Dual RL: Unification and New Methods for Reinforcement and Imitation Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

The goal of reinforcement learning (RL) is to maximize the expected cumulative return. It has been shown that this objective can be represented by an optimization problem of the state-action visitation distribution under linear constraints [52]. The dual problem of this formulation, which we refer to as *dual RL*, is unconstrained and easier to optimize. We show that several state-of-the-art off-policy deep reinforcement learning (RL) algorithms, under both online and offline, RL and imitation learning (IL) settings, can be viewed as dual RL approaches in a unified framework. This unification provides a common ground to study and identify the components that contribute to the success of these methods and also reveals the common shortcomings across methods with new insights for improvement. Our analysis shows that prior off-policy imitation learning methods are based on an unrealistic coverage assumption and are minimizing a particular f-divergence between the visitation distributions of the learned policy and the expert policy. We propose a new method using a simple modification to the dual RL framework that allows for performant imitation learning with arbitrary off-policy data to obtain near-expert performance, without learning a discriminator. Further, by framing a recent SOTA offline RL method XQL [23] in the dual RL framework, we propose alternative choices to replace the Gumbel regression loss, which achieve improved performance and resolve the training instability issue of XQL.

1 Introduction

A number of deep Reinforcement Learning (RL) algorithms optimize a regularized policy learning objective using approximate dynamic programming (ADP) [7]. Popular off-policy temporal difference algorithms spanning both imitation learning [39, 59] and RL [27, 20, 28, 69, 34] exemplify this class. As we will discuss in Section 3, one way to develop a principled off-policy algorithm is to ensure unbiased estimation of the on-policy policy gradient using off-policy data [55]. Unfortunately, many classical off-policy algorithms do not guarantee this property, resulting in issues like training instability and over-estimation of the value function [17, 20, 4]. To obtain high learning performance, these algorithms require that most data to be nearly on-policy, otherwise require special algorithmic treatments (e.g., importance sampling [65], layer normalization [5], prioritized sampling [78]) to avoid the aforementioned issues. Recently, there have been developments leading to new off-policy algorithms with improved performance for RL [43, 22, 41] and IL [82, 48, 22, 15]. These methods are derived via a variety of mathematical tools and attribute their success in different aspects. It remains an open question if we can inspect these algorithms under a unified framework to understand their core advantages and limitations, and subsequently propose better methods.

In this work, we consider a specific formulation for RL that writes the performance of a policy as a convex objective with linear constraints [52]. The convex program can be converted into unconstrained forms using Lagrangian duality, which is more amenable for stochastic optimization. We refer to approaches that admit the dual formulations as *Dual RL*. Dual RL approaches naturally provide unbiased estimation of the on-policy policy gradient using off-policy data, in a principled

Submitted to 16th European Workshop on Reinforcement Learning (EWRL 2023). Do not distribute.

	Dual RL Method	Gradient	Objective	dual-Q/V	Non-Adversarial?	Off-Policy Data	Coverage Assumption
RL	AlgaeDICE [56], GenDICE [81], CQL [43]	semi	reg. RL	Q	✗	Arbitrary	—
	OptiDICE [45]	full	reg. RL	V	✓	Arbitrary	—
	XQL [23], REPS [61], f-**DVL**	semi	reg. RL	V	✓	Arbitrary	—
	VIP [49], GoFAR [50]	full	reg. RL	V	✓	Arbitrary	—
	Logistic Q-learning [6]	full	reg. RL	QV[1]	✓	✗	—
IL	IQLearn [22], IBC [15]	semi	$D_f(\rho^\pi \Vert \rho^E)$	Q	✓	Expert-only	✗
	IVLearn	semi	$D_f(\rho^\pi \Vert \rho^E)$	V	✓	Expert-only	✗
	OPOLO [82], OPIRL [32]	semi	$D_{rkl}(\rho^\pi \Vert \rho^E)$	Q	✗	Arbitrary	✓
	ValueDICE [40]	semi	$D_{rkl}(\rho^\pi \Vert \rho^E)$	Q	✗	Arbitrary	✓
	SMODICE [48]	full	$D_{rkl}(\rho^\pi \Vert \rho^E)$	V	✓	Arbitrary	✓
	DemoDICE [38], LobsDICE [37]	full	$D_{rkl}(\rho^\pi \Vert \rho^E) + \alpha D_{rkl}(\rho^\pi \Vert \rho^R)$	V	✓	Arbitrary	✓
	P^2IL [79]	full	$D_C(\rho^\pi \Vert \rho^E)$[1]	QV[1]	✗	✗	✗
	ReCOIL-Q	full	$D_f(\rho^\pi_{mix} \Vert \rho^{E,R}_{mix})$	Q	✗	Arbitrary	✗
	ReCOIL-V	full	$D_f(\rho^\pi_{mix} \Vert \rho^{E,R}_{mix})$	V	✓	Arbitrary	✗

Table 1: A number of recent works can be studied together under the unified umbrella of **dual-RL**. These methods are instantiations of dual-RL with a choice of update strategy, objective, constraints, and their ability to handle off-policy data. **Bold** names correspond to the methods proposed in the paper.

way. They avoid explicit importance sampling that leads to high variance and ensures training stability and convergence [76]. Related approaches in this space have often been referred to as DICE (DIstribution Correction Estimation) methods in previous literature [56, 40, 45, 48, 81]. We note that the linear programming formulation of policy performance has been used and studied in [52, 13, 12, 8, 30, 11, 62, 51, 44]. The general duality framework was first introduced by Nachum and Dai [55]. Our work focus on formulating and studying properties of off-policy algorithms by utilizing this tool.

Our first contribution is that we show that many recent algorithms in deep reinforcement learning and imitation learning [23, 82, 43, 22, 15] can be all viewed as different instantiations of dual problems of regularized policy optimization, see Table 1 for the complete list. These algorithms have been motivated from a variety of perspectives. For example, XQL [23] focuses on introducing Gumbel regression into RL, CQL [43] aims at learning a pessimistic Q function, IQLearn [22] and OPOLO [82] use the change of variables for IL, and IBC [15] uses a contrastive loss for imitation learning. Even though these approaches have different derivations, we extend the work of Nachum and Dai [55] and show they can be unified under the framework of dual-RL in Sections 4 and 5.

Second, the presented unification provides a framework to evaluate and analyze which factors actually make the algorithm better or worse. We examine this in the context of XQL, whose success was attributed to better modeling of Bellman errors using Gumbel regression. On the other hand, XQL also suffers from the training instability of Gumbel regression. By situating the implicit policy improvement algorithms like XQL in the dual RL framework, in Section 5 we are able to propose a family of implicit algorithms f-Dual V Learning (f-`DVL`), which successfully addresses the training instabilities issue. The empirical experiments on the D4RL benchmarks establish the superior performance of f-`DVL`, see Section 6.

Third, building upon the dual framework, in Section 4 we propose a new algorithm for off-policy imitation learning that is able to leverage arbitrary off-policy data to learn near-expert policies, getting rid of the unrealistic coverage assumption (the suboptimal data covers the visitations of the expert data) required by previous works [48, 82, 38], and also eliminating the need for a discriminator. Our resulting algorithm, `ReCOIL`, is simple, theoretically principled, non-adversarial, and admits a single-player optimization in contrast to previous works in imitation [24, 31, 16, 68]. We empirically demonstrate the failure of previous IL methods based on the coverage assumption in a number of MuJoCo environments, and show substantial performance improvements of `ReCOIL` in Section 6.

2 Related Work

Off-Policy Methods for RL Off-policy RL methods promise a way to utilize data collected by arbitrary behavior policies to aid in learning an optimal policy and thus are advantageous over on-policy methods. This promise falls short, as previous off-policy algorithms are plagued with a number of issues such as overestimation of the value function, training instability, and various biases [74, 17, 20, 42]. Previous works have approached these issues for online RL using methods like double-Q learning [29], target networks [54], emphatic weightings [35, 33], and so on. Unfortunately, these approaches do not carry over well to the offline setting. For example, when deploying the policy online, the overestimation bias can be correctly by the environment feedbacks, which is infeasible for offline RL. A number of solutions exist for controlling overestimation in prior work—f-divergence regularization to the training distribution [80, 57, 21, 19], support regularization [70], implicit

[1]These methods use a different regularizer. More details in Appendix C.5.

maximization [41] and learning a Q function that penalizes OOD actions [43]. A recent method XQL [23] proposes Gumbel regression as a better tool to model Bellman errors and achieve significant gains in learning performance across online and offline RL.

Another common issue for previous off-policy algorithms is *distribution mismatch*. As we shall discuss later, the RL objective requires on-policy samples but is often estimated by off-policy samples in practice. Prior works have proposed fixing the distribution mismatch by using importance weights [64], which can lead to high variance policy gradients or ignoring the distribution mismatch completely [27, 20]. The dual RL framework [55] fixes the distribution mismatch issue in a principled way. It should come as no surprise that some of the most performant RL algorithms in the space are known dual methods (Online RL [56], Offline RL [45]).

Off-Policy Methods for IL Imitation learning has benefited greatly from using off-policy data to improve learning performance [39, 60, 68, 82]. Often, replacing the on-policy expectation common in most Inverse RL formulations [83, 73] by expectation under off-policy samples, which is unprincipled, has led to gains in sample efficiency [39]. Previous works have proposed a solution in the dual RL space for principled off-policy imitation but is based on a restrictive coverage assumption [48, 82, 38] and limit themselves to matching a particular f-divergence. In this work, we eliminate this assumption and allow for generalizing to all f-divergences, presenting a principled off-policy approach to imitation. Our work also presents an approach that allows for single-player non-adversarial optimization for imitation learning, in contrast to previous work [40].

3 Preliminaries

We consider an infinite horizon discounted Markov Decision Process denoted by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, r, \gamma, d_0)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, p is the transition probability function, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $\gamma \in (0, 1)$ is the discount factor, and d_0 is the distribution of initial state s_0. Let $\Delta(\mathcal{A})$ denote the probability simplex supported on $\mathcal{A}$. The goal of RL is to find a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ that maximizes the expected return: $\mathbb{E}_\pi\big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\big]$, where we use $\mathbb{E}_\pi$ to denote the expectation under the distribution induced by $a_t \sim \pi(\cdot|s_t), s_{t+1} \sim p(\cdot|s_t, a_t)$. We also define the discounted state-action visitation distribution $d^\pi(s, a) = \pi(a|s) \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\pi)$. The unique stationary policy that induces a visitation $d(s, a)$ is given by $\pi(a|s) = d(s, a)/\sum_a d(s, a)$. We will use d^O and d^E to denote the visitation distributions of the behavior policy of the offline dataset and the expert policy, respectively.

Value Functions and Bellman Operators Let V^π: $\mathcal{S} \to \mathbb{R}$ be the state value function of π. $V^\pi(s)$ is the expected return when starting from s and following π: $V^\pi(s) = \mathbb{E}_\pi\big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)|s_0 = s\big]$. Similarly, let $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ be the state-action value function of π, such that $Q^\pi(s, a) = \mathbb{E}_\pi\big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)|s_0 = s, a_0 = a\big]$. Let V^* and Q^* denote the value functions corresponding to an optimal policy π^*. Let $\mathcal{T}_r^\pi$ be the Bellman operator with policy π and reward function r such that $\mathcal{T}_r^\pi Q(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a), a' \sim \pi(\cdot|s')}[Q(s', a')]$. We also define the Bellman operator for the state value function $\mathcal{T}_r V(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)}[V(s')]$.

f-Divergence f-Divergence measures distance between two probability distribution P and Q given by: $D_f(P \mid\mid Q) = \mathbb{E}_{z\sim Q}[f\big(\frac{P(z)}{Q(z)}\big)]$. The convex conjugate of f is the function $f^*(y) = \sup_{x \in \mathbb{R}_+} [xy - f(x)]$. For a more formal overview of the above concepts, refer to Appendix C.1.

3.1 Reinforcement Learning via Lagrangian Duality

Reinforcement learning optimizes the expected return of a policy. We consider the linear programming formulation of the expected return [52], to which we can apply Lagrangian duality to obtain corresponding constraint-free problems. We here review the framework introduced by Nachum and Dai [55], which obtains the same formulations as ours via Fenchel-Rockfeller duality, yet we use Lagrangian duality for its simplicity and popularity. Consider the regularized policy learning problem

$$\max_\pi J(\pi) = \mathbb{E}_{d^\pi(s,a)}[r(s, a)] - \alpha D_f(d^\pi(s, a) \mid\mid d^O(s, a)), \tag{1}$$

where $D_f(d^\pi(s, a) \mid\mid d^O(s, a))$ is a conservatism regularizer that encourages the visitation distribution of π to stay close to some distribution d^O, and α is a temperature parameter that balances the expected return and the conservatism. An interesting fact is that $J(\pi)$ can be rewritten as a convex problem that searches for a visitation distribution that satisfies the *Bellman-flow* constraints. We refer to this form as `primal-Q`:

$$\texttt{primal-Q} \quad \max_{\pi} J(\pi) = \max_{\pi} \big[\max_{d} \mathbb{E}_{d(s,a)}[r(s,a)] - \alpha D_f(d(s,a) \,||\, d^O(s,a)) \\ \text{s.t } d(s,a) = (1-\gamma)d_0(s).\pi(a|s) + \gamma \textstyle\sum_{s',a'} d(s',a')p(s|s',a')\pi(a|s), \ \forall s \in \mathcal{S}, a \in \mathcal{A}. \big] \tag{2}$$

Applying Lagrangian duality and convex conjugate to this problem, we can convert it to an unconstrained problem with dual variables $Q(s,a)$ defined for all $s,a \in \mathcal{S} \times \mathcal{A}$, giving us the `dual-Q` formulation:

$$\texttt{dual-Q} \quad \max_{\pi} \min_{Q} (1-\gamma)\mathbb{E}_{s\sim d_0, a\sim\pi(s)}[Q(s,a)] + \alpha \mathbb{E}_{(s,a)\sim d^O}[f^*([\mathcal{T}_r^\pi Q(s,a) - Q(s,a)]/\alpha)], \tag{3}$$

where f^* is the convex conjugate of f. Problem (2) is overconstrained—the constraints determine the unique solution d^π rendering the inner maximization w.r.t d unnecessary. In fact, we can relax the constraints to obtain another problem with the same optimal solution π^* and d^*, which we call `primal-V` below:

$$\texttt{primal-V} \quad \max_{d \geqslant 0} \mathbb{E}_{d(s,a)}[r(s,a)] - \alpha D_f(d(s,a) \,||\, d^O(s,a)) \\ \text{s.t } \textstyle\sum_{a\in\mathcal{A}} d(s,a) = (1-\gamma)d_0(s) + \gamma \sum_{(s',a')\in\mathcal{S}\times\mathcal{A}} d(s',a')p(s|s',a'), \ \forall s \in \mathcal{S}. \tag{4}$$

Similarly, we consider the Lagrangian dual of (4), with dual variables $V(s)$ defined for all $s \in S$:

$$\texttt{dual-V} \quad \min_{V} (1-\gamma)\mathbb{E}_{s\sim d_0}[V(s)] + \alpha \mathbb{E}_{(s,a)\sim d^O}\big[f_p^*([\mathcal{T}V(s,a) - V(s))]/\alpha)\big], \tag{5}$$

where f_p^* is a variant of f^* defined in Eq. (45). Such modification is to cope with the nonnegativity constraint $d(s,a) \geqslant 0$ in `primal-V`. Note that in both cases for `dual-Q` and `dual-V`, the optimal solution is same as their primal formulations due to strong convexity. See Appendix C.1 for a detailed review, connections between Fenchel and Lagrangian duality and discussion of computing π^* from V^* for the `dual-V` formulation.

Remarks. The dual formulations have a few appealing properties. (a) They allow us to transform constrained distribution-matching problems, w.r.t previously logged data, into unconstrained forms. (b) One can show that the gradient of `dual-Q` w.r.t π, when Q is optimized for the inner problem, is the on-policy policy gradient computed by off-policy data [56, 55]. This key property relieves the instability or divergence issue in off-policy learning [74, 17, 20, 42]. (c) The dual framework can be extended to the max-entropy RL setting, where $J(\pi)$ consists of additional entropy regularization, by replacing Bellman-operator with their soft Bellman counterparts [26].

4 Imitation Learning from Dual Perspective

Imitation learning is the setting where an agent does not have access to the reward when interacting with the environment. Instead, it is given a set of reward-free demonstrations, i.e. state-action trajectories. For ease of presentation, we start with the standard *offline IL* setup in Section 4.1, where the demonstrations are generated by expert agents. We show how IQLearn [22] and IBC [15] can be written as `dual-Q` problems. Next, we consider the setting in which the agents receive additional suboptimal demonstrations, where we rewrite OPOLO [82] in the `dual-Q` formulation. Finally, we discuss how these formulations and algorithms further extend to *online IL* under mild modifications. The process of unifying those algorithms helps us identify shortcomings and unprincipled components in them, and we propose a novel algorithm `ReCOIL` that eliminates those downsides in Section 4.2.

4.1 Dual Formulation for Existing Off-Policy Imitation Learning Algorithms

Offline IL with Expert Data Only A straightforward application of our `dual-Q` formulation to offline IL is to simply set the reward to be uniformly 0 across the state-action space and set the regularization distribution d^O to be the expert visitation distribution d^E. That is,

$$\texttt{dual-Q} \quad \max_{\pi} \min_{Q} (1-\gamma)\mathbb{E}_{d_0(s),\pi(a|s)}[Q(s,a)] + \alpha \mathbb{E}_{s,a\sim d^E}[f^*([\mathcal{T}_0^\pi Q(s,a) - Q(s,a)]/\alpha)]. \tag{6}$$

Interestingly, this reduction directly leads us to a family of IL methods IQLearn [22], which was derived using a change of variables in the form of an inverse backup operator.

Lemma 1. *IQLearn [22] is an instance of* `dual-Q` *using the semi-gradient update rule with a soft-Bellman operator, where $r(s,a) = 0\, \forall s \in \mathcal{S}, a \in \mathcal{A}$, $d^O = d^E$.*

We also find that IBC [15], an offline IL method that performs behavior cloning using a contrastive objective, is a special case of IQLearn and consequently of the `dual-Q` form.

Corollary 1. *IBC [15] is an instance of* `dual-Q` *using the full-gradient update rule, where $r(s,a) = 0 \ \forall s \in \mathcal{S}, a \in \mathcal{A}$, $d^O = d^E$, and the f-divergence is the total variation distance.*

Offline IL with Additional Suboptimal Data The `dual-Q` and `dual-V` formulations do not naturally incorporate additional suboptimal data. To remedy this, prior methods have relied on careful selection of the f-divergence and a *coverage assumption* that allows them to craft an off-policy objective for imitation learning [82, 32, 48, 38, 37]. More precisely, under the *coverage assumption* that the suboptimal data visitation (denoted by d^S) covers the expert visitation ($d^S > 0$ wherever $d^E > 0$) [48], and using the reverse KL divergence, we get the following `dual-Q` problem:

$$\texttt{dual-Q} \quad \max_{\pi(a|s)} \min_{Q(s,a)} (1-\gamma)\mathbb{E}_{\rho_0(s),\pi(a|s)}[Q(s,a)] + \mathbb{E}_{s,a\sim d^S}\left[f^*(\mathcal{T}^{\pi}_{r^{imit}}Q(s,a) - Q(s,a))\right], \tag{7}$$

where $\mathcal{T}_{r^{\text{imit}}}$ denote Bellman operator under the *pseudo-reward* function $r^{\text{imit}}(s,a) = -\log \frac{d^S(s,a)}{d^E(s,a)}$. This leads us to a reduction of another IL method OPOLO [82] to `dual-Q`.

Lemma 2. *OPOLO [82] is an instance of* `dual-Q` *using the semi-gradient update rule, where* $r(s,a) = 0\ \forall \mathcal{S}, \mathcal{A}$, $d^O = d^E$*, and the f-divergence set to the reverse KL divergence.*

We note that the `dual-V` framework for off-policy imitation learning under coverage assumptions with a full-gradient update rule was studied in SMODICE [48].

From Offline to Online Problem (7) naturally extends to online IL, as the suboptimal data does not need to be static—it can be the replay buffer during online training. The corresponding algorithms generalize as well, since their key component is estimating the Q^π function using *off-policy* data. It is worth noting that d^S is dynamically changing for online IL. In contrast, Eq. (6) cannot be extended to online IL. Garg et al. [22] uses IQLearn in the online setting where they add additional regularization using bellman backups on d^S. Our results suggest this to be unprincipled (also pointed out by Al-Hafez et al. [3]), as only expert data samples can be leveraged in this formulation.

4.2 `ReCOIL`: Imitation Learning from Arbitrary Experience

As demonstrated in Section 4.1, previous off-policy IL methods often rely on the coverage assumption [48, 82, 38, 36], and many of them need to train a discriminator between the demonstration data and the policy generated data to obtain the pseudo-reward r^{imit}. We propose **RE**laxed **C**overage for **O**ff-policy **I**mitation **L**earning (`ReCOIL`), an off-policy IL algorithm that eliminates the need for both, the coverage assumption and the discriminator.

Let $d^S_{\text{mix}} := \beta d(s,a) + (1-\beta)d^S(s,a)$ and $d^{E,S}_{\text{mix}} := \beta d^E(s,a) + (1-\beta)d^S(s,a)$, where $\beta \in (0,1)$ is a fixed hyperparameter. We consider the following problem in `primal-V` form:

$$\texttt{primal-V} \quad \max_{d(s,a)\geqslant 0} -D_f(d^S_{\text{mix}}(s,a) \,||\, d^{E,S}_{\text{mix}}(s,a))$$
$$\text{s.t} \sum_{a\in\mathcal{A}} d(s,a) = (1-\gamma)d_0(s) + \gamma \sum_{(s',a')\in\mathcal{S}\times\mathcal{A}} d(s',a')p(s|s',a'),\ \forall s\in\mathcal{S}. \tag{8}$$

This is a valid imitation learning formulation [24, 36, 60, 68] since the global maximum of the objective is attained at $d = d^E$ which also satisfies the above constraints, irrespective of the suboptimal data distribution d^S. Therefore, the corresponding `dual-V` formulation, which we dub `ReCOIL-V`, can be leveraged to solve the IL problem:

$$\texttt{ReCOIL-V} \quad \min_V \beta(1-\gamma)\mathbb{E}_{s\sim d_0}[V(s)] + \mathbb{E}_{(s,a)\sim d^{E,S}_{\text{mix}}}\left[f^*_p(\mathcal{T}_0 V(s,a) - V(s))\right] - (1-\beta)\mathbb{E}_{(s,a)\sim d^S}[\mathcal{T}_0 V(s,a) - V(s)]. \tag{9}$$

Lemma 3. *For any visitation distribution* d^S *and any* $\beta \in (0,1)$*, the solution of off-policy objective* `ReCOIL-V` *is* V^* *that corresponds to an optimal policy* π^*.

In other words, imitation learning can be solved by optimizing the unconstrained problem `ReCOIL-V` with arbitrary off-policy data, without the coverage assumption. Besides, as opposed to many previous algorithms, `ReCOIL-V` uses the Bellman operator $\mathcal{T}_0$ which does not need the pseudo-reward r^{imit}, therefore it is discriminator-free. Although the pseudo-reward is not needed for training, `ReCOIL-V` allows for recovering the reward function using the learned V^* which corresponds to the intent of the expert. That is, $r(s,a) = V^*(s) - \mathcal{T}_0(V^*(s,a))$. Moreover, our method is generic to incorporate any f-divergence. The complete algorithm for `ReCOIL-V` can be found in Algorithm 1 in Appendix E. The `primal-Q` form for mixture distribution can be similarly specified, whose dual problem `ReCOIL-Q` also solves IL with any off-policy data, see Lemma 7 in Appendix D.

$$\texttt{ReCOIL-Q} \quad \max_\pi \min_Q \beta(1-\gamma)\mathbb{E}_{d_0,\pi}[Q(s,a)] + \mathbb{E}_{s,a\sim d^{E,S}_{\text{mix}}}\left[f^*_p(\mathcal{T}^\pi_0 Q(s,a) - Q(s,a))\right] - (1-\beta)\mathbb{E}_{s,a\sim d^S}[\mathcal{T}^\pi_0 Q(s,a) - Q(s,a)] \tag{10}$$

A Bellman Consistent Energy-Based Model (EBM) View for ReCOIL Instantiating `ReCOIL-Q` with Pearson χ^2 Divergence, we obtain the following problem:

$$\max_\pi \min_Q \beta(\mathbb{E}_{d^S,\pi(a|s)}[Q(s,a)] - \mathbb{E}_{d^E(s,a)}[Q(s,a)]) + \underbrace{\mathbb{E}_{s,a\sim d^{E,S}_{\text{mix}}(s,a)}\left[(\gamma Q(s',\pi(s')) - Q(s,a))^2\right]}_{\text{Bellman consistency}}. \tag{11}$$

One can see that `ReCOIL-Q` aims to learn a score function Q whose expected value is low over the suboptimal distribution but high over the expert distribution, while ensuring that Q is Bellman consistent over the mixture. The Bellman consistency is crucial to propagate the information of how to recover when the policy makes a mistake. The Q value can be interpreted as a score as it is not representative of any expected return, and we can view `ReCOIL-Q` as an energy-based model with Bellman consistency. Figure 1 illustrated this intuition.

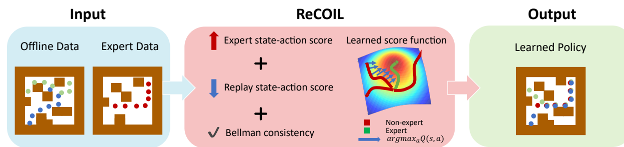

Figure 1: Recipe for ReCOIL: Learn a Bellman consistent EBM - A model which increases the score of expert transitions, and decreases the score of replay transitions while maintaining Bellman consistency throughout.

Theorem 1 (Suboptimality Bound for Offline `ReCOIL`)**.** *Let S^J denote the joint support of d^S and d^E. Let $r(s,a) = V(s) - \mathcal{T}_0 V(s,a)$ be the pseudo-reward implied by `ReCOIL` and $R_{\max} = \max_{s,a} r(s,a)$. Let $D_\delta = \left\{ d \,|\, \Pr_d\left((s,a) \in S^J\right) \geqslant 1-\delta \right\}$ be the set of visitation distributions that have $1-\delta$ coverage of S^J. Let π^*_δ be the best policy over all policies whose visitation distribution is in D_δ. Let $g(d,V) = (1-\gamma)\mathbb{E}_{d_0(s)}[V(s)] + \mathbb{E}_d[\mathcal{T}_0 V(s,a) - V(s)] - D_f(d(s,a) \,||\, d^E(s,a))$ be the imitation learning objective, and $h(V) = \max_{d \in D_\delta} g(d,V)$. Suppose that we can solve `ReCOIL` with the constraint $d \in D_\delta$, h is κ-strongly convex in V and $\beta \to 1$, then the output policy $\widehat{\pi}$ satisfies that $J(\pi^*_\delta) - J(\widehat{\pi}) \leqslant \frac{4}{1-\gamma}\sqrt{2\delta R_{\max}/\kappa}$.*

Theorem 1 bounds that the performance gap between the `ReCOIL` policy and the best imitation policy with visitation in the joint support of expert data distribution d^E and suboptimal distribution d^S. In Appendix D.1, we further discuss how `ReCOIL` obtains a stronger performance guarantee compared to `IQLearn` and how `ReCOIL` ensures a search among policies with the support constraint in practice. Moreover, our method implicitly learns a distribution ratio $d^S_{\text{mix}}/d^{E,S}_{\text{mix}}$ which is well-defined for all the suboptimal and expert transitions ($d^S > 0$ or $d^E > 0$) that the policy is trained on. While `ReCOIL` utilizes additional suboptimal data, we also leverage the `dual-V` formulation to obtain a novel method `IVLearn` for offline imitation learning with expert-data only. Due to space constraints, we defer the discussion to Appendix C.3.1.

5 Reinforcement Learning from Dual Perspective

Regularized policy optimization, in its various forms [21, 1, 69, 80], is a natural objective for off-policy algorithms, in both offline and online settings. In offline RL, various types of conservatism notions have been proposed to prevent overestimated Q-values for offline RL, which can lead to huge extrapolation error [17, 20]. Two notable frameworks for offline RL are pessimistic value learning e.g. CQL [43] and implicit policy improvement algorithms including IQL [41] and XQL [23]. These frameworks have seemingly been exceptions to the regularized policy optimization formulation (Eq. (1) and (4)). Nonetheless, our results in Section 5.1, first to our knowledge, formulate both of them as instances of dual methods, which are solving regularized policy optimization in essence. Such unification also inspires us to propose f**-DVL**, a new approach under this framework in Section 5.2.

5.1 Dual Formulations for Existing Off-Policy Reinforcement Learning Algorithms

Lemma 4. *CQL is an instance of `dual-Q` under the semi-gradient update rule, where the f-divergence is the Pearson χ^2 divergence, and d^O is the offline visitation distribution.*

Kumar et al. [43] shows that CQL outperforms a family of behavior-regularized offline RL methods [20, 80, 57], which solve different forms of `primal-Q` using approximate dynamic programming. The above result indicates that CQL's better performance is likely due to the choice of f-divergence and more amenable optimization afforded by the dual formulation. Moreover, the same `dual-Q` formulation has been previously studied for online RL in AlgaeDICE [56], and Lemma 4 suggests that CQL is an offline version of AlgaeDICE.

Next, we show that `dual-V` subsumes a family of implicit policy improvement methods for offline RL, thus tying together all three types of methods – policy regularized, pessimistic value function, and implicit maximization based as instances of `primal-Q`, `dual-Q` and `dual-V` respectively. We formalize the reduction of XQL, a recent implicit policy improvement method, to `dual-V` below.

Lemma 5. *XQL is an instance of `dual-V` under the semi-gradient update rule, where the f-divergence is the reverse Kullback-Liebler divergence, and d^O is the offline visitation distribution.*

We also highlight that the full-gradient variant of the `dual-V` framework for offline RL has been studied extensively in OptiDICE [45] and Lemma 5 highlights that XQL is a special case OptiDICE with a semi-gradient update rule.

From Offline to Online Again, all the above-discussed offline methods naturally extend to online settings [41, 23, 58], as their off-policy nature extends beyond the offline setup. Our analysis still holds, where the regularization distribution d^O becomes the visitation distribution of the replay buffer d^R. It is worth noting that d^R is dynamically changing over the course of training.

5.2 f-`DVL` (Dual-V Learning): Better Implicit Maximizers for Offline RL

The success of XQL was attributed to the property that Gumbel distribution better models the Bellman errors [23]. Despite its decent performance, XQL is prone to training instability (see e.g., Figure 3), since the Gumbel loss is an exponential function that can produce large gradient during training. Lemma 5 shows that XQL is a particular `dual-V` problem where the Gumbel loss is the conjugate f_p^* corresponding to reserve KL divergence. This inspires us to extend XQL by choosing different f-divergences, where the conjugate functions are more optimization amenable. We further show that the proposed methods enjoy both improved performance and better training stability in Section 6.

Implicit policy improvement algorithms iterate two steps alternately: 1) regress $Q(s,a)$ to $r(s,a) + \gamma V(s')$ for transition (s,a,s'), 2) estimate $V(s) = \max_{a\in A} Q(s,a)$. The learned Q, V functions can be used to extract policy as for the `dual-V` formulation, see Appendix C.1.6. Step 1) is akin to the *policy evaluation* step of generalized policy iteration (GPI), and step 2) acts like the *policy improvement* step without explicitly learning a policy $\pi(s) = \arg\max_a Q(s,a)$. The crux is to conservatively estimate the maximum of Q in step 2.

Consider a rewriting of `dual-V` with the temperature parameter λ:

$$\min_V (1-\lambda)\mathbb{E}_{s\sim d^O}[V(s)] + \lambda \mathbb{E}_{(s,a)\sim d^O}\left[f_p^*\left(\bar{Q}(s,a) - V(s)\right)\right], \tag{12}$$

where $\bar{Q}(s,a)$ denotes $\texttt{stop-gradient}(r(s,a) + \gamma \sum_{s'} p(s'|s,a)V(s'))$. Let x be a random variable of distribution D. Problem (12) can be considered as a special form of the problem below:

$$\min_v (1-\lambda)v + \lambda \mathbb{E}_{x\sim D}\left[f_p^*\left(x - v\right)\right], \tag{13}$$

where x is analogous to $\bar{Q}$ and v is analogous to V. As opposed to the handcrafted choices [41, 23], we show through Lemma 6 below that as $\lambda \to 1$, problem (13) naturally gives rise to a family of *implicit maximizers* that estimates $\sup_{x\sim D} x$.

Lemma 6. *Let x be a real-valued random variable such that $\Pr(x > x^*) = 0$. Let v_λ be the solution of Problem (13). It holds that $v_{\lambda_1} \leqslant v_{\lambda_2}, \forall\, 0 < \lambda_1 < \lambda_2 < 1$. Further, $\lim_{\lambda\to 1} v_\lambda = x^*$.*

We propose a family of maximizers associated with different f-divergences and apply them to `dual-V`. We call the resulting methods f-`DVL` (Dual-V Learning), and the complete algorithm can be found in Appendix E.3. Particularly, we consider the two maximizers that correspond to (1) Total Variation: $f(x) = \frac{1}{2}|x-1|, f_p^*(y) = \max(y, 0)$, (2) Pearson χ^2 divergence: $f(x) = (x-1)^2,\ f_p^*(y) = \max(\frac{1}{4}y^2 + y, 0)$. See Figure 6 for an illustration. Recall that XQL uses the implicit maximizer associated with reserve KL divergence where f_p^* is exponential. Compared with XQL, our f_p^* functions are low-order polynomials and are thus stable for optimization.

6 Experiments

Our experiments aim to answer the following four questions. **IL:** 1) How does `ReCOIL` perform and compare with previous IL methods? 2) Can `ReCOIL` accurately estimate the policy visitation distribution d^π and the reward function/intent of the expert? **RL:** 3) How does f-`DVL` perform and compare with previous RL methods? 4) Is the training of f-`DVL` more stable than XQL?

In order to circumvent the intricacies associated with exploration and direct our attention towards the intrinsic nature of dual RL formulation, we focus on the offline setting in this section, although the approaches can also be applied to online settings. We consider the locomotion and manipulation tasks from the D4RL benchmark [18], and report the results in Section 6.1 and 6.2, respectively. For each algorithm, we train 7 instances with different seeds and report their average return and standard derivation. Full experiment details can be found in Appendix E.

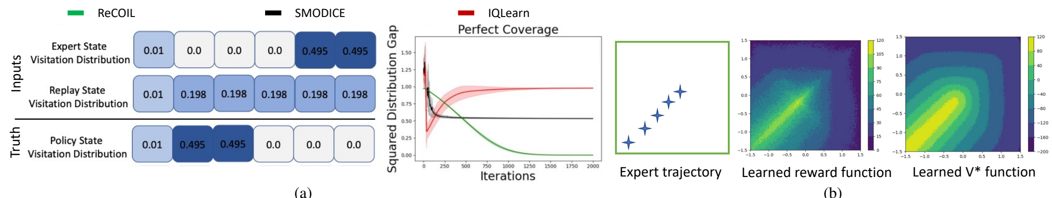

Figure 2: (a) The replay buffer distribution covers the policy visitation distribution d^{π}. `ReCOIL` perfectly infers d^{π} whereas a method that only relies on expert data or the replay data with the coverage assumption fails. Results averaged over 100 seeds. (b) Recovered R and V^* on a simple grid-world environment by `ReCOIL`.

6.1 Offline IL

Benchmark Comparisons For every task, our agent is given 1 expert demonstration and a set of suboptimal transitions, both extracted from the D4RL datasets. We follow the construction of suboptimal dataset in SMODICE [48]. For locomotion tasks, the suboptimal dataset consists of 1 million transitions of the `random` or `medium` D4RL datasets and 200 expert demonstrations, which we label as `random+expert` and `medium+expert`, respectively. We also consider suboptimal datasets mixed with only 30 expert demonstrations, which are called `random+few-expert` and `medium+few-expert`. Similarly, we construct datasets for the manipulation tasks. More details can be found in Appendix E.2.

Suboptimal Dataset	Env	RCE	ORIL	SMODICE	ReCOIL
random+ expert	hopper	51.41±38.63	73.93±11.06	101.61±7.69	**108.18±3.28**
	halfcheetah	64.19±11.06	60.49±3.53	80.16±7.30	**80.20±6.61**
	walker2d	20.90±26.80	2.86±3.39	**105.86±3.47**	102.16±7.19
	ant	105.38±14.15	73.67±12.69	**126.78±5.12**	**126.74±4.63**
random+ few-expert	hopper	25.31±18.97	42.04±13.76	60.11±18.28	**97.85±17.89**
	halfcheetah	2.99±1.07	2.84±5.52	2.28±0.62	**76.92±7.53**
	walker2d	40.49±26.52	3.22±3.29	**107.18±1.87**	83.23±19.00
	ant	**67.62±15.81**	25.41 ± 8.58	-6.10±7.85	**67.14± 8.30**
medium+ expert	hopper	58.71±34.06	61.68±7.61	49.74±3.62	**88.51±16.73**
	halfcheetah	65.14±13.82	54.66±0.88	59.50±0.82	**81.15±2.84**
	walker2d	96.24±14.04	8.19±7.70	2.62±0.93	**108.54±1.81**
	ant	86.14±38.59	102.74±6.63	104.95±6.43	**120.36±7.67**
medium few-expert	hopper	66.15±35.16	17.40±15.15	47.61±7.08	**50.01±10.36**
	halfcheetah	61.14±18.31	43.24±0.75	46.45±3.12	**75.96±4.54**
	walker2d	85.28±34.90	6.81±6.76	6.00±6.69	**91.25±17.63**
	ant	67.95±36.78	81.53±8.618	81.53±8.618	**110.38±10.96**
cloned+expert	pen	19.60±11.40	-3.10±0.40	-3.36±0.71	**95.04±4.48**
	door	0.08± 0.15	-0.33±0.01	0.25± 0.54	**102.75±4.05**
	hammer	1.95±3.89	0.25± 0.01	0.15± 0.078	**95.77±17.90**
	relocate	-0.25±0.04	-0.29±0.01	1.75±3.85	**67.43±14.60**
human+expert	pen	17.81±5.91	-3.38±2.29	-2.20±2.40	**103.72±2.90**
	door	-0.05±0.05	-0.33±0.01	-0.20± 0.11	**104.70±0.55**
	hammer	5.00±5.64	1.89±0.70	-0.07±0.39	**125.19±3.29**
	relocate	0.02±0.10	-0.29±0.01	-0.16±0.04	**91.98± 2.89**
partial+expert	kitchen	6.875±9.24	0.00±0.00	39.16± 1.17	**60.0±5.70**
mixed+expert	kitchen	1.66±2.35	0.00±0.00	42.5±2.04	**52.0±1.0**

Table 2: The normalized return obtained by different offline IL methods trained on the D4RL suboptimal datasets with 1000 expert transitions.

We compare `ReCOIL-V` against recent offline IL methods RCE [14], SMODICE [48] and ORIL [84]. We do not compare to DEMODICE [38] as SMODICE was shown to be competitive in Ma et al. [48]. Both SMODICE and ORIL require learning a discriminator, and SMODICE is built upon the coverage assumption. RCE also uses a recursive discriminator to test the proximity of the policy visitations to successful examples. In contrast, `ReCOIL-V` is discriminator-free and does not need this coverage assumption. Table 2 reports the results. `ReCOIL` strongly outperforms the baselines in most environments. SMODICE shows poor performance in cases when the combined offline dataset has low expert coverage (`random+few-expert`) or where the discriminator can easily overfit (high-dimensional environments like dextrous manipulation).

Estimation of the Policy Visitation Distribution and Reward Recovery Correctly estimating a given policy's visitation distribution d^{π} is key to testing its closeness to the expert visitation. For both `ReCOIL-Q` and `ReCOIL-V`, d^{π} can be computed via Eq (47) (appendix). Here we present the results obtained by `ReCOIL-Q` for simplicity. Figure 2a and Figure 11 show that `ReCOIL-Q` can estimate d^{π} more accurately than OPOLO [82] which relies on coverage assumption and IQLearn [22] which only utilize expert data. This validates our theoretical results in Theorem 1. Besides, Figure 2b shows the reward function recovered by `ReCOIL-V` for a simple grid-world task. For Hopper and Walker, we respectively observe a Pearson correlation of **0.98** and **0.92** between the recovered reward with the ground truth. See more details in Appendix F.9.

6.2 Offline RL

Benchmark Comparison Table 3 shows that f-`DVL` outperforms XQL and other prior offline RL methods [9, 42, 43, 41, 19] on a broad range of continuous control tasks. We note an inconsistency

Dataset	BC	10%BC	DT	TD3+BC	CQL	IQL	XQL(r)	f-DVL (χ^2)	f-DVL (TV)
halfcheetah-medium-v2	42.6	42.5	42.6	**48.3**	44.0	**47.4**	47.4	**47.7**	47.5
hopper-medium-v2	52.9	56.9	67.6	59.3	58.5	66.3	**68.5**	63.0	64.1
walker2d-medium-v2	75.3	75.0	74.0	**83.7**	72.5	78.3	81.4	80.0	81.5
halfcheetah-medium-replay-v2	36.6	40.6	36.6	**44.6**	**45.5**	**44.2**	44.1	42.9	**44.7**
hopper-medium-replay-v2	18.1	75.9	82.7	60.9	95.0	94.7	95.1	90.7	**98.0**
walker2d-medium-replay-v2	26.0	62.5	66.6	**81.8**	77.2	73.9	58.0	52.1	68.7
halfcheetah-medium-expert-v2	55.2	**92.9**	86.8	90.7	**91.6**	86.7	90.8	89.3	**91.2**
hopper-medium-expert-v2	52.5	110.9	**107.6**	98.0	105.4	91.5	94.0	**105.8**	93.3
walker2d-medium-expert-v2	107.5	**109.0**	108.1	**110.1**	108.8	**109.6**	**110.1**	**110.1**	109.6
antmaze-umaze-v0	54.6	62.8	59.2	78.6	74.0	**87.5**	47.7	83.7	**87.7**
antmaze-umaze-diverse-v0	45.6	50.2	53.0	71.4	**84.0**	62.2	51.7	50.4	48.4
antmaze-medium-play-v0	0.0	5.4	0.0	10.6	61.2	**71.2**	31.2	56.7	**71.0**
antmaze-medium-diverse-v0	0.0	9.8	0.0	3.0	53.7	**70.0**	0.0	48.2	60.2
antmaze-large-play-v0	0.0	0.0	0.0	0.2	15.8	39.6	10.7	36.0	**41.7**
antmaze-large-diverse-v0	0.0	6.0	0.0	0.0	14.9	**47.5**	31.28	**44.5**	39.3
kitchen-complete-v0	65.0	-	-	-	43.8	62.5	56.7	**67.5**	61.3
kitchen-partial-v0	38.0	-	-	-	49.8	46.3	48.6	58.8	**70.0**
kitchen-mixed-v0	51.5	-	-	-	51.0	51.0	40.4	**53.75**	52.5

Table 3: The normalized return of offline RL methods on D4RL tasks. XQL(r) denotes the results obtained under the standard evaluation protocol. Results aggregated over 7 seeds.

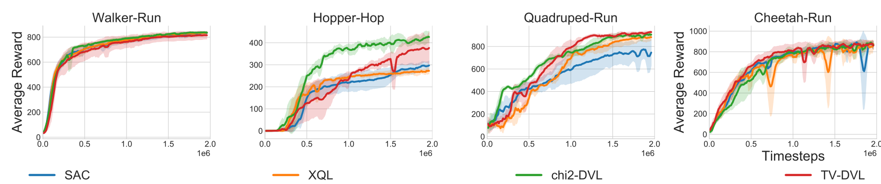

Figure 4: Online RL: f-DVL outperforms SAC and XQL, particularly for Hopper-Hop and Quadruped-Run tasks.

between our reproduced XQL results and the results reported in the original paper: their results were reported by taking the best average return during training as opposed to the standard practice of taking the average of the last iterate performance across different seeds at 1 million gradient steps. Such inconsistency can be validated by comparing their training plots and reported results (Fig 11 and Table 1 in [23]). XQL(r) shows the results for XQL under the standard evaluation protocol.

Training Stability As pointed out by the authors, the exponential loss function of XQL causes numerical instabilities during optimization. As discussed in Section 5.2, this is a by-product of reverse KL divergence. Fig. 3 confirms that this is fixed by f-DVL by using other f-divergences with more stable loss functions. Additionally, Fig. 4 demonstrates that f-DVL also outperforms XQL and SAC in the online setting as well. See Appendix E for additional experimental details.

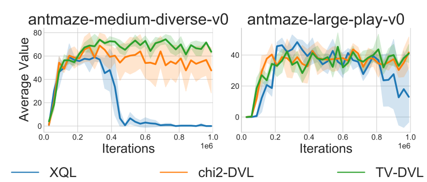

Figure 3: XQL training diverges due to the numerical instability of its loss function. f-DVL fixes this problem by using more well-behaved f-divergences.

6.3 Additional Experiments

We conduct additional experiments in Appendix F. We further demonstrate a) when incorporating off-policy data in online training, traditional ADP-based methods suffer from the over-estimation of value functions and the performance gain is limited, whereas dual-RL methods can leverage the same data to achieve better performance (Appendix F.1); b) the reward functions learned by ReCOIL are of high quality (Appendix F.9); c) the hyperparameter ablation for f-DVL (Appendix F.7) and qualitative results for ReCOIL (Appendix F.4).

7 Conclusion

Our work unifies a significant number of recent developments in RL and IL. Our insight calls for these methods to be studied under this unified lens to determine the core components that contribute to the success and limitations of these methods. Inspired by this unification, we propose: 1) a family of stable offline RL methods f-DVL relying on implicit value function maximization, 2) ReCOIL, a general off-policy IL method from arbitrary data that do not rely on the restrictive coverage assumption made by prior work, and 3) a non-adversarial offline IL method IVLearn using expert data only. We show that f-DVL and ReCOIL both outperform previous methods in online/offline RL and offline IL domains, respectively. We demonstrate that Dual-RL algorithms have great potential for developing performant algorithms and warrant further study.

References

[1] A. Abdolmaleki, J. T. Springenberg, Y. Tassa, R. Munos, N. Heess, and M. Riedmiller. Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*, 2018. 6

[2] A. Agarwal, N. Jiang, S. M. Kakade, and W. Sun. Reinforcement learning: Theory and algorithms. *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*, pages 10–4, 2019. 17

[3] F. Al-Hafez, D. Tateo, O. Arenz, G. Zhao, and J. Peters. Ls-iq: Implicit reward regularization for inverse reinforcement learning. *arXiv preprint arXiv:2303.00599*, 2023. 5, 26, 34

[4] L. Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pages 30–37. Elsevier, 1995. 1

[5] P. J. Ball, L. Smith, I. Kostrikov, and S. Levine. Efficient online reinforcement learning with offline data. *arXiv preprint arXiv:2302.02948*, 2023. 1

[6] J. Bas-Serrano, S. Curi, A. Krause, and G. Neu. Logistic q-learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3610–3618. PMLR, 2021. 2, 28

[7] D. P. Bertsekas and J. N. Tsitsiklis. Neuro-dynamic programming: an overview. In *Proceedings of 1995 34th IEEE conference on decision and control*, volume 1, pages 560–564. IEEE, 1995. 1

[8] V. S. Borkar. A convex analytic approach to markov decision processes. *Probability Theory and Related Fields*, 78(4):583–602, 1988. 2

[9] 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. *Advances in neural information processing systems*, 34:15084–15097, 2021. 8

[10] B. Dai, N. He, Y. Pan, B. Boots, and L. Song. Learning from conditional distributions via dual embeddings. In *Artificial Intelligence and Statistics*, pages 1458–1467. PMLR, 2017. 19

[11] D. P. De Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. *Operations research*, 51(6):850–865, 2003. 2

[12] G. T. de Ghellinck and G. D. Eppen. Linear programming solutions for separable markovian decision problems. *Management Science*, 13(5):371–394, 1967. 2

[13] E. V. Denardo. On linear programming in a markov decision problem. *Management Science*, 16(5):281–288, 1970. 2

[14] B. Eysenbach, S. Levine, and R. R. Salakhutdinov. Replacing rewards with examples: Example-based policy search via recursive classification. *Advances in Neural Information Processing Systems*, 34:11541–11552, 2021. 8, 36

[15] 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 *Conference on Robot Learning*, pages 158–168. PMLR, 2022. 1, 2, 4, 26, 27

[16] J. Fu, K. Luo, and S. Levine. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*, 2017. 2

[17] J. Fu, A. Kumar, M. Soh, and S. Levine. Diagnosing bottlenecks in deep q-learning algorithms. In *International Conference on Machine Learning*, pages 2021–2030. PMLR, 2019. 1, 2, 4, 6

[18] 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. 7, 35

[19] S. Fujimoto and S. S. Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021. 2, 8, 36

[20] S. Fujimoto, H. Hoof, and D. Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018. 1, 2, 3, 4, 6

[21] S. Fujimoto, D. Meger, and D. Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pages 2052–2062. PMLR, 2019. 2, 6

[22] D. Garg, S. Chakraborty, C. Cundy, J. Song, and S. Ermon. Iq-learn: Inverse soft-q learning for imitation. *Advances in Neural Information Processing Systems*, 34:4028–4039, 2021. 1, 2, 4, 5, 8, 22, 26, 36

[23] D. Garg, J. Hejna, M. Geist, and S. Ermon. Extreme q-learning: Maxent rl without entropy. *arXiv preprint arXiv:2301.02328*, 2023. 1, 2, 3, 6, 7, 9, 24, 34, 36, 37

[24] S. K. S. Ghasemipour, R. Zemel, and S. Gu. A divergence minimization perspective on imitation learning methods. In *Conference on Robot Learning*, pages 1259–1277. PMLR, 2020. 2, 5

[25] A. Gleave, M. Dennis, S. Legg, S. Russell, and J. Leike. Quantifying differences in reward functions. *arXiv preprint arXiv:2006.13900*, 2020. 41

[26] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pages 1352–1361. PMLR, 2017. 4

[27] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018. 1, 3, 16, 38

[28] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023. 1

[29] H. Hasselt. Double q-learning. *Advances in neural information processing systems*, 23, 2010. 2

[30] O. Hernández-Lerma and J. B. Lasserre. *Discrete-time Markov control processes: basic optimality criteria*, volume 30. Springer Science & Business Media, 2012. 2

[31] J. Ho and S. Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29:4565–4573, 2016. 2, 36

[32] H. Hoshino, K. Ota, A. Kanezaki, and R. Yokota. Opirl: Sample efficient off-policy inverse reinforcement learning via distribution matching. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 448–454. IEEE, 2022. 2, 5

[33] E. Imani, E. Graves, and M. White. An off-policy policy gradient theorem using emphatic weightings. *Advances in Neural Information Processing Systems*, 31, 2018. 2

[34] M. Janner, J. Fu, M. Zhang, and S. Levine. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32, 2019. 1

[35] R. Jiang, T. Zahavy, Z. Xu, A. White, M. Hessel, C. Blundell, and H. Van Hasselt. Emphatic algorithms for deep reinforcement learning. In *International Conference on Machine Learning*, pages 5023–5033. PMLR, 2021. 2

[36] L. Ke, S. Choudhury, M. Barnes, W. Sun, G. Lee, and S. Srinivasa. Imitation learning as f-divergence minimization. In *Algorithmic Foundations of Robotics XIV: Proceedings of the Fourteenth Workshop on the Algorithmic Foundations of Robotics 14*, pages 313–329. Springer International Publishing, 2021. 5

[37] G.-H. Kim, J. Lee, Y. Jang, H. Yang, and K.-E. Kim. Lobsdice: offline imitation learning from observation via stationary distribution correction estimation. *arXiv preprint arXiv:2202.13536*, 2022. 2, 5

[38] G.-H. Kim, S. Seo, J. Lee, W. Jeon, H. Hwang, H. Yang, and K.-E. Kim. Demodice: Offline imitation learning with supplementary imperfect demonstrations. In *International Conference on Learning Representations*, 2022. 2, 3, 5, 8, 36

[39] I. Kostrikov, K. K. Agrawal, D. Dwibedi, S. Levine, and J. Tompson. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. *arXiv preprint arXiv:1809.02925*, 2018. 1, 3

[40] I. Kostrikov, O. Nachum, and J. Tompson. Imitation learning via off-policy distribution matching. *arXiv preprint arXiv:1912.05032*, 2019. 2, 3

[41] I. Kostrikov, A. Nair, and S. Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021. 1, 3, 6, 7, 8, 34, 36

[42] A. Kumar, J. Fu, M. Soh, G. Tucker, and S. Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019. 2, 4, 8

[43] A. Kumar, A. Zhou, G. Tucker, and S. Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020. 1, 2, 3, 6, 8, 22, 23, 24

[44] D. Lee and N. He. Stochastic primal-dual q-learning algorithm for discounted mdps. In *2019 american control conference (acc)*, pages 4897–4902. IEEE, 2019. 2

[45] J. Lee, W. Jeon, B. Lee, J. Pineau, and K.-E. Kim. Optidice: Offline policy optimization via stationary distribution correction estimation. In *International Conference on Machine Learning*, pages 6120–6130. PMLR, 2021. 2, 3, 7

[46] J. Lei Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *ArXiv e-prints*, pages arXiv–1607, 2016. 35, 36

[47] S. Levine, A. Kumar, G. Tucker, and J. Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020. 31

[48] Y. J. Ma, A. Shen, D. Jayaraman, and O. Bastani. Smodice: Versatile offline imitation learning via state occupancy matching. *arXiv preprint arXiv:2202.02433*, 2022. 1, 2, 3, 5, 8, 21, 27, 28, 35, 36

[49] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022. 2

[50] Y. J. Ma, J. Yan, D. Jayaraman, and O. Bastani. How far i'll go: Offline goal-conditioned reinforcement learning via f-advantage regression. *arXiv preprint arXiv:2206.03023*, 2022. 2

[51] A. Malek, Y. Abbasi-Yadkori, and P. Bartlett. Linear programming for large-scale markov decision problems. In *International Conference on Machine Learning*, pages 496–504. PMLR, 2014. 2

[52] A. S. Manne. Linear programming and sequential decisions. *Management Science*, 6(3): 259–267, 1960. 1, 2, 3, 16

[53] P. Mehta and S. Meyn. Q-learning and pontryagin's minimum principle. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 3598–3605. IEEE, 2009. 28

[54] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013. 2

[55] O. Nachum and B. Dai. Reinforcement learning via fenchel-rockafellar duality. *arXiv preprint arXiv:2001.01866*, 2020. 1, 2, 3, 4, 16, 18, 19, 21

[56] O. Nachum, B. Dai, I. Kostrikov, Y. Chow, L. Li, and D. Schuurmans. Algaedice: Policy gradient from arbitrary experience. *arXiv preprint arXiv:1912.02074*, 2019. 2, 3, 4, 6, 39

[57] A. Nair, A. Gupta, M. Dalal, and S. Levine. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020. 2, 6, 38

[58] M. Nakamoto, Y. Zhai, A. Singh, M. S. Mark, Y. Ma, C. Finn, A. Kumar, and S. Levine. Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning. *arXiv preprint arXiv:2303.05479*, 2023. 7

[59] T. Ni, H. Sikchi, Y. Wang, T. Gupta, L. Lee, and B. Eysenbach. f-irl: Inverse reinforcement learning via state marginal matching. *arXiv preprint arXiv:2011.04709*, 2020. 1

[60] T. Ni, H. Sikchi, Y. Wang, T. Gupta, L. Lee, and B. Eysenbach. f-irl: Inverse reinforcement learning via state marginal matching. In *Conference on Robot Learning*, pages 529–551. PMLR, 2021. 3, 5

[61] J. Peters, K. Mulling, and Y. Altun. Relative entropy policy search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pages 1607–1612, 2010. 2

[62] M. Petrik and S. Zilberstein. Constraint relaxation in approximate linear programs. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 809–816, 2009. 2

[63] A. Picard-Weibel and B. Guedj. On change of measure inequalities for f-divergences. *arXiv preprint arXiv:2202.05568*, 2022. 25

[64] D. Precup. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, page 80, 2000. 3

[65] D. Precup, R. S. Sutton, and S. Dasgupta. Off-policy temporal-difference learning with function approximation. In *ICML*, pages 417–424, 2001. 1

[66] M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014. 17

[67] A. Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 547–561. University of California Press, 1961. 16

[68] H. Sikchi, A. Saran, W. Goo, and S. Niekum. A ranking game for imitation learning. *arXiv preprint arXiv:2202.03481*, 2022. 2, 3, 5, 34

[69] H. Sikchi, W. Zhou, and D. Held. Learning off-policy with online planning. In *Conference on Robot Learning*, pages 1622–1633. PMLR, 2022. 1, 6

[70] A. Singh, A. Kumar, Q. Vuong, Y. Chebotar, and S. Levine. Offline rl with realistic datasets: Heteroskedasticity and support constraints. *arXiv preprint arXiv:2211.01052*, 2022. 2

[71] S. P. Singh and R. C. Yee. An upper bound on the loss from approximate optimal-value functions. *Machine Learning*, 16:227–233, 1994. 33

[72] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018. 24

[73] G. Swamy, S. Choudhury, J. A. Bagnell, and S. Wu. Of moments and matching: A game-theoretic framework for closing the imitation gap. In *International Conference on Machine Learning*, pages 10022–10032. PMLR, 2021. 3

[74] S. Thrun and A. Schwartz. Issues in using function approximation for reinforcement learning. In *Proceedings of the Fourth Connectionist Models Summer School*, volume 255, page 263. Hillsdale, NJ, 1993. 2, 4

[75] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012. 35

[76] J. Tsitsiklis and B. Van Roy. An analysis of temporal-difference learning with function approximationtechnical. *Rep. LIDS-P-2322). Lab. Inf. Decis. Syst. Massachusetts Inst. Technol. Tech. Rep*, 1996. 2

[77] I. Uchendu, T. Xiao, Y. Lu, B. Zhu, M. Yan, J. Simon, M. Bennice, C. Fu, C. Ma, J. Jiao, et al. Jump-start reinforcement learning. *arXiv preprint arXiv:2204.02372*, 2022. 38

[78] M. Vecerik, T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. Heess, T. Rothörl, T. Lampe, and M. Riedmiller. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv:1707.08817*, 2017. 1

[79] L. Viano, A. Kamoutsi, G. Neu, I. Krawczuk, and V. Cevher. Proximal point imitation learning. *arXiv preprint arXiv:2209.10968*, 2022. 2, 28

[80] Y. Wu, G. Tucker, and O. Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019. 2, 6

[81] R. Zhang, B. Dai, L. Li, and D. Schuurmans. Gendice: Generalized offline estimation of stationary values. *arXiv preprint arXiv:2002.09072*, 2020. 2

[82] Z. Zhu, K. Lin, B. Dai, and J. Zhou. Off-policy imitation learning from observations. *Advances in Neural Information Processing Systems*, 33:12402–12413, 2020. 1, 2, 3, 4, 5, 8, 28

[83] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008. 3

[84] K. Zolna, A. Novikov, K. Konyushkova, C. Gulcehre, Z. Wang, Y. Aytar, M. Denil, N. de Freitas, and S. Reed. Offline learning from demonstrations and unlabeled experience. *arXiv preprint arXiv:2011.13885*, 2020. 8, 36

Appendix

Table of Contents

A Code Release

The accompanying code (in JAX) and instructions to reproduce the results for this work can be found at the link here.

B Limitations and Negative Societal Impacts

Limitations: One limitation of the paper is the assumption that the expert demonstrations used in the imitation learning process are always of high quality and provide the desired behavior. In practice, obtaining high-quality demonstrations can be challenging, especially in complex environments where the behavior of the expert is not always clear. The performance of the proposed approach could be limited in cases where the expert demonstrations are of poor quality or where the behavior of the expert does not correspond to the desired behavior. The second issue with dual-RL approaches is the training stability. Although the methods we propose are significantly more stable than prior works that use dual approaches, it still lacks heuristics which have made ADP-based primal methods quite robust to train (eg. [27]).

Negative Societal Impacts: As machine learning algorithms continue to grow in sophistication, it is important to consider the potential risks and harms associated with their use. One such area of concern is imitation learning, which involves training a model to imitate a desired behavior by providing it with examples of that behavior. However, this approach can be problematic if the demonstration data includes harmful behaviors, whether intentional or not. Even in cases where the demonstration data is of high quality and desirable behavior is learned, the algorithm may still fall short of providing sufficient guarantees of performance. In high-stakes domains, the use of such algorithms without appropriate safety checks on learned behaviors could lead to serious consequences. As such, it is crucial to carefully consider the potential risks and benefits of imitation learning, and to develop strategies for ensuring safe and effective use of these algorithms in real-world application

C Dual Reinforcement Learning

C.1 A Review of Dual-RL

In this section, we aim to give a self-contained review for Dual Reinforcement Learning. For a more thorough read, refer to [55].

C.1.1 Convex conjugates and f-divergence

We first review the basics of duality in reinforcement learning. Let $f : \mathbb{R}_+ \to \mathbb{R}$ be a convex function. The convex conjugate $f^* : \mathbb{R}_+ \to \mathbb{R}$ of f is defined by:

$$f^*(y) = \sup_{x \in \mathbb{R}_+}[xy - f(x)]. \tag{14}$$

The convex conjugates have the important property that f^* is also convex and the convex conjugate of f^* retrieves back the original function f. We also note an important relation regarding f and f^*: $(f^*)^{'} = (f')^{-1}$, where the $'$ notation denotes first derivative.

Going forward, we would be dealing extensively with f-divergences. Informally, f-divergences [67] are a measure of distance between two probability distributions. Here's a more formal definition:

Let P and Q be two probability distributions over a space $\mathcal{Z}$ such that P is absolutely continuous with respect to Q [1]. For a function $f : \mathbb{R}_+ \to \mathbb{R}$ that is a convex lower semi-continuous and $f(1) = 0$, the f-divergence of P from Q is

$$D_f(P \,||\, Q) = \mathbb{E}_{z\sim Q}\left[f\left(\frac{P(z)}{Q(z)}\right)\right]. \tag{15}$$

Table 4 lists some common f-divergences with their generator functions f and the conjugate functions f^*.

C.1.2 An Overview of Reinforcement Learning via Lagrangian Duality

We consider RL problems with their average return considered in the form of a convex program with linear constraints [52], to which we apply Lagrangian duality to obtain corresponding constraint-free problems. This framework was first introduced in the work of Nachum and Dai [55], which obtains the same formulations as ours via Fenchel-Rockfeller duality. Here we use Lagrangian duality for its simplicity and popularity.

Consider the following regularized policy learning problem

$$\max_{\pi} J(\pi) = \mathbb{E}_{d^\pi(s,a)}[r(s,a)] - \alpha D_f(d^\pi(s,a) \,||\, d^O(s,a)), \tag{16}$$

[1] Let z denote the random variable. For any measurable set $Z \subseteq \mathcal{Z}$, $Q(z \in Z) = 0$ implies $P(z \in Z) = 0$.

Divergence Name	Generator $f(x)$	Conjugate $f^*(y)$
Forward KL	$-\log x$	$-1-\log(-y)$
Reverse KL	$x \log x$	$e^{(y-1)}$
Squared Hellinger	$(\sqrt{x}-1)^2$	$\frac{y}{1-y}$
Pearson χ^2	$(x-1)^2$	$y+\frac{y^2}{4}$
Total Variation	$\frac{1}{2}\|x-1\|$	y if $y \in [-\frac{1}{2}, \frac{1}{2}]$ otherwise ∞
Jensen-Shannon	$-(x+1)\log(\frac{x+1}{2})+x\log x$	$-\log(2-e^y)$

Table 4: List of common f-divergences.

where $D_f(d^\pi(s,a) \,||\, d^O(s,a))$ is a conservatism regularizer that encourages the visitation distribution of π to stay close to some distribution d^O, and α is a temperature parameter that balances the expected return and the conservatism.

An interesting fact is that $J(\pi)$ can be rewritten as a convex problem that searches for an *achievable* visitation distribution that satisfies the *Bellman-flow* constraints:

$$\begin{aligned} J(\pi) = \max_d \; & \mathbb{E}_{d(s,a)}[r(s,a)] - \alpha D_f(d(s,a) \,||\, d^O(s,a)) \\ \text{s.t } & d(s,a) = (1-\gamma)d_0(s).\pi(a|s) + \gamma \textstyle\sum_{s',a'} d(s',a')p(s|s',a')\pi(a|s), \;\; \forall s \in \mathcal{S}, a \in \mathcal{A}. \end{aligned} \tag{17}$$

Applying Lagrangian duality and convex conjugate (14) to this problem, we can convert it to an unconstrained problem with dual variables $Q(s,a)$ defined for all $s,a \in \mathcal{S} \times \mathcal{A}$:

$$\min_Q (1-\gamma)\mathbb{E}_{s\sim d_0, a\sim\pi(s)}[Q(s,a)] + \alpha\mathbb{E}_{(s,a)\sim d^O}[f^*\left([\mathcal{T}_r^\pi Q(s,a) - Q(s,a)]/\alpha\right)], \tag{18}$$

where f^* is the convex conjugate of f. We defer the derivation to the next section. As problem (17) is convex, strong duality holds and problems (17) and (18) have the same optimal objective value up to a constant scaling[2]. We refer to the nested policy learning problem where $J(\pi)$ is of form (17) as `primal-Q` and the joint problem with scaled $J(\pi)$ of form (18) as `dual-Q`.

$$\texttt{primal-Q} \quad \max_\pi [J(\pi) \text{ in the form Eq. (2)}], \tag{19}$$

$$\texttt{dual-Q} \quad \max_\pi \min_Q (1-\gamma)\mathbb{E}_{s\sim d_0, a\sim\pi(s)}[Q(s,a)] + \alpha\mathbb{E}_{(s,a)\sim d^O}[f^*\left([\mathcal{T}_r^\pi Q(s,a) - Q(s,a)]/\alpha\right)]. \tag{20}$$

In fact, problem (17) is overconstrained – the maximization w.r.t d is unnecessary, as for a fixed π the $|\mathcal{S}| \times |\mathcal{A}|$ equality constraints already uniquely determine a solution d^π [66]. Let π^*, d^* be the optimal policy and corresponding visitation distribution. In fact, we can relax the constraints to get another problem [2] with the same optimal solution d^*, which we call `primal-V` below:

$$\begin{aligned} \texttt{primal-V} \quad \max_{d \geqslant 0} \; & \mathbb{E}_{d(s,a)}[r(s,a)] - \alpha D_f(d(s,a) \,||\, d^O(s,a)) \\ \text{s.t } & \textstyle\sum_{a\in\mathcal{A}} d(s,a) = (1-\gamma)d_0(s) + \gamma \sum_{(s',a')\in\mathcal{S}\times\mathcal{A}} d(s',a')p(s|s',a'), \;\; \forall s \in \mathcal{S}. \end{aligned} \tag{21}$$

Comparing with problem (17), the constraints are relaxed and there is no policy π in this formulation. In fact, as opposed to `primal-Q`, which needs to solve nested inner problems, `primal-V` solves a single problem to obtain d^*, from which we can recover π^* via Eq. (22)[3]:

$$\pi(a|s) = d^\pi(s,a) / \textstyle\sum_{a\in\mathcal{A}} d^\pi(s,a). \tag{22}$$

Similarly, we consider the Lagrangian dual of (21), with dual variables $V(s)$ defined for all $s \in S$:

$$\texttt{dual-V} \quad \min_V (1-\gamma)\mathbb{E}_{s\sim d_0}[V(s)] + \alpha\mathbb{E}_{(s,a)\sim d^O}\left[f_p^*\left([\mathcal{T}V(s,a) - V(s))]/\alpha\right)\right], \tag{23}$$

where f_p^* is a variant of f^* defined in Eq. (45). Such modification is to cope with the nonnegativity constraint $d(s,a) \geqslant 0$ in `primal-V`. This constraint is ignored in `primal-Q` because the constraints

[2] We scaled the dual problem by $1/\alpha$ for derivation simplicity.

[3] Eq. (22) can be easily computed for discrete actions, yet it is difficult for continuous actions. While our analysis focuses on the tabular case, we discuss two methods for recovering π^* for continuous actions in Appendix C.1.6.

of the inner problem (17) already uniquely identify the solution. See Appendix C.1.4 for the derivation. As before, strong duality holds here (up to a factor of $1/\alpha$), and we can compute the optimal policy π^* after obtaining V^*. We discuss this in detail in Appendix C.1.6.

Remark 1. The above formulations generalizes to the popular MaxEnt RL framework, where the objective $J(\pi)$ contains an extra policy entropy regularizer. One only needs to replace the Bellman operator $\mathcal{T}_r^\pi$ by its soft variant: $\mathcal{T}_{r,\text{soft}}^\pi Q(s,a) = r(s,a) + \gamma \mathbb{E}_{s',a'}[Q(s',a') - \log \pi(a'|s')]$.

Remark 2. We derive the dual problems via the Lagrangian duality. Taking the `primal-Q` problem as an example, the key step which bridges its Lagrangian dual problem $\min_Q \max_d L(Q,d)$ and the final formulation `dual-Q` is that the maximizer d^* of the inner problem has a closed form solution. Equivalently, we can rewrite the inner problem $\max_d L(Q,d)$ via the convex conjugate (30), which eliminates the variable d. The Fenchel-Rockerfeller duality provides an alternative way to directly reach the same formulation, where one first rewrites the linear constraints as part of the objective using the Dirac delta function [55].

Remark 3. The dual formulations have a few appealing properties. (a) They allow us to transform constrained distribution-matching problems, w.r.t previously logged data, into unconstrained forms. (b) One can show that the gradient of `dual-Q` w.r.t π, when Q is optimized for the inner problem, is the on-policy policy gradient computed by off-policy data. This key property relieves the instability or divergence issue in off-policy learning.

C.1.3 Deriving `dual-Q`

We again consider the RL problem as a maximization of a convex program for estimating policy performance $J(\pi)$ by considering optimization over *achievable* state-action visitations (i.e $\max_\pi J(\pi)$):

$$\max_{\pi} \left[\max_{d \geqslant 0} \mathbb{E}_{d(s,a)}[r(s,a)] - \alpha D_f(d(s,a) \,||\, d^O(s,a)) \right. \tag{24}$$

$$\left. \text{s.t } d(s,a) = (1-\gamma) d_0(s).\pi(a|s) + \gamma \sum_{s',a'} d(s',a') p(s|s',a') \pi(a|s) \right], \tag{25}$$

where α allows us to weigh policy improvement against conservatism from staying close to the state-action distribution d^O.

A careful reader may notice that the inner problem is overconstrained and overparameterized. The solution to the inner maximization problem with respect to d is uniquely determined by the $|\mathcal{S}| \times |\mathcal{A}|$ linear constraints, and the nonnegativity constraint $d \geqslant 0$ is not necessary. Moreover, given a fixed policy π, the solution of the inner problem is its visitation distribution d^π.

The constraints of the inner problem are known as the *Bellman flow equations* that an achievable stationary state-action distribution must satisfy. The next question is how can we solve it? Here is where Lagrangian duality comes into play. First, we form the Lagrangian dual of our original optimization problem, transforming our constrained optimization into an unconstrained form. This introduces additional optimization variables - the Lagrange multipliers Q.

As mentioned before, we can discard the nonnegativity constraint $d \geqslant 0$ as the other constraints imply a unique solution for d. Focusing on the inner optimization problem, we optimize the Lagrangian dual problem:

$$\min_{Q(s,a)} \max_{d} \mathbb{E}_{s,a \sim d(s,a)}[r(s,a)] - \alpha D_f(d(s,a) \,||\, d^O(s,a))$$
$$+ \sum_{s,a} Q(s,a) \left((1-\gamma) d_0(s).\pi(a|s) + \gamma \sum_{s',a'} d(s',a') p(s|s',a') \pi(a|s) - d(s,a) \right),$$

where $Q(s,a)$ are the Lagrange multipliers associated with the equality constraints. We can now do some simple algebraic manipulation to further simplify it:

$$\begin{aligned}
&\min_{Q(s,a)} \max_{d} \mathbb{E}_{s,a\sim d(s,a)}[r(s,a)] - \alpha D_f(d(s,a)\,||\,d^O(s,a)) \\
&+ \sum_{s,a} Q(s,a)\left((1-\gamma)d_0(s).\pi(a|s) + \gamma \sum_{s',a'} d(s',a')p(s|s',a')\pi(a|s) - d(s,a)\right) \qquad (26)\\
=&\min_{Q(s,a)} \max_{d} (1-\gamma)\mathbb{E}_{d_0(s),\pi(a|s)}[Q(s,a)] \\
&+ \mathbb{E}_{s,a\sim d}\left[r(s,a) + \gamma \sum_{s'} p(s'|s,a)\pi(a'|s')Q(s',a') - Q(s,a)\right] - \alpha D_f(d(s,a)\,||\,d^O(s,a)), \qquad (27)
\end{aligned}$$

where we swap the mamximum and minimum in the last step as strong duality holds for this problem. This is equivalent to solving the following scaled objective (scaled by $1/\alpha$).

$$\begin{aligned}
&\min_{Q(s,a)} \max_{d} \frac{(1-\gamma)}{\alpha}\mathbb{E}_{d_0(s),\pi(a|s)}[Q(s,a)] \\
&+ \mathbb{E}_{s,a\sim d}\left[(r(s,a) + \gamma \sum_{s'} p(s'|s,a)\pi(a'|s')Q(s',a') - Q(s,a))/\alpha\right] - D_f(d(s,a)\,||\,d^O(s,a)) \qquad (28)\\
=&\min_{Q(s,a)} \frac{(1-\gamma)}{\alpha}\mathbb{E}_{d_0(s),\pi(a|s)}[Q(s,a)] \\
&+ \mathbb{E}_{s,a\sim d^O}\left[f^*((r(s,a) + \gamma \sum_{s'} p(s'|s,a)\pi(a'|s')Q(s',a') - Q(s,a))/\alpha)\right], \qquad (29)
\end{aligned}$$

where we applied the convex conjugate (Eq. (14)) in the last step. To see this more clearly, let $y(s,a) = r(s,a) + \gamma \sum_{s'} p(s'|s,a)\pi(a'|s')Q(s',a') - Q(s,a)$. Then, under mild conditions that the interchangeability principle [10] is satisfied, and d^O has sufficient support over $\mathcal{S}\times\mathcal{A}$ [55], it holds that

$$\begin{aligned}
&\max_{d} \mathbb{E}_{s,a\sim d}[y(s,a)] - D_f(d(s,a)\,||\,d^O(s,a)) \qquad (30)\\
=&\max_{d} \mathbb{E}_{s,a\sim d^O}\left[\frac{d(s,a)}{d^O(s,a)}y(s,a) - f\left(\frac{d(s,a)}{d^O(s,a)}\right)\right] \qquad (31)\\
=&\mathbb{E}_{d^O}[f^*(y(s,a))]. \qquad (32)
\end{aligned}$$

We have transformed the problem of computing $J(\pi)$ to solving Eq. (29). Finally, the policy optimization problem $\max_\pi J(\pi)$ is reduced to solving the following min-max optimization problem, which we will refer to as `dual-Q`:

$$\max_\pi \min_Q \tfrac{(1-\gamma)}{\alpha}\mathbb{E}_{d_0(s),\pi(a|s)}[Q(s,a)] + \mathbb{E}_{s,a\sim d^O}[f^*((r(s,a) + \gamma \textstyle\sum_{s'} p(s'|s,a)\pi(a|s')Q(s',a') - Q(s,a))/\alpha)]. \qquad (33)$$

Table 4 lists the corresponding convex conjugates f^* for common f-divergences.

In the case of deterministic policy and deterministic dynamics, the above-obtained optimization takes a simpler form:

$$\max_{\pi(a|s)} \min_{Q(s,a)} \frac{(1-\gamma)}{\alpha}\mathbb{E}_{\rho_0(s)}[Q(s,\pi(s))] + \mathbb{E}_{s,a\sim d^O}\left[f^*((r(s,a) + \gamma Q(s',\pi(s')) - Q(s,a))/\alpha)\right] \qquad (34)$$

Now, we have seen how we can transform a regularized RL problem into its `dual-Q` form which uses Lagrange variables in the form of state-action functions. Interestingly, we can go further to transform the regularized RL problem into Lagrange variables (V) that only depend on the state, and in doing so we also get rid of the two-player nature (min-max optimization) in the `dual-Q`.

C.1.4 Deriving `dual-V`

One important constraint we have not discussed so far is that the variable d we are optimizing must be nonnegative. This constraint is not needed for `primal-Q`, as for the inner problem (2), the solution is uniquely determined by the constraints. Nonetheless, it is important we consider this constraint for `primal-V` and derive the correct dual problem.

In `primal-V`, we formulate the visitation constraints to depend solely on states rather than state-action pairs. Note that doing this does not change the solution π^* for the regularized RL problem (Eq (16)). We consider $\alpha = 1$ for the sake of exposition. Interested readers can derive the result for $\alpha \neq 1$ as in the `dual-Q` case above. Recall the formulation of `primal-V`:

$$\max_{d \geqslant 0} \; \mathbb{E}_{d(s,a)}[r(s,a)] - D_f(d(s,a) \,||\, d^O(s,a))$$
$$\text{s.t} \sum_{a \in \mathcal{A}} d(s,a) = (1-\gamma) d_0(s) + \gamma \sum_{s',a'} d(s',a') p(s|s',a'). \tag{35}$$

As before, we construct the Lagrangian dual to this problem. Note that our constraints now solely depend on s.

$$\min_{V(s)} \max_{d \geqslant 0} \mathbb{E}_{s \sim d(s,a)}[r(s,a)] - D_f(d(s,a) \,||\, d^O(s,a))$$
$$+ \sum_s V(s) \left((1-\gamma) d_0(s) + \gamma \sum_{s',a'} d(s',a') p(s|s',a') - d(s,a) \right) \tag{36}$$

Using similar algebraic manipulations we used to obtain `dual-Q` in Section C.1.3, we have :

$$\min_{V(s)} \max_{d(s,a) \geqslant 0} \mathbb{E}_{s,a \sim d(s,a)}[r(s,a)] - D_f(d(s,a) \,||\, d^O(s,a))$$
$$+ \mathbb{E}_{s,a \sim d}\left[r(s,a) + \gamma \sum_{s'} p(s'|s,a) V(s') - V(s) \right] - D_f(d(s,a) \,||\, d^O(s,a)) \tag{37}$$
$$= \min_{V(s)} \max_{d(s,a) \geqslant 0} (1-\gamma) \mathbb{E}_{d_0(s)}[V(s)]$$
$$+ \mathbb{E}_{s,a \sim d}\left[r(s,a) + \gamma \sum_{s'} p(s'|s,a) V(s') - V(s) \right] - D_f(d(s,a) \,||\, d^O(s,a)) \tag{38}$$
$$= \min_{V(s)} \max_{d(s,a) \geqslant 0} (1-\gamma) \mathbb{E}_{d_0(s)}[V(s)]$$
$$+ \mathbb{E}_{s,a \sim d^O}\left[\tfrac{d(s,a)}{d^O(s,a)} \left(r(s,a) + \gamma \sum_{s'} p(s'|s,a) V(s') - V(s) \right) \right] - \mathbb{E}_{s,a \sim d^O}\left[f\left(\tfrac{d(s,a)}{d^O(s,a)} \right) \right] \tag{39}$$

Let $w(s,a) = \frac{d(s,a)}{d^O(s,a)}$ and $\delta_V(s,a) = r(s,a) + \gamma \sum_{s'} p(s'|s,a) V(s') - V(s)$ denote the TD error. The last equation becomes

$$\min_{V(s)} \max_{w(s,a) \geqslant 0} (1-\gamma) \mathbb{E}_{d_0(s)}[V(s)] + \mathbb{E}_{s,a \sim d^O}[w(s,a)(\delta_V(s,a))] - \mathbb{E}_{s,a \sim d^O}[f(w(s,a))]. \tag{40}$$

We now direct the attention to the inner maximization problem and derive a closed-form solution for it. Consider the Lagrangian dual problem of it:

$$\min_{\lambda \geqslant 0} \max_{w(s,a)} \mathbb{E}_{s,a \sim d^O}[w(s,a)(\delta_V(s,a))] - \mathbb{E}_{s,a \sim d^O}[f(w(s,a))] + \sum_{s,a} \lambda(s,a) w(s,a) \tag{41}$$

where the parameters $\lambda(s,a)$ for all $s \in S$ and $a \in A$ are the Lagrange multipliers. Since strong duality holds, we can use the KKT constraints to find the optimal solutions $w^*(s,a)$ and $\lambda^*(s,a)$:

1. **Primal feasibility** $w^*(s,a) \geqslant 0 \;\; \forall\, s,a$

2. **Dual feasibility** $\lambda^*(s,a) \geqslant 0 \;\; \forall\, s,a$

3. **Stationarity** $d^O(s,a)(-f'(w^*(s,a)) + \delta_V(s,a) + \lambda^*(s,a)) = 0 \;\; \forall\; s,a$

4. **Complementary Slackness** $w^*(s,a)\lambda^*(s,a) = 0 \;\; \forall\; s,a$

Using stationarity we have the following:

$$f'(w^*(s,a)) = \delta_V(s,a) + \lambda^*(s,a) \;\; \forall\; s,a \tag{42}$$

Now using complementary slackness only two cases are possible $w^*(s,a) \geqslant 0$ or $\lambda^*(s,a) \geqslant 0$. Combining both cases we arrive at the following solution for this constrained optimization:

$$w^*(s,a) = \max\left(0, {f'}^{-1}(\delta_V(s,a))\right) \tag{43}$$

We refer to the resulting function after plugging the solution for w^* back in Eq. (40) and refer to the closed form solution for d in second and third term as f_p^*.

$$f_p^*(\delta_V(s,a)) = w^*(s,a)(\delta_V(s,a)) - f(w^*(s,a)) \tag{44}$$

Plugging in $w^*(s,a)$ from Eq. (43) to Eq. (44), we get:

$$f_p^*(\delta_V(s,a)) = \max\left(0, {f'}^{-1}(\delta_V(s,a))\right)(\delta_V(s,a)) - f\left(\max\left(0, {f'}^{-1}(\delta_V(s,a))\right)\right) \tag{45}$$

Note that we get the original conjugate f^* back if we do not consider the nonnegativity constraints:

$$f^*(s,a) = {f'}^{-1}(\delta_V(s,a))(\delta_V(s,a)) - f({f'}^{-1}(\delta_V(s,a))). \tag{46}$$

Finally, we have the following optimization to solve for `dual-V` when considering the nonnegativity constraints:

> `dual-V`: $\min_{V(s)} (1-\gamma)\mathbb{E}_{s\sim d_0}[V(s)] + \mathbb{E}_{(s,a)\sim d^O}\left[f_p^*(\delta_V(s,a))\right]$

Some works e.g. SMODICE [48], ignore the nonnegativity constraints and use the corresponding `dual-V` formulation

> `dual-V` (w/o nonneg. constraints): $\min_V (1-\gamma)\mathbb{E}_{s\sim d_0}[V(s)] + \mathbb{E}_{(s,a)\sim d^O}[f^*(\delta_V(s,a)]$.

C.1.5 Discussion on Dual Formulations

In summary, we have two dual formulations for regularized policy learning:

> `dual-Q`: $\max_\pi \min_Q (1-\gamma)\mathbb{E}_{d_0(s),\pi(a|s)}[Q(s,a)]$
> $+ \mathbb{E}_{s,a\sim d^O}[f^*\left(r(s,a) + \gamma \sum_{s'} p(s'|s,a)\pi(a'|s')Q(s',a') - Q(s,a)\right)]$

and

> `dual-V`: $\min_{V(s)} (1-\gamma)\mathbb{E}_{s\sim d_0}[V(s)] + \mathbb{E}_{(s,a)\sim d^O}\left[f_p^*(\delta_V(s,a))\right]$

The above derivations for dual of primal RL formulation - `dual-Q` and `dual-V` brings out some important observations

- `dual-Q` and `dual-V` present off-policy policy optimization solutions for regularized RL problems which requires sampling transitions only from the off-policy distribution the policy state-action visitation is being regularized against. The gradient with respect to policy π when d is optimized in `dual-Q` can be shown to be equivalent to the on-policy policy gradient under a regularized Q-function (see Section 5.1 from [55]).
- The above property allows us to solve not only RL problems but also imitation problems by setting the reward function to be zero everywhere and d^O to be the expert dataset, and also offline RL problems where we want to maximize reward with the constraint that our state-action visitation should not deviate too much from the replay buffer ($d^O =$ replay-buffer).

- `dual-V` formulation presents a way to solve the RL problem using a single optimization rather than a min-max optimization of the `primal-Q` or standard RL formulation. `dual-V` implicitly subsumes greedy policy maximization.

C.1.6 How to recover the optimal policy in `dual-V`?

In the above derivations for `dual-Q` and `dual-V` we leveraged the fact that the closed form solution for optimizing Eq. (14) w.r.t d is known. The value of d^* for which Eq. (30) is maximized can be found by setting the gradient to zero (stationary point) leading to:

$$\frac{d^*(s,a)}{d^O(s,a)} = \max\left(0, (f')^{-1}\left(\frac{y(s,a)}{\alpha}\right)\right) \tag{47}$$

This ratio can be utilized in two different ways to recover the optimal policy:

Method 1: Maximum likelihood on expert visitation distribution

Policy learning can be written as maximizing the likelihood of optimal actions under the optimal state-action visitation:

$$\max \mathbb{E}_{s,a\sim d^*}[\pi_\theta(a|s)] \tag{48}$$

Using importance sampling we can rewrite the optimization above in a form suitable for optimization:

$$\max_\theta \mathbb{E}_{s,a\sim d^O}\left[\frac{d^*(s,a)}{d^O(s,a)}\pi_\theta(a|s)\right] = \max_\theta \mathbb{E}_{s,a\sim d^O}[w^*(s,a)\pi_\theta(a|s)] \tag{49}$$

This way of policy learning is similar to weighted behavior cloning or advantage-weighted regression, but suffers from the issue that policy is not optimized at state-actions where the offline dataset d^O has no coverage but $d^* > 0$.

Method 2: Reverse KL matching on offline data distribution (Information Projection)

To allow the policy to be optimized at all that states in the offline dataset + actions outside the dataset we consider an alternate objective:

$$\min_\theta D_{\mathrm{KL}}(d^O(s)\pi_\theta(a|s) \,||\, d^O(s)\pi^*(a|s)) \tag{50}$$

The objective can be expanded as follows:

$$\min_\theta D_{\mathrm{KL}}(d^O(s)\pi_\theta(a|s) \,||\, d^O(s)\pi^*(a|s)) \tag{51}$$

$$= \min_\theta \mathbb{E}_{s\sim d^O(s),a\sim\pi_\theta}\left[\log\frac{\pi_\theta(a|s)}{\pi^*(a|s)}\right] \tag{52}$$

$$= \min_\theta \mathbb{E}_{s\sim d^O(s),a\sim\pi_\theta}\left[\log\frac{\pi_\theta(a|s)d^*(s)d^O(s)\pi^o(a|s)}{\pi^*(a|s)d^*(s)d^O(s)\pi^o(a|s)}\right] \tag{53}$$

$$= \min_\theta \mathbb{E}_{s\sim d^O(s),a\sim\pi_\theta}\left[\log\frac{\pi_\theta(a|s)}{\pi^o(a|s)} - \log(w^*(s,a)) + \log\frac{d^*(s)}{d^O(s)}\right] \tag{54}$$

$$= \min_\theta \mathbb{E}_{s\sim d^O(s),a\sim\pi_\theta}[\log(\pi_\theta(a|s)) - \log(\pi^o(a|s)) - \log(w^*(s,a))] \tag{55}$$

This method recovers the optimal policy at the states present in the dataset but has the added complexity of learning another policy $\pi^o(a|s)$. One way of obtaining $\pi^o(a|s)$ is by behavior cloning the replay buffer.

C.2 Dual Connections to Reinforcement Learning

We begin by showing reducing popular offline RL class of methods: pessimistic value learning (CQL [43]) and implicit policy improvement (XQL [22]) to the `dual-Q` and `dual-V` framework respectively. Then, we show how the `dual-V` framework under a semi-gradient update rule leads to a family of offline RL algorithms that do not sample OOD actions.

Lemma 4. *CQL is an instance of `dual-Q` under the semi-gradient update rule, where the f-divergence is the Pearson χ^2 divergence, and d^O is the offline visitation distribution.*

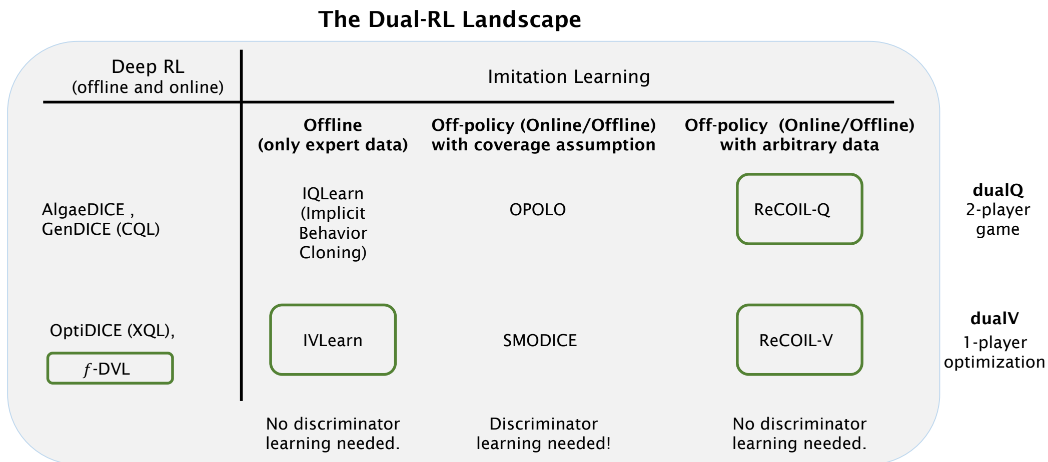

Figure 5: We show that a number of prior methods can be understood as a special case of the dual RL framework. Based on this framework, we also propose new methods addressing the shortcomings of previous works (boxed in green).

Proof. We show that CQL [43], a popular offline RL method is a special case of `dual-Q` for offline RL. Consider the χ^2 f-divergence with the generator function $f = (t-1)^2$. The dual function f^* is given by $f^* = (\frac{t^2}{4} + t)$. With this f-divergence the `dual-Q` optimization can be simplified as:

$$\frac{(1-\gamma)}{\alpha}\mathbb{E}_{d_0,\pi(a|s)}[Q(s,a)] + \mathbb{E}_{s,a\sim d^O}\left[\frac{y(s,a,r,s')^2}{4\alpha^2} + \frac{y(s,a,r,s')}{\alpha}\right] \tag{56}$$

$$= \frac{(1-\gamma)}{\alpha}\mathbb{E}_{d_0,\pi(a|s)}[Q(s,a)] + \mathbb{E}_{s,a\sim d^O}\left[\frac{y(s,a,r,s')}{\alpha}\right] + \mathbb{E}_{s,a\sim d^O}\left[\frac{y(s,a,r,s')^2}{4\alpha^2}\right] \tag{57}$$

Let's simplify the first two terms:

$$\frac{1}{\alpha}\left[(1-\gamma)\mathbb{E}_{d_0,\pi(a|s)}[Q(s,a)] + \mathbb{E}_{s,a\sim d^O}\left[r(s,a) + \gamma\sum_{s',a'} p(s'|s,a)\pi(a'|s')Q(s',a') - Q(s,a)\right]\right] \tag{58}$$

$$= \frac{1}{\alpha}\left[(1-\gamma)\mathbb{E}_{d_0,\pi(a|s)}[Q(s,a)] + \mathbb{E}_{s,a\sim d^O}\left[\gamma\sum_{s',a'} p(s'|s,a)\pi(a'|s')Q(s',a')\right] - \mathbb{E}_{s,a\sim d^O}[Q(s,a)] + \cancel{\mathbb{E}_{s,a\sim d^O}[r(s,a)]}\right] \tag{59}$$

$$= \frac{1}{\alpha}\left[(1-\gamma)\sum_{s,a} d_0(s)\pi(a|s)Q(s,a) + \gamma\sum_{s,a} d^O(s,a)\sum_{s'} p(s'|s,a)\pi(a'|s')Q(s',a') - \mathbb{E}_{s,a\sim d^O}[Q(s,a)]\right] \tag{60}$$

$$= \frac{1}{\alpha}\left[(1-\gamma)\sum_{s,a} d_0(s)\pi(a|s)Q(s,a) + \gamma\langle d^O, P^\pi Q\rangle - \mathbb{E}_{s,a\sim d^O}[Q(s,a)]\right] \tag{61}$$

$$= \frac{1}{\alpha}\left[(1-\gamma)\sum_{s,a} d_0(s)\pi(a|s)Q(s,a) + \gamma\langle P^\pi_* d^O, Q\rangle - \mathbb{E}_{s,a\sim d^O}[Q(s,a)]\right] \tag{62}$$

$$= \frac{1}{\alpha}\left[(1-\gamma)\sum_{s,a} d_0(s)\pi(a|s)Q(s,a) + \gamma\sum_{s,a}\pi(a|s)Q(s,a)\sum_{s',a'} p(s|s',a')d(s',a') - \mathbb{E}_{s,a\sim d^O}[Q(s,a)]\right] \tag{63}$$

$$= \frac{1}{\alpha}\left[\sum_{s,a}(d_0(s) + \gamma\sum_{s'a,'} p(s|s',a')d(s',a'))\pi(a|s)Q(s,a) - \mathbb{E}_{s,a\sim d^O}[Q(s,a)] + \mathbb{E}_{s,a\sim d^O}[r(s,a)]\right] \tag{64}$$

$$= \frac{1}{\alpha}\left[\sum_{s,a} d^O(s)\pi(a|s)Q(s,a) - \mathbb{E}_{s,a\sim d^O}[Q(s,a)] + \mathbb{E}_{s,a\sim d^O}[r(s,a)]\right] \tag{65}$$

$$= \frac{1}{\alpha}\left[\mathbb{E}_{s\sim d^O, a\sim\pi}[Q(s,a)] - \mathbb{E}_{s,a\sim d^O}[Q(s,a)]\right] \tag{66}$$

where P^π denotes the policy transition operator, P^π_* denotes the adjoint policy transition operator. Removing constant terms (Eq. (59)) with respect to optimization variables we end up with the following form for `dual-Q`:

$$\frac{1}{\alpha}\left[\underbrace{\mathbb{E}_{s\sim d^O, a\sim\pi}[Q(s,a)]}_{\text{reduce Q at OOD actions}} - \underbrace{\mathbb{E}_{s,a\sim d^O}[Q(s,a)]}_{\text{increase Q at in-distribution actions}}\right] + \underbrace{\mathbb{E}_{s,a\sim d^O}\left[\frac{y(s,a,r,s')^2}{4\alpha^2}\right]}_{\text{minimize Bellman Error}} \tag{67}$$

Hence the `dual-Q` optimization reduces to:

$$\max_\pi \min_Q \alpha\left[\mathbb{E}_{s\sim d^O, a\sim\pi}[Q(s,a)] - \mathbb{E}_{s,a\sim d^O}[Q(s,a)]\right] + \mathbb{E}_{s,a\sim d^O}\left[\frac{y(s,a,r,s')^2}{4}\right] \tag{68}$$

This update equation matches the unregularized CQL objective (Equation 3 in [43]). $\square$

Lemma 5. *XQL is an instance of* `dual-V` *under the semi-gradient update rule, where the f-divergence is the reverse Kullback-Liebler divergence, and d^O is the offline visitation distribution.*

Proof. We show that the Extreme Q-Learning [23] framework for offline and online RL is a special case of the dual framework, specifically the `dual-V` using the semi-gradient update rule.

Consider setting the f-divergence to be the KL divergence in the `dual-V` framework, the regularization distribution and the initial state distribution to be the replay buffer distribution ($d^O = d^R$ and $d_0 = d^R$). The conjugate of the generating function for KL divergence is given by $f^*(t) = e^{t-1}$.

$$\min_{V(s)} (1-\gamma)\mathbb{E}_{d_0(s)}[V(s)] + \mathbb{E}_{s,a\sim d^R}\left[f^*\left(\left[r(s,a) + \gamma\sum_{s'} p(s'|s,a)V(s') - V(s))\right]/\alpha\right)\right] \tag{69}$$

$$\min_{V(s)} (1-\gamma)\mathbb{E}_{d_0(s)}[V(s)] + \mathbb{E}_{s,a\sim d^S}\left[\exp(\left(\left[r(s,a) + \gamma\sum_{s'} p(s'|s,a)V(s') - V(s))\right]/\alpha - 1\right)\right] \tag{70}$$

A popular approach for stable optimization in temporal difference learning is the semi-gradient update rule which has been studied in previous works [72]. In this update strategy, we fix the targets for the temporal difference backup. The target in the above optimization is given by:

$$\bar{Q}(s,a) = r(s,a) + \gamma\sum_{s'} p(s'|s,a)V(s') \tag{71}$$

The update equation for V is now given by:

$$\min_{V(s)} (1-\gamma)\mathbb{E}_{d_0(s)}[V(s)] + \mathbb{E}_{s,a\sim d^R}\left[\exp(\left(\left[\bar{Q}(s,a) - V(s))\right]/\alpha - 1\right)\right] \tag{72}$$

where hat denotes the `stop-gradient` operation. We approximate this target by using mean-squared regression with the single sample unbiased estimate as follows:

$$\min_{Q} \mathbb{E}_{s,a,s'\sim d^R}\left[(Q(s,a) - (r(s,a) + V(s')))^2\right] \tag{73}$$

The procedure (alternating Eq. (72) and Eq. (73)) is now equivalent to the Extreme-Q learning and is a special case of the `dual-V` framework. □

C.2.1 f-`DVL`: A family of implicit policy improvement algorithms for RL

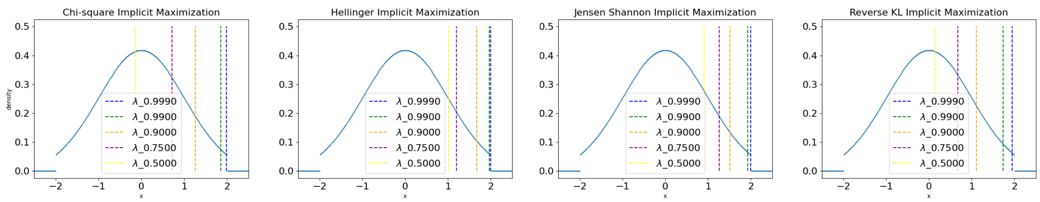

Figure 6: Illustration of a family of implicit maximizers corresponding to different f-divergences. The underlying data distribution is a truncated Gaussian `TN` with mean 0, variance 1 and a truncation range $(-2, 2)$. We sample 10000 data points from `TN` and compute the solution v_λ of Problem (13). As $\lambda \to 1$, the solution v_λ becomes a more accurate estimation for the supremum of the random variable x.

Lemma 6. *Let x be a real-valued random variable such that $\Pr(x > x^*) = 0$. Let v_λ be the solution of Problem (13). It holds that $v_{\lambda_1} \leqslant v_{\lambda_2}$, $\forall\, 0 < \lambda_1 < \lambda_2 < 1$. Further, $\lim_{\lambda\to 1} v_\lambda = x^*$.*

Proof. We analyze the behavior for the following optimization of interest.

$$\min_{v}(1-\lambda)\mathbb{E}_{x\sim D}[v] + \lambda\mathbb{E}_{x\sim D}\left[f_p^*\,(x-v)\right] \tag{74}$$

$f_p^*(t)$ is given by (using the definition in Eq. (45):

$$f_p^*(t) = -f\left(\max({f'}^{-1}(t), 0)\right) + t\max\left({f'}^{-1}(t), 0\right) \tag{75}$$

Accordingly, the function f_p^* admits two different behaviors given by:

$$f_p^* = \begin{cases} -f({f'}^{-1}(t)) + t{f'}^{-1}(t) = f^*(t), & \text{if } {f'}^{-1}(t) > 0 \\ -f(0), & \text{otherwise} \end{cases}$$

where f^* is the convex conjugate of f-divergence and is strictly increasing with t. We note other important properties related to f function for f-divergences: f^*, f', $(f')^{-1}$ is strictly increasing and ${f^*}' = {f'}^{-1}$. Even though f' does not admit a derivative to the right of 0, we define $f'(0) = \inf \cup_{x>0} f'(x)$ (similar to [63]). For all $x < 0$, $f^*(x) = -f(0)$, $f(0_+) > 0$ and $(f')^{-1}(t) > 0$ when $t > 0$ and 0 otherwise.

We analyze the second term in Eq. (74). It can be expanded as follows:

$$\lambda \int_{x:(f')^{-1}(x-v)>0} p(x)f^*(x-v)dx - \lambda \int_{x:(f')^{-1}(x-v)<0} f(0)p(x)dx \tag{76}$$

From the properties of f, we use the fact that $(f')^{-1}(x-v) > 0$ when $x - v > 0$ or equivalently $x > v$.

$$\lambda \int_{x>v} p(x)f^*(x-v)dx - \lambda \int_{x\leqslant v} f(0)p(x)dx \tag{77}$$

The first term in the above equation decreases monotonically and the second term increases monotonically (thus the combined terms decrease) as v increases until $v = x^*$ (supremum of the support of the distribution) after which the equation assumes a constant value of $-\lambda f(0)$.

Going back to our original optimization in Eq. (74), the first term decreases monotonically with v. As $\lambda \to 1$, the minimization of the second term takes precedence, with increasing v until saturation

($v = x^*$). We can go further to characterize the effect of λ on solution v_λ of the equation. The solution of the optimization can be written in closed form as (using stationarity):

$$\frac{(1-\lambda)}{\lambda} = \mathbb{E}_{x\sim D}\Big[f_p^{*\prime}(x-v)\Big] \tag{78}$$

Using the fact that $f_p^{*'}$ is non-decreasing, we can show that the right-hand term in the equation above increases as v decreases. This in turn implies that for all λ_1, λ_2 such that $\lambda_1 \leqslant \lambda_2$ we have that $v_{\lambda_1} \leqslant v_{\lambda_2}$. □

C.3 Dual Connections to Imitation Learning

This section outlines the reduction of a number of algorithms for Imitation Learning to the dual framework. Most prior methods can either take into account expert-only data for imitation whereas the other methods which do imitation from arbitrary offline data are limited by their assumptions and the form of f-divergence they optimize for. We walk through explaining how prior methods can be derived through the unified framework and also why they are limited.

C.3.1 Offline imitation learning with expert data only

We saw in Section 4.1, how using the `dual-Q` framework directly led to a reduction of IQ-Learn [22] as part of the dual framework. This was accomplished by simple setting the reward function to be 0 uniformly and setting the regularization distribution to the expert. Garg et al. [22] uses this method in the online imitation learning setting as well by incorporating the replay data as additional regularization which we suggest is unprincipled, also pointed out by others [3] (as only expert data samples can be leveraged in the above optimization) and provide a fix in the Section 4.2. In this section, we show how the same approach can directly lead to another method for learning to imitation from expert-only data avoiding the alternating min-max optimization of IQ-Learn.

IV-Learn: A new method for offline imitation learning: Analogous to `dual-Q` (offline imitation), we can leverage the `dual-V` (offline imitation) setting which avoids the min-max optimization given by:

`IV-Learn` or `dual-V` (offline imitation from expert-only data):

$$\min_{V(s)} (1-\gamma)\mathbb{E}_{d_0(s)}[V(s)] + \mathbb{E}_{s,a\sim d^E}[f^*\left([\mathcal{T}_0 V(s,a) - V(s))]/\alpha\right)] \tag{79}$$

We propose `dual-V` (offline imitation) to be a new method arising out of this framework which we leave for future exploration. This work primarily focuses on imitation learning from general off-policy data.

Proofs for this section:

Corollary 1. *IBC [15] is an instance of* `dual-Q` *using the full-gradient update rule, where* $r(s,a) = 0\ \forall s \in \mathcal{S}, a \in \mathcal{A}$, $d^O = d^E$, *and the f-divergence is the total variation distance.*

Eq. (6) suggests that intuitively IQ-Learn trains an energy-based model in the form of Q where it pushes down the Q-values for actions predicted by current policy and pushes up the Q-values at the expert state-action pairs. This becomes more clear when the divergence f is chosen to be Total-Variation ($f^* = \mathbb{I}$), IQ-Learn for Total-Variation divergence reduces to:

$$(1-\gamma)\mathbb{E}_{d_0(s),\pi(a|s)}[Q(s,a)] + \mathbb{E}_{s,a\sim d^E}\left[\gamma\sum_{s',a'} p(s'|s,a)\pi(a'|s')Q(s',a') - Q(s,a)\right] \tag{80}$$

$$= \left[(1-\gamma)\mathbb{E}_{d_0(s),\pi(a|s)}[Q(s,a)] + \mathbb{E}_{s,a\sim d^E}\left[\gamma\sum_{s',a'} p(s'|s,a)\pi(a'|s')Q(s',a')\right]\right] - \mathbb{E}_{s,a\sim d^E}[Q(s,a)] \tag{81}$$

First, we simplify the initial two terms:

$$
\begin{aligned}
&(1-\gamma)\mathbb{E}_{d_0(s),\pi(a|s)}[Q(s,a)] + \mathbb{E}_{s,a\sim d^E}\left[\gamma\sum_{s'} p(s'|s,a)\pi(a'|s')Q(s',a')\right] && (82)\\
&= (1-\gamma)\sum_{s,a} d_0(s)\pi(a|s)Q(s,a) + \gamma\sum_{s,a} d^E(s,a)\sum_{s',a'} p(s'|s,a)\pi(a'|s')Q(s',a') && (83)
\end{aligned}
$$

$$
\begin{aligned}
&= (1-\gamma)\sum_{s,a} d_0(s)\pi(a|s)Q(s,a) + \gamma\sum_{s',a'}\sum_{s,a} d^E(s,a)p(s'|s,a)\pi(a'|s')Q(s',a') && (84)\\
&= (1-\gamma)\sum_{s,a} d_0(s)\pi(a|s)Q(s,a) + \gamma\sum_{s',a'} \pi(a'|s')Q(s',a')(\sum_{s,a} d^E(s,a)p(s'|s,a)) && (85)\\
&= (1-\gamma)\sum_{s,a} d_0(s)\pi(a|s)Q(s,a) + \gamma\sum_{s',a'} \pi(a'|s')Q(s',a')(\sum_{s,a} d^E(s,a)p(s'|s,a)) && (86)
\end{aligned}
$$

$$
\begin{aligned}
&= (1-\gamma)\sum_{s,a} d_0(s)\pi(a|s)Q(s,a) + \gamma\sum_{s,a} \pi(a|s)Q(s,a)(\sum_{s',a'} d^E(s',a')p(s|s',a')) && (87)\\
&= \sum_{s,a}(1-\gamma)d_0(s)\pi(a|s)Q(s,a) + \pi(a|s)Q(s,a)(\sum_{s',a'} d^E(s',a')p(s|s',a')) && (88)\\
&= \sum_{s,a}\pi(a|s)Q(s,a)\left[(1-\gamma)d_0(s) + \gamma\sum_{s',a'} d^E(s',a')p(s|s',a')\right] && (89)\\
&= \sum_{s,a}\pi(a|s)Q(s,a)d^E(s) && (90)
\end{aligned}
$$

where the last step is due to the steady state property of the MDP (Bellman flow constraint).

Therefore IQ-Learn/`dual-Q` for offline imitation (in the special case of TV divergence) simplifies to (from Eq. (81)):

$$
\left[(1-\gamma)\mathbb{E}_{d_0(s),\pi(a|s)}[Q(s,a)] + \mathbb{E}_{s,a\sim d^E}\left[\gamma\sum_{s',a'} p(s'|s,a)\pi(a'|s')Q(s',a')\right]\right] - \mathbb{E}_{s,a\sim d^E}[Q(s,a)] \tag{91}
$$

$$
= \min_Q \mathbb{E}_{d_E(s),\pi(a|s)}[Q(s,a)] - \mathbb{E}_{s,a\sim d^E}[Q(s,a)] \tag{92}
$$

The update gradient w.r.t for the above optimization matches the gradient update of infoNCE objective in Implicit Behavior Cloning [15] with Q as the energy-based model.

C.4 Off-policy imitation learning (under coverage assumption)

Directly utilizing the dual-RL framework for imitation has its limitation as we see in the previous section – we cannot leverage off-policy suboptimal data. We first show that it is easy to see why choosing the f-divergence to reverse KL makes it possible to get an off-policy objective for imitation learning in the dual framework. We start with the `primal-Q` for imitation learning under the reverse KL-divergence regularization ($r(s,a) = 0$ and $d^O = d^E$):

$$
\begin{aligned}
&\max_{d(s,a)\geqslant 0,\pi(a|s)} -D_{\mathrm{KL}}(d(s,a)\ ||\ d^E(s,a))\\
&\text{s.t } d(s,a) = (1-\gamma)\rho_0(s).\pi(a|s) + \gamma\pi(a|s)\sum_{s',a'} d(s',a')p(s|s',a').
\end{aligned} \tag{93}
$$

Under the assumption that the suboptimal data visitation (denoted by d^S) covers the expert visitation ($d^S > 0$ wherever d^E>0) [48], which we refer to as the **coverage assumption**, the reverse KL divergence can be expanded as follows:

$$D_{\mathrm{KL}}(d(s,a) \mid\mid d^E(s,a)) = \mathbb{E}_{s,a\sim d(s,a)}\left[\log \frac{d(s,a)}{d^E(s,a)}\right] = \mathbb{E}_{s,a\sim d(s,a)}\left[\log \frac{d(s,a)}{d^E(s,a)}\frac{d^S(s,a)}{d^S(s,a)}\right] \tag{94}$$

$$= \mathbb{E}_{s,a\sim d(s,a)}\left[\log \frac{d(s,a)}{d^S(s,a)} + \log \frac{d^S(s,a)}{d^E(s,a)}\right] \tag{95}$$

$$= \mathbb{E}_{s,a\sim d(s,a)}\left[\log \frac{d^S(s,a)}{d^E(s,a)}\right] + D_{\mathrm{KL}}(d(s,a) \mid\mid d^S(s,a)). \tag{96}$$

Hence the `primal-Q` can now be written as:

$$\max_{d(s,a)\geqslant 0,\pi(a|s)} \mathbb{E}_{s,a\sim d(s,a)}\left[-\log \frac{d^S(s,a)}{d^E(s,a)}\right] - D_{\mathrm{KL}}(d(s,a) \mid\mid d^S(s,a)) \tag{97}$$

$$\text{s.t } d(s,a) = (1-\gamma)\rho_0(s).\pi(a|s) + \gamma \sum_{s',a'} d(s',a')p(s|s',a')\pi(a|s). \tag{98}$$

Now, in the optimization above the first term resembles the reward function and the second term resembles the divergence constraint with a new distribution $d^S(s,a)$ in the original regularized RL primal (Eq. (24)). Hence we can obtain respective `dual-Q` and `dual-V` in the setting for off-policy imitation learning using the reward function as $r^{\text{imit}}(s,a) = -\log \frac{d^S(s,a)}{d^E(s,a)}$ and the new regularization distribution as $d^S(s,a)$. Using $\mathcal{T}^\pi_{r^{\text{imit}}}$ and $\mathcal{T}_{r^{\text{imit}}}$ to denote backup operators under a new reward function r^{imit}, we have

`dual-Q` for off-policy imitation (coverage assumption) :

$$\max_{\pi(a|s)} \min_{Q(s,a)} (1-\gamma)\mathbb{E}_{\rho_0(s),\pi(a|s)}[Q(s,a)] + \mathbb{E}_{s,a\sim d^S}[f^*(\mathcal{T}^\pi_{r^{\text{imit}}}Q(s,a) - Q(s,a))]. \tag{99}$$

This choice of KL divergence leads us to a reduction of another method, OPOLO [82] for off-policy imitation learning to `dualQ` which we formalize in the lemma below:

Lemma 2. *OPOLO [82] is an instance of* `dual-Q` *using the semi-gradient update rule, where* $r(s,a) = 0\ \forall \mathcal{S},\mathcal{A}$, $d^O = d^E$*, and the f-divergence set to the reverse KL divergence.*

Proof. Proof is sketched in the above section, ie. Eq. (99) is the update equation for OPOLO.

Analogously we have `dual-V` for off-policy imitation (coverage assumption):

$$\min_{V(s)} (1-\gamma)\mathbb{E}_{\rho_0(s)}[V(s)] + \mathbb{E}_{s,a\sim d^S}[f^*(\mathcal{T}_{r^{\text{imit}}}V(s,a) - V(s))]. \tag{100}$$

We note that the `dual-V` framework for off-policy imitation learning under coverage assumptions was studied in the imitation learning work SMODICE [48].

C.5 Logistic Q-learning and P^{2}IL as dual-QV methods

Logistic Q-learning and Proximal Point Imitation Learning (P^{2}IL) uses a modified primal for regularized policy optimization:

$$\max_{d\geqslant 0} \mathbb{E}_{d(s,a)}[r(s,a)] - D_f(d(s,a) \mid\mid d^O(s,a)) - H(\mu(s,a)\|\mu^O(s,a)$$

$$\text{s.t } d(s,a) = (1-\gamma)d_0(s) + \pi(a|s)\gamma \sum_{s',a'} \mu(s',a')p(s|s',a'). \tag{101}$$

$$\text{and } d(s,a) = \mu(s,a) \tag{102}$$

where $H(\mu(s,a)\|\mu^O(s,a)) = \sum \mu(s,a)\log\frac{\pi_\mu(a|s)}{\pi_{\mu^O}(a|s)}$ denotes the conditional relative entropy and μ^O is another offline distribution of state-action transitions potentially the same as d^O. The optimization is overparameterized (setting $\mu = d$). This trick was popularized via [53] and leads to unbiased estimators and better downstream data driven algorithms. We call these two methods `dual-QV` as their dual requires estimating both Q and V as shown in [79, 6]

D ReCOIL: Off-policy imitation learning without the coverage assumption

Understanding the limitations of existing imitation learning methods in the dual framework, we now proceed to derive our proposed method for imitation learning with arbitrary (off-policy) data. The derivation for the `dual-Q` setting is shown below. `dual-V` derivation proceeds analogously.

Lemma 7. ***(`dual-Q` for off-policy imitation (relaxed coverage assumption))*** *Imitation learning using off-policy data can be solved by optimizing the following modified dual objective for `primal-Q` with $r(s,a) = 0\ \forall \mathcal{S}, \mathcal{A}$ and f-divergence considered between distributions $d^S_{mix}(s,a) := \beta d(s,a) + (1-\beta)d^S(s,a)$ and $d^{E,S}_{mix}(s,a) := \beta d^E(s,a) + (1-\beta)d^S(s,a)$, and is given by:*

$$\max_{\pi(a|s)} \min_{Q(s,a)} \beta(1-\gamma)\mathbb{E}_{d_0(s),\pi(a|s)}[Q(s,a)] + \mathbb{E}_{s,a\sim d^{E,S}_{mix}(s,a)}\left[f^*_p(\mathcal{T}^\pi_0 Q(s,a) - Q(s,a))\right] \\ - (1-\beta)\mathbb{E}_{s,a\sim d^S}[\mathcal{T}^\pi_0 Q(s,a) - Q(s,a)] \tag{103}$$

Proof. Let's define two mixture distributions that we are going to leverage to formulate the imitation learning problem: $d^S_{\text{mix}}(s,a) := \beta d(s,a) + (1-\beta)d^S(s,a)$ and $d^{E,S}_{\text{mix}}(s,a) := \beta d^E(s,a) + (1-\beta)d^S(s,a)$. $d^S_{\text{mix}}(s,a)$ is a mixture between the current agent's visitation distribution with suboptimal transition dataset obtained from a mixture of arbitrary policies and $d^{E,S}_{\text{mix}}(s,a)$ is the mixture between the expert's visitation distribution with arbitrary experience from the offline transition dataset. Minimizing the divergence between these visitation distributions still solves the imitation learning problem, i.e $d = d^E$. We again start with the new modified `primal-Q` under this mixture divergence regularization:

$$\max_{d(s,a)\geqslant 0,\pi(a|s)} -D_f(d^S_{\text{mix}}(s,a)(s,a) \mid\mid d^{E,S}_{\text{mix}}(s,a)(s,a)) \\ \text{s.t } d(s,a) = (1-\gamma)\rho_0(s).\pi(a|s) + \gamma\pi(a|s)\sum_{s',a'} d(s',a')p(s|s',a').$$

Using the same algebraic machinery of duality as before (Section C.1.3) to get an unconstrained tractable optimization problem, we obtain:

$$\max_{\pi,d\geqslant 0} \min_{Q(s,a)} -D_f(d^S_{\text{mix}}(s,a) \mid\mid d^{E,S}_{\text{mix}}(s,a)) \\ + \sum_{s,a} Q(s,a)\left((1-\gamma)d_0(s).\pi(a|s) + \gamma\sum_{s',a'} d(s',a')p(s|s',a')\pi(a|s) - d(s,a)\right) \tag{104}$$

$$= \max_{\pi,d\geqslant 0} \min_{Q(s,a)} (1-\gamma)\mathbb{E}_{d_0(s),\pi(a|s)}[Q(s,a)] \\ + \mathbb{E}_{s,a\sim d}\left[\gamma\sum_{s'} p(s'|s,a)\pi(a'|s')Q(s',a') - Q(s,a)\right] - D_f(d^S_{\text{mix}}(s,a) \mid\mid d^{E,S}_{\text{mix}}(s,a)) \tag{105}$$

$$= \max_{\pi,d\geqslant 0} \min_{Q(s,a)} (1-\gamma)\mathbb{E}_{d_0(s),\pi(a|s)}[Q(s,a)] \\ + \beta\mathbb{E}_{s,a\sim d}\left[\gamma\sum_{s'} p(s'|s,a)\pi(a'|s')Q(s',a') - Q(s,a)\right] \\ + (1-\beta)\mathbb{E}_{s,a\sim d^S}\left[\gamma\sum_{s'} p(s'|s,a)\pi(a'|s')Q(s',a') - Q(s,a)\right] \\ - (1-\beta)\mathbb{E}_{s,a\sim d^S}\left[\gamma\sum_{s'} p(s'|s,a)\pi(a'|s')Q(s',a') - Q(s,a)\right] - D_f(d^S_{\text{mix}}(s,a) \mid\mid d^{E,S}_{\text{mix}}(s,a)) \tag{106}$$

Before moving forward with the derivation, we summarize the result of the derivation so far:

Imitation from Arbitrary data (dualQ, no nonnegativity constraints)

$$= \max_{\pi(a|s)} \min_{Q(s,a)} \max_{d\geqslant 0} \alpha(1-\gamma)\mathbb{E}_{d_0(s),\pi(a|s)}[Q(s,a)]$$
$$+ \mathbb{E}_{s,a\sim d^S_{\text{mix}}(s,a)}\left[\gamma\sum_{s'} p(s'|s,a)\pi(a'|s')Q(s',a') - Q(s,a)\right] - D_f(d^S_{\text{mix}}(s,a) \,||\, d^{E,S}_{\text{mix}}(s,a))$$
$$- (1-\alpha)\mathbb{E}_{s,a\sim d^S}\left[\gamma\sum_{s'} p(s'|s,a)\pi(a'|s')Q(s',a') - Q(s,a)\right] \tag{107}$$

Note that the inner maximization with respect to d has the constraint that $d \geqslant 0$. In this setting, to get a tractable closed form we replace the optimization variable from d to $d^S_{\text{mix}}(s,a)$ with the constraint that $d \geqslant 0$. This prevents the optimization to result in values for $d^S_{\text{mix}}(s,a)$ which has $d(s,a) < 0$ for some s, a. This nonnegativity constraint was not necessary for the previous settings for `dual-Q` problems we have discussed in RL and IL (as the constraints implied a unique solution which is no longer the case). Ignoring this constraint ($d \geqslant 0$) results in the following dual-optimization for imitation from arbitrary data.

$$\max_{\pi(a|s)} \min_{Q(s,a)} \alpha(1-\gamma)\mathbb{E}_{d_0(s),\pi(a|s)}[Q(s,a)]$$
$$+ \mathbb{E}_{s,a\sim d^{E,S}_{\text{mix}}(s,a)}\left[f^*(\gamma\sum_{s'} p(s'|s,a)\pi(a'|s')Q(s',a') - Q(s,a))\right]$$
$$- (1-\alpha)\mathbb{E}_{s,a\sim d^S}\left[\gamma\sum_{s'} p(s'|s,a)\pi(a'|s')Q(s',a') - Q(s,a)\right] \tag{108}$$

To incorporate the nonnegativity constraints, we need to obtain the closed form solution for maximization w.r.t $d \geqslant 0$. To do that, we start with the inner maximization w.r.t $d^S_{\text{mix}}(s,a)$ and consider the terms dependent on $d^S_{\text{mix}}(s,a)$ below.

$$\max_{d^S_{\text{mix}}(s,a), d\geqslant 0} \mathbb{E}_{s,a\sim d^S_{\text{mix}}(s,a)}\left[\gamma\sum_{s'} p(s'|s,a)\pi(a'|s')Q(s',a') - Q(s,a)\right] - D_f(d^S_{\text{mix}}(s,a) \,||\, d^{E,S}_{\text{mix}}(s,a)) \tag{109}$$

Let $p(s,a) = \frac{(1-\alpha)d^S(s,a)}{\alpha d^E(s,a)+(1-\alpha)d^S(s,a)}$, $y(s,a) = \gamma\sum_{s'} p(s'|s,a)\pi(a'|s')Q(s',a') - Q(s,a)$ and $w(s,a) = \frac{d^S_{\text{mix}}(s,a)}{d^{E,S}_{\text{mix}}(s,a)}$. We construct the Lagrangian dual to incorporate the constraint $d \geqslant 0$ in its equivalent form $w(s,a) \geqslant p(s,a)$ and obtain the following:

$$\max_{w(s,a)} \max_{\lambda\geqslant 0} \mathbb{E}_{s,a\sim d^{E,S}_{\text{mix}}(s,a)}[w(s,a)y(s,a)] - \mathbb{E}_{d^{E,S}_{\text{mix}}(s,a)}[f(w(s,a))] + \sum_{s,a}\lambda(w(s,a) - p(s,a)) \tag{110}$$

Since strong duality holds, we can use the KKT constraints to find the solutions $w^*(s,a)$ and $\lambda^*(s,a)$.

1. Primal feasibility: $w^*(s,a) \geqslant p(s,a) \;\; \forall\, s,a$
2. Dual feasibility: $\lambda^* \geqslant 0 \;\; \forall\, s,a$
3. Stationarity: $d^{E,S}_{\text{mix}}(s,a)(f'(w^*(s,a)) + y(s,a) + \lambda^*(s,a)) = 0 \;\; \forall\, s,a$
4. Complementary Slackness: $(w^*(s,a) - p(s,a))\lambda^*(s,a) = 0 \;\; \forall\, s,a$

Using stationarity we have the following:

$$f'(w^*(s,a)) = y(s,a) + \lambda^*(s,a) \;\; \forall\, s,a \tag{111}$$

Now using complementary slackness, only two cases are possible $w^*(s,a) \geqslant p(s,a)$ or $\lambda^*(s,a) \geqslant 0$. Combining both cases we arrive at the following solution for this constrained optimization:

$$w^*(s,a) = \max\left(p(s,a), {f'}^{-1}(y(s,a))\right) \tag{112}$$

Plugging in the optimal solution for Eq. (110) (w^*) back in Eq. (107), we obtain

$$\begin{aligned}
&\max_{\pi(a|s)} \min_{Q(s,a)} \alpha(1-\gamma)\mathbb{E}_{d_0(s),\pi(a|s)}[Q(s,a)] \\
&+ \mathbb{E}_{s,a\sim d^{E,S}_{\text{mix}}(s,a)}\left[\max\left(p(s,a),(f')^{-1}(y(s,a))\right)y(s,a) - \alpha f\left(\max\left(p(s,a),(f')^{-1}(y(s,a))\right)\right)\right] \\
&- (1-\alpha)\mathbb{E}_{s,a\sim d^S}\left[r(s,a) + \gamma\sum_{s'} p(s'|s,a)\pi(a'|s')Q(s',a') - Q(s,a)\right]
\end{aligned} \tag{113}$$

Thus, the closed-form solution with the nonnegativity constraints requires us to use the ratio $p(s,a)$ to threshold the distribution ratio. We observed in our experiments that ignoring the nonnegativity constraints still resulted in a similarly performant method while having the benefits of being more stable. A similar derivation can be done in V-space to obtain an analogous result for `ReCOIL-V`. □

D.1 Suboptimality Bound for `ReCOIL-V`

Recall that `ReCOIL-V` admits a `dual-V` form (9). When deriving `dual-V`, there is one step (Eq. (39)) where we assumed the importance sampling is exact, i.e.,

$$\mathbb{E}_{(s,a)\sim d}[\mathcal{T}V(s,a) - V(s)] = \mathbb{E}_{(s,a)\sim d^O}\left[\tfrac{d(s,a)}{d^O(s,a)}(\mathcal{T}V(s,a) - V(s))\right]. \tag{114}$$

However, this assumption does not hold in general and is not practical, because d^O and d might have different support. The gap between the two terms greatly affects the performance of dual RL approaches. We shall bound the approximation error introduced by importance sampling for `ReCOIL-V` in Section D.1.1, and then bound the suboptimality of the learned policy in Section D.1.2, under mild conditions. This analysis also results in the suboptimality bound of `IV-Learn` and `IQ-Learn` methods.

Let S^J denote the joint support of d^S and d^E. Let $r(s,a) = V(s) - \gamma\mathcal{T}_0 V(s,a)$ be the pseudo-reward implied by `ReCOIL` and $R_{\max} = \max_{s,a}|r(s,a)|$. Let $D_\delta = \left\{d \,|\, \Pr_d\left((s,a)\in S^J\right) \geqslant 1-\delta\right\}$ be the set of visitation distributions that have $1-\delta$ coverage of S^J, where $\Pr_d\left((s,a)\in S^J\right)$ is the probably that (s,a) lies in S^J when sampling (s,a) from d.

We make the following assumptions for our proof:

Assumption 1 We consider imitation learning under the constraint $d \in D_\delta$. This is similar to pessimism assumption when learning from fixed datasets in offline RL [47].

Assumption 2 The hyperparameter β for defining $d^S_{\text{mix}}(s,a)$ and $d^{E,S}_{\text{mix}}(s,a)$ goes to 1: $\beta \to 1$.

Assumption 3 The function $h(V)$ defined in Section D.1.2 is κ-strongly convex.

For Assumption 1, `ReCOIL-V` (see Algorithm 1) is able to find a policy under the visitation constraint as a result of a combination of implicit maximization, which prevents overestimation and thus choosing OOD action, and weighted behavior cloning (Advantage-weighted regression), which keeps the output policy close to the dataset policy.

D.1.1 Approximation Error of the Imitation Learning Objective

The imitation learning problem can be written in the Lagrangian form of `primal-V` where $r(s,a) = 0$ everywhere:

$$\min_V \max_{d\in D_\delta} (1-\gamma)\mathbb{E}_{d_0(s)}[V(s)] + \mathbb{E}_d[\mathcal{T}_0 V(s,a) - V(s)] - D_f(d(s,a)\,||\,d^E(s,a)), \tag{115}$$

where we have a constraint $d \in D_\delta$ due to Assumption 1. `ReCOIL-V` optimizes a surrogate objective of Problem (115). To derive `ReCOIL-V`, consider the corresponding `primal-V` in its Lagrangian form

$$\min_V \max_{d \in D_\delta} (1-\gamma)\mathbb{E}_{d_0(s)}[V(s)] + \mathbb{E}_d[\mathcal{T}_0 V(s,a) - V(s)] - D_f(d^S_{\text{mix}}(s,a) \,||\, d^{E,S}_{\text{mix}}(s,a)). \tag{116}$$

Rewriting the second term, we obtain

$$\begin{aligned}\min_V \max_{d \in D_\delta} (1-\gamma)\mathbb{E}_{d_0(s)}[V(s)] + \frac{1}{\beta}\mathbb{E}_{s,a\sim d^S_{\text{mix}}(s,a)}[\mathcal{T}_0 V(s,a) - V(s)] \\ - D_f(d^S_{\text{mix}}(s,a) \,||\, d^{E,S}_{\text{mix}}(s,a)) - \frac{1-\beta}{\beta}\mathbb{E}_{d^S}[\mathcal{T}_0 V(s,a) - V(s)].\end{aligned} \tag{117}$$

Now we *approximate* the second term via importance sampling, which leads to

$$\begin{aligned}\min_V \max_{d \in D_\delta} (1-\gamma)\mathbb{E}_{d_0(s)}[V(s)] + \frac{1}{\beta}\mathbb{E}_{s,a\sim d^{E,S}_{\text{mix}}(s,a)}\left[\frac{d^S_{\text{mix}}(s,a)}{d^{E,S}_{\text{mix}}(s,a)}(\mathcal{T}_0 V(s,a) - V(s))\right] \\ - \mathbb{E}_{d^{E,S}_{\text{mix}}(s,a)}\left[f(\frac{d^S_{\text{mix}}(s,a)}{d^{E,S}_{\text{mix}}(s,a)})\right] - \frac{1-\beta}{\beta}\mathbb{E}_{d^S}[\mathcal{T}_0 V(s,a) - V(s)].\end{aligned} \tag{118}$$

By expanding $d^S_{\text{mix}}(s,a) = \beta d(s,a) + (1-\beta)d^S(s,a)$, we obtain

$$\begin{aligned}\min_V \max_{d \in D_\delta} (1-\gamma)\mathbb{E}_{d_0(s)}[V(s)] + \frac{1}{\beta}\mathbb{E}_{s,a\sim d^{E,S}_{\text{mix}}(s,a)}\left[\frac{d^S_{\text{mix}}(s,a)}{d^{E,S}_{\text{mix}}(s,a)}(\mathcal{T}_0 V(s,a) - V(s))\right] \\ - \mathbb{E}_{d^{E,S}_{\text{mix}}(s,a)}\left[f\left(\frac{d^S_{\text{mix}}(s,a)}{d^{E,S}_{\text{mix}}(s,a)}\right)\right] - \frac{1-\beta}{\beta}\mathbb{E}_{d^S}[\mathcal{T}_0 V(s,a) - V(s)],\end{aligned} \tag{119}$$

This can be further simplified to

$$\begin{aligned}\min_V \max_{d \in D_\delta} (1-\gamma)\mathbb{E}_{d_0(s)}[V(s)] + \mathbb{E}_{s,a\sim d^{E,S}_{\text{mix}}(s,a)}\left[\frac{d(s,a)}{d^{E,S}_{\text{mix}}(s,a)}(\gamma\mathcal{T}_0 V(s,a) - V(s))\right] \\ - \mathbb{E}_{d^{E,S}_{\text{mix}}(s,a)}\left[f\left(\frac{d^S_{\text{mix}}(s,a)}{d^{E,S}_{\text{mix}}(s,a)}\right)\right],\end{aligned} \tag{120}$$

where we used the fact

$$\mathbb{E}_{s,a\sim d^{E,S}_{\text{mix}}(s,a)}\left[\frac{d^S(s,a)}{d^{E,S}_{\text{mix}}(s,a)}(\mathcal{T}_0 V(s,a) - V(s))\right] = \mathbb{E}_{s,a\sim d^S}[\mathcal{T}_0 V(s,a) - V(s)]$$

as the support of $d^{E,S}_{\text{mix}}(s,a)$ contains the support of d^S.

Let $g(d,V)$ and $\widehat{g}_{\texttt{ReCOIL}}(d,V)$ be the objective functions of Problem (115) and (120). $g(d,V)$ is the original IL objective we want to solve, and $\widehat{g}_{\texttt{ReCOIL}}(d,V)$ is an approximation (with importance sampling) of $g(d,V)$ used by `ReCOIL-V`. To simplify the analysis, we consider the case when mixture ratio $\beta \to 1$ (Assumption 2), so that *the approximation error of the objective function reduces to the approximation error of importance sampling.* That is,

$$|g(d,V) - \widehat{g}_{\texttt{ReCOIL}}(d,V)| \to \left|\mathbb{E}_d[\mathcal{T}_0 V(s,a) - V(s)] - \mathbb{E}_{d^{E,S}_{\text{mix}}(s,a)}\left[\tfrac{d(s,a)}{d^{E,S}_{\text{mix}}(s,a)}(\mathcal{T}_0 V(s,a) - V(s))\right]\right|. \tag{121}$$

For any visitation distribution $d \in D_\delta$, it holds that

$$\begin{aligned}&\left|\mathbb{E}_d[(\mathcal{T}_0 V(s,a) - V(s))] - \mathbb{E}_{d^{E,S}_{\text{mix}}(s,a)}\left[\tfrac{d(s,a)}{d^{E,S}_{\text{mix}}(s,a)}(\mathcal{T}_0 V(s,a) - V(s))\right]\right| \\ &\leqslant \mathbb{E}_{s,a\in S^d\setminus S^J}[|\mathcal{T}_0 V(s,a) - V(s)|] \leqslant \max \delta\, |\mathcal{T}_0 V(s,a) - V(s)| \leqslant \delta R_{\max},\end{aligned} \tag{122}$$

where S^d is the support of d, and the second inequality follows from the definition of D_δ. As a consequence, we can bound the approximation error

$$\epsilon_{\texttt{ReCOIL}} = \max_{d\in D_\delta, V}\left|g(d,V) - \lim_{\beta\to 1}\widehat{g}_{\texttt{ReCOIL}}(d,V)\right| \leqslant \delta R_{\max}. \tag{123}$$

Similarly, one can show that for IV-Learn, we have

$$|g(d,V)-\widehat{g}_{\texttt{IVLearn}}(d,V)| \rightarrow \left|\mathbb{E}_d[\mathcal{T}_0 V(s,a)-V(s)]-\mathbb{E}_{s,a\sim d^E}\left[\frac{d(s,a)}{d^E(s,a)}\right](\mathcal{T}_0 V(s,a)-V(s))\right|. \tag{124}$$

Let S^E be the support of d^E. Unlike ReCOIL-V, the objective of IVLearn suffers from the following worst-case estimation error

$$\begin{aligned}&\left|\mathbb{E}_d[(\mathcal{T}_0 V(s,a)-V(s))]-\mathbb{E}_{d^E}\left[\frac{d(s,a)}{d^E(s,a)}(\mathcal{T}_0 V(s,a)-V(s))\right]\right|\\ &\leqslant \mathbb{E}_{(s,a)\in S^d\setminus S^E}[|\mathcal{T}_0 V(s,a)-V(s)|] \leqslant \max|\mathcal{T}_0 V(s,a)-V(s)| \leqslant R_{\max},\end{aligned} \tag{125}$$

and consequently

$$\epsilon_{\texttt{IVLearn}} = \max_{d\in D_\delta, V}\left|g(d,V)-\lim_{\beta\to 1}\widehat{g}_{\texttt{IVLearn}}(d,V)\right| \leqslant R_{\max}. \tag{126}$$

We note that the same approximation error bounds hold similarly for IQLearn as that of IVLearn. Thus ReCOIL has a smaller upper bound for the approximation error than IQLearn which we will see in the next sections leads to a better performance guarantee than IQLearn.

D.1.2 Performance Bound of the Learned Policy

Recall that $\epsilon_{\texttt{ReCOIL}}$ denotes the approximation error of the objective function by ReCOIL-V:

$$\epsilon_{\texttt{ReCOIL}} = \max_{d\in D_\delta, V}\left|g(d,V)-\lim_{\beta\to 1}\widehat{g}(d,V)\right|. \tag{127}$$

Let $h(V) = \max_{d\in D_\delta} g(d,V)$ and $\widehat{h}(V) = \max_{d\in D_\delta}\lim_{\beta\to 1}\widehat{g}(d,V)$. It directly follows from Eq. (127) that

$$|\widehat{h}(V)-h(V)| \leqslant 2\epsilon_{\texttt{ReCOIL}},\ \forall V. \tag{128}$$

We note that $\max_d g(d,V)$ (without the $d\in D_\delta$ constraint) is the standard dual-V form for imitation learning, but $h(V)$ here is defined as the same optimization under a constrained set $d\in D_\delta$.

Let $\widehat{V} = \arg\min_V \widehat{h}(V)$ and $V^* = \arg\min_V h(V)$. We are interested in bounding the gap $h(\widehat{V})-h(V^*)$. It holds that

$$\begin{aligned}h(\widehat{V})-h(V^*) &= h(\widehat{V})-\widehat{h}(\widehat{V})+\widehat{h}(\widehat{V})-h(V^*) &(129)\\ &= h(\widehat{V})-\widehat{h}(\widehat{V})+\widehat{h}(\widehat{V})-\widehat{h}(V^*)+\widehat{h}(V^*)-h(V^*) &(130)\\ &\leqslant 2\epsilon_{\texttt{ReCOIL}}+0+2\epsilon_{\texttt{ReCOIL}} &(131)\\ &= 4\epsilon_{\texttt{ReCOIL}}, &(132)\end{aligned}$$

where the inequality follows from Eq. (128) and the fact $\widehat{V} = \arg\min_V \widehat{h}(V)$.

As a consequence, we have

$$\begin{aligned}4\epsilon_{\texttt{ReCOIL}} &\geqslant h(\widehat{V})-h(V^*) &(133)\\ &\geqslant h(V^*)+(V^*-\widehat{V})\nabla h(V^*)+\frac{\kappa}{2}\|V^*-\widehat{V}\|_F^2-h(V^*) &(134)\\ &= \frac{\kappa}{2}\|V^*-\widehat{V}\|_F^2, &(135)\end{aligned}$$

where the second inequality comes from the fact that the function $h(V)$ is κ-strongly convex (Assumption 3) and $\nabla h(V^*) = 0$. It directly follows that

$$\|V^*-\widehat{V}\|_\infty \leqslant \|V^*-\widehat{V}\|_F \leqslant 2\sqrt{\tfrac{2}{\kappa}\epsilon_{\texttt{ReCOIL}}}. \tag{136}$$

Let π_δ^* be the policy that acts greedily with value function V^*, which is an optimal policy over all policies whose visitation distribution is within D_δ. Let $\widehat{\pi}$ denote the policy that acts greedily with value function $\widehat{V}$, i.e., the output policy of ReCOIL-V. We then use the results in Singh and Yee [71] to bound the performance gap between π_δ^* and $\widehat{\pi}$:

$$J^{\pi_\delta^*}-J^{\widehat{\pi}} \leqslant \frac{4}{1-\gamma}\sqrt{\frac{2\epsilon_{\texttt{ReCOIL}}}{\kappa}} \leqslant \frac{4}{1-\gamma}\sqrt{\frac{2\delta R_{\max}}{\kappa}}. \tag{137}$$

Algorithm 1: ReCOIL-V Idealized Algorithm (Under Stochastic Dynamics)

1: Initialize Q_ϕ, $\bar{Q}_\phi$(target Q-function), V_θ, and π_ψ, mixing ratio β
2: Let $\mathcal{D}^{\mathcal{S}} = (s, a, s')$ be data (possibly suboptimal) from the suboptimal transition dataset (online or offline)
3: Let $\mathcal{D}^{\mathcal{E}} = (s, a, s')$ be expert data transitions. Let $\mathcal{D}$ be a sampling distribution s.t $s, a \sim \mathcal{D} = \{s, a \sim \mathcal{D}^{\mathcal{S}}$ w.p $1-\beta, s, a \sim \mathcal{D}^{\mathcal{E}}$ w.p $\beta\}$
4: **for** $t = 1..T$ iterations **do**
5: Train Q_ϕ using $\min_\phi \mathcal{L}(\phi)$:

$$\begin{aligned}\mathcal{L}(\phi) = \beta(1-\gamma)\mathbb{E}_{\mathcal{D}}[V_\theta(s)] + \mathbb{E}_{s,a\sim\mathcal{D}}\big[f_p^*(Q_\phi(s,a) - V_\theta(s))\big] \\ - (1-\beta)\mathbb{E}_{s,a\sim\mathcal{D}^{\mathcal{S}}}[Q_\phi(s,a) - V_\theta(s)] \end{aligned} \tag{140}$$

6: Train V_θ using $\min_\theta \mathcal{J}(\theta)$

$$\begin{aligned}\mathcal{J}(\theta) = \beta(1-\gamma)\mathbb{E}_{\mathcal{D}}[V_\theta(s)] + \mathbb{E}_{s,a\sim\mathcal{D}}\big[f_p^*(Q_\phi(s,a) - V_\theta(s))\big] \\ - (1-\beta)\mathbb{E}_{s,a\sim\mathcal{D}^{\mathcal{S}}}[Q_\phi(s,a) - V_\theta(s)] \end{aligned} \tag{141}$$

7: Update π_ψ via $\max_\psi \mathcal{M}(\psi)$:

$$\mathcal{M}(\psi) = \mathbb{E}_{s,a\sim\mathcal{D}}[e^{(Q_\phi(s,a) - V_\theta(s))/\beta} \log \pi_\psi(s|a)]. \tag{142}$$

8: **end for**

The above results demonstrate that `ReCOIL` is able to leverage suboptimal data with an approximate in-distribution policy improvement and results in a policy close to the best policy with visitation almost in-support of the dataset.

E Implementation and Experiment Details

Rewriting of `dual-V` using temperature parameter λ instead of α : An implementation trick that we found particularly useful in reducing the number of hyperparameters to tune in order to obtain strong learning performance was replace the temperature parameter from α to λ. Notice that our initial `dual-V` formulation used the temperature parameter α as follows:

$$\texttt{dual-V} \quad \min_V (1-\gamma)\mathbb{E}_{s\sim d_0}[V(s)] + \alpha\mathbb{E}_{(s,a)\sim d^O}\left[f_p^*\left([\mathcal{T}V(s,a) - V(s))\right]/\alpha\right)\right], \tag{138}$$

The temperature parameter α captures the tradeoff between the first term which seeks to minimize V vs the second term which seeks to maximize V and set it to the maximum value possible when taking various actions from that state onwards. Depending on different f generator functions we would require tuning this parameter as it has a non-linear dependence on the entire optimization problem through the function f. Instead we consider a simpler objective, that we observe to empirically reduce hyperparameter tuning significantly by trading off linear between the first term and the second term using parameter λ. This modification is used in all of our experiments for RL and IL.

$$\texttt{dual-V (rewritten)} \quad \min_V (1-\lambda)\mathbb{E}_{s\sim d_0}[V(s)] + \lambda\mathbb{E}_{(s,a)\sim d^O}\left[f_p^*\left([\mathcal{T}V(s,a) - V(s))]\right)\right], \tag{139}$$

E.1 Offline IL: ReCOIL algorithm and implementation details

We give algorithms for two versions of ReCOIL: An idealized version and a practical version (Algorithm 1 and Algorithm 2 respectively). The practical version incorporates tricks like regressing to fixed targets for expert Q-values and learning bounded reward functions (corresponding to χ^2 divergence), that greatly increase training stability for the method inspired by [68, 3]. We base the ReCOIL implementation on the official implementation of XQL [23] and IQL [41]. Our network architecture mimics theirs and uses the same data preprocessing techniques.

In our set of environments, we keep the same hyper-parameters (except λ) across tasks - locomotion, adroit manipulation, and kitchen manipulation. For each environment, the values of λ are searched

Algorithm 2: ReCOIL-V Practical Algorithm (Under Stochastic Dynamics)

Initialize Q_ϕ,, $\bar{Q}_\phi$(target Q-function) V_θ, and π_ψ, $R_{\max} = k, R_{\min} = -k$

Let $\mathcal{D}^{\mathcal{S}} = (s, a, s')$ be data (possibly suboptimal) from the replay buffer (online or offline)

Let $\mathcal{D}^{\mathcal{E}} = (s, a, s')$ be expert data transitions. Let $\mathcal{D}$ be a sampling distribution s.t $s, a \sim \mathcal{D} = \{s, a \sim \mathcal{D}^{\mathcal{S}}$ w.p $1 - \beta, s, a \sim \mathcal{D}^{\mathcal{E}}$ w.p $\beta\}$

for $t = 1..T$ iterations **do**

Train Q_ϕ using $\min_\phi \mathcal{L}(\phi)$:

$$\mathcal{L}(\phi) = \mathbb{E}_{s,a,s' \sim \mathcal{D}^{\mathcal{S}}}\Big[(Q_\phi(s,a) - (R_{\min}(s,a) + V_\theta(s')))^2\Big] + \mathbb{E}_{s,a\sim\mathcal{D}^{\mathcal{E}}}\Big[Q_\phi(s,a) - \tfrac{R_{\max}}{1-\gamma}\Big]^2$$

Train V_θ using $\min_\theta \mathcal{J}(\theta)$:

$$\mathcal{J}(\theta) = \begin{cases} (1-\lambda)\mathbb{E}_{s,a\sim D}[V_\theta(s)] + \lambda\mathbb{E}_{s,a\sim D}\big[\max(\bar{Q}_\phi(s,a) - V_\theta(s), 0)\big] & \text{TV} \\ (1-\lambda)\mathbb{E}_{s,a\sim D}[V_\theta(s)] + \lambda\mathbb{E}_{s,a\sim D}\big[\max((\bar{Q}_\phi(s,a) - V_\theta(s)) + 0.5(\bar{Q}_\phi(s,a) - V_\theta(s))^2, 0)\big] & \chi^2 \\ (1-\lambda)\mathbb{E}_{s,a\sim D}[V_\theta(s)] + \lambda\mathbb{E}_{s,a\sim D}\big[\exp(\big(\big[\bar{Q}_\phi(s,a) - V_\theta(s))\big] - 1\big)\big] & \text{RKL/XQL} \end{cases}$$

Update π_ψ via $\max_\psi \mathcal{M}(\psi)$:

$$\mathcal{M}(\psi) = \mathbb{E}_{s,a\sim\mathcal{D}}[e^{(Q_\phi(s,a) - V_\theta(s))/\beta} \log \pi_\psi(s|a)]. \tag{143}$$

end for

between [2.5,5,10]. We keep a constant batch size of 256 across all environments. For all tasks we average mean returns over 10 evaluation trajectories and 7 random seeds. We add Layer Normalization [46] to the value networks for all environments. Full hyper-parameters we used for experiments are given in Table 5. We found the RKL update for training the V function to be the most performant for the imitation setting.

Hyperparameters for our proposed off-policy imitation learning method `ReCOIL` are shown in Table 5.

Hyperparameter	**Value**
Policy learning rate	3e-4
Value learning rate	3e-4
f-divergence	RKL/ (χ^2,TV)
max-clip (loss clipping)	7 (for RKL)
MLP layers	(256,256)
LR decay schedule	cosine

Table 5: Hyperparameters for `ReCOIL`.

E.2 Offline Imitation Learning Experiments

Environments: For the offline imitation learning experiments we focus on 10 locomotion and manipulation environments from the MuJoCo physics engine [75]. These environments include Hopper, Walker2d, HalfCheetah, Ant, Kitchen, Pen, Door, Hammer, and Relocate. The MuJoCo environments used in this work are licensed under CC BY 4.0 and the datasets used from D4RL are also licensed under Apache 2.0.

Suboptimal Datasets: For the offline imitation learning task, we utilize offline datasets consisting of environment interactions from the D4RL framework [18]. Specifically, we construct suboptimal datasets following the composition approach introduced in SMODICE [48]. The suboptimal datasets, denoted as 'random+expert', 'random+few-expert', 'medium+expert', and 'medium+few-expert' combine expert trajectories with low-quality trajectories obtained from the "random-v2" and "medium-v2" datasets, respectively. For locomotion tasks, the 'x+expert' dataset (where x is 'random' or 'medium') contains a mixture of some number of expert trajectories ($\leqslant 200$) and $\approx$1 million transitions from the "x" dataset. The 'x+few-expert' dataset is similar to 'x+expert,' but with only 30 expert trajectories included. For manipulation environments we consider only 30 expert trajectories mixed with the complete 'x' dataset of transitions obtained from D4RL.

Algorithm 3: f-DVL (Under Stochastic Dynamics)

Initialize Q_ϕ, $\bar{Q}_\phi$(target Q-function), V_θ, and π_ψ

Let $\mathcal{D} = (s, a, r, s')$ be data from $\pi_\mathcal{D}$ (offline) or replay buffer (online)

for $t = 1..T$ iterations **do**

  Train Q_ϕ using $\min_\phi \mathcal{L}(\phi)$:

$$\mathcal{L}(\phi) = \mathbb{E}_{s,a,s' \sim d^S}\left[(Q_\phi(s,a) - (r(s,a) + V(s')))^2\right]$$

  Train V_θ using $\min_\theta \mathcal{J}(\theta)$

$$\mathcal{J}(\theta) = \begin{cases} (1-\lambda)\mathbb{E}_{s,a\sim D}[V_\theta(s)] + \lambda\mathbb{E}_{s,a\sim D}\left[\max(\bar{Q}_\phi(s,a) - V_\theta(s), 0)\right] & \text{TV} \\ (1-\lambda)\mathbb{E}_{s,a\sim D}[V_\theta(s)] + \lambda\mathbb{E}_{s,a\sim D}\left[\max((\bar{Q}_\phi(s,a) - V_\theta(s)) + 0.5(\bar{Q}_\phi(s,a) - V_\theta(s))^2, 0)\right] & \chi^2 \\ (1-\lambda)\mathbb{E}_{s,a\sim D}[V_\theta(s)] + \lambda\mathbb{E}_{s,a\sim D}\left[\exp(([\bar{Q}_\phi(s,a) - V_\theta(s))] - 1)\right] & \text{RKL/XQL} \end{cases}$$

  Update π_ψ via $\max_\psi \mathcal{M}(\psi)$:

$$\mathcal{M}(\psi) = \mathbb{E}_{s,a\sim\mathcal{D}}[e^{(Q_\phi(s,a) - V_\theta(s))/\beta} \log \pi_\psi(s|a)]. \tag{144}$$

end for

Expert Dataset: To enable imitation learning, an offline expert dataset is required. In this work, we use 1 expert trajectory obtained from the "expert-v2" dataset for each respective environment.

Baselines: To benchmark and analyze the performance of our proposed methods for offline imitation learning with suboptimal data, we consider four representative baselines in this work: SMODICE [48], RCE [14], ORIL [84], and IQLearn [22]. We exclude DEMODICE [38] from the comparison, as SMODICE has been shown to be competitive [48]. SMODICE is an imitation learning method based on the dual framework, assuming a restrictive coverage. ORIL adapts the generative adversarial imitation learning (GAIL) [31] algorithm to the offline setting, employing an offline RL algorithm for policy optimization. The RCE baseline combines RCE, an online example-based RL method proposed by Eysenbach et al. [14]. RCE also uses a recursive discriminator to test the proximity of the policy visitations to successful examples. [14], with TD3-BC [19]. Both ORIL and RCE utilize a state-action based discriminator similar to SMODICE, and TD3-BC serves as the offline RL algorithm. All the compared approaches only have access to the expert state-action trajectory.

The open-source implementations of the baselines SMODICE, RCE, and ORIL provided by the authors [48] are employed in our experiments. We use the hyperparameters provided by the authors, which are consistent with those used in the original SMODICE paper [48], for all the MuJoCo locomotion and manipulation environments.

E.3 Online and Offline RL: f-DVL Algorithm and implementation details

Offline RL: Algorithm E.3 gives the algorithm for f-DVL. This section provides additional offline RL experiences along with complete hyper-parameter and implementation details. Figure 13 shows learning curves for all the environments. f-DVL exhibits as fast convergence as XQL but avoids the numerical instability of XQL with one hyperparameter across each set of environments. We base our implementation of f-DVL off the official implementation of XQL [23] and IQL from Kostrikov et al. [41]. Our network architecture mimics theirs and uses the same data preprocessing techniques.

In our set of environments, we keep the same hyper-parameter across sets of tasks - locomotion, adroit manipulation, kitchen-manipulation, and antmaze. Contrary to XQL, we find no need to use tricks like gradient clipping to stabilize learning. For each set of environment, the values of λ were tuned via hyper-parameter sweeps over a fixed set of values $[0.65, 0.7, 0.75, 0.8, 0.9]$. We keep a constant batch size of 256 across all environments. For MuJoCo locomotion tasks we average mean returns over 10 evaluation trajectories and 7 random seeds. For the AntMaze tasks, we average over 1000 evaluation trajectories. We add Layer Normalization [46] to the value networks for all environments. Full hyper-parameters we used for experiments are given in Table 6.

Env	Lambda λ	Batch Size	v_updates
halfcheetah-medium-v2	0.7	256	1
hopper-medium-v2	0.7	256	1
walker2d-medium-v2	0.7	256	1
halfcheetah-medium-replay-v2	0.7	256	1
hopper-medium-replay-v2	0.7	256	1
walker2d-medium-replay-v2	0.7	256	1
halfcheetah-medium-expert-v2	0.7	256	1
hopper-medium-expert-v2	0.7	256	1
walker2d-medium-expert-v2	0.7	256	1
antmaze-umaze-v0	0.8	256	1
antmaze-umaze-diverse-v0	0.8	256	1
antmaze-medium-play-v0	0.8	256	1
antmaze-medium-diverse-v0	0.8	256	1
antmaze-large-play-v0	0.8	256	1
antmaze-large-diverse-v0	0.8	256	1
kitchen-complete-v0	0.8	256	1
kitchen-partial-v0	0.8	256	1
kitchen-mixed-v0	0.8	256	1
pen-human-v0	0.8	256	1
hammer-human-v0	0.8	256	1
door-human-v0	0.8	256	1
relocate-human-v0	0.8	256	1
pen-cloned-v0	0.8	256	1
hammer-cloned-v0	0.8	256	1
door-human-v0	0.8	256	1
relocate-human-v0	0.8	256	1

Table 6: Offline RL Hyperparameters used for f-DVL. Lambda λ is the value that controls the strength of the implicit maximizer. V-updates gives the number of value updates per Q updates.

E.4 Online RL Experiments

Online RL: We base the implementation of SAC off pytorch_sac and XQL [23]. Like in offline experiments, hyper-parameters were left as default except for λ, which we tuned between $[0.6, 0.7, 0.8]$ and found a single value to work best across all environments. This was in contrast to XQL's finding which required per environment different hyperparameter. Also, as opposed to XQL we required no clipping of the loss function. We test our method on 7 random seeds for each environment.

Hyperparameter	**Value**
Policy updates n_{pol}	1
Policy learning rate	3e-4
Value learning rate	3e-4
MLP layers	(256,256)
LR decay schedule	cosine

Table 7: Common hyperparameters for f-DVL.

Hyperparameter	**Value**
Batch Size	1024
Learning Rate	0.0001
Critic Freq	1
Actor Freq	1
Actor and Critic Arch	1024, 1024
Buffer Size	1,000,000
Actor Noise	Auto-tuned
Target Noise	–

Table 8: Hyperparameters for SAC.

Compute We ran all our experiments on a machine with AMD EPYC 7J13 64-Core Processor and NVIDIA A100 with a GPU memory consumption of <1000 MB per experiment. Our offline RL and IL experiments for locomotion tasks take 10-20 min and the online IL experiments took around 5-6 hours for 1 million timesteps.

F Additional Experimental Results

F.1 Why Dual-RL Methods are a Better Alternative to Traditional Off-Policy Algorithms

Our experimental evaluation aims to illustrate the benefits of the dual RL framework and analyze our proposed method for off-policy imitation learning. In the RL setting, we first present a case study on the failure of ADP-based methods like SAC [27] to make the most when bootstrapped with additional (helpful) data. This setting is what motivates the use of off-policy algorithms in the first place and is invaluable in domains like robotics [77, 57]. Our results validate the benefit of utilizing the dual RL framework for off-policy learning.

The limitations of classical off-policy algorithms: Our experiments with the popular off-policy method SAC [27] reveal its brittleness to off-policy data. At the beginning of training, each learning agent is provided with expert or human-demonstrated trajectories for completing the task. We add 1000 transitions from this dataset to the replay buffer for the off-policy algorithm to bootstrap from. SAC is able to leverage this helpful data and shows improved performance in Hopper-v2, where the action dimension is small. As the action dimension increases, the brittleness of SAC becomes more apparent (see SAC+off policy data and SACfD plots in Figure 7). We hypothesize that this failure in the online RL setting is primarily due to the training instabilities caused by TD-backups resulting in overestimation in regions where the agent's current policy does not visit. In Figure 8, we observe that overestimation indeed happens in environments with larger action dimensions and these overestimations take longer to get corrected and in the process destabilize the training.

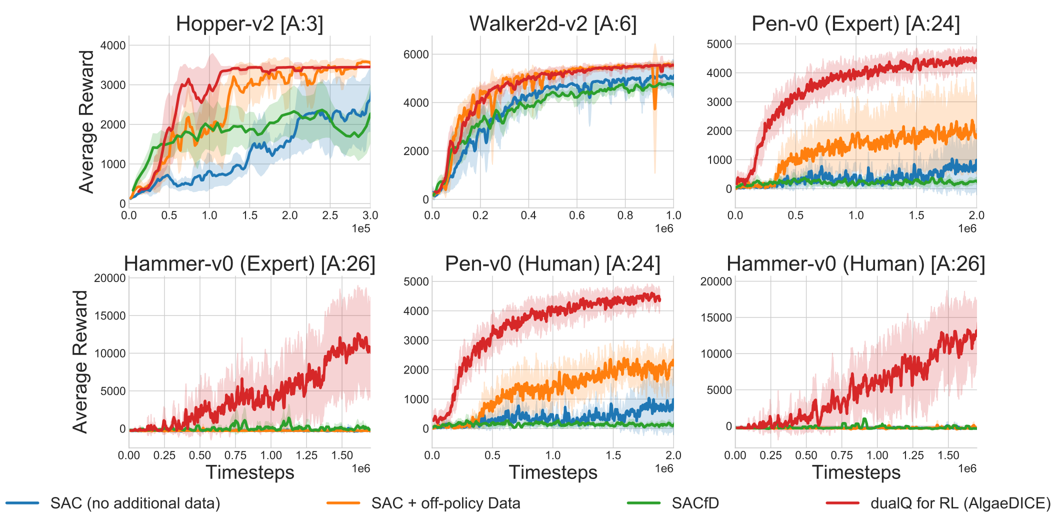

Figure 7: Despite the promise of off-policy methods, current methods based on ADP such as SAC fail when the dimension of action space, denoted by A, increases even when helpful data is added to their replay buffer. On other hand, dual-Q methods are able to leverage off-policy data to increase their learning performance

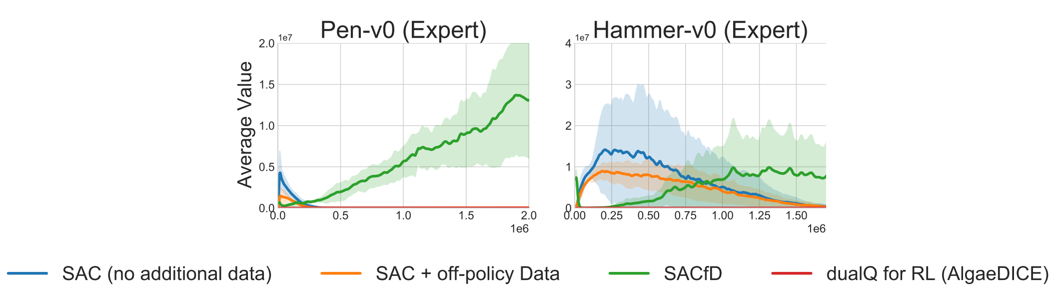

Figure 8: SAC and SACfD suffer from overestimation when off-policy data is added to the replay buffer. We hypothesize this to cause instabilities during training while dualQ has no overestimation.

Figure 7 shows that the dual-RL method (AlgaeDICE) is able to leverage off-policy data to increase learning performance without any signs of destabilization. This can be attributed to the distribution correction estimation property of dual RL methods which updates the current policy using the

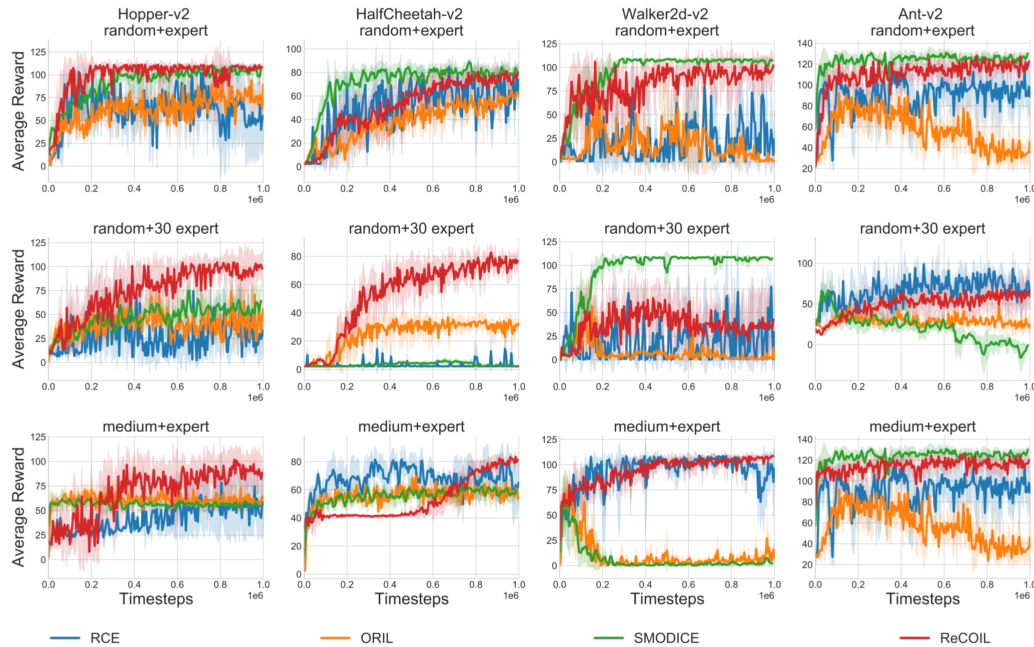

Figure 9: Learning curves for ReCOIL showing that it outperforms baselines in the setting of learning to imitate from diverse offline data. The results are averaged over 7 seeds

corrected on-policy policy visitation [56]. Note, that we set the temperature α to a low value (0.001) to disentangle the effect of pessimism which is an alternate way to avoid overestimation.

F.2 Training Curves for `ReCOIL` on MuJoCo tasks

We show learning curves for `ReCOIL` in Figure 9 for locomotion tasks and Figure 10 for manipulation tasks below. `ReCOIL` training curves are reasonably stable while also being performant, especially in the manipulation setting where other methods completely fail.

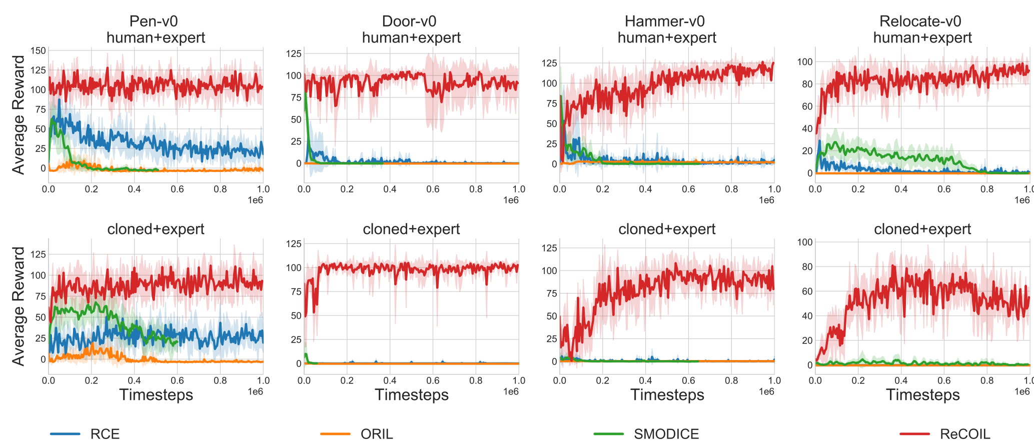

Figure 10: Learning curves for ReCOIL showing that it outperforms baselines in the setting of learning to imitate from diverse offline data. The results are averaged over 7 seeds

F.3 Does `ReCOIL` Allow for Better Estimation of Agent Visitation Distribution?

We consider an additional 2-D gridworld environment that demonstrate the failures of a method that either do not utilize all available suboptimal data (IQ-Learn) or relies on a coverage assumption (SMODICE). We saw that `ReCOIL` is able to perfectly infer the agent's visitation when the replay buffer covers agent ground truth visitation perfectly (Fig 2a) and here we see that `ReCOIL` is able to outperform baselines when the replay buffer has imperfect coverage over the agent's ground truth visitation (Fig 11). In this task, the agent starts at (0,0) which is the top-left corner. The agent can only move in cardinal directions with deterministic dynamics. The agent has access to two sources of off-policy data - expert visitation and replay visitation. The problem is to estimate the agent's visitation distribution given access to the agent's policy using all the available transition data. IQLearn and SMODICE predict an agent's visitation that wildly differs from Agent's ground truth visitation distribution. While ReCOIL is not perfect as the coverage of the offline data is limited, we can estimate some visitation which is qualitatively very similar to the agent's ground truth visitation.

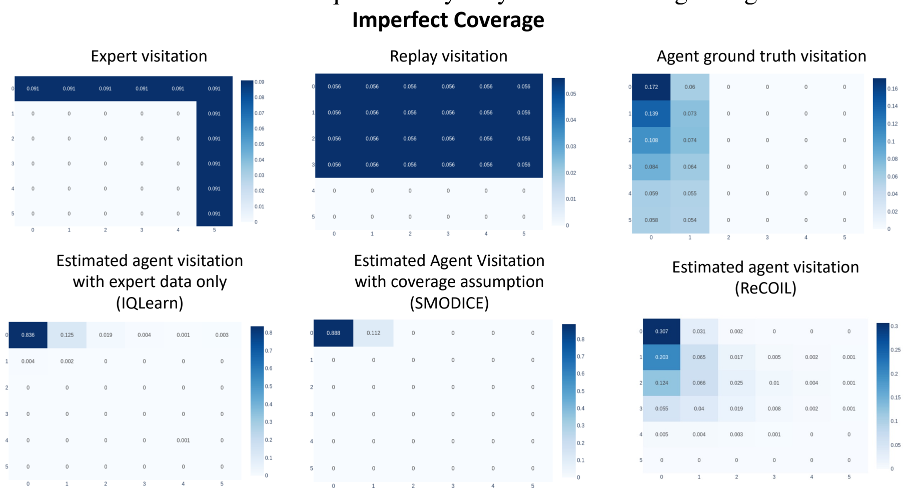

Figure 11: Replay buffer consists of data that visits near the initial state (0,0), a setting commonly observed when training RL agents. We estimate the agent's policy visitation and observe ReCOIL to outperform both methods which rely on expert data only or use the replay data with coverage assumption

F.4 ReCOIL: Qualitative Comparison with a Baseline

In Figure 12, we investigate qualitatively why other baselines fail where ReCOIL succeeds in high-dimensional tasks. A surprising finding is that the baseline we consider 'SMODICE' almost learns to imitate. It follows nearly the same actions as an expert but makes small mistakes along the way - eg. 'gripping the hammer too loose' or 'picking up the ball at a slightly wrong location'. SMODICE is unable to recover from such mistakes and ends up having low performance. ReCOIL, on the other hand, learns a performant task-solving policy from the same data.

F.5 Training Curves for f-DVL on MuJoCo Tasks (Offline)

Figure 13 shows the learning curves during training for f-DVL. f-DVL is able to leverage low-order conjugate f-divergences to give offline RL algorithms that more stable compared to XQL. XQL frequently crashes in the antmaze environment.

F.6 f-DVL: Complete Offline RL Results

Table 9 and Table 10 show complete results for benchmarking f-DVL on MuJoCo D4RL environments. Here we also show the author-reported results for XQL and the reproduced results (XQL(r)) using the metric of taking the average of the last iterate performance across seeds.

F.7 Sensitivity of f-DVL (offline) with varying λ on MuJoCo tasks

We ablate the temperature parameter, λ for offline RL experiments using f-DVL in Figure 15 and Figure 14. The temperature λ controls the strength of KL penalization between the learned policy and the dataset behavior policy, and a small λ is beneficial for datasets with lots of random noisy actions. In contrast, a high λ favors more expert-like datasets. We observe that significantly less hyperparameter tuning is required compared to XQL as a single temperature value works well across a broad range of experiments.

F.8 Sensitivity of f-DVL (online) with varying λ on MuJoCo tasks

We ablate the temperature parameter λ for online RL experiments using f-DVL in Figure 17 (chi-square) and Figure 16 (TV). We observe that significantly less hyperparameter tuning is required compared to XQL as a single temperature value works well across a broad range of experiments.

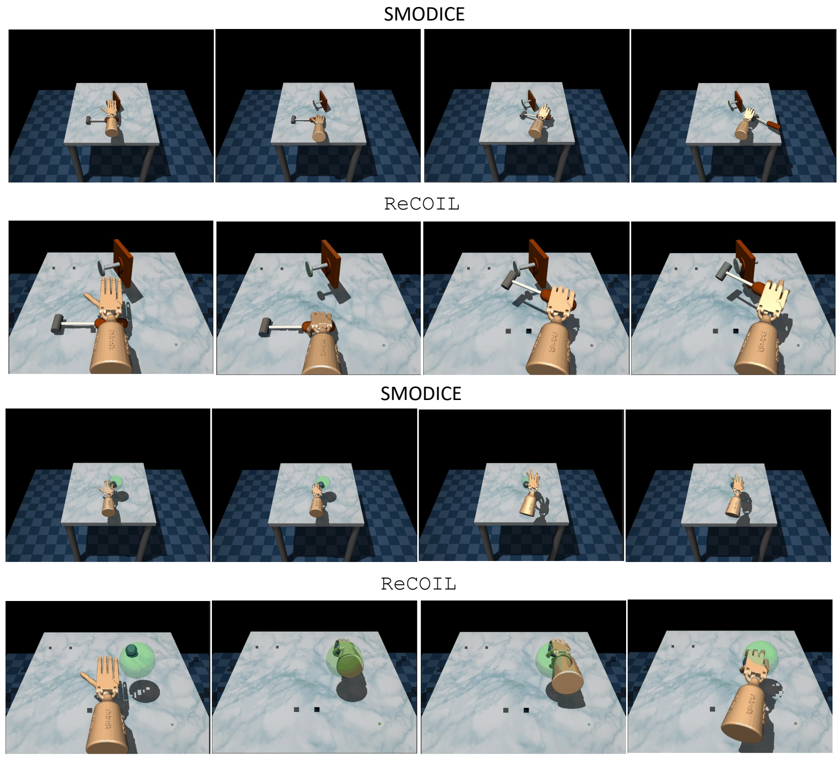

Figure 12: Errors compound in imitation learning and recovery is of crucial importance. Figure demonstrate how SMODICE 'almost' imitates, figures out roughly what actions to take but does not realise once it has made a mistake. In Hammer environment, it grips the hammer too loose causing it to get thrown away and for relocate picks up just beside the ball missing the original task the expert intended to solve.

Table 9: Averaged normalized scores on MuJoCo locomotion and Ant Maze tasks. XQL(r) denotes the reproduced results with author's implementation.

	Dataset	BC	10%BC	DT	TD3+BC	CQL	IQL	XQL	XQL(r)	f-DVL χ^2	f-DVL TV
Gym	halfcheetah-medium-v2	42.6	42.5	42.6	**48.3**	44.0	**47.4**	47.7	47.4	**47.7**	47.5
	hopper-medium-v2	52.9	56.9	67.6	59.3	58.5	66.3	71.1	**68.5**	63.0	64.1
	walker2d-medium-v2	75.3	75.0	74.0	**83.7**	72.5	78.3	81.5	81.4	80.0	**81.5**
	halfcheetah-medium-replay-v2	36.6	40.6	36.6	**44.6**	**45.5**	**44.2**	44.8	44.1	42.9	**44.7**
	hopper-medium-replay-v2	18.1	75.9	82.7	60.9	95.0	94.7	97.3	95.1	90.7	**98.0**
	walker2d-medium-replay-v2	26.0	62.5	66.6	**81.8**	77.2	73.9	75.9	58.0	52.1	68.7
	halfcheetah-medium-expert-v2	55.2	92.9	86.8	90.7	**91.6**	86.7	89.8	90.8	89.3	**91.2**
	hopper-medium-expert-v2	52.5	110.9	**107.6**	98.0	105.4	91.5	107.1	94.0	**105.8**	93.3
	walker2d-medium-expert-v2	107.5	**109.0**	108.1	**110.1**	108.8	**109.6**	110.1	**110.1**	**110.1**	109.6
AntMaze	antmaze-umaze-v0	54.6	62.8	59.2	78.6	74.0	**87.5**	87.2	47.7	**83.7**	**87.7**
	antmaze-umaze-diverse-v0	45.6	50.2	53.0	71.4	**84.0**	62.2	69.17	51.7	50.4	48.4
	antmaze-medium-play-v0	0.0	5.4	0.0	10.6	61.2	**71.2**	73.5	31.2	56.7	**71.0**
	antmaze-medium-diverse-v0	0.0	9.8	0.0	3.0	53.7	**70.0**	67.8	0.0	48.2	60.2
	antmaze-large-play-v0	0.0	0.0	0.0	0.2	15.8	39.6	41	10.7	36.0	**41.7**
	antmaze-large-diverse-v0	0.0	6.0	0.0	0.0	14.9	**47.5**	47.3	31.28	**44.5**	39.3
Franka	kitchen-complete-v0	65.0	-	-	-	43.8	62.5	72.5	56.7	**67.5**	61.3
	kitchen-partial-v0	38.0	-	-	-	49.8	46.3	73.8	48.6	58.8	**70.0**
	kitchen-mixed-v0	51.5	-	-	-	51.0	51.0	54.6	40.4	**53.75**	52.5

F.9 Recovering Reward functions from `ReCOIL`

We study the quality of reward functions recovered from `ReCOIL` using the hopper-medium-expert and Walker2d-medium-expert datasets and the setup described in Section 6.1. For all trajectories in this dataset, we calculate the ground truth return (sum of rewards) and the predicted cumulative reward using `ReCOIL`. The scatter plot in figure 18 shows the correlation between predicted rewards. We note that `ReCOIL` is an IRL method and suffers from the reward ambiguity problems as rest of the IRL methods— we can only expect a reward function that induces an optimal policy whose visitation is close to an expert and cannot guarantee that we recover the expert's exact reward function. To test the quality of rewards functions output by IRL methods, Pearson correlation is not the accurate metric and metrics like EPIC [25] might be used instead.

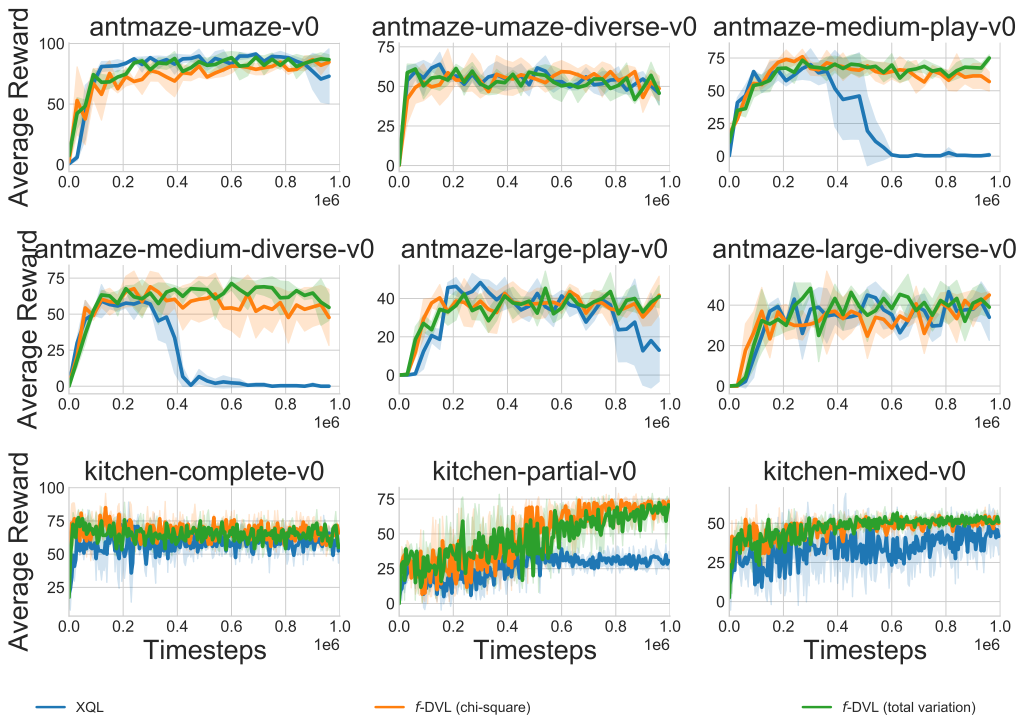

Figure 13: Learning curves for f-DVL showing that it is able to leverage low-order conjugate f-divergences to give offline RL algorithms that more stable compared to XQL. The results are averaged over 7 seeds

Table 10: Evaluation on Adroit tasks from D4RL.XQL-C (r) denotes the reproduced results with author's implementation.

Dataset	BC	BRAC-p	BEAR	Onestep RL	CQL	IQL	XQL	XQL(r)	f-DVL (χ^2)	f-DVL (TV)
pen-human-v0	63.9	8.1	-1.0	-	37.5	71.5	85.5	63.5	67.1	64.1
hammer-human-v0	1.2	0.3	0.3	-	4.4	1.4	2.2	1.4	2.6	1.8
door-human-v0	2	-0.3	-0.3	-	9.9	4.3	11.5	6.63	5.7	6.77
relocate-human-v0	0.1	-0.3	-0.3	-	**0.2**	0.1	0.17	0.2	**0.37**	0.12
pen-cloned-v0	37	1.6	26.5	**60.0**	39.2	37.3	38.6	25.25	36.1	38.1
hammer-cloned-v0	0.6	0.3	0.3	2.1	2.1	2.1	4.3	1.58	1.64	1.65
door-cloned-v0	0.0	-0.1	-0.1	0.4	0.4	1.6	5.9	0.69	0.45	0.87
relocate-cloned-v0	-0.3	-0.3	-0.3	-0.1	-0.1	-0.2	-0.2	-0.24	-0.24	-0.24

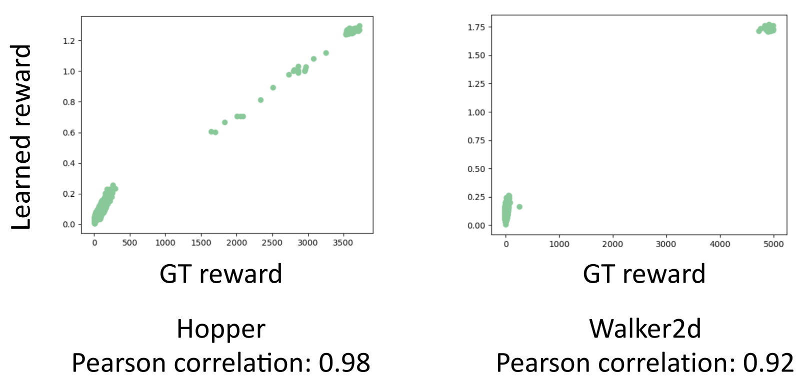

Figure 18: Correlation of the rewards inferred by `ReCOIL` with respect to the ground truth reward function of the expert.

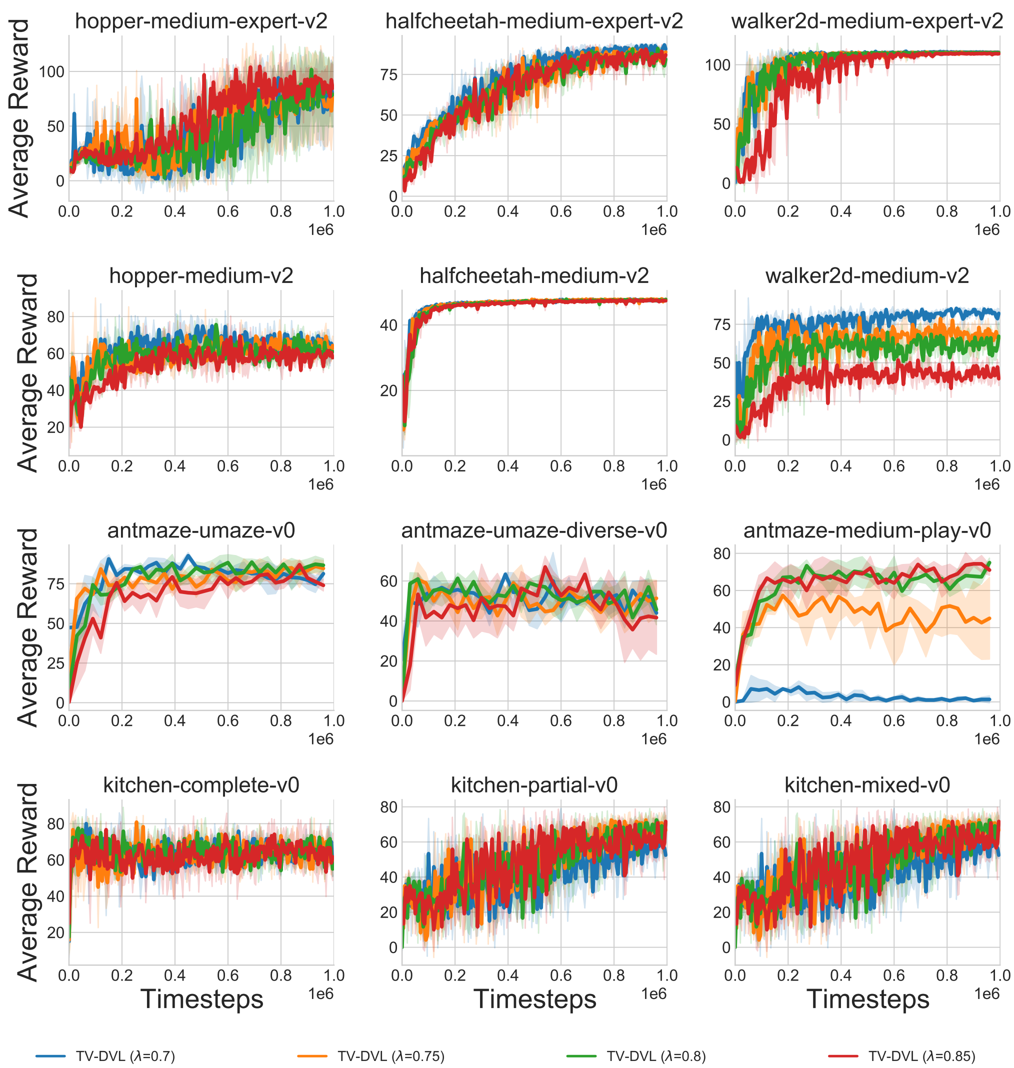

Figure 14: Offline RL: Ablating the temperature parameter for f-DVL (Total variation). The plot shows the effect of temperature parameters on learning performance.

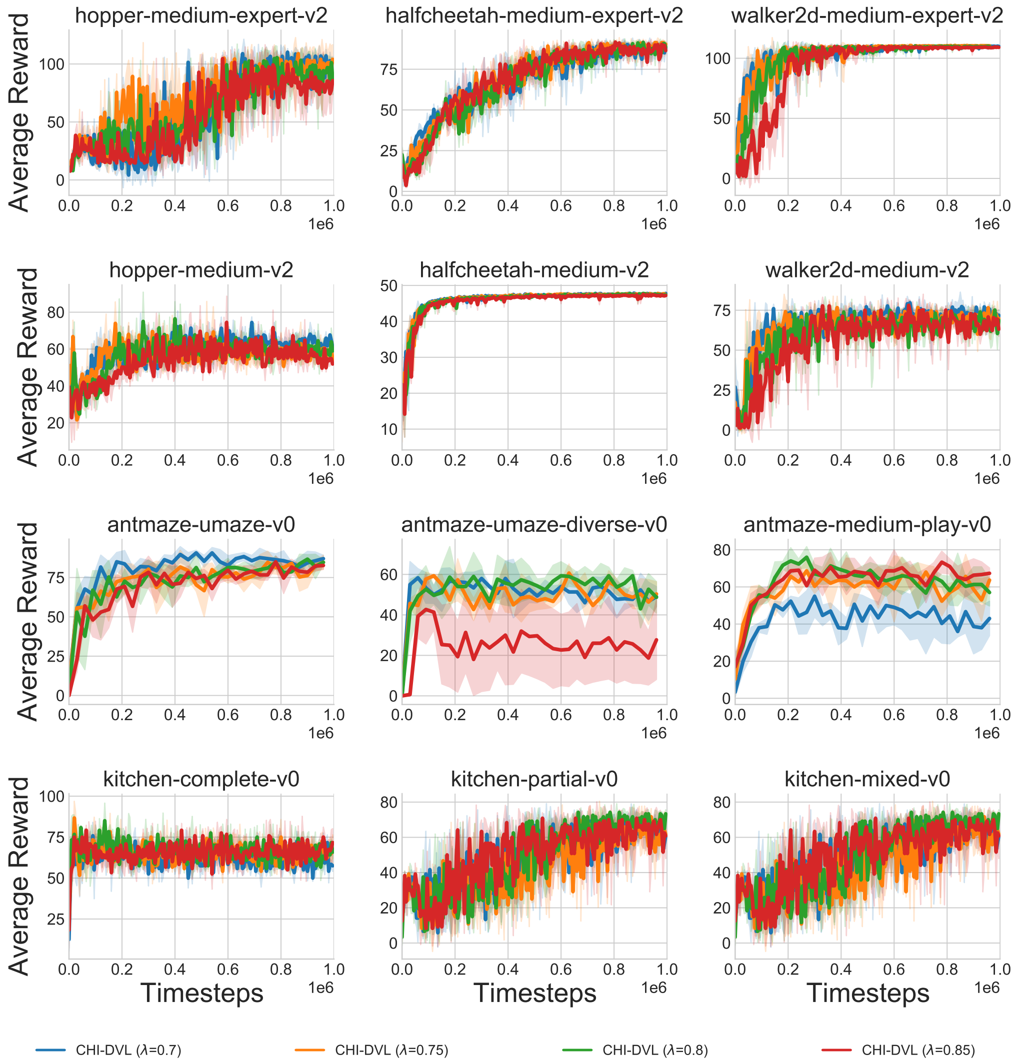

Figure 15: Offline RL: Ablating the temperature parameter for f-DVL (Chi-square). The plot shows the effect of temperature parameters on learning performance.

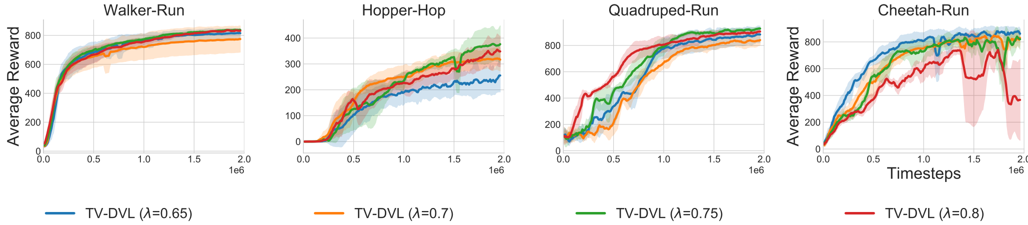

Figure 16: Online RL: Ablating the temperature parameter for f-DVL (Total variation). The plot shows the effect of temperature parameters on learning performance.

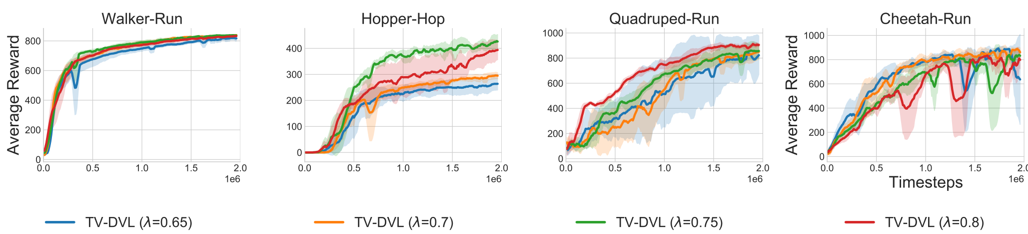

Figure 17: Online RL: Ablating the temperature parameter for f-DVL (Chi-square). The plot shows the effect of temperature parameters on learning performance.