}$$

where

$$Z_k(s) = \int_a \pi_k(a \mid s)\exp\left(\frac{A^{\pi_k}_{\tau_k}(s,a)}{\lambda}\right) da,$$

and

$$A^{\pi_k}_{\tau_k}(s,a) = Q^{\pi_k}_{\tau_k}(s,a) - V^{\pi_k}_{\tau_k}(s).$$

Based on wBC-style policy extraction, Eq. (7.7) is the optimal solution to

$$\begin{aligned} \pi_{k+1} = \arg\max_{\pi\in\Pi} & \mathbb{E}_{a\sim\pi(\cdot|s)}\left[Q^{\pi_k}_{\tau_k}(s,a)\right], \\ \text{s.t.} \quad & D_{\mathrm{KL}}\left[\pi(\cdot \mid s) \,\|\, \pi_k(\cdot \mid s)\right] \leq \epsilon, \\ & \int_a \pi(a \mid s)\, da = 1. \end{aligned} \tag{7.8}$$

Since

$$D_{\mathrm{KL}}\left[\pi_k(\cdot \mid s) \,\|\, \pi_k(\cdot \mid s)\right] = 0 < \epsilon, \qquad \forall s,$$

π_k is a strictly feasible solution. Therefore,

$$\mathbb{E}_{a\sim\pi_{k+1}(\cdot|s)}\left[Q_{\tau_k}^{\pi_k}(s,a)\right] \geq \mathbb{E}_{a\sim\pi_k(\cdot|s)}\left[Q_{\tau_k}^{\pi_k}(s,a)\right], \qquad \forall s.$$

This implies

$$\begin{aligned}
Q_{\tau_k}^{\pi_k}(s,a) &= r(s,a) + \gamma \mathbb{E}_{s_{t+1}\sim\mathcal{T}(\cdot|s_t,a_t)} \mathbb{E}_{a\sim\pi_k}^{\tau_k}\left[Q_{\tau_k}^{\pi_k}(s_{t+1},a_{t+1}) \mid s_t = s, a_t = a\right] \\
&\leq r(s,a) + \gamma \mathbb{E}_{s_{t+1}\sim\mathcal{T}(\cdot|s_t,a_t)} \mathbb{E}_{a\sim\pi_{k+1}}^{\tau_k}\left[Q_{\tau_k}^{\pi_k}(s_{t+1},a_{t+1}) \mid s_t = s, a_t = a\right] \\
&\leq \cdots \leq \mathbb{E}_{a\sim\pi_{k+1}}^{\tau_k}\left[\sum_{n=0}^{\infty} \gamma^n r(s_{t+n},a_{t+n}) \mid s_t = s, a_t = a\right] \\
&= Q_{\tau_k}^{\pi_{k+1}}(s,a).
\end{aligned}$$

Hence,

$$Q_{\tau_k}^{\pi_{k+1}}(s,a) \geq Q_{\tau_k}^{\pi_k}(s,a), \qquad \forall s,a.$$

Next, rewrite $Q_{\tau_k}^{\pi_{k+1}}(s,a)$ as

$$\begin{aligned}
Q_{\tau_k}^{\pi_{k+1}}(s,a) &= r(s,a) + \gamma \mathbb{E}_{s_{t+1}\sim\mathcal{T}(\cdot|s_t,a_t)}\left[V_{\tau_k}^{\pi_{k+1}}(s_{t+1}) \mid s_t = s, a_t = a\right] \\
&\leq r(s,a) + \gamma \mathbb{E}_{s_{t+1}\sim\mathcal{T}(\cdot|s_t,a_t)}\left[V_{\tau_{k+1}}^{\pi_{k+1}}(s_{t+1}) \mid s_t = s, a_t = a\right] \qquad \text{(by Lemma 3.1)} \\
&= Q_{\tau_{k+1}}^{\pi_{k+1}}(s,a).
\end{aligned}$$

Therefore,

$$Q_{\tau_{k+1}}^{\pi_{k+1}}(s,a) \geq Q_{\tau_k}^{\pi_{k+1}}(s,a) \geq Q_{\tau_k}^{\pi_k}(s,a), \qquad \forall s,a.$$

□

7.1.5 Proof of Theorem 3.5

Theorem 3.5 (Expectation Rigorous Criterion for Advantageous Actions) *For any $s \in \mathcal{D}$ and $a \sim \pi_{k+1}(\cdot \mid s)$, if*

$$0.5 \leq \tau_k(a \mid s),\ \tau_{k+1}(a \mid s) \leq 1,$$

then

$$\mathbb{E}_{a\sim\pi_{k+1}(\cdot|s)}\left[Q_{\tau_{k+1}}^{\pi_{k+1}}(s,a) - V_{\tau_{k+1}}^{\pi_{k+1}}(s)\right] \leq \mathbb{E}_{a\sim\pi_{k+1}(\cdot|s)}\left[Q_{\tau_k}^{\pi_k}(s,a) - V_{\tau_k}^{\pi_k}(s)\right].$$

Proof. For the left-hand side,

$$\begin{aligned}
&\mathbb{E}_{a\sim\pi_{k+1}(\cdot|s)}\left[Q_{\tau_{k+1}}^{\pi_{k+1}}(s,a) - V_{\tau_{k+1}}^{\pi_{k+1}}(s)\right] \\
&= \mathbb{E}_{a\sim\pi_{k+1}(\cdot|s)}\left[Q_{\tau_{k+1}}^{\pi_{k+1}}(s,a)\right] - V_{\tau_{k+1}}^{\pi_{k+1}}(s) \\
&= V^{\pi_{k+1}}(s) - V_{\tau_{k+1}}^{\pi_{k+1}}(s) \\
&\leq 0 \qquad \text{(by Lemma 3.1).}
\end{aligned} \tag{7.9}$$

For the right-hand side,

$$
\begin{aligned}
&\mathbb{E}_{a\sim\pi_{k+1}(\cdot|s)}\left[Q_{\tau_k}^{\pi_k}(s,a)-V_{\tau_k}^{\pi_k}(s)\right]\\
&=\mathbb{E}_{a\sim\pi_{k+1}(\cdot|s)}\left[Q_{\tau_k}^{\pi_k}(s,a)\right]-V_{\tau_k}^{\pi_k}(s)\\
&=\mathbb{E}_{a\sim\pi_k(\cdot|s)}^{\tau\to 1}\left[Q_{\tau_k}^{\pi_k}(s,a)\right]-V_{\tau_k}^{\pi_k}(s) \qquad \text{(by Eq. (7.8) and Theorem 3.4)}\\
&=V_{\tau\to 1}^{\pi_k}(s)-V_{\tau_k}^{\pi_k}(s)\\
&\geq 0 \qquad \text{(by Lemma 3.1).}
\end{aligned}
\tag{7.10}
$$

Therefore, Theorem 3.5 holds. □

7.1.6 Proof of Lemma 3.6

Lemma 3.6 *For any random variable X, if $0.5 \leq \tau_1 \leq \tau_2 \leq 1$, then*

$$\mathrm{Var}^{\tau_1}(X) \leq \mathrm{Var}^{\tau_2}(X),$$

where

$$\mathrm{Var}^{\tau}(X) = \mathbb{E}\left[\left(X-\mathbb{E}^{\tau}[X]\right)^2\right].$$

Proof. We calculate

$$
\begin{aligned}
&\mathrm{Var}^{\tau_1}(X)-\mathrm{Var}^{\tau_2}(X)\\
&=\mathbb{E}(X^2)-2\mathbb{E}(X)\mathbb{E}^{\tau_1}(X)+\left(\mathbb{E}^{\tau_1}(X)\right)^2\\
&\quad-\mathbb{E}(X^2)+2\mathbb{E}(X)\mathbb{E}^{\tau_2}(X)-\left(\mathbb{E}^{\tau_2}(X)\right)^2\\
&=\underbrace{\left[\mathbb{E}^{\tau_1}(X)+\mathbb{E}^{\tau_2}(X)-2\mathbb{E}(X)\right]}_{\geq 0}\underbrace{\left[\mathbb{E}^{\tau_1}(X)-\mathbb{E}^{\tau_2}(X)\right]}_{\leq 0} \qquad \text{(by Lemma 3.1)}\\
&\leq 0.
\end{aligned}
\tag{7.11}
$$

Hence,

$$\mathrm{Var}^{\tau_1}(X) \leq \mathrm{Var}^{\tau_2}(X).$$

□

7.1.7 Proof of Theorem 3.7

Theorem 3.7 (Maximum Rigorous Criterion for Advantageous Actions) *For $\forall s \in \mathcal{D}$, if $0.5 \leq \tau_k(a|s) \leq \tau_{k+1}(a|s) \leq 1$, we have*

$$\max_{a\sim\pi_1(\cdot|s)} P\left\{Q_{\tau_{k+1}}^{\pi_{k+1}}(s,a)-V_{\tau_{k+1}}^{\pi_{k+1}}(s)\geq 0\right\} \leq \max_{a\sim\pi_2(\cdot|s)} P\left\{Q_{\tau_k}^{\pi_k}(s,a)-V_{\tau_k}^{\pi_k}(s)\geq 0\right\},$$

where $\pi_1(\cdot|s)$ and $\pi_2(\cdot|s)$ are arbitrary policies.

Proof. First define

$$\mathbb{E}_{a\sim\pi_{k+1}}^{\tau_{k+1}}\left[Q_{\tau_{k+1}}^{\pi_{k+1}}(s,a)\right]-\mathbb{E}_{a\sim\pi_{k+1}}\left[Q_{\tau_{k+1}}^{\pi_{k+1}}(s,a)\right]=\epsilon_1, \tag{7.12}$$

and

$$\mathbb{E}_{a\sim\pi_k}^{\tau_k}\left[Q_{\tau_k}^{\pi_k}(s,a)\right]-\mathbb{E}_{a\sim\pi_k}\left[Q_{\tau_k}^{\pi_k}(s,a)\right]=\epsilon_2. \tag{7.13}$$

Since $0.5\le\tau_k(a\mid s)\le\tau_{k+1}(a\mid s)\le 1$, Lemma 3.1 implies that $\epsilon_1,\epsilon_2\ge 0$.

Next, for any random variable X,

$$\begin{aligned}\mathrm{Var}^{\tau}(X)-\mathrm{Var}(X)&=\mathbb{E}(X)^2-2\mathbb{E}(X)\mathbb{E}^{\tau}(X)+\left(\mathbb{E}^{\tau}(X)\right)^2\\&=\left[\mathbb{E}(X)-\mathbb{E}^{\tau}(X)\right]^2\\&\ge 0.\end{aligned} \tag{7.14}$$

By Eqs. (7.12) and (7.13), this gives

$$\mathrm{Var}^{\tau_{k+1}}\left[Q_{\tau_{k+1}}^{\pi_{k+1}}(s,a)\right]-\mathrm{Var}\left[Q_{\tau_{k+1}}^{\pi_{k+1}}(s,a)\right]=\epsilon_1^2, \tag{7.15}$$

and

$$\mathrm{Var}^{\tau_k}\left[Q_{\tau_k}^{\pi_k}(s,a)\right]-\mathrm{Var}\left[Q_{\tau_k}^{\pi_k}(s,a)\right]=\epsilon_2^2. \tag{7.16}$$

Now apply Cantelli's inequality (one-sided Chebyshev inequality):

$$\begin{aligned}&P\left\{Q_{\tau_{k+1}}^{\pi_{k+1}}(s,a)-V_{\tau_{k+1}}^{\pi_{k+1}}(s)\ge 0\right\}\\&=P\left\{Q_{\tau_{k+1}}^{\pi_{k+1}}(s,a)-\mathbb{E}_{a\sim\pi_{k+1}}^{\tau_{k+1}}\left[Q_{\tau_{k+1}}^{\pi_{k+1}}(s,a)\right]\ge 0\right\}\\&=P\left\{Q_{\tau_{k+1}}^{\pi_{k+1}}(s,a)-\mathbb{E}_{a\sim\pi_{k+1}}\left[Q_{\tau_{k+1}}^{\pi_{k+1}}(s,a)\right]\ge\epsilon_1\right\}\qquad\text{(by Eq. (7.12))}\\&\le\frac{\mathrm{Var}\left[Q_{\tau_{k+1}}^{\pi_{k+1}}(s,a)\right]}{\mathrm{Var}\left[Q_{\tau_{k+1}}^{\pi_{k+1}}(s,a)\right]+\epsilon_1^2}\qquad\text{(Cantelli's inequality)}\\&=\frac{\mathrm{Var}\left[Q_{\tau_{k+1}}^{\pi_{k+1}}(s,a)\right]}{\mathrm{Var}^{\tau_{k+1}}\left[Q_{\tau_{k+1}}^{\pi_{k+1}}(s,a)\right]}.\end{aligned} \tag{7.17}$$

Similarly,

$$P\left\{Q_{\tau_k}^{\pi_k}(s,a)-V_{\tau_k}^{\pi_k}(s)\ge 0\right\}\le\frac{\mathrm{Var}\left[Q_{\tau_k}^{\pi_k}(s,a)\right]}{\mathrm{Var}^{\tau_k}\left[Q_{\tau_k}^{\pi_k}(s,a)\right]}. \tag{7.18}$$

Since

$$0.5\le\tau_k(a\mid s)\le\tau_{k+1}(a\mid s)\le 1,$$

Lemma 3.6 implies

$$\mathrm{Var}^{\tau_{k+1}}(X)\ge\mathrm{Var}^{\tau_k}(X)\ge\mathrm{Var}(X)\ge 0.$$

Combining Eqs. (7.17) and (7.18), Theorem 3.7 follows. □

7.2 Experimental Details – PIQL

The vast majority of the hyperparameters in our experiments are consistent with those used in IQL, as summarized in Tab. 7.1.

For the experiments corresponding to Fig. 3.1. These experiments are conducted for the IQL baseline rather than PIQL. We vary only the expectile parameter τ, setting it to 0.3, 0.6, and 0.9. All other hyperparameters are kept the same as those reported in Tab. 7.1.

For the experiments corresponding to Fig. 3.2. We use the hyperparameter settings listed in Tab. 7.1. In addition, following the recommendation of the dataset authors, we subtract 1 from the rewards on the AntMaze-v0 datasets.

For the experiments corresponding to Fig. 3.3. We adopt the hyperparameter settings in Tab. 7.1, except that the batch size is varied over 16, 64, 128, and 256.

For the experiments corresponding to Tab. 3.1, 3.2, and 3.3. We use the hyperparameter settings reported in Tab. 7.1.

Table 7.1: Hyperparameters in PIQL

Hyperparameter	Value	Environment / Network
Hidden Layers	2	All
Layer Width	256	All
Activations	ReLU	All
Learning Rate	3e-4	All
Soft Update	5e-3	All
Training Gradient Steps	1e6	Gym-MuJoCo-v2 & AntMaze-v0 tasks
	2e5	Kitchen-complete-v0 task
	5e5	Kitchen-partial-v0 & Kitchen-mixed-v0 tasks
Evaluation Epochs	10	All
Evaluation Frequency	5e3	All
Random Seed	[0, 2, 4, 6, 8]	All
Batch Size	256	All
Inverse Temperature $\frac{1}{\lambda}$	10.0	AntMaze-v0
	3.0	Gym-MuJoCo-v2
	0.5	Kitchen-v0
Behavior Policy Training Step	1e5	All
τ	0.9	AntMaze-v0
	0.7	Gym-MuJoCo-v2
	0.7	Kitchen-v0
Exponentiated Advantage	$(-\infty, 100]$	All
Dropout	0	All Q networks and target-Q networks
	0	All value networks
	0	Policy networks for AntMaze-v0 and Gym-MuJoCo-v2 tasks
	0.1	Policy networks for Kitchen-v0 tasks

7.3 Proofs – ACPO

7.3.1 Continuous Constraint Interpolation Framework

We start from the following constrained maximum-entropy policy optimization problem,

$$\begin{aligned} \max_{\pi} \mathbb{E}_{s\sim\mathcal{D},a\sim\pi(\cdot|s)}\big[Q(s,a) - \alpha \log \pi(a|s)\big], \\ \text{s.t. } \mathbb{E}_{s\sim\mathcal{D},a\sim\pi(\cdot|s)}\big[\log \pi_\beta(a|s)\big] \geq \epsilon, \\ \int_a \pi(a|s)\, da = 1,\ \forall s. \end{aligned} \tag{7.19}$$

Solving the above constrained problem Eq. (7.19) via the Lagrangian,

$$\begin{aligned} \mathcal{L}(\pi,\lambda,\nu) =& \mathbb{E}_{s\sim\mathcal{D},a\sim\pi(\cdot|s)}\left[Q(s,a) - \alpha \log \pi(a|s)\right] + \lambda\left[\mathbb{E}_{s\sim\mathcal{D},a\sim\pi(\cdot|s)}\left[\log \pi_\beta(a|s) - \epsilon\right]\right] \\ &+ \mathbb{E}_{s\sim\mathcal{D}}\left[\nu(s)\big[\int_a \pi(a|s)\, da - 1\big]\right], \end{aligned} \tag{7.20}$$

To obtain the optimal non-parametric policy, we take the functional derivative of $\mathcal{L}(\pi,\lambda,\nu)$ w.r.t. $\pi(\cdot|s)$, and setting it to zero yields:

$$\begin{aligned} 0 &= \frac{\partial \mathcal{L}(\pi,\lambda,\nu)}{\partial \pi} \\ &= Q(s,a) - \alpha\big(1 + \log \pi(a|s)\big) + \lambda \log \pi_\beta(a|s) + \nu(s), \end{aligned} \tag{7.21}$$

Rearranging gives

$$\log \pi(a|s) = \frac{1}{\alpha}\Big(Q(s,a) + \lambda \log \pi_\beta(a|s) + \nu(s) - \alpha\Big). \tag{7.22}$$

Exponentiating both sides yields

$$\pi^*_\lambda(a|s) = \exp\left(\frac{\nu(s) - \alpha}{\alpha}\right) \exp\left(\frac{Q(s,a)}{\alpha}\right)\ \pi_\beta(a|s)^{\lambda/\alpha}. \tag{7.23}$$

Since the prefactor $\exp((\nu(s) - \alpha)/\alpha)$ does not depend on a, it can be absorbed into the normalizer, giving

$$\pi^*_\lambda(a|s)\ \propto\ \exp\left(\frac{Q(s,a)}{\alpha}\right)\ \pi_\beta(a|s)^{\lambda/\alpha}. \tag{7.24}$$

Enforcing $\int \pi^*_\lambda(a|s)\, da = 1$ yields the normalized closed-form solution

$$\pi^*_\lambda(a|s) = \exp\left(\frac{Q(s,a) - Z_\lambda(s)}{\alpha}\right)\ \pi_\beta(a|s)^{\lambda/\alpha}, \tag{7.25}$$

where $\lambda \geq 0$ and $\nu(s)$ are dual variables, and $Z_\lambda(s) = \alpha \log \int_a \exp\left(\frac{Q(s,a)}{\alpha}\right) \pi_\beta(a|s)^{\lambda/\alpha}\, da$.

Similarly within InAC [64], we also replace $Z_\lambda(s)$ with value function $V(s)$. Therefore, the closed-form solution Eq. (7.25) can be transformed into the following form,

$$\pi^*_\lambda(a|s) \propto \exp\left(\frac{A(s,a)}{\alpha}\right) \pi_\beta(a|s)^{\lambda/\alpha}, \tag{7.26}$$

Then, we project the non-parametric optimal policy π^*_λ into the parametric space π_θ by minimizing the reverse KL divergence under the state distribution in dataset,

$$\begin{aligned}
\pi_\theta(a|s) &= \arg\min_{\pi_\theta} \mathbb{E}_{s\sim\mathcal{D}}\left[\, D_{\mathrm{KL}}\left[\pi^*_\lambda(\cdot|s)||\pi_\theta(\cdot|s)\right]\,\right] \\
&= \arg\min_{\theta} \mathbb{E}_{s\sim\mathcal{D}}\Big[\mathbb{E}_{a\sim\pi^*_\lambda(\cdot|s)}\left[\log \pi^*_\lambda(a|s) - \log \pi_\theta(a|s)\right]\Big] \\
&= \arg\min_{\theta} \mathbb{E}_{s\sim\mathcal{D}}\Big[\mathbb{E}_{a\sim\pi_\beta(\cdot|s)}\Big[-\exp\left(\frac{A(s,a)}{\alpha}\right)\pi_\beta(a|s)^{(\lambda-\alpha)/\alpha}\log \pi_\theta(a|s)\Big]\Big] \\
&= \arg\max_{\theta} \mathbb{E}_{s\sim\mathcal{D}}\Big[\mathbb{E}_{a\sim\pi_\beta(\cdot|s)}\Big[\exp\left(\frac{A(s,a)}{\alpha} + \frac{\lambda-\alpha}{\alpha}\log \pi_\beta(a|s)\right)\log \pi_\theta(a|s)\Big]\Big].
\end{aligned} \tag{7.27}$$

From Eq. (7.27), the policy improvement objective can be written in the dataset form

$$\max_\theta\ \mathcal{J}^{\mathrm{CCI}}(\theta;\lambda) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\Big[w_\lambda(s,a)\ \log \pi_\theta(a|s)\Big], \tag{7.28}$$

where

$$w_\lambda(s,a) = \exp\left(\frac{A(s,a)}{\alpha} + \frac{\lambda-\alpha}{\alpha}\log \pi_\beta(a|s)\right).$$

This yields the CCI policy improvement objective Eq. (4.14).

Varying λ continuously interpolates the influence of the weight term $w_\lambda(s,a)$, yielding several representative regimes:

① **Support Constraint** $\lambda = 0$**.** When $\lambda = 0$, the objective becomes

$$\max_\theta\ \mathcal{J}^{\mathrm{CCI}}(\theta;0) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\Big[\exp\left(\frac{A(s,a)}{\alpha} - \log \pi_\beta(a|s)\right)\log \pi_\theta(a|s)\Big], \tag{7.29}$$

which matches policy improvement objective used by InAC [64].

② **Support-to-Density Constraint** $0 < \lambda < \alpha$**.** When $0 < \lambda < \alpha$, the objective becomes

$$\max_\theta\ \mathcal{J}^{\mathrm{CCI}}(\theta;\lambda) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\Big[\exp\left(\frac{A(s,a)}{\alpha} - \frac{\alpha-\lambda}{\alpha}\log \pi_\beta(a|s)\right)\log \pi_\theta(a|s)\Big], \tag{7.30}$$

As λ increases toward α, the update gradually shifts from support constraint toward KL-based density constraint.

③ **KL-based Density Constraint $\lambda = \alpha$.** When $\lambda = \alpha$, the objective becomes

$$\max_{\theta} \mathcal{J}^{\mathrm{CCI}}(\theta;\alpha) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\Big[\exp\Big(\frac{A(s,a)}{\alpha}\Big)\log\pi_\theta(a|s)\Big], \tag{7.31}$$

which matches the policy improvement objective in AWAC [21].

④ **Density-to-wBC Constraint $\lambda > \alpha$.** When $\lambda > \alpha$, the objective becomes

$$\max_{\theta} \mathcal{J}^{\mathrm{CCI}}(\theta;\lambda) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\Big[\exp\Big(\underbrace{\tfrac{A(s,a)}{\alpha}}_{\text{KL-density term}} + \underbrace{\tfrac{\lambda-\alpha}{\alpha}\log\pi_\beta(a|s)}_{\text{wBC term}}\Big)\log\pi_\theta(a|s)\Big]. \tag{7.32}$$

⑤ **Practical wBC $\lambda \gg \alpha + A(s,a)(\log\pi_\beta(a|s))^{-1}$.** When $\lambda \gg \alpha + A(s,a)(\log\pi_\beta(a|s))^{-1}$, then $\frac{\lambda-\alpha}{\alpha}\log\pi_\beta(a|s) \gg \frac{A(s,a)}{\alpha}$, the objective becomes

$$\max_{\theta} \mathcal{J}^{\mathrm{CCI}}(\theta;\lambda) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\Big[\exp\big(\frac{\lambda-\alpha}{\alpha}\log\pi_\beta(a|s)\big)\log\pi_\theta(a|s)\Big]. \tag{7.33}$$

We note that the practical wBC regime corresponds to choosing

$$\lambda \gg \alpha + A(s,a)\big(\log\pi_\beta(a|s)\big)^{-1}.$$

Next, we provide an explicit upper bound on $\big|A(s,a)\,(\log\pi_\beta(a|s))^{-1}\big|$ under the maximum-entropy setting.

In practice, we stabilize training by clipping log-probabilities for any stochastic policy: $\log\pi(a|s) \in [\ell_{\min}, \ell_{\max}]$. Define

$$C_{\max} \triangleq \max\{|\ell_{\min}|, |\ell_{\max}|\} \Rightarrow |-\alpha\log\pi(a|s)| \le \alpha C_{\max},$$

In common RL reward setting, the reward function is bounded by $|r(s,a)| \le R_{\max}$. Then the per-step soft reward satisfies

$$\big|r(s,a) - \alpha\log\pi(a|s)\big| \le R_{\max} + \alpha C_{\max}.$$

By the discounted sum, we have $|Q^\pi(s,a)| \le \frac{R_{\max}+\alpha C_{\max}}{1-\gamma}$, $|V^\pi(s)| \le \frac{R_{\max}+\alpha C_{\max}}{1-\gamma}$. Therefore,

$$|A^\pi(s,a)| \le |Q^\pi(s,a)| + |V^\pi(s)| \le \frac{2\big(R_{\max}+\alpha C_{\max}\big)}{1-\gamma}. \tag{7.34}$$

It follows that

$$\left|\frac{A(s,a)}{\log\pi_\beta(a|s)}\right| \le \frac{|A(s,a)|}{C^\beta_{\min}+\delta} \le \frac{2\big(R_{\max}+\alpha C_{\max}\big)}{(1-\gamma)(C^\beta_{\min}+\delta)}. \tag{7.35}$$

Therefore, a sufficient condition for the practical wBC constraint is to choose

$$\lambda \gg \alpha + \frac{2\big(R_{\max} + \alpha C_{\max}\big)}{(1-\gamma)(C^{\beta}_{\min} + \delta)},$$

where $C^{\beta}_{\min} \triangleq \min |\log \pi_\beta(a|s)|$, and δ is a small constant introduced to avoid division by zero.

Summarizing the above, we can obtain Tab. 4.1.

In addition, SAC [139] is another extreme initiation of the CCI framework, minimizing the forward KL divergence $D_{\mathrm{KL}}\left[\pi_\theta(\cdot|s)||\pi^*_\lambda(\cdot|s)\right]$, and setting $\lambda = 0$, then the policy improvement objective is

$$\max_\theta \ \mathcal{J}^{\mathrm{SAC}}(\theta; 0) = \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\big[\exp\big(\frac{A(s,a)}{\alpha} - \log \pi_\theta(a|s)\big)\big], \tag{7.36}$$

which matches the policy improvement objective in SAC.

7.3.2 Proof for Proposition 4.1

Proposition 4.1 *Fix any state s and temperature $\alpha > 0$. Let $\pi^*_\lambda(\cdot \mid s)$ denote the closed-form optimizer in Eq. (4.12) for a given $\lambda \geq 0$. Define*

$$g_s(\lambda) := \mathbb{E}_{a\sim\pi^*_\lambda(\cdot|s)}\big[\log \pi_\beta(a \mid s)\big]. \tag{7.37}$$

Then $g_s(\lambda)$ is monotone non-decreasing in λ, and

$$\frac{d}{d\lambda} g_s(\lambda) = \frac{1}{\alpha}\mathrm{Var}_{a\sim\pi^*_\lambda(\cdot|s)}\big(\log \pi_\beta(a \mid s)\big) \geq 0. \tag{7.38}$$

Consequently, the dataset-averaged quantity

$$g(\lambda) := \mathbb{E}_{s\sim\mathcal{D}}[g_s(\lambda)]$$

also satisfies $g'(\lambda) \geq 0$.

Proof. Fix s and abbreviate $Q(a) := Q(s,a)$ and $\pi_\beta(a) := \pi_\beta(a|s)$. Define the unnormalized density and the partition function

$$w_\lambda(a) := \exp\left(\frac{Q(a)}{\alpha}\right) \pi_\beta(a)^{\lambda/\alpha}, \qquad \mathcal{Z}(\lambda) := \int w_\lambda(a)\, da, \tag{7.39}$$

so that $\pi^*_\lambda(a|s) = w_\lambda(a)/\mathcal{Z}(\lambda)$. By definition,

$$g_s(\lambda) = \int \pi^*_\lambda(a|s)\, \log \pi_\beta(a)\, da. \tag{7.40}$$

Differentiating Eq. (7.40) w.r.t. λ and exchanging derivative and integral,

$$\frac{d}{d\lambda} g_s(\lambda) = \int \frac{\partial}{\partial \lambda} \pi^*_\lambda(a|s)\ \log \pi_\beta(a)\, da. \tag{7.41}$$

Since $\pi^*_\lambda = w_\lambda / \mathcal{Z}(\lambda)$, we have

$$\frac{\partial}{\partial \lambda} \pi^*_\lambda(a|s) = \frac{w'_\lambda(a)}{\mathcal{Z}(\lambda)} - \frac{w_\lambda(a)\mathcal{Z}'(\lambda)}{\mathcal{Z}(\lambda)^2} = \pi^*_\lambda(a|s)\Big(\frac{w'_\lambda(a)}{w_\lambda(a)} - \frac{\mathcal{Z}'(\lambda)}{\mathcal{Z}(\lambda)}\Big). \tag{7.42}$$

Using $w_\lambda(a) = \exp(Q(a)/\alpha)\exp((\lambda/\alpha)\log \pi_\beta(a))$, we obtain

$$\frac{w'_\lambda(a)}{w_\lambda(a)} = \frac{1}{\alpha} \log \pi_\beta(a). \tag{7.43}$$

Moreover,

$$\begin{aligned} \mathcal{Z}'(\lambda) = \int w'_\lambda(a)\, da &= \int w_\lambda(a) \cdot \frac{1}{\alpha} \log \pi_\beta(a)\, da, \\ \Rightarrow \frac{\mathcal{Z}'(\lambda)}{\mathcal{Z}(\lambda)} &= \frac{1}{\alpha} \int \pi^*_\lambda(a|s) \log \pi_\beta(a)\, da = \frac{1}{\alpha} g_s(\lambda). \end{aligned} \tag{7.44}$$

Substituting Eqs. (7.43), (7.44) into Eq. (7.42) yields

$$\frac{\partial}{\partial \lambda} \pi^*_\lambda(a|s) = \frac{1}{\alpha} \pi^*_\lambda(a|s)\Big(\log \pi_\beta(a) - g_s(\lambda)\Big). \tag{7.45}$$

Plugging Eq. (7.45) into Eq. (7.41) gives

$$\begin{aligned} \frac{d}{d\lambda} g_s(\lambda) &= \frac{1}{\alpha} \int \pi^*_\lambda(a|s)\Big(\log \pi_\beta(a) - g_s(\lambda)\Big) \log \pi_\beta(a)\, da \\ &= \frac{1}{\alpha}\Big(\mathbb{E}_{\pi^*_\lambda}\big[(\log \pi_\beta)^2\big] - \mathbb{E}_{\pi^*_\lambda}[\log \pi_\beta]^2\Big) = \frac{1}{\alpha} \mathrm{Var}_{a \sim \pi^*_\lambda(\cdot|s)}\big(\log \pi_\beta(a)\big)\ \geq\ 0, \end{aligned}$$

which proves Eq. (7.38). The statement for $g(\lambda) = \mathbb{E}_{s\sim\mathcal{D}}[g_s(\lambda)]$ follows by linearity of expectation. □

7.3.3 Proof for Lemma 4.2

Lemma 4.2 (Maximum Entropy Performance Difference) *Let π and π_β be two stationary policies. Consider the maximum-entropy performance*

$$\mathcal{J}(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t \big(r(s_t, a_t) - \alpha \log \pi(a_t|s_t)\big)\right],$$

then the maximum entropy performance difference satisfies

$$\mathcal{J}(\pi) - \mathcal{J}(\pi_\beta) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\pi}\Big[\mathbb{E}_{a\sim\pi(\cdot|s)}\big[\tilde{A}^{\pi_\beta}(s,a)\big] - \alpha \mathrm{KL}\big(\pi(\cdot|s)\|\pi_\beta(\cdot|s)\big)\Big], \tag{7.46}$$

where $\tilde{A}^{\pi_\beta}(s,a)$ *is the advantage function under the shaped reward* $\tilde{r}(s,a) = r(s,a) - \alpha \log \pi_\beta(a|s)$.

Proof. Introduce π_β into the entropy-regularized return:

$$\begin{aligned}\mathcal{J}(\pi) &= \mathbb{E}_{\tau\sim\pi}\Big[\sum_{t=0}^{\infty}\gamma^t\big(r(s_t,a_t) - \alpha\log\pi(a_t|s_t)\big)\Big] \\ &= \mathbb{E}_{\tau\sim\pi}\Big[\sum_{t=0}^{\infty}\gamma^t\big(r(s_t,a_t) - \alpha\log\pi_\beta(a_t|s_t)\big)\Big] - \alpha\,\mathbb{E}_{\tau\sim\pi}\Big[\sum_{t=0}^{\infty}\gamma^t\log\frac{\pi(a_t|s_t)}{\pi_\beta(a_t|s_t)}\Big]. \end{aligned}\tag{7.47}$$

Define the shaped return

$$\widetilde{\mathcal{J}}(\pi) := \mathbb{E}_{\tau\sim\pi}\big[\sum_{t=0}^{\infty}\gamma^t\,\tilde{r}(s_t,a_t)\big]$$

with $\tilde{r}(s,a) = r(s,a) - \alpha\log\pi_\beta(a|s)$. Since $\log\frac{\pi_\beta(a|s)}{\pi_\beta(a|s)} = 0$, we have $\mathcal{J}(\pi_\beta) = \widetilde{\mathcal{J}}(\pi_\beta)$ and thus

$$\mathcal{J}(\pi) - \mathcal{J}(\pi_\beta) = \big(\widetilde{\mathcal{J}}(\pi) - \widetilde{\mathcal{J}}(\pi_\beta)\big) - \alpha\,\mathbb{E}_{\tau\sim\pi}\Big[\sum_{t=0}^{\infty}\gamma^t\log\frac{\pi(a_t|s_t)}{\pi_\beta(a_t|s_t)}\Big]. \tag{7.48}$$

For the first term, $\widetilde{\mathcal{J}}$ is a standard discounted return with reward $\tilde{r}$. By the performance difference theorem (e.g., Lemma 6.1 in [136]),

$$\widetilde{\mathcal{J}}(\pi) - \widetilde{\mathcal{J}}(\pi_\beta) = \frac{1}{1-\gamma}\,\mathbb{E}_{s\sim d_\pi}\,\mathbb{E}_{a\sim\pi(\cdot|s)}\big[\tilde{A}^{\pi_\beta}(s,a)\big]. \tag{7.49}$$

For the second term, using the discounted occupancy identity $d_\pi(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t\,\Pr(s_t = s \mid \pi)$, we obtain

$$\begin{aligned}\mathbb{E}_{\tau\sim\pi}\Big[\sum_{t=0}^{\infty}\gamma^t\log\frac{\pi(a_t|s_t)}{\pi_\beta(a_t|s_t)}\Big] &= \frac{1}{1-\gamma}\,\mathbb{E}_{s\sim d_\pi}\,\mathbb{E}_{a\sim\pi(\cdot|s)}\Big[\log\frac{\pi(a|s)}{\pi_\beta(a|s)}\Big] \\ &= \frac{1}{1-\gamma}\,\mathbb{E}_{s\sim d_\pi}\Big[D_{\mathrm{KL}}\big(\pi(\cdot|s)\,\|\,\pi_\beta(\cdot|s)\big)\Big]. \end{aligned}\tag{7.50}$$

Substituting Eq. (7.49) and Eq. (7.50) into Eq. (7.48) yields Eq. (7.46). □

7.3.4 Proof for Theorem 4.3

Theorem 4.3 (Performance Lower Bound for the Optimal Policy) *Define* $\epsilon_\beta(\pi) := \max_s \left|\mathbb{E}_{a\sim\pi(\cdot|s)}\left[\tilde{A}^{\pi_\beta}(s,a)\right]\right|$, $\kappa_\beta(\pi) := \max_s \mathrm{KL}\big(\pi(\cdot|s)\|\pi_\beta(\cdot|s)\big)$ *and the average total variation distance under* ν *by* $\Delta_{\mathrm{TV}}(\pi,\pi_\beta;\nu) := \mathbb{E}_{s\sim\nu}\big[D_{\mathrm{TV}}(\pi(\cdot|s),\pi_\beta(\cdot|s))\big]$. *Assume* $|\log\pi_\beta(a|s)| \le B$ *for all* (s,a). *For a given* $\lambda \ge 0$ *and* $\alpha > 0$, π^*_λ *satisfies the following maximum-entropy performance lower bound:*

$$\begin{aligned}\mathcal{J}(\pi^*_\lambda) \ge \mathcal{J}(\pi_\beta) &- \frac{2B|\lambda-\alpha|}{1-\gamma}\Delta_{\mathrm{TV}}(\pi^*_\lambda,\pi_\beta;d_{\pi_\beta}) \\ &- \frac{4\alpha B + 2\gamma\Big(\epsilon_\beta(\pi^*_\lambda)+\alpha\kappa_\beta(\pi^*_\lambda)\Big)}{(1-\gamma)^2}\Delta_{\mathrm{TV}}(\pi^*_\lambda,\pi_\beta;d_{\pi_\beta}).\end{aligned} \tag{7.51}$$

Proof. By the maximum-entropy performance difference lemma (Lemma 4.2),

$$\mathcal{J}(\pi) - \mathcal{J}(\pi_\beta) = \frac{1}{1-\gamma}\,\mathbb{E}_{s\sim d_\pi}\Big[\mathbb{E}_{a\sim\pi(\cdot|s)}\left[\tilde{A}^{\pi_\beta}(s,a)\right] - \alpha\, D_{\mathrm{KL}}\big(\pi(\cdot|s)\,\|\,\pi_\beta(\cdot|s)\big)\Big]. \tag{7.52}$$

Let

$$f_\pi(s) := \mathbb{E}_{a\sim\pi(\cdot|s)}\left[\tilde{A}^{\pi_\beta}(s,a)\right] - \alpha\, D_{\mathrm{KL}}\big(\pi(\cdot|s)\,\|\,\pi_\beta(\cdot|s)\big),$$

so that the RHS of (7.52) equals $\frac{1}{1-\gamma}\mathbb{E}_{d_\pi}[f_\pi(s)]$. Decomposing around d_{π_β} yields

$$\mathbb{E}_{d_\pi}[f_\pi] = \mathbb{E}_{d_{\pi_\beta}}[f_\pi] + \Big(\mathbb{E}_{d_\pi}[f_\pi] - \mathbb{E}_{d_{\pi_\beta}}[f_\pi]\Big). \tag{7.53}$$

By definition of $\epsilon_\beta(\pi)$ and $\kappa_\beta(\pi)$, for all s we have

$$|f_\pi(s)| \le \left|\mathbb{E}_{a\sim\pi(\cdot|s)}\left[\tilde{A}^{\pi_\beta}(s,a)\right]\right| + \alpha\, D_{\mathrm{KL}}\big(\pi(\cdot|s)\|\pi_\beta(\cdot|s)\big) \le \epsilon_\beta(\pi) + \alpha\kappa_\beta(\pi),$$

hence

$$\left|\mathbb{E}_{d_\pi}[f_\pi] - \mathbb{E}_{d_{\pi_\beta}}[f_\pi]\right| \le \|d_\pi - d_{\pi_\beta}\|_1\Big(\epsilon_\beta(\pi) + \alpha\kappa_\beta(\pi)\Big). \tag{7.54}$$

Moreover, the divergence between discounted state visitation distributions (Lemma 3 in [163]) gives

$$\|d_\pi - d_{\pi_\beta}\|_1 \le \frac{2\gamma}{1-\gamma}\,\Delta_{\mathrm{TV}}(\pi,\pi_\beta;d_{\pi_\beta}). \tag{7.55}$$

Combining (7.54) and (7.55) yields

$$\mathbb{E}_{d_\pi}[f_\pi] \ge \mathbb{E}_{d_{\pi_\beta}}[f_\pi] - \frac{2\gamma}{1-\gamma}\Big(\epsilon_\beta(\pi) + \alpha\kappa_\beta(\pi)\Big)\Delta_{\mathrm{TV}}(\pi,\pi_\beta;d_{\pi_\beta}). \tag{7.56}$$

Substituting (7.56) into (7.52) and setting $\pi = \pi_\lambda^*$ yields

$$\mathcal{J}(\pi_\lambda^*) - \mathcal{J}(\pi_\beta) \geq \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_{\pi_\beta}} f_{\pi_\lambda^*}(s) - \frac{2\gamma}{(1-\gamma)^2}\Big(\epsilon_\beta(\pi_\lambda^*) + \alpha\kappa_\beta(\pi_\lambda^*)\Big)\Delta_{\mathrm{TV}}(\pi_\lambda^*, \pi_\beta; d_{\pi_\beta}). \tag{7.57}$$

For the first term,

$$\mathbb{E}_{s\sim d_{\pi_\beta}} f_{\pi_\lambda^*}(s) = \mathbb{E}_{s\sim d_{\pi_\beta}}\Big[\mathbb{E}_{a\sim\pi_\lambda^*(\cdot|s)}\tilde{A}^{\pi_\beta}(s,a) - \alpha\, D_{\mathrm{KL}}\big(\pi_\lambda^*(\cdot|s)\|\pi_\beta(\cdot|s)\big)\Big].$$

Next, define the auxiliary objective

$$F(\pi) := \mathbb{E}_{s\sim d_{\pi_\beta}}\Big[\mathbb{E}_{a\sim\pi(\cdot|s)}A^{\pi_\beta}(s,a) - \alpha\, D_{\mathrm{KL}}\big(\pi(\cdot|s)\|\pi_\beta(\cdot|s)\big) + (\lambda-\alpha)\,\mathbb{E}_{a\sim\pi(\cdot|s)}\log\pi_\beta(a|s)\Big],$$

where $A^{\pi_\beta}(s,a)$ denotes the advantage under the original reward $r(s,a)$. Since π_λ^* maximizes F in the CCI optimization problem, we have $F(\pi_\lambda^*) \geq F(\pi_\beta)$. For π_β, note that $\mathbb{E}_{a\sim\pi_\beta(\cdot|s)}[A^{\pi_\beta}(s,a)] = 0$ and $D_{\mathrm{KL}}(\pi_\beta(\cdot|s)\|\pi_\beta(\cdot|s)) = 0$, hence

$$F(\pi_\beta) = (\lambda-\alpha)\, g_{d_{\pi_\beta}}(\pi_\beta), \qquad g_{d_{\pi_\beta}}(\pi) := \mathbb{E}_{s\sim d_{\pi_\beta}, a\sim\pi(\cdot|s)}[\log\pi_\beta(a|s)].$$

Rearranging gives

$$\mathbb{E}_{s\sim d_{\pi_\beta}}\Big[\mathbb{E}_{a\sim\pi_\lambda^*(\cdot|s)}A^{\pi_\beta}(s,a) - \alpha\, D_{\mathrm{KL}}\big(\pi_\lambda^*(\cdot|s)\,\|\,\pi_\beta(\cdot|s)\big)\Big] \geq -(\lambda-\alpha)\big(g_{d_{\pi_\beta}}(\pi_\lambda^*) - g_{d_{\pi_\beta}}(\pi_\beta)\big). \tag{7.58}$$

We now relate $\tilde{A}^{\pi_\beta}$ to A^{π_β}. Let $c(s,a) := -\alpha\log\pi_\beta(a|s)$ and $\tilde{r}(s,a) = r(s,a) + c(s,a)$. By linearity of value functions in reward,

$$\tilde{A}^{\pi_\beta}(s,a) = A^{\pi_\beta}(s,a) + A_c^{\pi_\beta}(s,a),$$

where $A_c^{\pi_\beta}$ is the advantage under the reward c alone. Since $|\log\pi_\beta(a|s)| \leq B$, we have $|c(s,a)| \leq \alpha B$, which implies $\|A_c^{\pi_\beta}\|_\infty \leq \frac{2\alpha B}{1-\gamma}$. Moreover, $\mathbb{E}_{a\sim\pi_\beta(\cdot|s)}[A_c^{\pi_\beta}(s,a)] = 0$ for all s, hence for any π and s,

$$\begin{aligned}\Big|\mathbb{E}_{a\sim\pi(\cdot|s)}[A_c^{\pi_\beta}(s,a)]\Big| &= \Big|\int\big(\pi(a|s) - \pi_\beta(a|s)\big)\,A_c^{\pi_\beta}(s,a)\,da\Big| \\ &\leq 2\|A_c^{\pi_\beta}\|_\infty\, D_{\mathrm{TV}}\big(\pi(\cdot|s), \pi_\beta(\cdot|s)\big) \\ &\leq \frac{4\alpha B}{1-\gamma}\, D_{\mathrm{TV}}\big(\pi(\cdot|s), \pi_\beta(\cdot|s)\big).\end{aligned} \tag{7.59}$$

Therefore, for $\pi = \pi_\lambda^*$,

$$\mathbb{E}_{s\sim d_{\pi_\beta}}\mathbb{E}_{a\sim\pi_\lambda^*(\cdot|s)}\tilde{A}^{\pi_\beta}(s,a) \geq \mathbb{E}_{s\sim d_{\pi_\beta}}\mathbb{E}_{a\sim\pi_\lambda^*(\cdot|s)}A^{\pi_\beta}(s,a) - \frac{4\alpha B}{1-\gamma}\,\Delta_{\mathrm{TV}}(\pi_\lambda^*, \pi_\beta; d_{\pi_\beta}). \tag{7.60}$$

Combining (7.58) and (7.60) yields

$$\mathbb{E}_{s\sim d_{\pi_\beta}} f_{\pi_\lambda^*}(s) \geq -(\lambda-\alpha)\big(g_{d_{\pi_\beta}}(\pi_\lambda^*) - g_{d_{\pi_\beta}}(\pi_\beta)\big) - \frac{4\alpha B}{1-\gamma}\,\Delta_{\mathrm{TV}}(\pi_\lambda^*, \pi_\beta; d_{\pi_\beta}). \tag{7.61}$$

Finally, for each state s, define

$$\Delta g_s := \mathbb{E}_{a\sim\pi_\lambda^*(\cdot|s)}[\log \pi_\beta(a|s)] - \mathbb{E}_{a\sim\pi_\beta(\cdot|s)}[\log \pi_\beta(a|s)].$$

Then

$$|\Delta g_s| \leq \int \big|\pi_\lambda^*(a|s) - \pi_\beta(a|s)\big| \cdot |\log \pi_\beta(a|s)|\, da \leq 2B\, D_{\mathrm{TV}}\big(\pi_\lambda^*(\cdot|s), \pi_\beta(\cdot|s)\big), \tag{7.62}$$

and averaging over $s \sim d_{\pi_\beta}$ gives

$$\big|g_{d_{\pi_\beta}}(\pi_\lambda^*) - g_{d_{\pi_\beta}}(\pi_\beta)\big| \leq 2B\,\Delta_{\mathrm{TV}}(\pi_\lambda^*, \pi_\beta; d_{\pi_\beta}). \tag{7.63}$$

Substituting (7.63) into (7.61) yields

$$\mathbb{E}_{s\sim d_{\pi_\beta}} f_{\pi_\lambda^*}(s) \geq -2B|\lambda-\alpha|\,\Delta_{\mathrm{TV}}(\pi_\lambda^*, \pi_\beta; d_{\pi_\beta}) - \frac{4\alpha B}{1-\gamma}\,\Delta_{\mathrm{TV}}(\pi_\lambda^*, \pi_\beta; d_{\pi_\beta}). \tag{7.64}$$

Plugging Eq. (7.64) into Eq. (7.57) completes the proof and yields Eq. (7.51). □

7.3.5 Proof for Theorem 4.4

Theorem 4.4 (Performance Lower Bound for a Parametric Policy) *Define the suboptimality gap of a parametric policy π_θ by*

$$\delta_\theta^{\mathrm{sub}}(\lambda) := F_\lambda(\pi_\lambda^*) - F_\lambda(\pi_\theta) \;\geq\; 0, \tag{7.65}$$

where

$$F_\lambda(\pi) = \mathbb{E}_{s\sim d_{\pi_\beta}}\Big[\mathbb{E}_{a\sim\pi(\cdot|s)}\tilde{A}^{\pi_\beta}(s,a) - \alpha \mathrm{KL}\big(\pi(\cdot|s)\|\pi_\beta(\cdot|s)\big) + (\lambda-\alpha)\mathbb{E}_{a\sim\pi(\cdot|s)}\log\pi_\beta(a|s)\Big].$$

Then π_θ satisfies

$$\begin{aligned}\mathcal{J}(\pi_\theta) \geq\; \mathcal{J}(\pi_\beta) &- \frac{2B|\lambda-\alpha|}{1-\gamma}\,\Delta_{\mathrm{TV}}\big(\pi_\theta, \pi_\beta; d_{\pi_\beta}\big) - \frac{\delta_\theta^{\mathrm{sub}}(\lambda)}{1-\gamma} \\ &- \frac{4\alpha B + 2\gamma\Big(\epsilon_\beta(\pi_\theta) + \alpha\kappa_\beta(\pi_\theta)\Big)}{(1-\gamma)^2}\Delta_{\mathrm{TV}}(\pi_\theta, \pi_\beta; d_{\pi_\beta}).\end{aligned} \tag{7.66}$$

Proof. The derivation follows the same decomposition as in the proof of Theorem 4.3. Applying Eq. (7.52) with $\pi = \pi_\theta$ and using the same occupancy-shift bound yields

$$\mathcal{J}(\pi_\theta) - \mathcal{J}(\pi_\beta) \geq \frac{1}{1-\gamma}\,\mathbb{E}_{s\sim d_{\pi_\beta}} f_{\pi_\theta}(s) - \frac{2\gamma}{(1-\gamma)^2}\Big(\epsilon_\beta(\pi_\theta) + \alpha\kappa_\beta(\pi_\theta)\Big)\Delta_{\mathrm{TV}}(\pi_\theta, \pi_\beta; d_{\pi_\beta}), \tag{7.67}$$

where

$$f_\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}\big[\tilde{A}^{\pi_\beta}(s,a)\big] - \alpha\, D_{\mathrm{KL}}\big(\pi(\cdot|s)\,\|\,\pi_\beta(\cdot|s)\big).$$

We first lower bound the shaped-advantage term by the original advantage term, as in Eq. (7.60):

$$\mathbb{E}_{s\sim d_{\pi_\beta}, a\sim\pi(\cdot|s)}[\tilde{A}^{\pi_\beta}(s,a)] \geq \mathbb{E}_{s\sim d_{\pi_\beta}, a\sim\pi(\cdot|s)}[A^{\pi_\beta}(s,a)] - \frac{4\alpha B}{1-\gamma}\,\Delta_{\mathrm{TV}}(\pi, \pi_\beta; d_{\pi_\beta}). \tag{7.68}$$

Next, we convert the suboptimality gap in Eq. (7.65) into a baseline-anchored inequality. Since π_λ^* maximizes F_λ, we have $F_\lambda(\pi_\lambda^*) \geq F_\lambda(\pi_\beta)$. Combining this with Eq. (7.65) gives

$$F_\lambda(\pi_\theta) = F_\lambda(\pi_\lambda^*) - \delta_\theta^{\mathrm{sub}}(\lambda) \geq F_\lambda(\pi_\beta) - \delta_\theta^{\mathrm{sub}}(\lambda). \tag{7.69}$$

For π_β, $\mathbb{E}_{a\sim\pi_\beta}[A^{\pi_\beta}(s,a)] = 0$ and $D_{\mathrm{KL}}(\pi_\beta\|\pi_\beta) = 0$, hence $F_\lambda(\pi_\beta) = (\lambda-\alpha)\, g_{d_{\pi_\beta}}(\pi_\beta)$ where

$$g_{d_{\pi_\beta}}(\pi) := \mathbb{E}_{s\sim d_{\pi_\beta}, a\sim\pi(\cdot|s)}[\log \pi_\beta(a|s)].$$

Rearranging Eq. (7.69) yields

$$\begin{aligned}\mathbb{E}_{s\sim d_{\pi_\beta}, a\sim\pi_\theta(\cdot|s)}\big[A^{\pi_\beta}(s,a)\big] - \alpha\,\mathbb{E}_{s\sim d_{\pi_\beta}}\Big[D_{\mathrm{KL}}\big(\pi_\theta(\cdot|s)\,\|\,\pi_\beta(\cdot|s)\big)\Big] \\ \geq -(\lambda-\alpha)\Big(g_{d_{\pi_\beta}}(\pi_\theta) - g_{d_{\pi_\beta}}(\pi_\beta)\Big) - \delta_\theta^{\mathrm{sub}}(\lambda).\end{aligned} \tag{7.70}$$

Finally, we bound the density-shaping difference by total variation. For each state s, define

$$\Delta g_s := \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[\log \pi_\beta(a|s)] - \mathbb{E}_{a\sim\pi_\beta(\cdot|s)}[\log \pi_\beta(a|s)].$$

Then

$$|\Delta g_s| \leq 2B\, D_{\mathrm{TV}}\big(\pi_\theta(\cdot|s), \pi_\beta(\cdot|s)\big), \tag{7.71}$$

and averaging over $s \sim d_{\pi_\beta}$ yields

$$\big|g_{d_{\pi_\beta}}(\pi_\theta) - g_{d_{\pi_\beta}}(\pi_\beta)\big| \leq 2B\,\Delta_{\mathrm{TV}}(\pi_\theta, \pi_\beta; d_{\pi_\beta}). \tag{7.72}$$

Combining Eq. (7.68), Eq. (7.70), and Eq. (7.72) yields

$$\begin{aligned}\mathbb{E}_{s\sim d_{\pi_\beta}} f_{\pi_\theta}(s) &= \mathbb{E}_{s\sim d_{\pi_\beta}}\Big[\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\tilde{A}^{\pi_\beta}(s,a) - \alpha\, D_{\mathrm{KL}}\big(\pi_\theta(\cdot|s)\,\|\,\pi_\beta(\cdot|s)\big)\Big] \\ &\geq -2B|\lambda-\alpha|\,\Delta_{\mathrm{TV}}(\pi_\theta, \pi_\beta; d_{\pi_\beta}) - \frac{4\alpha B}{1-\gamma}\,\Delta_{\mathrm{TV}}(\pi_\theta, \pi_\beta; d_{\pi_\beta}) - \delta_\theta^{\mathrm{sub}}(\lambda).\end{aligned} \tag{7.73}$$

Plugging Eq. (7.73) into Eq. (7.67) completes the proof and yields Eq. (7.66). □

7.4 Experimental Details – ACPO

7.4.1 Implementation Details for D4RL Experiments

The vast majority of hyperparameters used in our D4RL experiments are summarized in Tab. 7.2.

Table 7.2: D4RL hyperparameters in ACPO, where s_{dim} and a_{dim} are the state and action dimensions, respectively.

Hyperparameter	Value	Task
Actor architecture	$s_{\text{dim}} - 256 - 256 - 2a_{\text{dim}}$	All
Q-network	$(s_{\text{dim}}+a_{\text{dim}}) - 256 - 256 - 1$	All
Value network	$s_{\text{dim}} - 256 - 256 - 1$	All
Behavior network	$s_{\text{dim}} - 512 - 512 - 2a_{\text{dim}}$	All
Activation	ReLU	All
Learning rate (actor & critic)	$5e-4 \rightarrow 1e-4$ (CosineAnnealingLR)	All
Learning rate (behavior policy)	$1e-4$	All
Learning rate (λ)	$1e-5$	All
Target Q soft update rate τ	$5e-3$	All
Evaluation epochs	10	Gym-MuJoCo & Kitchen
	50	AntMaze
Evaluation frequency	$5e3$	All
Batch size	256	All
Discount factor γ	0.99	All
Temperature α	0.1	All
Initial λ	0.1 ($\lambda = \alpha$)	All
ϵ in Eq. (4.10)	-5.0	Hopper
	-0.2	HalfCheetah & AntMaze
	-0.7	Walker2d
	-1.0	Kitchen

The dataset abbreviations used in Tab. 4.2 and their corresponding full names are listed in Tab. 7.3.

Training curves for all D4RL tasks are provided in Fig. 7.3.

7.4.2 Implementation Details for NeoRL2 Experiments

The vast majority of hyperparameters used in our NeoRL2 experiments are summarized in Tab. 7.4.

Training curves for all NeoRL2 tasks are provided in Fig. 7.2.

Table 7.3: Dataset abbreviations for D4RL.

Dataset abbreviation	Dataset full name
halfcheetah-med	halfcheetah-medium-v2
hopper-med	hopper-medium-v2
walker2d-med	walker2d-medium-v2
halfcheetah-med-rep	halfcheetah-medium-replay-v2
hopper-med-rep	hopper-medium-replay-v2
walker2d-med-rep	walker2d-medium-replay-v2
halfcheetah-med-exp	halfcheetah-medium-expert-v2
hopper-med-exp	hopper-medium-expert-v2
walker2d-med-exp	walker2d-medium-expert-v2
antmaze-u	antmaze-umaze-v2
antmaze-u-d	antmaze-umaze-diverse-v2
antmaze-m-p	antmaze-medium-play-v2
antmaze-m-d	antmaze-medium-diverse-v2
antmaze-l-p	antmaze-large-play-v2
antmaze-l-d	antmaze-large-diverse-v2
kitchen-m	kitchen-mixed-v0
kitchen-p	kitchen-partial-v0

7.4.3 Additional Details for Special-Case Comparisons and λ Dynamics

In Section 4.5.3, we keep all hyperparameters consistent with Tab. 7.2 and Tab. 7.4. We additionally specify the key settings in the main text: *"For Gym-MuJoCo tasks, we set* $\alpha = 0.1$. *For NeoRL2 tasks, we set* $\alpha = 0.5$. *We instantiate three special cases by choosing: (i) practical wBC:* $\lambda = 100$ *to approximate the wBC regime, (ii) AWAC:* $\lambda = \alpha$, *and (iii) InAC:* $\lambda = 0$."

7.4.4 Additional Details for Behavior Density Estimator Ablation

In the experiments of Subsection 4.5.4, we perform inference with the trained Gaussian and CVAE behavior models on the `walker2d-medium-v2` and `walker2d-medium-replay-v2` datasets, obtaining the distribution curves shown in Fig. 4.3. In addition to the case of $\xi = 0.3$, we further evaluate $\xi \in \{0.1, 0.5, 0.8\}$, and report the corresponding curves in Fig. 7.1. The conclusions remain the same: compared with the Gaussian estimator, the CVAE does not consistently assign sufficiently low likelihood to perturbed OOD actions.

One plausible explanation is that the CVAE is trained by maximizing the ELBO rather than the exact conditional likelihood. Because this objective relies on a variational approx-

Table 7.4: NeoRL2 hyperparameters. Architectures are denoted by MLP layer sizes, where s_{dim} and a_{dim} are the state and action dimensions, respectively.

Hyperparameter	Value	Task
Actor architecture	s_{dim} – 256 – 256 – $2a_{\text{dim}}$	All
Q-network	$(s_{\text{dim}}+a_{\text{dim}})$ – 256 – 256 – 1	All
Value network	s_{dim} – 256 – 256 – 1	All
Behavior network	s_{dim} – 512 – 512 – $2a_{\text{dim}}$	All
Activation	ReLU	All
Learning rate (actor & critic)	$5e-4 \rightarrow 1e-4$ (CosineAnnealingLR)	All
Learning rate (behavior policy)	$1e-4$	All
Learning rate (λ)	$1e-5$	All
Target Q soft update rate τ	$5e-3$	All
Evaluation epochs	10	All
Evaluation frequency	$5e3$	All
Batch size	256	All
Discount factor γ	0.99	All
Temperature α	0.5	All
Initial λ	0.5 ($\lambda = \alpha$)	All
ϵ in Eq. (4.10)	-5.0	Simglucose & Pipeline
	-10.0	SafetyHalfCheetah
	-10.0	RocketRecovery
	-10.0	Fusion
	-2.0	RandomFrictionHopper
	-0.5	DMSD

imation, it may yield an over-smoothed estimate of the behavior distribution, causing the model to assign non-negligible probability mass to OOD actions.

In addition, the latent-variable structure of the CVAE encourages the decoder to account for diverse behaviors through shared latent representations. This may blur the boundary between in-distribution and out-of-distribution actions and produce overly smooth interpolations across modes. As a consequence, actions that are not truly supported by the behavior data, but lie between or around high-density regions, may still be assigned relatively large likelihood estimates.

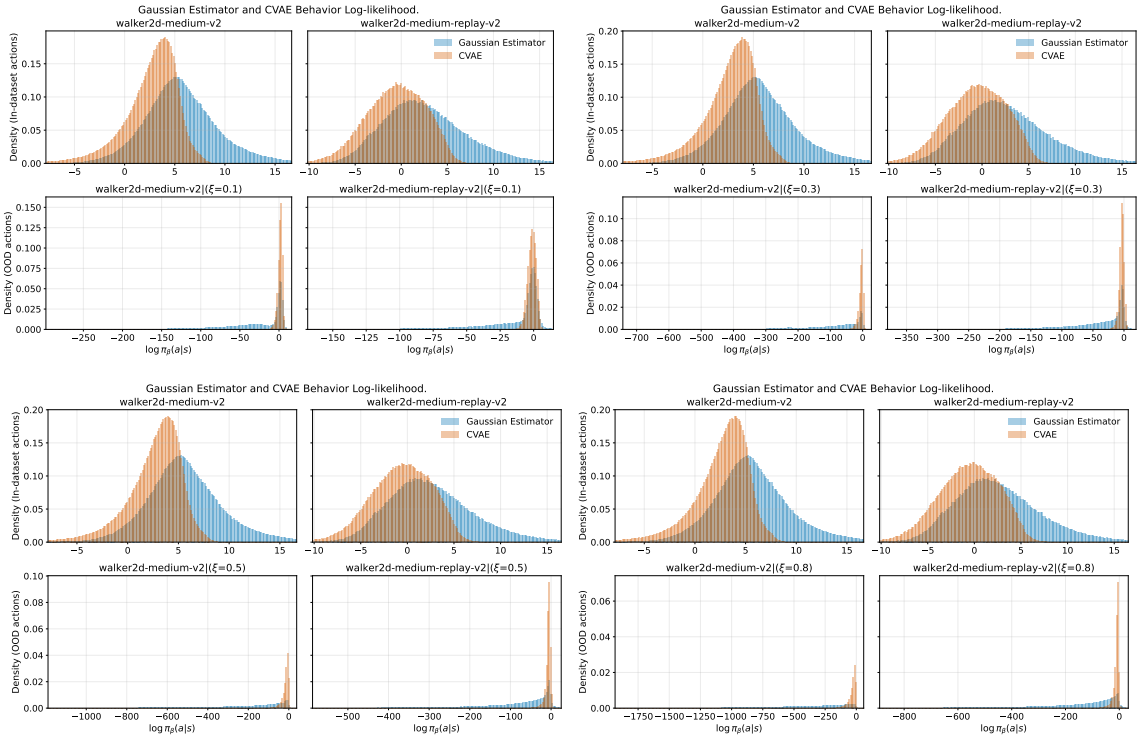


Figure 7.1: Gaussian and CVAE behavior models under $\xi \in \{0.1, 0.3, 0.5, 0.8\}$.

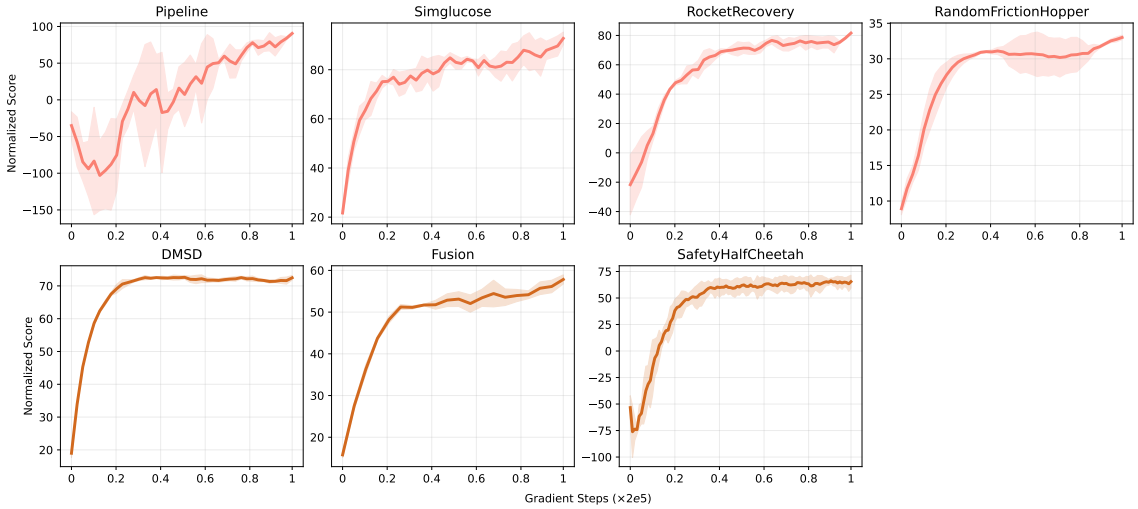


Figure 7.2: NeoRL2 Learning Curves.

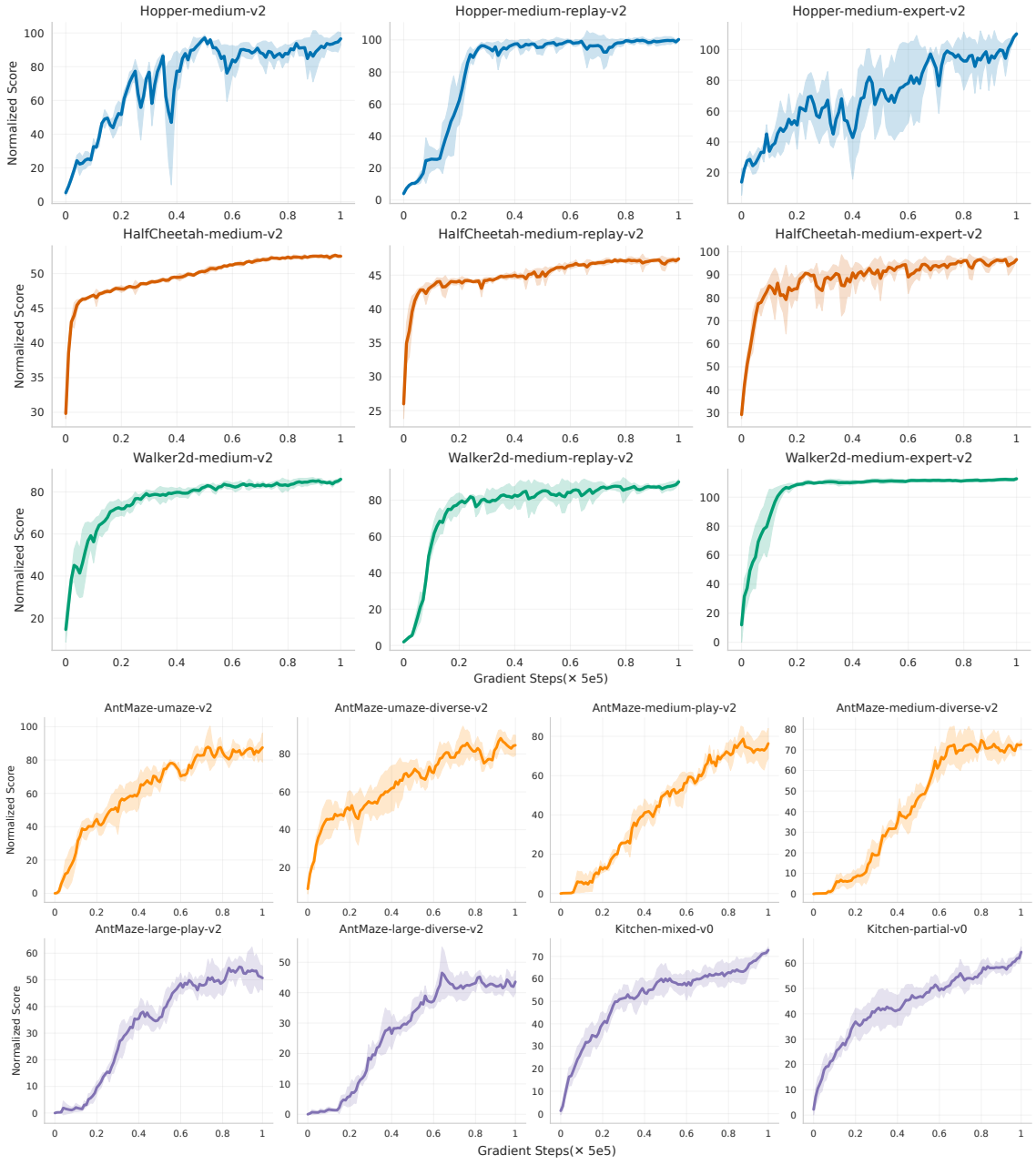


Figure 7.3: D4RL Learning Curves.

7.5 Proofs – ELAPSE

7.5.1 ELAPSE-LN

We implemented ELAPSE by introducing the BN method. Nevertheless, if Eq. (5.16) $\mathcal{L}_D = D_{KL}\left[q_\xi(z|a;s)||p(z|s)\right]$ is rewritten as follows,

$$\begin{aligned}\mathcal{L}_D &= \frac{1}{2m}\sum_{j=1}^{m}\sum_{i=1}^{n}\left(\mu_{ij}^2 + \sigma_{ij}^2 - log\,\sigma_{ij}^2 - 1\right) \\ &= \frac{1}{2}\sum_{j=1}^{m}\left(\frac{\sum_{i=1}^{n}\mu_{ij}^2}{n} + \frac{\sum_{i=1}^{n}\sigma_{ij}^2}{n} - \frac{\sum_{i=1}^{n} log\,\sigma_{ij}^2}{n} - 1\right),\end{aligned} \tag{7.74}$$

where m is the number of mini-batch samples, n is the dimension of the latent variable z, μ_{ij} and σ_{ij} are the mean and standard deviation of the approximate posterior for the i_{th} latent dimension of the j_{th} sample.

Similarly, we can calculate the lower bound of $\mathbb{E}[\mathcal{L}_D]$ as follows in terms of latent variable dimension, rather than based on the number of mini-batch samples.

$$\mathbb{E}\left[\mathcal{L}_D\right] \geq \frac{1}{2}\sum_{j=1}^{m}\left(Var[\mu_j] + \mathbb{E}^2[\mu_j]\right). \tag{7.75}$$

According to the Eq. (7.75), we can employ the Layer Normalization (LN) technique [164] to control $\hat{\mu}_j$ and thus $\mathbb{E}[\mathcal{L}_D]$. The $\hat{\mu}_j$ after LN can be written as $\hat{\mu}_j = w_L\frac{\mu_j - \text{mean}(\mu_j)}{\text{std}(\mu_j)} + b_L$, where $\text{mean}(x)$ and $\text{std}(x)$ denote the mean and standard deviations of x, w_L and b_L are the weight and bias parameters in LN layer. Based on $\mathbb{E}\left[\hat{\mu}_j\right] = b_L$ and $Var\left[\hat{\mu}_j\right] = w_L^2$, hence we can calculate the lower bound of $\mathbb{E}[\hat{\mathcal{L}}_D]$ as follows,

$$\mathbb{E}\left[\hat{\mathcal{L}}_D\right] \geq \frac{m}{2}(w_L^2 + b_L^2). \tag{7.76}$$

This method is named as *ELAPSE-LN*, summarized in Algorithm 4.

However, ELAPSE-LN is not used in practice, because our goal is to make each latent variable z_i discriminative, rather than to emphasize the discriminativeness of individual dimensions within z_i. Enhancing the discriminateness of z_i is beneficial for policy optimization in the latent action space and facilitates mapping it into the latent action projection space.

7.5.2 ELAPSE-TR

Next, we discuss *ELAPSE-TR*, another version of the ELAPSE algorithm. Considering that the prior is assumed $p(z_i|s_i) \sim \mathcal{N}(0,1)$ and the posterior $p(z_i|a_i;s_i)$ is approximated

Algorithm 4 ELAPSE-LN

Input: Dataset $\mathcal{D} = \{(s, a, r, s')\}$.
Parameters: Q-networks $Q_{\phi_{1,2}}$, target Q-networks $Q_{\phi'_{1,2}}$, value network V_ψ, policy network π_θ, LN weight w.

// Pre-train CVAE model for N_β steps.
for $t = 1$ to N_β **do**
  Sample mini-batch $(s, a) \sim \mathcal{D}$.
  Apply layer normalization:

$$\hat{\mu} = w\frac{\mu - \text{mean}(\mu)}{\text{std}(\mu)} + b.$$

  Optimize CVAE model using the ELAPSE-modified LN weight w by Eq. (5.3).
end for
// Policy evaluation and Policy improvement
for each gradient step **do**
  Sample $(s, a, r, s') \sim \mathcal{D}$.
  Decode latent actions using the ELAPSE-CVAE decoder: $\hat{a} = p_\rho(\cdot|z = \pi_\theta(s); s)$.
  Update the critic, actor, and other algorithm-specific components according to the chosen base method (e.g., PLAS (5.6), LAPO (5.9), or SPOT (5.13)).
end for
return Policy π_θ.

by encoder $q_\xi(z_i|a_i;s_i)$. Hence,

$$p(z_i|s_i) = \int_{a_i} p(a_i) q_\xi(z_i|a_i;s_i) da_i. \tag{7.77}$$

If we calculate the expectation of z_i for both sides of Eq. (7.77),

$$\begin{aligned} 0 &= \int_{z_i} \int_{a_i} z_i \cdot p(a_i) q_\xi(z_i|a_i;s_i) da_i dz_i \\ &= \mathbb{E}_{a_i}\left[\mu_i\right]. \end{aligned} \tag{7.78}$$

where μ_i is the mean of the approximate posterior for the i_{th} latent dimension of latent variable z.

Similarly, if we calculate the variance of z_i for both sides of Eq. (7.77),

$$\begin{aligned} 1 &= \int_{z_i} \int_{a_i} z_i^2 \cdot p(a_i) q_\xi(z_i|a_i;s_i) da_i dz_i \\ &= \mathbb{E}_{a_i}\left[\mu_i^2\right] + \mathbb{E}_{a_i}\left[\sigma_i^2\right]. \end{aligned} \tag{7.79}$$

where σ_i is the standard deviation of the approximate posterior for the i_{th} latent dimension of latent variable z.

We then apply the BN to the μ_i and σ_i,

$$\begin{cases} \hat{\mu_i} = w_\mu \dfrac{\mu_i - \text{mean}(\mu_i)}{\text{std}(\mu_i)} + b_\mu. \\ \hat{\sigma_i} = w_\sigma \dfrac{\sigma_i - \text{mean}(\sigma_i)}{\text{std}(\sigma_i)} + b_\sigma. \end{cases} \tag{7.80}$$

Thus, combining Eqs. (7.78), (7.79), and (7.80),

$$\begin{cases} b_\mu = 0, \\ w_\mu^2 + b_\mu^2 + w_\sigma^2 + b_\sigma^2 = 1. \end{cases} \tag{7.81}$$

If we set the bias $b_\sigma = 0$, then we can calculate $w_\mu^2 + w_\sigma^2 = 1$. Naturally, we can think of the fundamental trigonometric identity $\sin^2(x) + \cos^2(x) = 1$. Thus, the parameter can be calculated as follows,

$$\begin{cases} w_\mu = \sin(\dfrac{\pi}{2} \cdot sigmoid(\alpha)), \\ w_\sigma = \cos(\dfrac{\pi}{2} \cdot sigmoid(\alpha)), \\ b_\mu = b_\sigma = 0. \end{cases} \tag{7.82}$$

where α is a learnable parameter in BN.

Algorithm 5 ELAPSE-TR

Input: Dataset $\mathcal{D} = \{(s, a, r, s')\}$.
Parameters: Q-networks $Q_{\phi_{1,2}}$, target Q-networks $Q_{\phi'_{1,2}}$, value network V_ψ, policy network π_θ, TR weight w.

// Pre-train CVAE model for N_β steps.
for $t = 1$ to N_β **do**
  Sample mini-batch $(s, a) \sim \mathcal{D}$.
  Optimize CVAE model by Eq. (5.3) and optimizing parameters with Eq. (7.82).
end for
// Policy evaluation and Policy improvement.
for each gradient step **do**
  Sample $(s, a, r, s') \sim \mathcal{D}$.
  Decode latent actions using the ELAPSE-CVAE decoder: $\hat{a} = p_\rho(\cdot|z = \pi_\theta(s); s)$.
  Update the critic, actor, and other algorithm-specific components according to the chosen base method (e.g., PLAS (5.6), LAPO (5.9), or SPOT (5.13)).
end for
return Policy π_θ.

7.6 Experimental Details – ELAPSE

Data. We adopt the Gym-MuJoCo-v2 and Antmaze-v1 datasets from the D4RL benchmark, following the dataset choices in the original LAPO paper. Since the Gym-MuJoCo and Antmaze datasets used in PLAS, as well as the Antmaze datasets used in SPOT, differ from these versions, we reimplement both algorithms on the Gym-MuJoCo-v2 and Antmaze-v1 datasets for a consistent comparison.

Training Curves. The implementations of ELAPSE-PLAS, ELAPSE-LAPO, and ELAPSE-SPOT are built upon the official codebases of PLAS, LAPO, and SPOT, respectively. The training curves of ELAPSE-PLAS, ELAPSE-LAPO, and ELAPSE-SPOT are presented in Fig. 7.4.

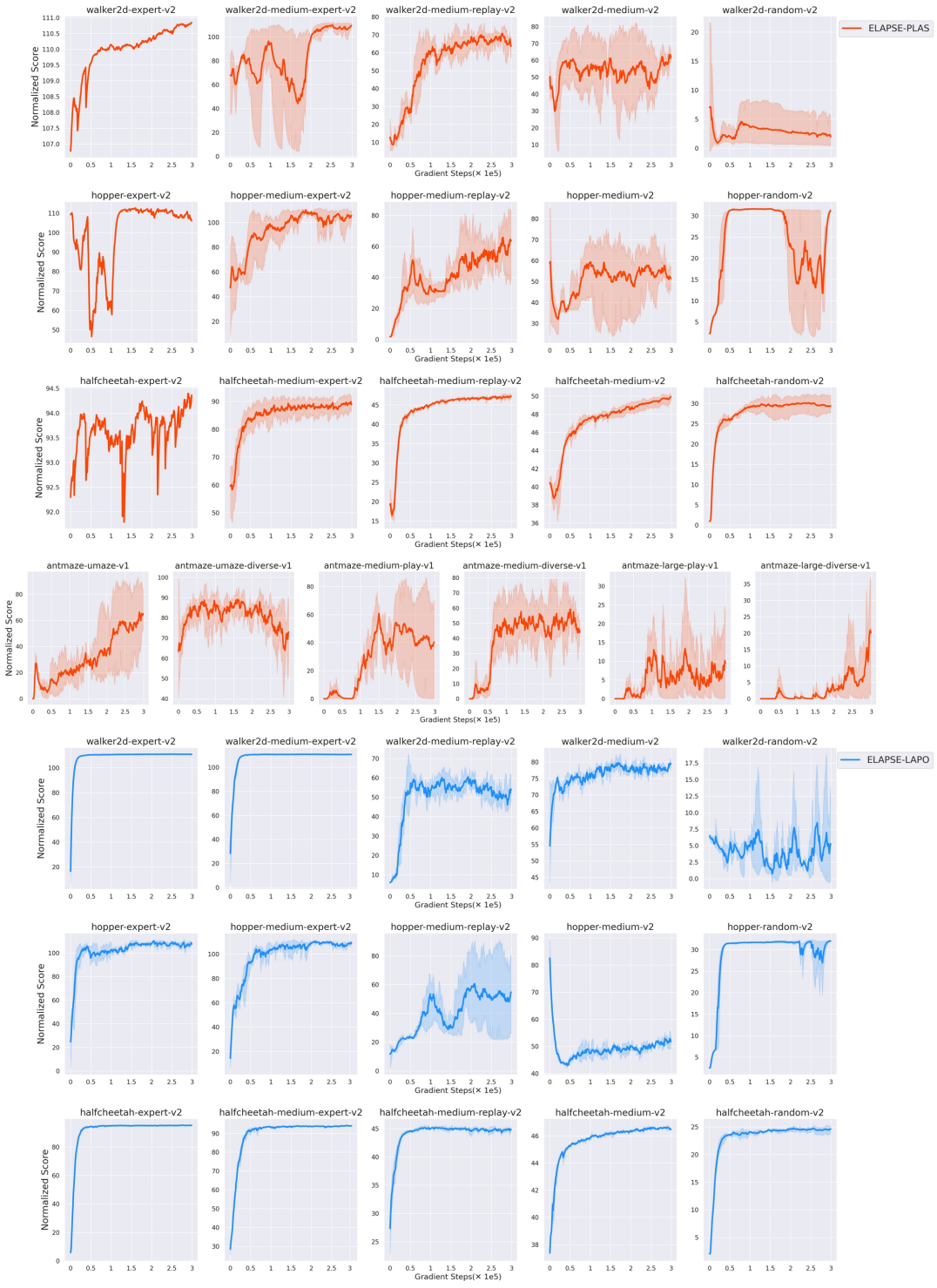
For Fig. 5.7 For each evaluation iteration, we average 10 normalized scores to obtain the reported score. In addition, the solid line and the shaded region denote the mean and standard deviation, respectively, over 3 random seeds.

For Tab. 5.1 The reported scores are averaged over the final 50 evaluations across 3 random seeds.

The dataset abbreviations used in Tab. 5.1 and Tab. 5.2 follow the standard D4RL naming convention. For example, `exp`, `med`, `med-rep`, and `med-exp` denote `expert`,

`medium`, `medium-replay`, and `medium-expert`, respectively, while `u`, `u-d`, `m-p`, `m-d`, `l-p`, and `l-d` denote `umaze`, `umaze-diverse`, `medium-play`, `medium-diverse`, `large-play`, and `large-diverse`, respectively.

Baselines. The normalized scores of the baseline algorithms are taken from their corresponding references.


walker2d-expert-v2
walker2d-medium-expert-v2
walker2d-medium-replay-v2
walker2d-medium-v2
walker2d-random-v2
ELAPSE-PLAS
hopper-expert-v2
hopper-medium-expert-v2
hopper-medium-replay-v2
hopper-medium-v2
hopper-random-v2
halfcheetah-expert-v2
halfcheetah-medium-expert-v2
halfcheetah-medium-replay-v2
halfcheetah-medium-v2
halfcheetah-random-v2
antmaze-umaze-v1
antmaze-umaze-diverse-v1
antmaze-medium-play-v1
antmaze-medium-diverse-v1
antmaze-large-play-v1
antmaze-large-diverse-v1
Normalized Score
Gradient Steps(× 1e5)
ELAPSE-LAPO

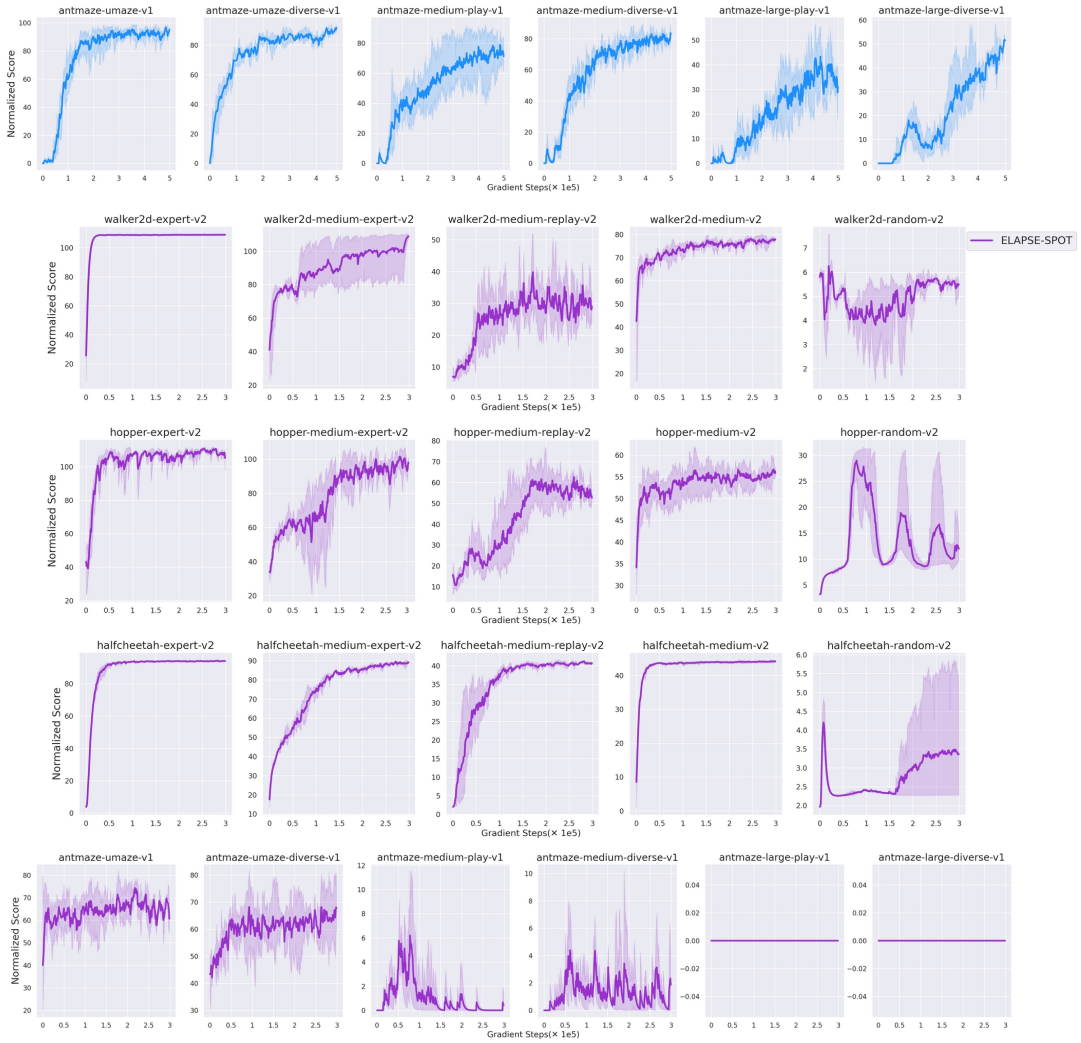


Figure 7.4: The training curves of ELAPSE-PLAS, ELAPSE-LAPO, and ELAPSE-SPOT algorithms on D4RL datasets.

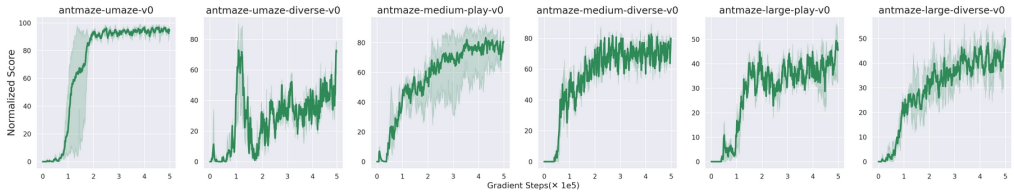


Figure 7.5: The training curves of ELAPSE-LAPO on Antmaze datasets.

Bibliography

[1] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," 2020. [Online]. Available: https://arxiv.org/abs/2005.01643

[2] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[3] M. Ghasemi, A. H. Moosavi, and D. Ebrahimi, "A comprehensive survey of reinforcement learning: From algorithms to practical challenges," 2025. [Online]. Available: https://arxiv.org/abs/2411.18892

[4] M. L. H. D. Lemos, R. e Silva Vieira, A. R. Tavares, L. S. Marcolino, and L. Chaimowicz, "Enhancing deep reinforcement learning for scale flexibility in real-time strategy games," *Entertainment Computing*, vol. 52, p. 100843, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1875952124002118

[5] K. Zhang, Y. Zuo, B. He, Y. Sun, R. Liu, C. Jiang, Y. Fan, K. Tian, G. Jia, P. Li, Y. Fu, X. Lv, Y. Zhang, S. Zeng, S. Qu, H. Li, S. Wang, Y. Wang, X. Long, F. Liu, X. Xu, J. Ma, X. Zhu, E. Hua, Y. Liu, Z. Li, H. Chen, X. Qu, Y. Li, W. Chen, Z. Yuan, J. Gao, D. Li, Z. Ma, G. Cui, Z. Liu, B. Qi, N. Ding, and B. Zhou, "A survey of reinforcement learning for large reasoning models," 2025. [Online]. Available: https://arxiv.org/abs/2509.08827

[6] M. M. Afsar, T. Crump, and B. Far, "Reinforcement learning based recommender systems: A survey," 2022. [Online]. Available: https://arxiv.org/abs/2101.06286

[7] X. Han, H. Afifi, H. Moungla, and M. Marot, "Leaky ppo: A simple and efficient rl algorithm for autonomous vehicles," in *2024 International Joint Conference on Neural Networks (IJCNN)*, 2024, pp. 1–7.

[8] M. S. Benhelal, X. Han, H. Afifi, B. Jouaber, H. Moungla, and A. M. Said, "Autonomous uav edge architecture for road hazards extended reality warnings," *IEEE Internet of Things Magazine*, vol. 6, no. 1, pp. 26–30, 2023.

[9] D. Ernst, P. Geurts, and L. Wehenkel, "Tree-based batch mode reinforcement learning," *Journal of Machine Learning Research*, vol. 6, no. 18, pp. 503–556, 2005. [Online]. Available: http://jmlr.org/papers/v6/ernst05a.html

[10] S. Lange, T. Gabel, and M. Riedmiller, *Batch Reinforcement Learning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 45–73. [Online]. Available: https://doi.org/10.1007/978-3-642-27645-3_2

[11] S. Liu, C. Bai, Z. Guo, H. Zhang, G. Sharma, and Y. Liu, "Selfbc: Self behavior cloning for offline reinforcement learning," in *ECAI*, 2024, pp. 2306–2313. [Online]. Available: https://doi.org/10.3233/FAIA240754

[12] Z. Peng, C. Han, Y. Liu, and Z. Zhou, "Weighted policy constraints for offline reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 8, 2023, pp. 9435–9443.

[13] Y. Mao, H. Zhang, C. Chen, Y. Xu, and X. Ji, "Supported trust region optimization for offline reinforcement learning," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 202. PMLR, 23–29 Jul 2023, pp. 23 829–23 851. [Online]. Available: https://proceedings.mlr.press/v202/mao23c.html

[14] X. Han, H. Afifi, and M. Marot, "PIQL: Projective implicit q-learning with support constraint for offline reinforcement learning," in *The 25th International Conference on Autonomous Agents and Multi-Agent Systems*, 2026. [Online]. Available: https://openreview.net/forum?id=K4mEvxumLB

[15] S. Park, K. Frans, S. Levine, and A. Kumar, "Is value learning really the main bottleneck in offline rl?" in *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, Eds., vol. 37. Curran Associates, Inc., 2024, pp. 79 029–79 056. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/file/8ffb4e3118280a66b192b6f06e0e2596-Paper-Conference.pdf

[16] H. Chen, C. Lu, Z. Wang, H. Su, and J. Zhu, "Score regularized policy optimization through diffusion behavior," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=xCRr9DrolJ

[17] S. Fujimoto, D. Meger, and D. Precup, "Off-policy deep reinforcement learning without exploration," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 97. PMLR, 09–15 Jun 2019, pp. 2052–2062. [Online]. Available: https://proceedings.mlr.press/v97/fujimoto19a.html

[18] A. Kumar, J. Fu, M. Soh, G. Tucker, and S. Levine, "Stabilizing off-policy q-learning via bootstrapping error reduction," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/c2073ffa77b5357a498057413bb09d3a-Paper.pdf

[19] J. Peters and S. Schaal, "Reinforcement learning by reward-weighted regression for operational space control," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 745–750.

[20] X. B. Peng, A. Kumar, G. Zhang, and S. Levine, "Advantage-weighted regression: Simple and scalable off-policy reinforcement learning," *arXiv preprint arXiv:1910.00177*, 2019.

[21] A. Nair, A. Gupta, M. Dalal, and S. Levine, "Awac: Accelerating online reinforcement learning with offline datasets," *arXiv preprint arXiv:2006.09359*, 2020.

[22] Z. Wang, A. Novikov, K. Zolna, J. S. Merel, J. T. Springenberg, S. E. Reed, B. Shahriari, N. Siegel, C. Gulcehre, N. Heess *et al.*, "Critic regularized regression," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7768–7778, 2020.

[23] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit q-learning," in *Deep RL Workshop NeurIPS 2021*, 2021. [Online]. Available: https://openreview.net/forum?id=EblVBDNalKu

[24] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative q-learning for offline reinforcement learning," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 1179–1191. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/0d2b2061826a5df3221116a5085a6052-Paper.pdf

[25] X. Han, H. Afifi, and M. Marot, "Cosine similarity based adaptive implicit q-learning for offline reinforcement learning," in *2025 IEEE Wireless Communications and Networking Conference (WCNC)*, 2025, pp. 1–6.

[26] K. Wu, Y. Zhao, Z. Xu, Z. Che, C. Yin, C. Harold Liu, F. Feng, and J. Tang, "Acl-ql: Adaptive conservative level in q-learning for offline reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 6, pp. 11 399–11 413, 2025.

[27] D. Brandfonbrener, W. F. Whitney, R. Ranganath, and J. Bruna, "Offline RL without off-policy evaluation," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021. [Online]. Available: https://openreview.net/forum?id=LU687itn08w

[28] B. Eysenbach, M. Geist, S. Levine, and R. Salakhutdinov, "A connection between one-step RL and critic regularization in reinforcement learning," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 202. PMLR, 23–29 Jul 2023, pp. 9485–9507. [Online]. Available: https://proceedings.mlr.press/v202/eysenbach23a.html

[29] H. Xu, L. Jiang, J. Li, Z. Yang, Z. Wang, V. W. K. Chan, and X. Zhan, "Offline RL with no OOD actions: In-sample learning via implicit value regularization," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=ueYYgo2pSSU

[30] K. LEI, Z. He, C. Lu, K. Hu, Y. Gao, and H. Xu, "Uni-o4: Unifying online and offline deep reinforcement learning with multi-step on-policy optimization," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=tbFBh3LMKi

[31] R. Figueiredo Prudencio, M. R. O. A. Maximo, and E. L. Colombini, "A survey on offline reinforcement learning: Taxonomy, review, and open problems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 8, pp. 10 237–10 257, 2024.

[32] J. Chen, B. Ganguly, Y. Xu, Y. Mei, T. Lan, and V. Aggarwal, "Deep generative models for offline policy learning: Tutorial, survey, and perspectives on future directions," *Transactions on Machine Learning Research*, 2024, survey Certification. [Online]. Available: https://openreview.net/forum?id=Mm2cMDl9r5

[33] C. Lu, P. J. Ball, T. G. J. Rudner, J. Parker-Holder, M. A. Osborne, and Y. W. Teh, "Challenges and opportunities in offline reinforcement learning from visual observations," *Transactions on Machine Learning Research*, 2023. [Online]. Available: https://openreview.net/forum?id=1QqIfGZOWu

[34] D. Garg, J. Hejna, M. Geist, and S. Ermon, "Extreme q-learning: Maxent RL without entropy," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=SJ0Lde3tRL

[35] X. Chen, Z. Zhou, Z. Wang, C. Wang, Y. Wu, and K. Ross, "Bail: Best-action imitation learning for batch deep reinforcement learning," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 18 353–18 363.

[36] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson, "Implicit behavioral cloning," in *Proceedings of the 5th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 164. PMLR, 08–11 Nov 2022, pp. 158–168.

[37] H. Xu, X. Zhan, H. Yin, and H. Qin, "Discriminator-weighted offline imitation learning from suboptimal demonstrations," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 162. PMLR, 17–23 Jul 2022, pp. 24 725–24 742.

[38] S. Fujimoto and S. S. Gu, "A minimalist approach to offline reinforcement learning," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 20 132–20 145.

[39] J. Wu, H. Wu, Z. Qiu, J. Wang, and M. Long, "Supported policy optimization for offline reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 31 278–31 291, 2022.

[40] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2022. [Online]. Available: https://arxiv.org/abs/1312.6114

[41] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in Neural Information Processing Systems*, S. Solla, T. Leen, and K. Müller, Eds., vol. 12. MIT Press, 1999. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf

[42] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[43] H. v. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI'16. AAAI Press, 2016, p. 2094–2100.

[44] M. Hessel, J. Modayil, H. van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, "Rainbow: combining improvements in deep reinforcement learning," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, ser. AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018.

[45] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3, pp. 229–256, 1992.

[46] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proceedings of The 33rd International Conference on Machine Learning*, ser.

Proceedings of Machine Learning Research, vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1928–1937. [Online]. Available: https://proceedings.mlr.press/v48/mniha16.html

[47] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proceedings of the 31st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 32, no. 1. Bejing, China: PMLR, 22–24 Jun 2014, pp. 387–395. [Online]. Available: https://proceedings.mlr.press/v32/silver14.html

[48] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[49] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 10–15 Jul 2018, pp. 1587–1596. [Online]. Available: https://proceedings.mlr.press/v80/fujimoto18a.html

[50] P. Thomas and E. Brunskill, "Data-efficient off-policy policy evaluation for reinforcement learning," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 2139–2148. [Online]. Available: https://proceedings.mlr.press/v48/thomasa16.html

[51] N. Kallus and M. Uehara, "Intrinsically efficient, stable, and bounded off-policy evaluation for reinforcement learning," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/59bcda7c438bad7d2afffe9e2fed00be-Paper.pdf

[52] O. Nachum, Y. Chow, B. Dai, and L. Li, "Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/cf9a242b70f45317ffd281241fa66502-Paper.pdf

[53] H. van Hasselt, Y. Doron, F. Strub, M. Hessel, N. Sonnerat, and J. Modayil, "Deep reinforcement learning and the deadly triad," 2018. [Online]. Available: https://arxiv.org/abs/1812.02648

[54] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 15. Fort Lauderdale, FL, USA: PMLR, 11–13 Apr 2011, pp. 627–635. [Online]. Available: https://proceedings.mlr.press/v15/ross11a.html

[55] G. An, S. Moon, J.-H. Kim, and H. O. Song, "Uncertainty-based offline reinforcement learning with diversified q-ensemble," *Advances in neural information processing systems*, vol. 34, pp. 7436–7447, 2021.

[56] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," in *Advances in Neural Information Processing Systems*, D. Touretzky, Ed., vol. 1. Morgan-Kaufmann, 1988. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/1988/file/812b4ba287f5ee0bc9d43bbf5bbe87fb-Paper.pdf

[57] I. Csiszár, "i-divergence geometry of probability distributions and minimization problems," *The Annals of Probability*, vol. 3, no. 1, pp. 146–158, 1975. [Online]. Available: http://www.jstor.org/stable/2959270

[58] A. Beeson and G. Montana, "Improving TD3-BC: Relaxed policy constraint for offline learning and stable online fine-tuning," in *3rd Offline RL Workshop: Offline RL as a "Launchpad"*, 2022. [Online]. Available: https://openreview.net/forum?id=roLMh1mcp6

[59] Y. Wu, G. Tucker, and O. Nachum, "Behavior regularized offline reinforcement learning," 2020. [Online]. Available: https://openreview.net/forum?id=BJg9hTNKPH

[60] M. Omura, Y. Mukuta, K. Ota, T. Osa, and T. Harada, "Offline reinforcement learning with wasserstein regularization via optimal transport maps," in *Reinforcement Learning Conference*, 2025. [Online]. Available: https://openreview.net/forum?id=RxdQBYKjtE

[61] A. Asadulaev, R. Korst, A. Korotin, V. Egiazarian, A. Filchenkov, and E. Burnaev, "Rethinking optimal transport in offline reinforcement learning," in *Advances in Neural Information Processing Systems*, vol. 37. Curran Associates, Inc., 2024, pp. 123 592–123 607.

[62] I. Kostrikov, R. Fergus, J. Tompson, and O. Nachum, "Offline reinforcement learning with fisher divergence critic regularization," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 139. PMLR, 18–24 Jul 2021, pp. 5774–5783.

[63] S. K. S. Ghasemipour, D. Schuurmans, and S. S. Gu, "Emaq: Expected-max q-learning operator for simple yet effective offline and online rl," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 139. PMLR, 18–24 Jul 2021, pp. 3682–3691.

[64] C. Xiao, H. Wang, Y. Pan, A. White, and M. White, "The in-sample softmax for offline reinforcement learning," in *The Eleventh International Conference on Learning Representations*, 2023.

[65] J. Lyu, X. Ma, X. Li, and Z. Lu, "Mildly conservative q-learning for offline reinforcement learning," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 1711–1724. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/0b5669c3b07bb8429af19a7919376ff5-Paper-Conference.pdf

[66] J. Liu, Y. Zhang, C. Li, Y. Yang, Y. Liu, and W. Ouyang, "Adaptive pessimism via target q-value for offline reinforcement learning," *Neural Networks*, vol. 180, p. 106588, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608024005124

[67] L. Huang, B. Dong, J. Lu, and W. Zhang, "Mild policy evaluation for offline actor–critic," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 12, pp. 17 950–17 964, 2024.

[68] L. Huang, B. Dong, and W. Zhang, "Efficient offline reinforcement learning with relaxed conservatism," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 8, pp. 5260–5272, 2024.

[69] L. Huang, B. Dong, W. Xie, and W. Zhang, "Offline reinforcement learning with behavior value regularization," *IEEE Transactions on Cybernetics*, vol. 54, no. 6, pp. 3692–3704, 2024.

[70] W. R. Clements, B. V. Delft, B.-M. Robaglia, R. B. Slaoui, and S. Toth, "Estimating risk and uncertainty in deep reinforcement learning," 2020. [Online]. Available: https://arxiv.org/abs/1905.09638

[71] Y. Wu, S. Zhai, N. Srivastava, J. M. Susskind, J. Zhang, R. Salakhutdinov, and H. Goh, "Uncertainty weighted actor-critic for offline reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2021, pp. 11 319–11 328.

[72] C. Bai, L. Wang, Z. Yang, Z.-H. Deng, A. Garg, P. Liu, and Z. Wang, "Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning," in *International Conference on Learning Representations*, 2022.

[73] K. Ghasemipour, S. S. Gu, and O. Nachum, "Why so pessimistic? estimating uncertainties for offline rl through ensembles, and why their independence matters," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS '22. Red Hook, NY, USA: Curran Associates Inc., 2022.

[74] D. Wang and L. Zhang, "Ensemble-based offline reinforcement learning with adaptive behavior cloning," in *Adaptive Foundation Models: Evolving AI for Personalized and Efficient Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=PwRWbBLOmr

[75] Z. Peng, C. Han, Y. Liu, J. Tang, and Z. Zhou, "Pessimistic policy iteration with bounded uncertainty," *Expert Systems with Applications*, vol. 282, p. 127651, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417425012734

[76] X. Han, H. Afifi, H. Moungla, and M. Marot, "Attention ensemble mixture: a novel offline reinforcement learning algorithm for autonomous vehicles," *Applied Intelligence*, vol. 55, no. 7, 2025.

[77] H. Zhang, Y. Mao, B. Wang, S. He, Y. Xu, and X. Ji, "In-sample actor critic for offline reinforcement learning," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=dfDv0WU853R

[78] H. Xu, S. Li, H. Sikchi, S. Niekum, and A. Zhang, "Iterative dual-RL: An optimal discriminator weighted imitation perspective for reinforcement learning," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=9JtG4nN7ql

[79] R. Zhang*, B. Dai*, L. Li, and D. Schuurmans, "Gendice: Generalized offline estimation of stationary values," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=HkxlcnVFwB

[80] S. Zhang, B. Liu, and S. Whiteson, "GradientDICE: Rethinking generalized offline estimation of stationary values," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 119. PMLR, 13–18 Jul 2020, pp. 11 194–11 203. [Online]. Available: https://proceedings.mlr.press/v119/zhang20r.html

[81] J. Lee, W. Jeon, B. Lee, J. Pineau, and K.-E. Kim, "Optidice: Offline policy optimization via stationary distribution correction estimation," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine

Learning Research, vol. 139. PMLR, 18–24 Jul 2021, pp. 6120–6130. [Online]. Available: https://proceedings.mlr.press/v139/lee21f.html

[82] L. Mao, H. Xu, W. Zhang, and X. Zhan, "ODICE: Revealing the mystery of distribution correction estimation via orthogonal-gradient update," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=L8UNn7Llt4

[83] L. Mao, H. Xu, X. Zhan, W. Zhang, and A. Zhang, "Diffusion-dice: In-sample diffusion guidance for offline reinforcement learning," in *Advances in Neural Information Processing Systems*, vol. 37. Curran Associates, Inc., 2024, pp. 98 806–98 834. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/file/b2fea79b1137d917e8b7cce9434ab5fa-Paper-Conference.pdf

[84] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[85] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf

[86] M. Janner, Q. Li, and S. Levine, "Offline reinforcement learning as one big sequence modeling problem," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 1273–1286. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/099fe6b0b444c23836c4a5d07346082b-Paper.pdf

[87] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 15 084–15 097. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/7f489f642a0ddb10272b5c31057f0663-Paper.pdf

[88] T. Yamagata, A. Khalil, and R. Santos-Rodriguez, "Q-learning decision transformer: Leveraging dynamic programming for conditional sequence modelling in offline RL," in *Proceedings of the 40th International Conference on Machine Learning*,

ser. Proceedings of Machine Learning Research, vol. 202. PMLR, 23–29 Jul 2023, pp. 38 989–39 007. [Online]. Available: https://proceedings.mlr.press/v202/yamagata23a.html

[89] J. Kim, S. Lee, W. Kim, and Y. Sung, "Adaptive q-aid for conditional supervised learning in offline reinforcement learning," in *Advances in Neural Information Processing Systems*, vol. 37. Curran Associates, Inc., 2024, pp. 87 104–87 135. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/file/9e72fc628edeb29f7aa64ac81b7ec6ce-Paper-Conference.pdf

[90] Z. Zhuang, D. Peng, J. Liu, Z. Zhang, and D. Wang, "Reinformer: max-return sequence modeling for offline rl," in *Proceedings of the 41st International Conference on Machine Learning*, ser. ICML'24. JMLR.org, 2024.

[91] A. Ajay, Y. Du, A. Gupta, J. B. Tenenbaum, T. S. Jaakkola, and P. Agrawal, "Is conditional generative modeling all you need for decision making?" in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=sP1fo2K9DFG

[92] M. Janner, Y. Du, J. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 162. PMLR, 17–23 Jul 2022, pp. 9902–9915. [Online]. Available: https://proceedings.mlr.press/v162/janner22a.html

[93] W. Li, X. Wang, B. Jin, and H. Zha, "Hierarchical diffusion for offline decision making," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 202. PMLR, 23–29 Jul 2023, pp. 20 035–20 064. [Online]. Available: https://proceedings.mlr.press/v202/li23ad.html

[94] C. Chen, F. Deng, K. Kawaguchi, C. Gulcehre, and S. Ahn, "Simple hierarchical planning with diffusion," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=kXHEBK9uAY

[95] W. Zhou, S. Bajracharya, and D. Held, "Plas: Latent action space for offline reinforcement learning," in *Proceedings of the 2020 Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 155. PMLR, 16–18 Nov 2021, pp. 1719–1735. [Online]. Available: https://proceedings.mlr.press/v155/zhou21b.html

[96] X. Chen, A. Ghadirzadeh, T. Yu, J. Wang, A. Y. Gao, W. Li, L. Bin, C. Finn, and C. Zhang, "Lapo: Latent-variable advantage-weighted policy optimization for offline reinforcement learning," in *Advances in Neural Information*

Processing Systems, vol. 35. Curran Associates, Inc., 2022, pp. 36 902–36 913. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/efb2072a358cefb75886a315a6fcf880-Paper-Conference.pdf

[97] S. Rezaeifar, R. Dadashi, N. Vieillard, L. Hussenot, O. Bachem, O. Pietquin, and M. Geist, "Offline reinforcement learning as anti-exploration," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 7, pp. 8106–8114, Jun. 2022. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/20783

[98] T. Yu, G. Thomas, L. Yu, S. Ermon, J. Y. Zou, S. Levine, C. Finn, and T. Ma, "Mopo: Model-based offline policy optimization," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 14 129–14 142. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/a322852ce0df73e204b7e67cbbef0d0a-Paper.pdf

[99] T. Yu, A. Kumar, R. Rafailov, A. Rajeswaran, S. Levine, and C. Finn, "Combo: Conservative offline model-based policy optimization," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 28 954–28 967. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/f29a179746902e331572c483c45e5086-Paper.pdf

[100] Y. Sun, J. Zhang, C. Jia, H. Lin, J. Ye, and Y. Yu, "Model-Bellman inconsistency for model-based offline reinforcement learning," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 202. PMLR, 23–29 Jul 2023, pp. 33 177–33 194. [Online]. Available: https://proceedings.mlr.press/v202/sun23q.html

[101] H. Chen and Z. Chen, "Mildly conservative regularized evaluation for offline reinforcement learning," 2025. [Online]. Available: https://arxiv.org/abs/2508.05960

[102] Y. Mao, Y. Qu, C. Wang, and X. Ji, "Adaptive neighborhood-constrained q learning for offline reinforcement learning," in *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. [Online]. Available: https://openreview.net/forum?id=qgi5TfBXBw

[103] S. Park, Q. Li, and S. Levine, "Flow q-learning," in *Forty-second International Conference on Machine Learning*, 2025. [Online]. Available: https://openreview.net/forum?id=KVf2SFL1pi

[104] Z. Wang, J. J. Hunt, and M. Zhou, "Diffusion policies as an expressive policy class for offline reinforcement learning," *arXiv preprint arXiv:2208.06193*, 2022.

[105] P. Hansen-Estruch, I. Kostrikov, M. Janner, J. G. Kuba, and S. Levine, "Idql: Implicit q-learning as an actor-critic method with diffusion policies," 2023. [Online]. Available: https://arxiv.org/abs/2304.10573

[106] Z. Peng, Y. Liu, C. Han, and Z. Zhou, "Implicit policy constraint for offline reinforcement learning," *CAAI Transactions on Intelligence Technology*, vol. 9, no. 4, pp. 973–981, 2024. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/cit2.12304

[107] M. Wang, Y. Jin, and G. Montana, "Learning on one mode: Addressing multi-modality in offline reinforcement learning," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=upkxzurnLC

[108] C. Zhou, W. Luo, H. Li, C. Han, T. Guo, and Z. Zhang, "Dual alignment maximin optimization for offline model-based rl," 2025. [Online]. Available: https://arxiv.org/abs/2502.00850

[109] H. Zhang, J. Shao, Y. Jiang, S. He, G. Zhang, and X. Ji, "State deviation correction for offline reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 8, 2022, pp. 9022–9030.

[110] K. Jiang, J.-Y. Yao, and X. Tan, "Recovering from out-of-sample states via inverse dynamics in offline reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 38 815–38 826, 2023.

[111] K. Jiang, W. Jiang, and X. Tan, "Variational ood state correction for offline reinforcement learning," *arXiv preprint arXiv:2505.00503*, 2025.

[112] C. A. Hepburn, Y. Jin, and G. Montana, "State-constrained offline reinforcement learning," *arXiv preprint arXiv:2405.14374*, 2024.

[113] H. Xu, L. Jiang, L. Jianxiong, and X. Zhan, "A policy-guided imitation approach for offline reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 4085–4098, 2022.

[114] J. Lyu, A. Gong, L. Wan, Z. Lu, and X. Li, "State advantage weighting for offline rl," *arXiv preprint arXiv:2210.04251*, 2022.

[115] W. Jiang, S. Li, Z. Li, Y. Ke, Z. Jiang, Y. Li, and Y. Liu, "Dual-policy-guided offline reinforcement learning with optimal stopping," in *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, 2024, pp. 2315–2317.

[116] P. Cheng, Z. Wu, J. Li, Z. He, H. Xu, W. Sun, Y. Lin, and X. Zhan, "Pushing the limit of sample-efficient offline reinforcement learning," in *ICLR 2025 Workshop on World Models: Understanding, Modelling and Scaling*, 2025.

[117] S. Park, K. Frans, D. Mann, B. Eysenbach, A. Kumar, and S. Levine, "Horizon reduction makes RL scalable," in *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. [Online]. Available: https://openreview.net/forum?id=hguaupzLCU

[118] K. Wang, I. Javali, M. Bortkiewicz, T. Trzcinski, and B. Eysenbach, "1000 layer networks for self-supervised RL: Scaling depth can enable new goal-reaching capabilities," in *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. [Online]. Available: https://openreview.net/forum?id=s0JVsx3bx1

[119] Y. Chebotar, Q. Vuong, K. Hausman, F. Xia, Y. Lu, A. Irpan, A. Kumar, T. Yu, A. Herzog, K. Pertsch, K. Gopalakrishnan, J. Ibarz, O. Nachum, S. A. Sontakke, G. Salazar, H. T. Tran, J. Peralta, C. Tan, D. Manjunath, J. Singh, B. Zitkovich, T. Jackson, K. Rao, C. Finn, and S. Levine, "Q-transformer: Scalable offline reinforcement learning via autoregressive q-functions," in *Proceedings of The 7th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 229. PMLR, 06–09 Nov 2023, pp. 3909–3928. [Online]. Available: https://proceedings.mlr.press/v229/chebotar23a.html

[120] J. Cheng, R. Qiao, Y. MA, B. Li, G. Xiong, Q. Miao, Y. Li, and Y. Lv, "Scaling offline model-based RL via jointly-optimized world-action model pretraining," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=T1OvCSFaum

[121] A. Kumar, R. Agarwal, X. Geng, G. Tucker, and S. Levine, "Offline q-learning on diverse multi-task data both scales and generalizes," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=4-k7kUavAj

[122] J. T. Springenberg, A. Abdolmaleki, J. Zhang, O. Groth, M. Bloesch, T. Lampe, P. Brakel, S. M. E. Bechtle, S. Kapturowski, R. Hafner, N. Heess, and M. Riedmiller, "Offline actor-critic reinforcement learning scales to large models," in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 235. PMLR, 21–27 Jul 2024, pp. 46 323–46 350.

[123] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine, "D4rl: Datasets for deep data-driven reinforcement learning," *arXiv preprint arXiv:2004.07219*, 2020.

[124] S. Gao, Z. Tu, R.-J. Qin, Y.-H. Sun, X.-H. Chen, and Y. Yu, "Neorl-2: Near real-world benchmarks for offline reinforcement learning with extended realistic scenarios," 2025. [Online]. Available: https://arxiv.org/abs/2503.19267

[125] K. Wu, Y. Zhao, Z. Xu, Z. Che, C. Yin, C. H. Liu, Q. Qiu, F. Feng, and J. Tang, "Acl-ql: Adaptive conservative level in q-learning for offline reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[126] L. Chen, J. Yan, Z. Shao, L. Wang, Q. Lin, S. Rajmohan, T. Moscibroda, and D. Zhang, "Conservative state value estimation for offline reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[127] C. Zhang, S. Kuppannagari, and P. Viktor, "Brac+: Improved behavior regularized actor critic for offline reinforcement learning," in *Asian Conference on Machine Learning*. PMLR, 2021, pp. 204–219.

[128] X. Han, H. Afifi, and M. Marot, "Elapse: Expand latent action projection space for policy optimization in offline reinforcement learning," *Neurocomputing*, vol. 631, p. 129665, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231225003376

[129] Q. Vuong, A. Kumar, S. Levine, and Y. Chebotar, "Dasco: Dual-generator adversarial support constrained offline reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 38 937–38 949, 2022.

[130] S. Emmons, B. Eysenbach, I. Kostrikov, and S. Levine, "Rvs: What is essential for offline RL via supervised learning?" in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=S874XAIpkR-

[131] Y. Zhou, X. Li, and Q. Qu, "Offline reinforcement learning with adaptive behavior regularization," 2022.

[132] Q.-W. Luo, M.-K. Xie, Y.-W. Wang, and S.-J. Huang, "Learning to trust bellman updates: Selective state-adaptive regularization for offline RL," in *Forty-second International Conference on Machine Learning*, 2025.

[133] Z. Peng, Y. Liu, H. Chen, and Z. Zhou, "Conservative network for offline reinforcement learning," *Knowledge-Based Systems*, vol. 282, p. 111101, 2023.

[134] T. Chen, R. Cai, F. Wu, and X. Zhang, "ACTIVE: Offline reinforcement learning via adaptive imitation and in-sample v-ensemble," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=qiluFujVc8

[135] M. Omura, Y. Mukuta, K. Ota, T. Osa, and T. Harada, "Offline reinforcement learning with wasserstein regularization via optimal transport maps," in *Reinforcement Learning Conference*, 2025. [Online]. Available: https://openreview.net/forum?id=RxdQBYKjtE

[136] S. Kakade and J. Langford, "Approximately optimal approximate reinforcement learning," in *Proceedings of the Nineteenth International Conference on Machine Learning*, ser. ICML '02. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, p. 267–274.

[137] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, "Reinforcement learning with deep energy-based policies," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 06–11 Aug 2017, pp. 1352–1361.

[138] K. Asadi and M. L. Littman, "An alternative softmax operator for reinforcement learning," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 06–11 Aug 2017, pp. 243–252.

[139] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 10–15 Jul 2018, pp. 1861–1870.

[140] C. Xiao, R. Huang, J. Mei, D. Schuurmans, and M. Müller, "Maximum entropy monte-carlo planning," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019.

[141] N. Siegel, J. T. Springenberg, F. Berkenkamp, A. Abdolmaleki, M. Neunert, T. Lampe, R. Hafner, N. Heess, and M. Riedmiller, "Keep doing what worked: Behavior modelling priors for offline reinforcement learning," in *International Conference on Learning Representations*, 2020.

[142] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 10–15 Jul 2018, pp. 1587–1596.

[143] H. Xu, L. Jiang, J. Li, Z. Yang, Z. Wang, V. W. K. Chan, and X. Zhan, "Offline RL with no OOD actions: In-sample learning via implicit value regularization," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=ueYYgo2pSSU

[144] Y. Luo, zhengyao jiang, S. Cohen, E. Grefenstette, and M. P. Deisenroth, "Optimal transport for offline imitation learning," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=MhuFzFsrfvH

[145] M. Rigter, B. Lacerda, and N. Hawes, "Rambo-rl: Robust adversarial model-based offline reinforcement learning," in *Advances in Neural Information Processing Systems*, vol. 35. Curran Associates, Inc., 2022, pp. 16 082–16 097.

[146] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in Neural Information Processing Systems*, vol. 28, 2015. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf

[147] Z. Zhuang, K. LEI, J. Liu, D. Wang, and Y. Guo, "Behavior proximal policy optimization," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=3c13LptpIph

[148] D. Tarasov, V. Kurenkov, A. Nikulin, and S. Kolesnikov, "Revisiting the minimalist approach to offline reinforcement learning," in *Advances in Neural Information Processing Systems*, vol. 36. Curran Associates, Inc., 2023, pp. 11 592–11 620. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/26cce1e512793f2072fd27c391e04652-Paper-Conference.pdf

[149] S. Venkatraman, S. Khaitan, R. T. Akella, J. Dolan, J. Schneider, and G. Berseth, "Reasoning with latent diffusion in offline reinforcement learning," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=tGQirjzddO

[150] B. Kang, X. Ma, C. Du, T. Pang, and S. Yan, "Efficient diffusion policies for offline reinforcement learning," in *Advances in Neural Information Processing Systems*, vol. 36, 2023, pp. 67 195–67 212. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/d45e0bfb5a39477d56b55c0824200008-Paper-Conference.pdf

[151] S. Yang, Z. Wang, H. Zheng, Y. Feng, and M. Zhou, "A behavior regularized implicit policy for offline reinforcement learning," 2022. [Online]. Available: https://arxiv.org/abs/2202.09673

[152] Y. Burda, R. Grosse, and R. Salakhutdinov, "Importance weighted autoencoders," 2016. [Online]. Available: https://arxiv.org/abs/1509.00519

[153] S. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," in *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, 2016, pp. 10–21.

[154] H. Fu, C. Li, X. Liu, J. Gao, A. Celikyilmaz, and L. Carin, "Cyclical annealing schedule: A simple approach to mitigating KL vanishing," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 240–250. [Online]. Available: https://aclanthology.org/N19-1021/

[155] Q. Zhu, W. Bi, X. Liu, X. Ma, X. Li, and D. Wu, "A batch normalized inference network keeps the KL vanishing away," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 2636–2649. [Online]. Available: https://aclanthology.org/2020.acl-main.235/

[156] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*. PMLR, 2015, pp. 448–456.

[157] Z. Yu and X. Zhang, "Actor-critic alignment for offline-to-online reinforcement learning," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 202. PMLR, 23–29 Jul 2023, pp. 40 452–40 474. [Online]. Available: https://proceedings.mlr.press/v202/yu23k.html

[158] L. Shi, R. Dadashi, Y. Chi, P. S. Castro, and M. Geist, "Offline reinforcement learning with on-policy q-function regularization," in *Machine Learning and Knowledge Discovery in Databases: Research Track*. Cham: Springer Nature Switzerland, 2023, pp. 455–471.

[159] Y. Ma, J. HAO, X. Hu, Y. ZHENG, and C. Xiao, "Iteratively refined behavior regularization for offline reinforcement learning," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [Online]. Available: https://openreview.net/forum?id=RbS7RWxw3r

[160] Y. Yue, B. Kang, X. Ma, Q. Yang, G. Huang, S. Song, and S. Yan, "Decoupled prioritized resampling for offline rl," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15., 2024.

[161] S. Y. Arnob, R. Islam, and D. Precup, "Importance of empirical sample complexity analysis for offline reinforcement learning," 2021. [Online]. Available: https://arxiv.org/abs/2112.15578

[162] I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework." *ICLR (Poster)*, vol. 3, 2017.

[163] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 06–11 Aug 2017, pp. 22–31.

[164] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016. [Online]. Available: https://arxiv.org/abs/1607.06450

DOCTORAL SCHOOL

Titre : Étude systématique de l'apprentissage par renforcement hors ligne : évolution méthodologique et conception algorithmique

Mots clés : Apprentissage par renforcement hors ligne ; Évolution méthodologique ; Évaluation de politique ; Amélioration de politique ; Contraintes sur les politiques ; Autoencodeurs variationnels conditionnels

Résumé : L'apprentissage par renforcement hors ligne (offline RL) vise à apprendre des politiques de décision à partir de données fixes, sans interaction avec l'environnement. Il est pertinent lorsque la collecte en ligne est coûteuse, risquée ou impossible. Toutefois, l'offline RL se heurte à des défis fondamentaux, notamment le décalage de distribution, les actions hors distribution (out-of-distribution, OOD) et l'erreur d'extrapolation, qui compliquent l'évaluation et l'amélioration fiables des politiques. Cette thèse étudie l'offline RL sans modèle avec deux objectifs : comprendre son évolution méthodologique et développer des algorithmes plus efficaces.

Premièrement, cette thèse réexamine l'offline RL à la lumière de la deadly triad et en analyse les défis. Sur cette base, elle développe une taxonomie algorithmique de l'offline RL et retrace l'évolution de ses principes de conception. Plutôt que de considérer les méthodes existantes comme des catégories isolées, la thèse montre comment les familles algorithmiques émergent et évoluent face à des défis tels que la surestimation des valeurs et le décalage de distribution.

Deuxièmement, afin de répondre à la tension entre l'apprentissage in-sample et l'apprentissage multi-pas, Projective Implicit Q-Learning (PIQL) est proposé. PIQL remplace l'expectile fixe d'Implicit Q-Learning par un paramètre adaptatif fondé sur la projection, autorisant une interprétation multi-pas tout en préservant la propriété d'apprentissage in-sample. En outre, PIQL adopte un objectif d'amélioration de politique sous contrainte de support, mieux aligné sur l'étape d'évaluation. L'analyse théorique établit une amélioration monotone de la politique ainsi qu'un critère plus strict pour les actions avantageuses, tandis que les expériences sur des benchmarks montrent de bonnes performances.

Troisièmement, la thèse introduit le cadre Continuous Constraint Interpolation (CCI) et l'algorithme Automatic Constraint Policy Optimization (ACPO) pour unifier et adapter les contraintes de politique. CCI fournit une vision unifiée de l'optimisation où le clonage comportemental pondéré (weighted behavior cloning, wBC), la régularisation de densité fondée sur la divergence de Kullback-Leibler et les contraintes de support apparaissent comme des cas particuliers d'un spectre continu. Sur cette base, ACPO formalise l'apprentissage adaptatif des contraintes comme un problème primal-dual et ajuste automatiquement le paramètre d'interpolation. La thèse établit aussi un lemme de différence de performance à entropie maximale ainsi que des bornes inférieures pour la politique optimale et sa projection paramétrique. Les expériences montrent qu'ACPO obtient des résultats cohérents sur divers benchmarks d'offline RL.

Quatrièmement, cette thèse étudie les modes de défaillance des méthodes d'offline RL fondées sur les CVAE et identifie le phénomène de latent action projection space collapse, par lequel l'effondrement du posterior affaiblit l'optimisation de politique en aval. Sur cette base, la thèse propose Expand Latent Action Projection SpacE (ELAPSE), une méthode qui élargit l'espace de projection. ELAPSE améliore l'articulation entre la modélisation générative du comportement et l'optimisation de politique, et les résultats empiriques montrent qu'elle améliore substantiellement les performances de méthodes représentatives d'offline RL fondées sur les CVAE.

Dans l'ensemble, cette thèse étudie l'offline RL sans modèle sous des angles méthodologique et algorithmique. Sur le plan analytique, elle clarifie les défis et l'évolution méthodologique de l'offline RL. Sur le plan algorithmique, elle développe de nouvelles méthodes pour l'évaluation et l'amélioration de politique ainsi que pour la modélisation générative du comportement. Ensemble, ces contributions font progresser l'offline RL vers une prise de décision sur données statiques plus fiable, plus adaptative et mieux fondée théoriquement.

Title : A Systematic Study of Offline Reinforcement Learning : Methodological Evolution and Algorithmic Design

Keywords : Offline Reinforcement Learning ; Methodological Evolution ; Policy Evaluation ; Policy Improvement ; Policy Constraints ; Conditional Variational Autoencoders

Abstract : Offline Reinforcement Learning (offline RL) seeks to learn decision-making policies from fixed datasets without further interaction with the environment. It is especially appealing in domains where online collection is costly, risky, or infeasible. However, offline RL faces fundamental challenges, including distribution shift, out-of-distribution (OOD) actions, and extrapolation error, which make reliable policy evaluation and improvement difficult. This thesis studies model-free offline RL with two goals : to understand its methodological evolution and to develop more effective offline RL algorithms.

First, this thesis revisits offline RL through the lens of the deadly triad and analyzes its central challenges. Building on this perspective, it develops an algorithmic taxonomy of offline RL and traces the evolution of offline RL design principles. Rather than treating existing methods as isolated categories, the thesis highlights how major algorithmic families emerge and evolve in response to challenges such as value overestimation and distribution shift.

Second, to address the tension between in-sample learning and multi-step learning, Projective Implicit Q-Learning (PIQL) is proposed. PIQL replaces the fixed expectile in Implicit Q-Learning with a projection-based adaptive parameter, enabling a multi-step interpretation while preserving the in-sample learning property. In addition, PIQL adopts a support-constrained policy-improvement objective that is better aligned with the policy-evaluation stage. Theoretical analysis establishes monotonic policy improvement and a progressively stricter criterion for advantageous actions, while experiments on standard benchmarks demonstrate strong performance.

Third, the thesis introduces the Continuous Constraint Interpolation (CCI) framework and the Automatic Constraint Policy Optimization (ACPO) algorithm to unify and adapt policy constraints. CCI provides a unified optimization view in which weighted behavior cloning (wBC), KL-based density regularization, and support constraints arise as special cases along a continuous constraint spectrum. Building on CCI, ACPO formulates adaptive constraint learning as a primal-dual optimization problem and automatically tunes the interpolation parameter. The thesis also derives a maximum-entropy performance difference lemma and performance lower bounds for both the optimal policy and its parametric projection. Experiments show that ACPO achieves strong and consistent results across diverse offline RL benchmarks.

Fourth, this thesis studies failure modes of CVAE-based offline RL and identifies latent action projection space collapse, a phenomenon whereby posterior collapse weakens downstream policy optimization. Based on this analysis, the thesis proposes Expand Latent Action Projection SpacE (ELAPSE), a simple yet effective method that enlarges the projection space. ELAPSE improves the cooperation between generative behavior modeling and policy optimization, and empirical results show that it substantially enhances the performance of representative CVAE-based offline RL methods.

Overall, this thesis studies model-free offline RL from both methodological and algorithmic perspectives. At the analytical level, it clarifies the key challenges and the methodological evolution of offline RL. At the algorithmic level, it develops new methods for policy evaluation, policy improvement, and generative behavior modeling. Together, these contributions advance offline RL toward more reliable, adaptive, and theoretically grounded decision-making from static datasets.

Institut Polytechnique de Paris
91120 Palaiseau, France