Batch Policy Learning under Constraints

Hoang M. Le [1] **Cameron Voloshin** [1] **Yisong Yue** [1]

Abstract

When learning policies for real-world domains, two important questions arise: (i) how to efficiently use pre-collected off-policy, non-optimal behavior data; and (ii) how to mediate among different competing objectives and constraints. We thus study the problem of batch policy learning under multiple constraints, and offer a systematic solution. We first propose a flexible meta-algorithm that admits any batch reinforcement learning and online learning procedure as subroutines. We then present a specific algorithmic instantiation and provide performance guarantees for the main objective and all constraints. As part of off-policy learning, we propose a simple method for off-policy policy evaluation (OPE) and derive PAC-style bounds. Our algorithm achieves strong empirical results in different domains, including in a challenging problem of simulated car driving subject to multiple constraints such as lane keeping and smooth driving. We also show experimentally that our OPE method outperforms other popular OPE techniques on a standalone basis, especially in a high-dimensional setting.

1. Introduction

We study the problem of policy learning under multiple constraints. Contemporary approaches to learning sequential decision making policies have largely focused on optimizing some cost objective that is easily reducible to a scalar value function. However, in many real-world domains, choosing the right cost function to optimize is often not a straightforward task. Frequently, the agent designer faces multiple competing objectives. For instance, consider the aspirational task of designing autonomous vehicle controllers: one may care about minimizing the travel time while making sure the driving behavior is safe, consistent, or fuel efficient. Indeed, many such real-world applications require the primary objective function be augmented with an appropriate set of constraints (Altman, 1999).

Contemporary policy learning research has largely focused on either online reinforcement learning (RL) with a focus on exploration, or imitation learning (IL) with a focus on learning from expert demonstrations. However, many real-world settings already contain large amounts of pre-collected data generated by existing policies (e.g., existing driving behavior, power grid control policies, etc.). We thus study the complementary question: *can we leverage this abundant source of (non-optimal) behavior data in order to learn sequential decision making policies with provable guarantees on both primary objective and constraint satisfaction*?

We thus propose and study the problem of batch policy learning under multiple constraints. Historically, batch RL is regarded as a subfield of approximate dynamic programming (ADP) (Lange et al., 2012), where a set of transitions sampled from the existing system is given and fixed. From an interaction perspective, one can view many online RL methods (e.g., DDPG (Lillicrap et al., 2016)) as running a growing batch RL subroutine per round of online RL. In that sense, batch policy learning is complementary to any exploration scheme. To the best of our knowledge, the study of constrained policy learning in the batch setting is novel.

We present an algorithmic framework for learning sequential decision making policies from off-policy data. We employ multiple learning reductions to online and supervised learning, and present an analysis that relates performance in the reduced procedures to the overall performance with respect to both the primary objective and constraint satisfaction.

Constrained optimization is a well studied problem in supervised machine learning and optimization. In fact, our approach is inspired by the work of Agarwal et al. (2018) in the context of fair classification. In contrast to supervised learning for classification, batch policy learning for sequential decision making introduces multiple additional challenges. First, setting aside the constraints, batch policy learning itself presents a layer of difficulty, and the analysis is significantly more complicated. Second, verifying whether the constraints are satisfied is no longer as straightforward as passing the training data through the learned classifier. In sequential decision making, certifying constraint satisfac-

[1]California Institute of Technology, Pasadena, CA. Correspondence to: Hoang M. Le <hmle@caltech.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

tion amounts to an off-policy policy evaluation problem, which is a challenging problem and the subject of active research. In this paper, we develop a systematic approach to address these challenges, provide a careful error analysis, and experimentally validate our proposed algorithms. In summary, our contributions are:

- We formulate the problem of batch policy learning under multiple constraints, and present the first approach of its kind to solve this problem. The definition of constraints is general and can subsume many objectives. Our approach utilizes multi-level learning reductions, and we show how to instantiate it using various batch RL and online learning subroutines. We show that guarantees from the subroutines provably lift to provide end-to-end guarantees on the original constrained batch policy learning problem.
- While leveraging techniques from batch RL as a subroutine, we provide a refined theoretical analysis for general non-linear function approximation that improves upon the previously known sample complexity result (Munos & Szepesvári, 2008).
- To evaluate off-policy learning performance and constraint satisfaction, we propose a simple new technique for off-policy policy evaluation (OPE), which is used as a subroutine in our main algorithm. We show that it is competitive to other OPE methods.
- We validate our algorithm and analysis with two experimental settings. First, a simple navigation domain where we consider safety constraint. Second, we consider a high-dimensional racing car domain with smooth driving and lane centering constraints.

2. Problem Formulation

We first introduce notation. Let $\mathrm{X} \subset \mathbb{R}^d$ be a bounded and closed d-dimensional state space. Let A be a finite action space. Let $c : \mathrm{X} \times \mathrm{A} \mapsto [0, \overline{C}]$ be the primary objective cost function that is bounded by $\overline{C}$. Let there be m constraint cost functions, $g_i : \mathrm{X} \times \mathrm{A} \mapsto [0, \overline{G}]$, each bounded by $\overline{G}$. To simplify the notation, we view the set of constraints as a vector function $g : \mathrm{X} \times \mathrm{A} \mapsto [0, \overline{G}]^m$ where $g(x, a)$ is the column vector of individual $g_i(x, a)$. Let $p(\cdot|x, a)$ denote the (unknown) transition/dynamics model that maps state/action pairs to a distribution over the next state. Let $\gamma \in (0, 1)$ denote the discount factor. Let χ be the initial states distribution.

We consider the discounted infinite horizon setting. An MDP is defined using the tuple $(\mathrm{X}, \mathrm{A}, c, g, p, \gamma, \chi)$. A policy $\pi \in \Pi$ maps states to actions, i.e., $\pi(x) \in \mathrm{A}$. The value function $C^\pi : \mathrm{X} \mapsto \mathbb{R}$ corresponding to the primary cost function c is defined in the usual way: $C^\pi(x) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t c(x_t, a_t) \mid x_0 = x\right]$, over the randomness of the policy π and transition dynamics p. We similarly define the vector-value function for the constraint costs $G^\pi : \mathrm{X} \mapsto \mathbb{R}^m$ as $G^\pi(x) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t g(x_t, a_t) | x_0 = x\right]$. Define $C(\pi)$ and $G(\pi)$ as the expectation of $C^\pi(x)$ and $G^\pi(x)$, respectively, over the distribution χ of initial states.

2.1. Batch Policy Learning under Constraints

In batch policy learning, we have a pre-collected dataset, $\mathrm{D} = \{(x_i, a_i, x_i', c(x_i, a_i), g_{1:m}(x_i, a_i)\}_{i=1}^n$, generated from (a set of) historical behavioral policies denoted jointly by π_D. The goal of batch policy learning under constraints is to learn a policy $\pi \in \Pi$ from D that minimizes the primary objective cost while satisfying m different constraints:

$$\begin{aligned} \min_{\pi \in \Pi} \quad & C(\pi) \\ \text{s.t.} \quad & G(\pi) \leq \tau \end{aligned} \tag{OPT}$$

where $G(\cdot) = [g_1(\cdot), \ldots, g_m(\cdot)]^\top$ and $\tau \in \mathbb{R}^m$ is a vector of known constants. We assume that (OPT) is feasible. However, the dataset D might be generated from multiple policies that violate the constraints.

2.2. Examples of Policy Learning with Constraints

Counterfactual & Safe Policy Learning. In conventional online RL, the agent needs to "re-learn" from scratch when the cost function is modified. Our framework enables counterfactual policy learning assuming the ability to compute the new cost objective from the same historical data. A simple example is *safe* policy learning (Garcıa & Fernández, 2015). Define safety cost $g(x, a) = \phi(x, a, c)$ as a new function of existing cost c and features associated with current state-action pair. The goal here is to counterfactually avoid undesirable behaviors observed from historical data. We experimentally study this safety problem in Section 5.

Other examples from the literature that belong to this safety perspective include planning under chance constraints (Ono et al., 2015; Blackmore et al., 2011). The constraint here is $G(\pi) = \mathbb{E}[\mathbb{I}(x \in \mathrm{X}_{error})] = \mathrm{P}(x \in \mathrm{X}_{error}) \leq \tau$.

Multi-objective Batch Learning. Traditional policy learning (RL or IL) presupposes that the agent optimizes a single cost function. In reality, we may want to satisfy multiple objectives that are not easily reducible to a scalar objective function. One example is learning fast driving policies under multiple behavioral constraints such as smooth driving and lane keeping consistency (see Section 5).

2.3. Equivalence between Constraint Satisfaction and Regularization

Our constrained policy learning framework accommodates several existing regularized policy learning settings. Regularization typically encodes prior knowledge, and has been used extensively in the RL and IL literature to improve

learning performance. Many instances of regularized policy learning can be naturally cast into (OPT):

- *Entropy regularized RL* (Haarnoja et al., 2017; Ziebart, 2010) maps to policy-dependent constraint cost $g(x) = \mathbb{H}\big(\pi(\cdot|x)\big)$, where $\mathbb{H}$ measures conditional entropy.[1]
- *Conservative policy improvement* (Levine & Abbeel, 2014; Schulman et al., 2015; Achiam et al., 2017) is equivalent to $G(\pi) = D_{KL}(\pi, \pi_k)$, where π_k is some "well-behaving" policy.
- *Smooth imitation learning* (Le et al., 2016) is equivalent to $G(\pi) = \min_{h\in \mathrm{H}} \Delta(h, \pi)$, where H is a class of provably smooth policies and Δ is a divergence metric.
- *Regularizing RL with expert demonstration* (Hester et al., 2018) is equivalent to $G(\pi) = \mathbb{E}[\ell(\pi(x), \pi^*(x))]$, where π^* is the expert policy.

We provide further equivalence derivation of the above examples in Appendix A. Of course, some problems are more naturally described using the regularization perspective, while others using constraint satisfaction.

More generally, one can establish the equivalence between regularized and constrained policy learning via a simple appeal to Lagrangian duality as shown in Proposition 2.1 below. This Lagrangian duality also has a game-theoretic interpretation (Section 5.4 of Boyd & Vandenberghe (2004)), which serves as an inspiration for developing our approach.

Proposition 2.1. *Let Π be a convex set of policies. Let $C : \Pi \mapsto \mathbb{R}, C : \Pi \mapsto \mathbb{R}^K$ be value functions. Consider the two policy optimization tasks:*

$$\texttt{Regularization:} \quad \min_{\pi\in\Pi} \; C(\pi) + \lambda^\top G(\pi)$$

$$\texttt{Constraint:} \quad \min_{\pi\in\Pi} \; C(\pi) \quad \textit{s.t. } G(\pi) \le \tau$$

Assume that the Slater's condition is satisfied in the `Constraint` *problem (i.e., $\exists \pi$ s.t. $G(\pi) < \tau$). Assume also that the constraint cannot be removed without changing the optimal solution, i.e., $\inf_{\pi\in\Pi} C(\pi) < \inf_{\pi\in\Pi: G(\pi)\le\tau} C(\pi)$. Then $\forall\ \lambda > 0$, $\exists\ \tau$, and vice versa, such that* `Regularization` *and* `Constraint` *share the same optimal solutions. (Proof in Appendix A.)*

3. Proposed Approach

To make use of strong duality, we first *convexify* the policy class Π by allowing stochastic combinations of policies, which effectively expands Π into its convex hull $\texttt{Conv}(\Pi)$. Formally, $\texttt{Conv}(\Pi)$ contains *randomized policies*, which we denote $\pi = \sum_{t=1}^T \alpha_t \pi_t$ for $\pi_t \in \Pi$ and $\sum_{t=1}^T \alpha_t = 1$. Executing a mixed π consists of first sampling *one* policy π_t from $\pi_{1:T}$ according to distribution $\alpha_{1:T}$, and then executing π_t. Note that we still have $\mathbb{E}[\pi] = \sum_{t=1}^T \alpha_t \mathbb{E}[\pi_t]$ for any first-moment statistic of interest (e.g., state distribution, expected cost). It is easy to see that the augmented version of (OPT) over $\texttt{Conv}(\Pi)$ has a solution at least as good as the original (OPT). As such, to lighten the notation, we will equate Π with its convex hull for the rest of the paper.

Algorithm 1 Meta-algo for Batch Constrained Learning

for each round t **do**
  $\pi_t \leftarrow \texttt{Best-response}(\lambda_t)$
  $\widehat{\pi}_t \leftarrow \frac{1}{t}\sum_{t'=1}^t \pi_{t'}$, $\widehat{\lambda}_t \leftarrow \frac{1}{t}\sum_{t'=1}^t \lambda_{t'}$
  $\mathrm{L}_{\max} = \max_\lambda L(\widehat{\pi}_t, \lambda)$
  $\mathrm{L}_{\min} = L(\texttt{Best-response}(\widehat{\lambda}_t), \widehat{\lambda}_t)$
  if $\mathrm{L}_{\max} - \mathrm{L}_{\min} \le \omega$ **then**
    Return $\widehat{\pi}_t$
  end if
  $\lambda_{t+1} \leftarrow \texttt{Online-algorithm}(\pi_1, \ldots, \pi_{t-1}, \pi_t)$
end for

3.1. Meta-Algorithm

The Lagrangian of (OPT) is $L(\pi, \lambda) = C(\pi) + \lambda^\top (G(\pi) - \tau)$ for $\lambda \in \mathbb{R}^m_+$. Clearly (OPT) is equivalent to the min-max problem: $\min_{\pi\in\Pi} \max_{\lambda\in\mathbb{R}^k_+} L(\pi, \lambda)$. We assume (OPT) is feasible and that Slater's condition holds (otherwise, we can simply increase the constraint τ by a tiny amount). Slater's condition and policy class convexification ensure that strong duality holds (Boyd & Vandenberghe, 2004), and (OPT) is also equivalent to the max-min problem: $\max_{\lambda\in\mathbb{R}^k_+} \min_{\pi\in\Pi} L(\pi, \lambda)$.

Since $L(\pi, \lambda)$ is linear in both λ and π (due to stochastic mixture[2]) , strong duality is also a consequence of von Neumann's celebrated convex-concave minimax theorem for zero-sum games (Von Neumann & Morgenstern, 2007). From a game-thoeretic perspective, the problem becomes finding the equilibrium of a two-player game between the $\pi-$player and the $\lambda-$player (Freund & Schapire, 1999). In this repeated game, the $\pi-$player minimizes $L(\pi, \lambda)$ given the current λ, and the $\lambda-$player maximizes it given the current (mixture over) π.

We first present a meta-algorithm (Algorithm 1) that uses any no-regret online learning algorithm (for λ) and batch policy optimization (for π). At each iteration, given λ_t, the π-player runs `Best-response` to get the best response:

$$\begin{aligned}\texttt{Best-response}(\lambda_t) &= \arg\min_{\pi\in\Pi} L(\pi, \lambda_t) \\ &= \arg\min_{\pi\in\Pi} C(\pi) + \lambda_t^\top (G(\pi) - \tau).\end{aligned}$$

This is equivalent to a standard batch reinforcement learning problem where we learn a policy that is optimal with respect to $c + \lambda_t^\top g$. The corresponding mixed strategy is the uniform distribution over all previous π_t. In response to the

[1] Constraint value function $G(\pi)$ can be viewed as the expectation over discounted state visitation distribution. The lack of explicit discount rate does not intefere with our overall approach.

[2] This places no restrictions on the individual policies. Individual policy can be non-linear and cost function can be non-convex.

Algorithm 2 Constrained Batch Policy Learning

Input: Dataset $\mathrm{D} = \{x_i, a_i, x'_i, c_i, g_i\}_{i=1}^n \sim \pi_{\mathrm{D}}$. Online algorithm parameters: ℓ_1 norm bound B, learning rate η

1: Initialize $\lambda_1 = (\frac{B}{m+1}, \ldots, \frac{B}{m+1}) \in \mathbb{R}^{m+1}$
2: **for** each round t **do**
3: Learn $\pi_t \leftarrow \texttt{FQI}(c + \lambda_t^\top g)$ *// FQI with cost $c + \lambda_t^\top g$*
4: Evaluate $\widehat{C}(\pi_t) \leftarrow \texttt{FQE}(\pi_t, c)$ *// Algo 3 with π_t, cost c*
5: Evaluate $\widehat{G}(\pi_t) \leftarrow \texttt{FQE}(\pi_t, g)$ *// Algo 3 with π_t, cost g*
6: $\widehat{\pi}_t \leftarrow \frac{1}{t}\sum_{t'=1}^t \pi_{t'}$
7: $\widehat{C}(\widehat{\pi}_t) \leftarrow \frac{1}{t}\sum_{t'=1}^t \widehat{C}(\pi_{t'}), \widehat{G}(\widehat{\pi}_t) \leftarrow \frac{1}{t}\sum_{t'=1}^t \widehat{G}(\pi_{t'})$
8: $\widehat{\lambda}_t \leftarrow \frac{1}{t}\sum_{t'=1}^t \lambda_{t'}$
9: Learn $\widetilde{\pi} \leftarrow \texttt{FQI}(c + \widehat{\lambda}_t^\top g)$ *// FQI with cost $c + \widehat{\lambda}_t^\top g$*
10: Evaluate $\widehat{C}(\widetilde{\pi}) \leftarrow \texttt{FQE}(\widetilde{\pi}, c), \widehat{G}(\widetilde{\pi}) \leftarrow \texttt{FQE}(\widetilde{\pi}, g)$
11: $\widehat{\mathrm{L}}_{\max} = \max_{\lambda, \|\lambda\|_1 = B} \left(\widehat{C}(\widehat{\pi}_t) + \lambda^\top \left[(\widehat{G}(\widehat{\pi}_t) - \tau)^\top, 0 \right]^\top \right)$
12: $\widehat{\mathrm{L}}_{\min} = \widehat{C}(\widetilde{\pi}) + \widehat{\lambda}_t^\top \left[(\widehat{G}(\widetilde{\pi}) - \tau)^\top, 0 \right]^\top$
13: **if** $\widehat{\mathrm{L}}_{\max} - \widehat{\mathrm{L}}_{\min} \leq \omega$ **then**
14: Return $\widehat{\pi}_t$
15: **end if**
16: Set $z_t = \left[(\widehat{G}(\pi_t) - \tau)^\top, 0 \right]^\top \in \mathbb{R}^{m+1}$
17: $\lambda_{t+1}[i] = B \frac{\lambda_t[i] e^{-\eta z_t[i]}}{\sum_j \lambda_t[j] e^{-\eta z_t[j]}} \forall i$ *// $\lambda[i]$ the i^{th} coordinate*
18: **end for**

$\pi-$player, the $\lambda-$player employs `Online-algorithm`, which can be *any* no-regret algorithm that satisfies:

$$\sum_t L(\pi_t, \lambda_t) \geq \max_\lambda \sum_t L(\pi_t, \lambda) - o(T)$$

Finally, the algorithm terminates when the estimated primal-dual gap is below a threshold ω (Lines 7-8).

Leaving aside (for the moment) issues of generalization, Algorithm 1 is guaranteed to converge assuming: (i) `Best-response` gives the best single policy in the class, and (ii) $\mathrm{L}_{\max}$ and $\mathrm{L}_{\min}$ can be evaluated exactly.

Proposition 3.1. *Assuming (i) and (ii) above, Algorithm 1 is guaranteed to stop and the convergence depends on the regret of `Online-algorithm`. (Proof in Appendix B.)*

3.2. Specific Instantiation of Meta-Algorithm

We now focus on a specific instantiation of Algorithm 1. Algorithm 2 is our main algorithm in this paper.

Policy Learning. We instantiate `Best-response` with Fitted Q Iteration (FQI), a model-free off-policy learning approach (Ernst et al., 2005). FQI relies on a series of reductions to supervised learning. The key idea is to approximate the true action-value function Q^* by a sequence $\{Q_k \in \mathrm{F}\}_{k=0}^K$, where F is a chosen function class.

In Lines 3 & 9, $\texttt{FQI}(c + \lambda^\top g)$ is defined as follows. With Q_0 randomly initialized, for each $k = 1, \ldots, K$, we form a new training dataset $\widetilde{\mathrm{D}}_k = \{(x_i, a_i), y_i\}_{i=1}^n$ where:

$$\forall i: \quad y_i = (c_i + \lambda^\top g_i) + \gamma \min_a Q_{k-1}(x'_i, a),$$

and $(x_i, a_i, x'_i, c_i, g_i) \sim \mathrm{D}$ (original dataset). A supervised regression procedure is called to solve for:

$$Q_k = \arg\min_{f \in \mathrm{F}} \frac{1}{n} \sum_{i=1}^n (f(x_i, a_i) - y_i)^2.$$

FQI returns the policy: $\pi_K = \arg\min_a Q_K(\cdot, a)$. FQI has been shown to work well with several empirical domains: spoken dialogue systems (Pietquin et al., 2011), physical robotic soccer (Riedmiller et al., 2009), and cart-pole swing-up (Riedmiller, 2005), and clinical treatment (Prasad et al., 2017). Another possible model-free subroutine is Least-Squares Policy Iteration (LSPI) (Lagoudakis & Parr, 2003). One can also consider model-based alternatives (Ormoneit & Sen, 2002).

Off-policy Policy Evaluation. A crucial difference between constrained policy learning and existing work on constrained supervised learning is the technical challenge of evaluating the objective and constraints. First, estimating $\widehat{L}(\pi, \lambda)$ (Lines 11-12) requires estimating $\widehat{C}(\pi)$ and $\widehat{G}(\pi)$. Second, any gradient-based approach to `Online-learning` requires passing in $\widehat{G}(\pi) - \tau$ as part of gradient estimate (line 15). This problem is known as the off-policy policy evaluation (OPE) problem: we need to evaluate $\widehat{C}(\pi)$ and $\widehat{G}(\pi)$ having only access to data $\mathrm{D} \sim \pi_{\mathrm{D}}$

There are three main contemporary approaches to OPE: (i) importance weighting (IS) (Precup et al., 2000; 2001), which is unbiased but often has high-variance; (ii) regression-based direct methods (DM), which are typically model-based (Thomas & Brunskill, 2016),and can be biased but have much lower variance than IS; and (iii) doubly-robust techniques (Jiang & Li, 2016; Dudík et al., 2011), which combine IS and DM.

We propose a simple model-free technique using function approximation, called Fitted Q Evaluation (FQE). FQE is based on an iterative reductions scheme similar to FQI, but for the problem of off-policy evaluation. Algorithm 3 lays out the steps. The key difference with FQI is that the min operator is replaced by $Q_{k-1}(x'_i, \pi(x'_i))$ (Line 3 of Algorithm 3). Each x'_i comes from the original D. Since we know $\pi(x'_i)$, each $\widetilde{\mathrm{D}}_k$ is well-defined. Note that FQE can be plugged-in as a direct method if one wishes to augment the policy evaluation with a doubly-robust technique.

Algorithm 3 Fitted Q Evaluation: $\texttt{FQE}(\pi, c)$

Input: Dataset $\mathrm{D} = \{x_i, a_i, x'_i, c_i\}_{i=1}^n \sim \pi_{\mathrm{D}}$. Function class F. Policy π to be evaluated

1: Initialize $Q_0 \in \mathrm{F}$ randomly
2: **for** $k = 1, 2, \ldots, K$ **do**
3: Compute target $y_i = c_i + \gamma Q_{k-1}(x'_i, \pi(x'_i)) \ \forall i$
4: Build training set $\widetilde{\mathrm{D}}_k = \{(x_i, a_i), y_i\}_{i=1}^n$
5: Solve a supervised learning problem:
$Q_k = \arg\min_{f \in \mathrm{F}} \frac{1}{n} \sum_{i=1}^n (f(x_i, a_i) - y_i)^2$
6: **end for**

Output: $\widehat{C}^\pi(x) = Q_K(x, \pi(x)) \quad \forall x$

Online Learning Subroutine. As $L(\pi_t, \lambda)$ is linear in λ, many online convex optimization approaches can be used for `Online-algorithm`. Perhaps the simplest choice is Online Gradient Descent (OGD) (Zinkevich, 2003). We include an instantiation using OGD in Appendix G.

For our main Algorithm 2, similar to (Agarwal et al., 2018), we use Exponentiated Gradient (EG) (Kivinen & Warmuth, 1997), which has a regret bound of $O(\sqrt{\log(m)T})$ instead of $O(\sqrt{mT})$ as in OGD. One can view EG as a variant of Online Mirror Descent (Nemirovsky & Yudin, 1983) with a softmax link function, or of Follow-the-Regularized-Leader with entropy regularization (Shalev-Shwartz et al., 2012). Gradient-based algorithms generally require bounded λ. We thus force $\|\lambda\|_1 \leq B$ using hyperparameter B. Solving (OPT) exactly requires $B = \infty$. We will analyze Algorithm 2 with respect to finite B. With some abuse of notation, we augment λ into a $(m+1)-$dimensional vector by appending $B - \|\lambda\|_1$, and augment the constraint cost vector g by appending 0 (Lines 11, 12 & 15 of Algorithm 2).[3]

4. Theoretical Analysis

4.1. Convergence Guarantee

The convergence rate of Algorithm 2 depends on the radius B of the dual variables λ, the maximal constraint value $\overline{G}$, and the number of constraints m. In particular, we can show $O(\frac{B^2}{\omega^2})$ convergence for primal-dual gap ω.

Theorem 4.1 (Convergence of Algorithm 2)**.** *After T iterations, the empirical duality gap is bounded by*

$$\widehat{\mathrm{L}}_{\max} - \widehat{\mathrm{L}}_{\min} \leq 2\frac{B\log(m+1)}{\eta T} + 2\eta B\overline{G}^2$$

Consequently, to achieve the primal-dual gap of ω, setting $\eta = \frac{\omega}{4\overline{G}^2 B}$ will ensure that Algorithm 2 converges after $\frac{16B^2\overline{G}^2\log(m+1)}{\omega^2}$ iterations. (Proof in Appendix B.)

Convergence analysis of our main Algorithm 2 is an extension of the proof to Proposition 3.1, leveraging the no-regret property of the EG procedure (Shalev-Shwartz et al., 2012).

4.2. Generalization Guarantee of FQE and FQI

In this section, we provide sample complexity analysis for FQE and FQI as *standalone* procedures for off-policy evaluation and off-policy learning. We use the notion of pseudo-dimension as capacity measure of non-linear function class F (Friedman et al., 2001). Pseudo-dimension $\mathtt{dim}_{\mathrm{F}}$, which naturally extends VC dimension into the regression setting, is defined as the VC dimension of the function class induced by the sub-level set of functions of F: $\mathtt{dim}_{\mathrm{F}} = \mathtt{VC\text{-}dim}(\{(x,y) \mapsto \mathrm{sign}(f(x) - y) : f \in \mathrm{F}\})$. Pseudo-dimension is finite for a large class of function approximators. For example, Bartlett et al. (2017) bounded the pseudo-dimension of piece-wise linear deep neural networks (e.g., with ReLU activations) as $O(WL\log W)$, where W is the number of weights, and L is the number of layers.

Both FQI and FQE rely on reductions to supervised learning to update the value functions. In both cases, the learned policy and evaluation policy induces a different state-action distribution compared to the data generating distribution μ. We use the notion of concentration coefficient for the worst case, proposed by (Munos, 2003), to measure the degree of distribution shift. The following standard assumption from analysis of related ADP algorithms limits the severity of distribution shift over future time steps:

Assumption 1 (Concentrability coefficient of future state-action distribution)**.** *(Munos, 2003; 2007; Munos & Szepesvári, 2008; Antos et al., 2008a;b; Lazaric et al., 2010; 2012; Farahmand et al., 2009; Maillard et al., 2010)*
Let P^π be the operator acting on $f : \mathrm{X} \times \mathrm{A} \mapsto \mathbb{R}$ s.t. $(P^\pi f)(x,a) = \int_{\mathrm{X}} f(x', \pi(x'))p(dx'|x,a)$. Given data generating distribution μ, initial state distribution χ, for $m \geq 0$ and an arbitrary sequence of stationary policies $\{\pi_m\}_{m\geq 1}$ define the concentration coeffient:

$$\beta_\mu(m) = \sup_{\pi_1,\ldots,\pi_m} \left\| \frac{d(\chi P^{\pi_1} P^{\pi_2} \ldots P^{\pi_m})}{d\mu} \right\|_\infty$$

We assume $\beta_\mu = (1-\gamma)^2 \sum_{m\geq 1} m\gamma^{m-1}\beta_\mu(m) < \infty$.

This assumption is valid for a fairly large class of MDPs (Munos, 2007). For instance β_μ is finite for any finite MDP, or any infinite state-space MDP with bounded transition density.[4] Having a finite concentration coefficient is equivalent the top-Lyapunov exponent $\Gamma \leq 0$ (Bougerol & Picard, 1992), which means the underlying stochastic system is stable. We show below a simple sufficient condition for Assumption 1 (albeit stronger than necessary).

Example 4.1. Consider an MDP such that for any non-stationary distribution ρ, the marginals over states satisfy $\frac{\rho_x(x)}{\mu_x(x)} \leq L$ (i.e., transition dynamics are sufficiently stochastic), and $\exists M : \forall x, a : \mu(a|x) > \frac{1}{M}$ (i.e., the behavior policy is sufficiently exploratory). Then $\beta_\mu \leq LM$.

Recall that for a given policy π, the Bellman (evaluation) operator is defined as $(\mathbb{T}^\pi Q)(x,a) = r(x,a) + \gamma \int_{\mathrm{X}} Q(x', \pi(x'))p(dx'|x,a)$. In general $\mathbb{T}^\pi f$ may not belong to F for $f \in \mathrm{F}$. For FQE (and FQI), the main operation in the algorithm is to iteratively project $\mathbb{T}^\pi Q_{k-1}$ back to F via $Q_k = \arg\min_{f\in\mathrm{F}} \|f - \mathbb{T}^\pi Q_{k-1}\|$. The performance

[3] The $(m+1)^{th}$ coordinate of g is thus always satisfied. This augmentation is only necessary when executing EG.

[4] This assumption ensures sufficient data diversity, even when the executing policy is deterministic. An example of how learning can fail without this assumption is based on the "combination lock" MDP (Koenig & Simmons, 1996). In this deterministic MDP example, β_μ can grow arbitrarily large, and we need an exponential number of samples for both FQE and FQI. See Appendix D.

of both FQE and FQI thus depend on how well the function class F approximates the Bellman operator. We measure the ability of function class F to approximate the Bellman evaluation operator via the worst-case Bellman error:

Definition 4.1 (inherent Bellman evaluation error)**.** Given a function class F and policy π, the *inherent Bellman evaluation error* of F is defined as $d_{\mathrm{F}}^{\pi} = \sup_{g\in \mathrm{F}} \inf_{f\in \mathrm{F}} \|f - \mathbb{T}^{\pi} g\|_{\pi}$ where $\|\cdot\|_{\pi}$ is the ℓ_2 norm weighted by the state-action distribution induced by π.

We are now ready to state the generalization bound for FQE:

Theorem 4.2 (Generalization error of FQE)**.** *Under Assumption 1, for $\epsilon > 0$ & $\delta \in (0,1)$, after K iterations of Fitted Q Evaluation (Algorithm 3), for $n = O\big(\frac{\overline{C}^4}{\epsilon^2}(\log\frac{K}{\delta} + \mathtt{dim}_{\mathrm{F}} \log\frac{\overline{C}^2}{\epsilon^2} + \log \mathtt{dim}_{\mathrm{F}})\big)$, we have with probability $1-\delta$:*

$$\big|C(\pi) - \widehat{C}(\pi)\big| \leq \frac{\gamma^{1/2}}{(1-\gamma)^{3/2}}\Big(\sqrt{\beta_{\mu}}\,(2d_{\mathrm{F}}^{\pi} + \epsilon) + \frac{2\gamma^{K/2}\overline{C}}{(1-\gamma)^{1/2}}\Big).$$

This result shows a dependency on ϵ of $\widetilde{O}(\frac{1}{\epsilon^2})$, compared to $\widetilde{O}(\frac{1}{\epsilon^4})$ from other related ADP algorithms (Munos & Szepesvári, 2008; Antos et al., 2008b). The price that we pay is a multiplicative constant 2 in front of the inherent error d_{F}^{π}. The error from second term on RHS decays exponentially with iterations K. The proof is in Appendix E.

We can show an analogous generalization bound for FQI. While FQI has been widely used, to the best of our knowledge, a complete analysis of FQI for non-linear function approximation has not been previously reported.[5]

Definition 4.2 (inherent Bellman optimality error)**.** (Munos & Szepesvári, 2008) Recall that the Bellman optimality operator is defined as $(\mathbb{T}Q)(x,a) = r(x,a) + \gamma \int_{\mathrm{X}} \min_{a'\in \mathrm{A}} Q(x',a') p(dx'|x,a)$. Given a function class F, the *inherent Bellman error* is defined as $d_{\mathrm{F}} = \sup_{g\in \mathrm{F}} \inf_{f\in \mathrm{F}} \|f - \mathbb{T}g\|_{\mu}$, where $\|\cdot\|_{\mu}$ is the ℓ_2 norm weighted by μ, the state-action distribution induced by π_{D}.

Theorem 4.3 (Generalization error of FQI)**.** *Under Assumption 1, for $\epsilon > 0$ & $\delta \in (0,1)$, after K iterations of Fitted Q Iteration, for $n = O\big(\frac{\overline{C}^4}{\epsilon^2}(\log\frac{K}{\delta} + \mathtt{dim}_{\mathrm{F}} \log\frac{\overline{C}^2}{\epsilon^2} + \log \mathtt{dim}_{\mathrm{F}})\big)$, we have with probability $1-\delta$:*

$$\big|C^* - C(\pi_K)\big| \leq \frac{2\gamma}{(1-\gamma)^3}\Big(\sqrt{\beta_{\mu}}\,(2d_{\mathrm{F}} + \epsilon) + 2\gamma^{K/2}\overline{C}\Big)$$

where π_K is the policy acting greedy with respect to the returned function Q_K. (Proof in Appendix F.)

4.3. End-to-End Generalization Guarantee

We are ultimately interested in the test-time performance and constraint satisfaction of the returned policy from Algorithm 2. We now connect the previous analyses from Theorems 4.1, 4.2 & 4.3 into an end-to-end error analysis.

Since Algorithm 2 uses FQI and FQE as subroutines, the inherent Bellman error terms d_{F} and d_{F}^{π} will enter our overall performance bound. Estimating the inherent Bellman error caused by function approximation is not possible in general (chapter 11 of Sutton & Barto (2018)). We assume existence of a sufficiently expressive F that can generally make d_{F} and d_{F}^{π} arbitrarily small. To simplify our end-to-end analysis, consider $d_{\mathrm{F}} = 0$ and $d_{\mathrm{F}}^{\pi} = 0$, i.e., the function class F is closed under applying the Bellman operator.

Assumption 2 (Bellman operator realizability)**.** *We consider function classes* F *sufficiently rich so that $\forall f : \mathbb{T}f \in \mathrm{F}$ & $\mathbb{T}^{\pi} f \in \mathrm{F}$ for the policies π returned by Algorithm 2.*

With Assumptions 1 & 2, we have the following error bound:

Theorem 4.4 (Generalization guarantee of Algorithm 2)**.** *Let π^* be the optimal policy to* (OPT)*. Denote $\overline{V} = \overline{C} + B\overline{G}$. Let K be the number of iterations of FQE and FQI. Let $\widehat{\pi}$ be the policy returned by Algorithm 2, with termination threshold ω. For $\epsilon > 0$ & $\delta \in (0,1)$, when $n = O\big(\frac{\overline{V}^4}{\epsilon^2}(\log\frac{K(m+1)}{\delta} + \mathtt{dim}_{\mathrm{F}} \log\frac{\overline{V}^2}{\epsilon^2} + \log \mathtt{dim}_{\mathrm{F}})\big)$, we have with probability at least $1-\delta$:*

$$C(\widehat{\pi}) \leq C(\pi^*) + \omega + \frac{(4+B)\gamma}{(1-\gamma)^3}\Big(\sqrt{\beta_{\mu}}\epsilon + 2\gamma^{K/2}\overline{V}\Big),$$

and

$$G(\widehat{\pi}) \leq \tau + 2\frac{\overline{V}+\omega}{B} + \frac{\gamma^{1/2}}{(1-\gamma)^{3/2}}\Big(\sqrt{\beta_{\mu}}\epsilon + \frac{2\gamma^{K/2}\overline{V}}{(1-\gamma)^{1/2}}\Big).$$

The proof is in Appendix C. This result guarantees that, upon termination of Algorithm 2, the true performance on the main objective can be arbitrarily close to that of the optimal policy. At the same time, each constraint will be approximately satisfied with high probability, assuming sufficiently large B & K, and sufficiently small ϵ.

5. Empirical Analysis

We perform experiments on two different domains: a grid-world domain (from OpenAI's FrozenLake) under a safety constraint, and a challenging high-dimensional car racing domain (from OpenAI's CarRacing) under multiple behavior constraints. We seek to answer the following questions in our experiments: (i) whether the empirical convergence behavior of Algorithm 2 is consistent with our theory; and (ii) how the returned policy performs with respect to the main objective and constraint satisfaction. Appendix H includes a more detailed discussion of our experiments.

5.1. Frozen Lake.

Environment & Data Collection. The environment is an 8x8 grid. The agent has 4 actions N,S,E,W at each state. The main goal is to navigate from a starting position to

[5] FQI for continuous action space from (Antos et al., 2008a) is a variant of fitted policy iteration and not the version of FQI under consideration. The appendix of (Lazaric & Restelli, 2011) contains a proof of FQI specifically for linear function class.

the goal. Each episode terminates when the agent reaches the goal or falls into a hole. The main cost function is defined as $c = -1$ if goal is reached, otherwise $c = 0$ everywhere. We simulate a non-optimal data gathering policy π_{D} by adding random sub-optimal actions to the shortest path policy from any given state to goal. We run π_D for 5000 trajectories to collect the behavior dataset D (with constraint cost measurement specified below).

Counterfactual Safety Constraint. We augment the main objective c with safety constraint cost defined as $g = 1$ if the agent steps into a hole, and $g = 0$ otherwise. We set the constraint threshold $\tau = 0.1$, roughly 75% of the accumulated constraint cost of behavior policy π_{D}. The threshold can be interpreted as a counterfactually acceptable probability that we allow the learned policy to fail.

Results. The empirical primal dual gap $\widehat{\mathrm{L}}_{\max} - \widehat{\mathrm{L}}_{\min}$ in Figure 1 (left) quickly decreases toward the optimal gap of zero. The convergence is fast and monotonic, supporting the predicted behavior from our theory. The test-time performance in Figure 1 (middle) shows the safety constraint is always satisfied, while the main objective cost also smoothly converges to the optimal value achieved by an online RL baseline (DQN) trained without regard to the constraint. The returned policy significantly outperformed the data gathering policy π_{D} on both main and safety cost.

5.2. Car Racing.

Environment & Data Collection. The car racing environment (OpenAI), is a high-dimensional domain where the state is a $96 \times 96 \times 3$ tensor of raw pixels. The action space $\mathrm{A} = \{\text{steering} \times \text{gas} \times \text{brake}\}$ takes 12 discretized values. The goal in this episodic task is to traverse over 95% of the track, measured by a given number of "tiles" as a proxy for distance coverage. The agent receives a reward (negative cost) for each unique tile crossed and no reward if the agent is off-track. A small positive cost applies at every time step, with maximum horizon of 1000 for each episode. With these costs given by the environment, one can train online RL agent using DDQN (Van Hasselt et al., 2016). We collect ≈ 5000 trajectories from DDQN's randomization, resulting in data set D with ≈ 94000 transition tuples.

Fast Driving under Behavioral Constraints. We study the problem of minimizing environment cost while subject to two behavioral constraints: smooth driving and lane centering. The first constraint G_0 approximates smooth driving by $g_0(x, a) = 1$ if a contains braking action, and 0 otherwise. The second constraint cost g_1 measures the distance between the agent and center of lane at each time step. This is a highly challenging setup since three objectives and constraints are in direct conflict with one another, e.g., fast driving encourages the agent to cut corners and apply frequent brakes to make turns. Outside of this work, we are not aware of previous work in policy learning with 2 or more constraints in high-dimensional settings.

Baseline and Procedure. As a naïve baseline, DDQN achieves low cost, but exhibits "non-smooth" driving behavior (see our supplementary videos). We set the threshold for each constraint to 75% of the DDQN benchmark. We also compare against regularized batch RL algorithms (Farahmand et al., 2009), specifically regularized LSPI. We instantiate our subroutines, FQE and FQI, with multi-layered CNNs. We augment LSPI's linear policy with non-linear features derived from a well-performing FQI model.

Results. The returned mixture policy from our algorithm achieves low main objective cost, comparable with online RL policy trained without regard to constraints. After several initial iterations violating the braking constraint, the returned policy - corresponding to the appropriate λ trade-off - satisties both constraints, while improving the main objective. The improvement over data gathering policy is significant for both constraints and main objective.

Regularized policy learning is an alternative approach to (OPT) (section 2). We provide the regularized LSPI baseline the same set of λ found by our algorithm for one-shot regularized learning (Figures 2 (left & middle)). While regularized LSPI obtains good performance for the main objective, it does not achieve acceptable constraint satisfaction. By default, regularized policy learning requires parameter tuning heuristics. In principle, one can perform a grid-search over a range of parameters to find the right combination - we include such an example for both regularized LSPI and FQI in Appendix H. However, since our objective and constraints are in conflict, main objective and constraint satisfaction of policies returned by one-shot regularized learning are sensitive to step changes in λ. In constrast, our approach is systematic, and is able to avoid the curse-of-dimensionality of brute-force search that comes with multiple constraints.

In practice, one may wish to deterministically extract a single policy from the returned mixture for execution. A de-randomized policy can be obtained naturally from our algorithm by selecting the best policy from the existing FQE's estimates of individual `Best-response` policies.

5.3. Off-Policy Evaluation

The off-policy evaluation by FQE is critical for updating policies in our algorithm, and is ultimately responsible for certifying constraint satisfaction. While other OPE methods can also be used in place of FQE, we find that the estimates from popular methods are not sufficiently accurate in a high-dimensional setting. As a standalone comparison, we select an individual policy and compare FQE against PDIS (Precup et al., 2000), DR (Jiang & Li, 2016) and WDR (Thomas & Brunskill, 2016) with respect to the constraint cost evaluation. To compare both accuracy and

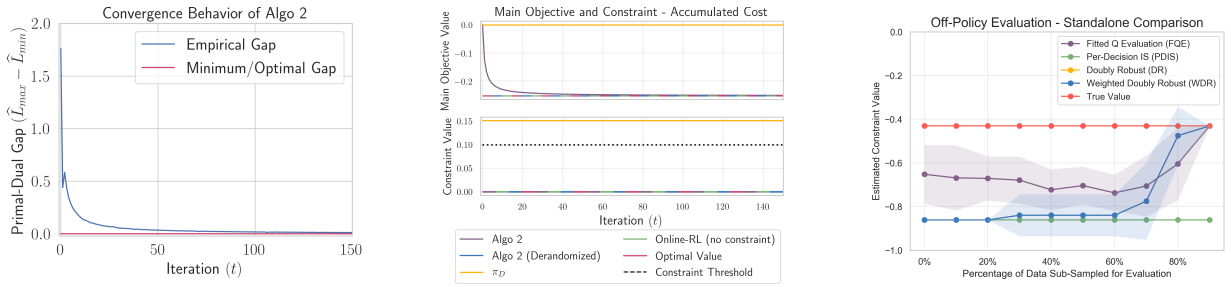

Figure 1. FrozenLake Results. (Left) Empirical duality gap of algorithm 2 vs. optimal gap. *(Middle)* Comparison of returned policy and others w.r.t. (top) main objective value and (bottom) safety constraint value. *(Right)* FQE vs. other OPE methods on a standalone basis.

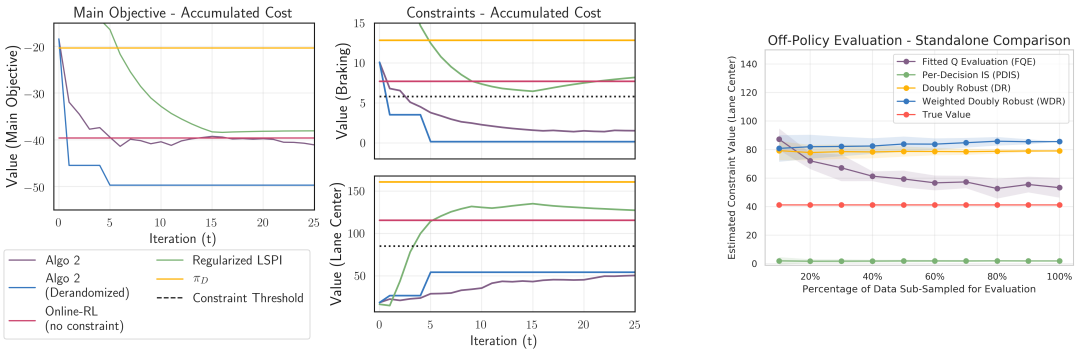

Figure 2. CarRacing Results. (Left) & (Middle) (Lower is better) Comparing our algorithm, regularized LSPI, online RL w/o constraints, behavior policy π_D w.r.t. main cost objectives and two constraints. *(Right)* FQE vs. other OPE methods on a standalone basis.

data-efficiency, for each domain we randomly sample different subsets of dataset D (from 10% to 100% transitions, 30 trials each). Figure 1 (right) and 2 (right) illustrate the difference in quality. In the FrozenLake domain, FQE performs competitively with the top baseline method (DR and WDR), converging to the true value estimate as the data subsample grows close to 100%. In the high-dimensional car domain, FQE signficantly outperforms other methods.

6. Other Related Work

Constrained MDP (CMDP). Among the most important techniques for solving CMDP are the Lagrangian approach and solving the dual LP program via occupation measure(Altman, 1999). However, these approaches require known MDP, and small state dimension so that solving via an LP is tractable. More recently, the constrained policy optimization approach (CPO) by (Achiam et al., 2017) learns a policy when the model is not initially known. The focus of CPO is on online safe exploration, and thus is not directly comparable to our setting. Other approaches (Cheng et al., 2019; Dalal et al., 2018) address safe exploration by building the constraint directly into the policy.

Multi-objective Reinforcement Learning (MORL). (Van Moffaert & Nowé, 2014; Roijers et al., 2013) Approaches to MORL have largely focused on approximating the Pareto frontier that trades-off competing objectives (Van Moffaert & Nowé, 2014; Roijers et al., 2013). The underlying approach to MORL frequently relies on linear or non-linear scalarization of rewards to heuristically turns the problem into a standard RL problem. Our proposed approach represents another systematic paradigm to solve MORL, whether in batch or online settings.

7. Discussion and Conclusion

Our implementation complies with the steps laid out in Algorithm 2. In very large scale or high-dimensional problems, one could consider a noisy update version for both policy learning and evaluation. We leave the theorerical and practical exploration of this extension to future work. In our high-dimensional domain with long horizon, our proposed FQE algorithm for OPE achieves strong results. More extensive comparisons between FQE and other contemporary OPE methods deserve further study.

We have presented a systematic approach for batch policy learning under multiple constraints. Our problem formulation can accommodate general definition of constraints, as partly illustrated by our experiments. We provide guarantees for our algorithm for both the main objective and constraint satisfaction. Our empirical results show a promise of making constrained batch policy learning applicable for real-world domains, where behavior data is abundant.

References

Achiam, J., Held, D., Tamar, A., and Abbeel, P. Constrained policy optimization. In *International Conference on Machine Learning*, pp. 22–31, 2017.

Agarwal, A., Beygelzimer, A., Dudík, M., Langford, J., and Wallach, H. A reductions approach to fair classification. In *International Conference on Machine Learning*, 2018.

Altman, E. *Constrained Markov decision processes*, volume 7. CRC Press, 1999.

Antos, A., Szepesvári, C., and Munos, R. Fitted q-iteration in continuous action-space mdps. In *Advances in neural information processing systems*, pp. 9–16, 2008a.

Antos, A., Szepesvári, C., and Munos, R. Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71(1): 89–129, 2008b.

Bartlett, P. L., Harvey, N., Liaw, C., and Mehrabian, A. Nearly-tight vc-dimension bounds for piecewise linear neural networks. In *Proceedings of the 22nd Annual Conference on Learning Theory (COLT 2017)*, 2017.

Blackmore, L., Ono, M., and Williams, B. C. Chance-constrained optimal path planning with obstacles. *IEEE Transactions on Robotics*, 27(6):1080–1094, 2011.

Bougerol, P. and Picard, N. Strict stationarity of generalized autoregressive processes. *The Annals of Probability*, pp. 1714–1730, 1992.

Boyd, S. and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004.

Cheng, R., Verma, A., Orosz, G., Chaudhuri, S., Yue, Y., and Burdick, J. W. Control regularization for reduced variance reinforcement learning. In *International Conference on Machine Learning*, 2019.

Dalal, G., Dvijotham, K., Vecerik, M., Hester, T., Paduraru, C., and Tassa, Y. Safe exploration in continuous action spaces. *arXiv preprint arXiv:1801.08757*, 2018.

Dudík, M., Langford, J., and Li, L. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pp. 1097–1104. Omnipress, 2011.

Ernst, D., Geurts, P., and Wehenkel, L. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6(Apr):503–556, 2005.

Farahmand, A. M., Ghavamzadeh, M., Mannor, S., and Szepesvári, C. Regularized policy iteration. In *Advances in Neural Information Processing Systems*, pp. 441–448, 2009.

Farajtabar, M., Chow, Y., and Ghavamzadeh, M. More robust doubly robust off-policy evaluation. *arXiv preprint arXiv:1802.03493*, 2018.

Freund, Y. and Schapire, R. E. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29: 79–103, 1999.

Friedman, J., Hastie, T., and Tibshirani, R. *The elements of statistical learning*. Springer, 2001.

Garcıa, J. and Fernández, F. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Guo, Z., Thomas, P. S., and Brunskill, E. Using options and covariance testing for long horizon off-policy policy evaluation. In *Advances in Neural Information Processing Systems*, pp. 2492–2501, 2017.

Györfi, L., Kohler, M., Krzyzak, A., and Walk, H. *A distribution-free theory of nonparametric regression*. Springer Science & Business Media, 2006.

Ha, D. and Schmidhuber, J. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*, pp. 1352–1361, 2017.

Haussler, D. Sphere packing numbers for subsets of the boolean n-cube with bounded vapnik-chervonenkis dimension. *Journal of Combinatorial Theory, Series A*, 69(2):217–232, 1995.

Henaff, M., Canziani, A., and LeCun, Y. Model-predictive policy learning with uncertainty regularization for driving in dense traffic. *arXiv preprint arXiv:1901.02705*, 2019.

Hester, T., Vecerik, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., Horgan, D., Quan, J., Sendonaris, A., Osband, I., et al. Deep q-learning from demonstrations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Jiang, N. and Li, L. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, pp. 652–661, 2016.

Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *ICML*, volume 2, pp. 267–274, 2002.

Kivinen, J. and Warmuth, M. K. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997.

Koenig, S. and Simmons, R. G. The effect of representation and knowledge on goal-directed exploration with reinforcement-learning algorithms. *Machine Learning*, 22(1-3):227–250, 1996.

Lagoudakis, M. G. and Parr, R. Least-squares policy iteration. *Journal of machine learning research*, 4(Dec):1107–1149, 2003.

Lange, S., Gabel, T., and Riedmiller, M. Batch reinforcement learning. In *Reinforcement learning*, pp. 45–73. Springer, 2012.

Lazaric, A. and Restelli, M. Transfer from multiple mdps. In *Advances in Neural Information Processing Systems*, pp. 1746–1754, 2011.

Lazaric, A., Ghavamzadeh, M., and Munos, R. Finite-sample analysis of lstd. In *ICML-27th International Conference on Machine Learning*, pp. 615–622, 2010.

Lazaric, A., Ghavamzadeh, M., and Munos, R. Finite-sample analysis of least-squares policy iteration. *Journal of Machine Learning Research*, 13(Oct):3041–3074, 2012.

Le, H. M., Kang, A., Yue, Y., and Carr, P. Smooth imitation learning for online sequence prediction. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48*, pp. 680–688. JMLR. org, 2016.

Lee, W. S., Bartlett, P. L., and Williamson, R. C. Efficient agnostic learning of neural networks with bounded fan-in. *IEEE Transactions on Information Theory*, 42(6):2118–2132, 1996.

Levine, S. and Abbeel, P. Learning neural network policies with guided policy search under unknown dynamics. In *Advances in Neural Information Processing Systems*, pp. 1071–1079, 2014.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2016.

Liu, Q., Li, L., Tang, Z., and Zhou, D. Breaking the curse of horizon: Infinite-horizon off-policy estimation. In *Advances in Neural Information Processing Systems*, pp. 5361–5371, 2018.

Maillard, O.-A., Munos, R., Lazaric, A., and Ghavamzadeh, M. Finite-sample analysis of bellman residual minimization. In *Proceedings of 2nd Asian Conference on Machine Learning*, pp. 299–314, 2010.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of machine learning*. MIT press, 2012.

Montgomery, W. H. and Levine, S. Guided policy search via approximate mirror descent. In *Advances in Neural Information Processing Systems*, pp. 4008–4016, 2016.

Munos, R. Error bounds for approximate policy iteration. In *ICML*, volume 3, pp. 560–567, 2003.

Munos, R. Performance bounds in l_p-norm for approximate value iteration. *SIAM journal on control and optimization*, 46(2): 541–561, 2007.

Munos, R. and Szepesvári, C. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(May):815–857, 2008.

Nemirovsky, A. S. and Yudin, D. B. *Problem complexity and method efficiency in optimization.* Wiley, 1983.

Oh, J., Guo, Y., Singh, S., and Lee, H. Self-imitation learning. In *International Conference on Machine Learning*, 2018.

Ono, M., Pavone, M., Kuwata, Y., and Balaram, J. Chance-constrained dynamic programming with application to risk-aware robotic space exploration. *Autonomous Robots*, 39(4): 555–571, 2015.

Ormoneit, D. and Sen, Ś. Kernel-based reinforcement learning. *Machine learning*, 49(2-3):161–178, 2002.

Pietquin, O., Geist, M., Chandramohan, S., and Frezza-Buet, H. Sample-efficient batch reinforcement learning for dialogue management optimization. *ACM Transactions on Speech and Language Processing (TSLP)*, 7(3):7, 2011.

Prasad, N., Cheng, L.-F., Chivers, C., Draugelis, M., and Engelhardt, B. E. A reinforcement learning approach to weaning of mechanical ventilation in intensive care units. *arXiv preprint arXiv:1704.06300*, 2017.

Precup, D., Sutton, R. S., and Singh, S. P. Eligibility traces for off-policy policy evaluation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 759–766. Morgan Kaufmann Publishers Inc., 2000.

Precup, D., Sutton, R. S., and Dasgupta, S. Off-policy temporal difference learning with function approximation. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pp. 417–424. Morgan Kaufmann Publishers Inc., 2001.

Riedmiller, M. Neural fitted q iteration–first experiences with a data efficient neural reinforcement learning method. In *European Conference on Machine Learning*, pp. 317–328. Springer, 2005.

Riedmiller, M., Gabel, T., Hafner, R., and Lange, S. Reinforcement learning for robot soccer. *Autonomous Robots*, 27(1):55–73, 2009.

Roijers, D. M., Vamplew, P., Whiteson, S., and Dazeley, R. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

Ross, S. and Bagnell, J. A. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*, 2014.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International Conference on Machine Learning*, pp. 1889–1897, 2015.

Shalev-Shwartz, S. et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4 (2):107–194, 2012.

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

Swaminathan, A. and Joachims, T. Batch learning from logged bandit feedback through counterfactual risk minimization. *Journal of Machine Learning Research*, 16(1):1731–1755, 2015.

Thomas, P. and Brunskill, E. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pp. 2139–2148, 2016.

Van Hasselt, H., Guez, A., and Silver, D. Deep reinforcement learning with double q-learning. In *AAAI*, volume 2, pp. 5. Phoenix, AZ, 2016.

Van Moffaert, K. and Nowé, A. Multi-objective reinforcement learning using sets of pareto dominating policies. *The Journal of Machine Learning Research*, 15(1):3483–3512, 2014.

Von Neumann, J. and Morgenstern, O. *Theory of games and economic behavior (commemorative edition)*. Princeton university press, 2007.

Ziebart, B. D. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. PhD thesis, CMU, 2010.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.

Zinkevich, M. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 928–936, 2003.

A. Equivalence between Regularization and Constraint Satisfaction

A.1. Formulating Different Regularized Policy Learning Problems as Constrained Policy Learning

In this section, we provide connections between regularized policy learning and our constrained formulation (OPT). Although the main paper focuses on batch policy learning, here we are agnostic between online and batch learning settings.

Entropy regularized RL. The standard reinforcement learning objective, either in online or batch setting, is to find a policy π^*_{std} that minimizes the long-term cost (equivalent to maximizing the accumuted rewards): $\pi^*_{\text{std}} = \arg\min_\pi \sum_t \mathbb{E}_{(x_t,a_t)\sim\pi}[c(x_t,a_t)] = \arg\min_\pi \mathbb{E}_{(x,a)\sim\mu_\pi}[c(x,a)]$. Maximum entropy reinforcement learning (Haarnoja et al., 2017) augments the cost with an entropy term, such that the optimal policy maximizes its entropy at each visited state: $\pi^*_{\text{MaxEnt}} = \arg\min_\pi \mathbb{E}_{(x,a)\sim\mu_\pi}[c(x,a)] - \lambda\mathbb{H}(\pi(\cdot|x))$. As discuseed by (Haarnoja et al., 2017), the goal is for the agent to maximize the entropy of the entire trajectory, and not greedily maximizing entropy at the current time step (i.e., Boltzmann exploration). Maximum entropy policy learning was first proposed by (Ziebart et al., 2008; Ziebart, 2010) in the context of learning from expert demonstrations. Entropy regulazed RL/IL is equivalent to our problem (OPT) by simply set $C(\pi) = \mathbb{E}_{(x_t,a_t)\sim\pi}[c(x_t,a_t)]$ (standard RL objective), and $g(x,a) = \pi(a|x)\log\pi(a|x)$, thus $G(\pi) = -\mathbb{H}(\pi) \le \tau$

Smooth imitation learning (& Regularized imitation learning). This is a constrained imitation learning problem studied by (Le et al., 2016): learning to mimic smooth behavior in continuous space from human desmonstrations. The data collected from human demonstrations is considered to be fixed and given a priori, thus the imitation learning task is also a batch policy learning problem. The proposed approach from (Le et al., 2016) is to view policy learning as a function regularization problem: policy $\pi = (f,g)$ is a combination of functions f and h, where f belongs to some expressive function class F (e.g., decision trees, neural networks) and $h \in$ H with certifiable smoothness property (e.g., linear models). Policy learning is the solution to the functional regularization problem $\pi = \arg\min_{f,g} \mathbb{E}_{x\sim\mu_\pi} \|f(x) - \pi_E(x)\| + \lambda\|h(x) - \pi_E(x)\|$, where π_E is the expert policy. This constrained imitation learning setting is equivalent to our problem (OPT) as follows: $C(\pi) = C((f,h)) = \mathbb{E}_{x\sim\mu_\pi}\|f(x) - \pi_E(x)\|$ and $G(\pi) = G((f,h)) = \min_{h'\in\text{H}}\|h'(x) - \pi_E(x)\| \le \tau$

Regularizing RL with expert demonstrations / Learning from imperfect demonstrations. Efficient exploration in RL is a well-known challenge. Expert demonstrations provide a way around online exploration to reduce the sample complexity for learning. However, the label budget for expert demonstrations may be limited, resulting in a sparse coverage of the state space compared to what the online RL agent can explore (Hester et al., 2018). Additionally, expert demonstrations may be imperfect (Oh et al., 2018). Some recent work proposed to regularize standard RL objective with some deviation measure between the learning policy and (sparse) expert data (Hester et al., 2018; Oh et al., 2018; Henaff et al., 2019).

For clarity we focus on the regularized RL objective for Q-learning in (Hester et al., 2018), which is defined as $J(\pi) = J_{DQ}(Q) + \lambda_1 J_n(Q) + \lambda_2 J_E(Q) + \lambda_3 J_{L2}(Q)$, where $J_{DQ}(Q)$ is the standard deep Q-learning loss, $J_n(Q)$ is the n-step return loss, $J_E(Q)$ is the imitation learning loss defined as $J_E(Q) = \max_{a\in\text{A}}[Q(x,a) + \ell(a_E,a) - Q(x,a_E)]$, and $J_{L2}(Q)$ is an L2 regularization loss applied to the Q-network to prevent overfitting to a small expert dataset. The regularization parameters λ's are obtained by hyperparameter tuning. This approach provides a bridge between RL and IL, whose objective functions are fundamentally different (see AggreVate by (Ross & Bagnell, 2014) for an alternative approach).

We can cast this problem into (OPT) as: $C(\pi) = C_{DQ}(Q) + \lambda_3 C_{L2}(Q)$ (standard RL objective), and two constraints: $g_1(\pi) = \mathbb{E}_{x\sim\mu_\pi}[\max_{a\in\text{A}} Q(x,a) + \ell(a_E,a) - Q(x,a_E)]$, and $g_2(x,a) = \mathbb{E}_{x\sim\mu_\pi}[c_t + \gamma c_{t+1} + \ldots + \gamma^{n-1}c_{t+n-1} + \min'_a \gamma^n Q(x_{t+n},a') - Q(x_t,a)]$. Here g_1 captures the loss w.r.t. expert demonstrations and g_2 reflects the n-step return constraint.

More generally, one can define the imitation learning constraint as $G(\pi) = \mathbb{E}_{x\sim\mu_\pi}\ell(\pi(x),\pi_E(x))$ for an appropriate divergence definition between $\pi(x)$ and $\pi_E(x)$ (defined at states where expert demonstrations are available).

Conservative policy improvement. Many policy search algorithms perform small policy update steps, requiring the new policy π to stay within a neighborhood of the most recent policy iterate π_k to ensure learning stability (Levine & Abbeel, 2014; Schulman et al., 2015; Montgomery & Levine, 2016; Achiam et al., 2017). This simply corresponds to the definition of $G(\pi) = \texttt{distance}(\pi,\pi_k) \le \tau$, where `distance` is typically KL-divergence or total variation distance between the distribution induced by π and π_k. For KL-divergence, the single timestep cost $g(x,a) = -\pi(a|x)\log(\frac{\pi_k(a|x)}{\pi(a|x)})$

A.2. Equivalence of Regularization and Constraint Viewpoint - Proof of Proposition 2.1

`Regularization` $\Longrightarrow$ `Constraint`: Let $\lambda > 0$ and π^* be optimal policy in `Regularization`. Set $\tau = G(\pi^*)$. Suppose that π^* is not optimal in `Constraint`. Then $\exists \pi \in \Pi$ such that $G(\pi) \leq \tau$ and $C(\pi) < C(\pi^*)$. We then have

$$C(\pi) + \lambda^\top G(\pi) < C(\pi^*) + \lambda^\top \tau = C(\pi^*) + \lambda^\top G(\pi^*)$$

which contradicts the optimality of π^* for `Regularization` problem. Thus π^* is also the optimal solution of the `Constraint` problem.

`Constraint` $\Longrightarrow$ `Regularization`: Given τ and let π^* be the corresponding optimal solution of the `Constraint` problem. The Lagrangian of `Constraint` is given by $L(\pi, \lambda) = C(\pi) + \lambda^\top G(\pi), \lambda \geq 0$. We then have $\pi^* = \arg\min_{\pi \in \Pi} \max_{\lambda \geq 0} L(\pi, \lambda)$. Let

$$\lambda^* = \arg\max_{\lambda \geq 0} \min_{\pi \in \Pi} L(\pi, \lambda)$$

Slater's condition implies strong duality. By strong duality and the strong max-min property (Boyd & Vandenberghe, 2004), we can exchange the order of maximization and minimization. Thus π^* is the optimal solution of

$$\min_{\pi \in \Pi} \quad C(\pi) + (\lambda^*)^\top (G(\pi) - \tau)$$

Removing the constaint $(\lambda^*)^\top \tau$, we have that π^* is the optimal solution of the `Regularization` problem with $\lambda = \lambda^*$. And since $\pi^* \neq \arg\min_{\pi \in \Pi} C(\pi)$, we must have $\lambda^* \geq 0$.

B. Convergence Proofs

B.1. Convergence of Meta-algorithm - Proof of Proposition 3.1

Let us evaluate the empirical primal-dual gap of the Lagrangian after T iterations:

$$\max_{\lambda} L(\widehat{\pi}_T, \lambda) = \max_{\lambda} \frac{1}{T} \sum_t L(\pi_t, \lambda) \tag{1}$$

$$\leq \frac{1}{T} \sum_t L(\pi_t, \lambda_t) + \frac{o(T)}{T} \tag{2}$$

$$\leq \frac{1}{T} \sum_t L(\pi, \lambda_t) + \frac{o(T)}{T} \quad \forall \pi \in \Pi \tag{3}$$

$$= L(\pi, \widehat{\lambda}_T) + \frac{o(T)}{T} \quad \forall \pi \tag{4}$$

Equations (1) and (4) are due to the definition of $\widehat{\pi}_T$ and $\widehat{\lambda}_T$ and linearity of $L(\pi, \lambda)$ wrt λ and the distribution over policies in Π. Equation (2) is due to the no-regret property of `Online-algorithm`. Equation (3) is true since π_t is best response wrt λ_t. Since equation (4) holds for all π, we can conclude that for T sufficiently large such that $\frac{o(T)}{T} \leq \omega$, we have $\max_\lambda L(\widehat{\pi}_T, \lambda) \leq \min_\pi L(\pi, \widehat{\lambda}_T) + \omega$, which will terminate the algorithm.

Note that we always have $\max_\lambda L(\widehat{\pi}_T, \lambda) \geq L(\widehat{\pi}_T, \widehat{\lambda}_T) \geq \min_\pi L(\pi, \widehat{\lambda}_T)$. Algorithm 1's convergence rate depends on the regret bound of the `Online-algorithm` procedure. Multiple algorithms exist with regret scaling as $\Omega(\sqrt{T})$ (e.g., online gradient descent with regularizer, variants of online mirror descent). In that case, the algorithm will terminate after $O(\frac{1}{\omega^2})$ iterations.

B.2. Empirical Convergence Analysis of Main Algorithm - Proof of Theorem 4.1

By choosing normalized exponentiated gradient as the online learning subroutine, we have the following regret bound after T iterations of the main algorithm 2 (chapter 2 of (Shalev-Shwartz et al., 2012)) for any $\lambda \in \mathbb{R}_+^{m+1}, \|\lambda\|_1 = B$:

$$\frac{1}{T} \sum_{t=1}^{T} \widehat{L}(\pi_t, \lambda) \leq \frac{1}{T} \sum_{t=1}^{T} \widehat{L}(\pi_t, \lambda_t) + \frac{\frac{B \log(m+1)}{\eta} + \eta \overline{G}^2 BT}{T} \tag{5}$$

Denote $\omega_T = \frac{\frac{B \log(m+1)}{\eta} + \eta \overline{G}^2 BT}{T}$ to simplify notations. By the linearity of $\widehat{L}(\pi, \lambda)$ in both π and λ, we have for any λ that

$$\widehat{L}(\widehat{\pi}_T, \lambda) \overset{\text{linearity}}{=} \frac{1}{T} \sum_{t=1}^{T} \widehat{L}(\pi_t, \lambda) \overset{\text{eqn (5)}}{\leq} \frac{1}{T} \sum_{t=1}^{T} \widehat{L}(\pi_t, \lambda_t) + \omega_T \overset{\text{best response } \pi_t}{\leq} \frac{1}{T} \sum_{t=1}^{T} \widehat{L}(\widehat{\pi}_T, \lambda_t) + \omega_T \overset{\text{linearity}}{=} \widehat{L}(\widehat{\pi}_T, \widehat{\lambda}_T) + \omega_T$$

Since this is true for any λ, $\max_\lambda \widehat{L}(\widehat{\pi}_T, \lambda) \leq \widehat{L}(\widehat{\pi}_T, \widehat{\lambda}_T) + \omega_T$.

On the other hand, for any policy π, we also have

$$\widehat{L}(\pi, \widehat{\lambda}_T) \overset{\text{linearity}}{=} \frac{1}{T} \sum_{t=1}^{T} \widehat{L}(\pi, \lambda_t) \overset{\text{best response } \pi_t}{\geq} \frac{1}{T} \sum_{t=1}^{T} \widehat{L}(\pi_t, \lambda_t) \overset{\text{eqn (5)}}{\geq} \frac{1}{T} \sum_{t=1}^{T} \widehat{L}(\pi_t, \widehat{\lambda}_T) - \omega_T \overset{\text{linearity}}{=} \widehat{L}(\widehat{\pi}_T, \widehat{\lambda}_T) - \omega_T$$

Thus $\min_\pi \widehat{L}(\pi, \widehat{\lambda}_T) \geq \widehat{L}(\widehat{\pi}_T, \widehat{\lambda}_T) - \omega_T$, leading to

$$\max_{\lambda} \widehat{L}(\widehat{\pi}_T, \lambda) - \min_{\pi} \widehat{L}(\pi, \widehat{\lambda}_T) \leq \widehat{L}(\widehat{\pi}_T, \widehat{\lambda}_T) + \omega_T - (\widehat{L}(\widehat{\pi}_T, \widehat{\lambda}_T) - \omega_T) = 2\omega_T$$

After T iterations of the main algorithm 2, therefore, the empirical primal-dual gap is bounded by

$$\max_{\lambda} \widehat{L}(\widehat{\pi}_T, \lambda) - \min_{\pi} \widehat{L}(\pi, \widehat{\lambda}_T) \leq \frac{2\frac{B \log(m+1)}{\eta} + 2\eta \overline{G}^2 BT}{T}$$

In particular, if we want the gap to fall below a desired threshold ω, setting the online learning rate $\eta = \frac{\omega}{4\overline{G}^2 B}$ will ensure that the algorithm converges after $\frac{16 B^2 \overline{G}^2 \log(m+1)}{\omega^2}$ iterations.

C. End-to-end Generalization Analysis of Main Algorithm

In this section, we prove the following full statement of theorem 4.4 of the main paper. Note that to lessen notation, we define $\overline{V} = \overline{C} + B\overline{G}$ to be the bound of value functions under considerations in algorithm 2.

Theorem C.1 (Generalization bound of algorithm 2)**.** *Let π^* be the optimal policy to problem OPT. Let K be the number of iterations of FQE and FQI. Let $\widehat{\pi}$ be the policy returned by our main algorithm 2, with termination threshold ω. For any $\epsilon > 0, \delta \in (0,1)$, when $n \geq \frac{24 \cdot 214 \cdot \overline{V}^4}{\epsilon^2}\big(\log\frac{K(m+1)}{\delta} + \mathtt{dim}_{\mathrm{F}}\log\frac{320\overline{V}^2}{\epsilon^2} + \log(14e(\mathtt{dim}_{\mathrm{F}}+1))\big)$, we have with probability at least $1-\delta$:*

$$C(\widehat{\pi}) \leq C(\pi^*) + \omega + \frac{(4+B)\gamma}{(1-\gamma)^3}\big(\sqrt{C_\rho}\epsilon + 2\gamma^{K/2}\overline{V}\big)$$

and

$$G(\widehat{\pi}) \leq \tau + 2\frac{\overline{V}+\omega}{B} + \frac{\gamma^{1/2}}{(1-\gamma)^{3/2}}\big(\sqrt{C_\rho}\epsilon + \frac{2\gamma^{K/2}\overline{V}}{(1-\gamma)^{1/2}}\big)$$

Let $\widehat{\pi} = \frac{1}{T}\sum_t \pi_t$ be the returned policy T iterations, with corresponding dual variable $\widehat{\lambda} = \frac{1}{T}\sum_t \lambda_t$.

By the stopping condition, the empirical duality gap is less than some threshold ω, i.e., $\max_{\lambda \in \mathbb{R}_+^{m+1}, \|\lambda\|_1 = B} \widehat{L}(\widehat{\pi}, \lambda) - \min_{\pi \in \Pi} \widehat{L}(\pi, \widehat{\lambda}) \leq \omega$ where $\widehat{L}(\pi, \lambda) = \widehat{C}(\pi) + \lambda^\top(\widehat{G}(\pi) - \tau)$. We first show that the returned policy approximately satisfies the constraints. The proof of theorem C.1 will make use of the following empirical constraint satisfaction bound:

Lemma C.2 (Empirical constraint satisfactions)**.** *Assume that the constraints $\widehat{G}(\pi) \leq \tau$ are feasible. Then the returned policy $\widehat{\pi}$ approximately satisfies all constraints*

$$\max_{i=1:m+1}(\widehat{g}_i(\widehat{\pi}) - \tau_i) \leq 2\frac{\overline{C}+\omega}{B}$$

Proof. We consider $\max_{i=1:m+1}(\widehat{g}_i(\widehat{\pi}) - \tau_i) > 0$ (otherwise the lemma statement is trivially true). The termination condition implies that $\widehat{L}(\widehat{\pi}, \widehat{\lambda}) - \max_{\lambda \in \mathbb{R}_+^{m+1}, \|\lambda\|_1 = B} \widehat{L}(\widehat{\pi}, \lambda) \geq -\omega$

$$\implies \widehat{\lambda}^\top(\widehat{G}(\widehat{\pi}) - \widehat{\tau}) \geq \max_{\lambda \in \mathbb{R}_+^{m+1}, \|\lambda\|_1 = B} \lambda^\top(\widehat{G}(\widehat{\pi}) - \widehat{\tau}) - \omega \tag{6}$$

Relaxing the RHS of equation (6) by setting $\lambda[j] = B$ for $j = \arg\max_{i=1:m+1}[\widehat{g}_i(\widehat{\pi}) - \tau_i]$ and $\lambda[i] = 0 \ \ \forall i \neq j$ yields:

$$B \max_{i=1:m+1}[\widehat{g}_i(\widehat{\pi}) - \tau_i] - \omega \leq \widehat{\lambda}^\top(\widehat{G}(\widehat{\pi}) - \tau) \tag{7}$$

Given π such that $\widehat{G}(\pi) \leq \tau$, also by the termination condition:

$$\widehat{L}(\widehat{\pi}, \widehat{\lambda}) - \widehat{L}(\pi, \widehat{\lambda}) \leq \max_{\lambda \in \mathbb{R}_+^{m+1}, \|\lambda\|_1 = B} \widehat{L}(\widehat{\pi}, \lambda) - \min_{\pi \in \Pi} \widehat{L}(\pi, \widehat{\lambda}) \leq \omega$$

Thus implies

$$\widehat{L}(\widehat{\pi}, \widehat{\lambda}) \leq \widehat{L}(\pi, \widehat{\lambda}) + \omega = \widehat{C}(\pi) + \widehat{\lambda}^\top(\widehat{G}(\pi) - \tau) \leq \widehat{C}(\pi) + \omega \tag{8}$$

combining what we have from equation (8) and (7):

$$B \max_{i=1:m+1}[\widehat{g}_i(\widehat{\pi}) - \widehat{\tau}_i] - \omega \leq \widehat{\lambda}^\top(\widehat{G}(\widehat{\pi}) - \widehat{\tau}) = \widehat{L}(\widehat{\pi}, \widehat{\lambda}) - \widehat{C}(\widehat{\pi}) \leq \widehat{C}(\pi) + \omega - \widehat{C}(\widehat{\pi})$$

Rearranging and bounding $\widehat{C}(\pi) \leq \overline{C}$ and $\widehat{C}(\widehat{\pi}) \leq -\overline{C}$ finishes the proof of the lemma. □

We now return to the proof of theorem C.1, our task is to lift empirical error to generalization bound for main objective and constraints.

Denote by ϵ_{FQE} the (generalization) error introduced by the Fitted Q Evaluation procedure (algorithm 3) and ϵ_{FQI} the (generalization) error introduced by the Fitted Q Iteration procedure (algorithm 4). For now we keep ϵ_{FQE} and ϵ_{FQI} unspecified (to be specified shortly). That is, for each $t = 1, 2, \ldots, T$, we have with probability at least $1-\delta$:

$$C(\pi_t) + \lambda_t^\top(G(\pi_t) - \tau) \leq C(\pi^*) + \lambda_t^\top(G(\pi^*) - \tau) + \epsilon_{FQI}$$

Since π^* satisfies the constraints, i.e., $G(\pi^*) - \tau \leq 0$ componentwise, and $\lambda_t \geq 0$, we also have with probability $1-\delta$

$$L(\pi_t, \lambda_t) = C(\pi_t) + \lambda_t^\top (G(\pi_t) - \tau) \leq C(\pi^*) + \epsilon_{FQI} \tag{9}$$

Similarly, with probability $1-\delta$, all of the following inequalities are true

$$\widehat{C}(\pi_t) + \epsilon_{FQE} \geq C(\pi_t) \geq \widehat{C}(\pi_t) - \epsilon_{FQE} \tag{10}$$

$$\widehat{G}(\pi_t) + \epsilon_{FQE}\mathbf{1} \geq G(\pi_t) \geq \widehat{G}(\pi_t) - \epsilon_{FQE}\mathbf{1} \text{ (row wise for all } m \text{ constraints)} \tag{11}$$

Thus with probability at least $1-\delta$

$$\begin{aligned} L(\pi_t, \lambda_t) = C(\pi_t) + \lambda_t^\top (G(\pi_t) - \tau) &\geq \widehat{C}(\pi_t) + \lambda_t^\top (\widehat{G}(\pi_t) - \tau) - \epsilon_{FQE}(1 + \lambda_t^\top \mathbf{1}) \\ &\geq \widehat{C}(\pi_t) + \lambda_t^\top (\widehat{G}(\pi_t) - \tau) - \epsilon_{FQE}(1 + B) \\ &= \widehat{L}(\pi_t, \lambda_t) - \epsilon_{FQE}(1 + B) \end{aligned} \tag{12}$$

Recall that the execution of mixture policy $\widehat{\pi}$ is done by uniformly sampling one policy π_t from $\{\pi_1, \ldots, \pi_T\}$, and rolling-out with π_t. Thus from equations (9) and (12), we have $\mathbb{E}_{t\sim U[1:T]} \widehat{L}(\pi_t, \lambda_t) \leq C(\pi^*) + \epsilon_{FQI} + (1+B)\epsilon_{FQE}$ w.p. $1-\delta$. In other words, with probability $1-\delta$:

$$\frac{1}{T}\sum_{t=1}^{T} \widehat{L}(\pi_t, \lambda_t) \leq C(\pi^*) + \epsilon_{FQI} + (1+B)\epsilon_{FQE}$$

Due to the no-regret property of our online algorithm (EG in this case):

$$\frac{1}{T}\sum_{t=1}^{T} \widehat{L}(\pi_t, \lambda_t) \geq \max_{\lambda} \widehat{L}(\widehat{\pi}, \lambda) - \omega = \widehat{C}(\widehat{\pi}) + \max_{\lambda} \lambda^\top (\widehat{G}(\widehat{\pi}) - \tau) - \omega$$

If $\widehat{G}(\widehat{\pi}) - \tau \leq 0$ componentwise, choose $\lambda[i] = 0, i = 1, 2, \ldots, m$ and $\lambda[m+1] = B$. Otherwise, we can choose $\lambda[j] = B$ for $j = \underset{i=1:m+1}{\arg\max} [\widehat{g}_i(\widehat{\pi}) - \tau[i]]$ and $\lambda[i] = 0 \ \ \forall i \neq j$. We can see that $\max_{\lambda \in \mathbb{R}_+^{m+1}, \|\lambda\|_1 = B} \lambda^\top (\widehat{G}(\widehat{\pi}) - \tau) \geq 0$. Therefore:

$$\widehat{C}(\widehat{\pi}) - \omega \leq C(\pi^*) + \epsilon_{FQI} + (1+B)\epsilon_{FQE} \text{ with probability at least } 1-\delta$$

Combined with the first term from equation (10):

$$C(\widehat{\pi}) - \epsilon_{FQE} - \omega \leq C(\pi^*) + \epsilon_{FQI} + (1+B)\epsilon_{FQE}$$

or

$$C(\widehat{\pi}) \leq C(\pi^*) + \omega + \epsilon_{FQI} + (2+B)\epsilon_{FQE} \tag{13}$$

We now bring in the generalization error results from our standalone analysis of FQI (appendix F) and FQE (appendix E) into equation (13).

Specifically, when $n \geq \frac{24\cdot 214\cdot \overline{V}^4}{\epsilon^2}\left(\log\frac{K(m+1)}{\delta} + \mathtt{dim}_{\mathrm{F}} \log\frac{320\overline{V}^2}{\epsilon^2} + \log(14e(\mathtt{dim}_{\mathrm{F}}+1))\right)$, when FQI and FQE are run with K iterations, we have the guarantee that for any $\epsilon > 0$, with probability at least $1-\delta$

$$\begin{aligned} C(\widehat{\pi}) &\leq C(\pi^*) + \omega + \underbrace{\frac{2\gamma}{(1-\gamma)^3}\left(\sqrt{C_\mu}\epsilon + 2\gamma^{K/2}\overline{V}\right)}_{\text{FQI generalization error}} + \underbrace{\frac{\gamma^{1/2}(2+B)}{(1-\gamma)^{3/2}}\left(\sqrt{C_\mu}\epsilon + \frac{\gamma^{K/2}}{(1-\gamma)^{1/2}}2\overline{V}\right)}_{(2+B)\times \text{ FQE generalization error}} \\ &\leq C(\pi^*) + \omega + \frac{(4+B)\gamma}{(1-\gamma)^3}\left(\sqrt{C_\mu}\epsilon + 2\gamma^{K/2}\overline{V}\right) \end{aligned} \tag{14}$$

From lemma C.2, $\widehat{G}(\widehat{\pi}) \leq \tau + 2\frac{\overline{C}+\omega}{B} \leq \tau + 2\frac{\overline{V}+\omega}{B}$. From equation (11), for each t=1,2,…,T, we have $\widehat{G}(\pi_t) \geq G(\pi_t) - \epsilon_{FQE}\mathbf{1}$ with probability $1-\delta$. Thus

$$\mathbf{P}\left(\widehat{G}(\widehat{\pi}) \geq G(\widehat{\pi}) - \epsilon_{FQE}\mathbf{1}\right) = \sum_{t=1}^{T} \mathbf{P}(\widehat{G}(\pi_t) \geq G(\pi_t) - \epsilon_{FQE}\mathbf{1} | \widehat{\pi} = \pi_t)\mathbf{P}(\widehat{\pi} = \pi_t) \geq T(1-\delta)\frac{1}{T} = 1-\delta$$

Therefore, we have the following generalization guarantee for the approximate satisfaction of all constraints:

$$G(\widehat{\pi}) \leq \tau + 2\frac{\overline{V}+\omega}{B} + \frac{\gamma^{1/2}}{(1-\gamma)^{3/2}}\left(\sqrt{C_\mu}\epsilon + \frac{\gamma^{K/2}}{(1-\gamma)^{1/2}}2\overline{V}\right) \tag{15}$$

Inequalities (14) and (15) complete the proof of theorem C.1 (and theorem 4.4 of the main paper)

D. Preliminaries to Analysis of Fitted Q Evaluation (FQE) and Fitted Q Iteration (FQI)

In this section, we set-up necessary notations and definitions for the theoretical analysis of FQE and FQI. To simplify the presentation, we will focus exclusively on weighted ℓ_2 norm for error analysis.

With the definitions and assumptions presented in this section, we will present the sample complexity guarantee of Fitted-Q-Evaluation (FQE) in appendix E. The proof for FQI will follow similarly in appendix F.

While it is possible to adapt proofs from related algorithms (Munos & Szepesvári, 2008; Antos et al., 2008b) to analyze FQE and FQI, in the next two sections we show improved convergence rate from $O(n^{-4})$ to $O(n^{-2})$, where n is the number of samples in data set D.

To be consistent with the notations in the main paper, we use the convention $C(\pi)$ as the value function that denotes long-term accumulated cost, instead of using $V(\pi)$ denoting long-term rewards in the traditional RL literature. Our notation for Q function is similar to the RL literature - the only difference is that the optimal policy minimizes $Q(x,a)$ instead of maximizing. We denote the bound on the value function as $\overline{C}$ (alternatively if the single timestep cost is bounded by $\bar{c}$, then $\overline{C} = \frac{\bar{c}}{1-\gamma}$). For simplicity, the standalone analysis of FQE and FQI concerns only with the cost objective c. Dealing with cost $c + \lambda^\top g$ offers no extra difficulty - in that case we simply augment the bound of the value function to $\overline{V} = \overline{C} + B\overline{G}$.

D.1. Bellman operators

The *Bellman optimality operator* $\mathbb{T} : \mathcal{B}(\mathrm{X} \times \mathrm{A}; \overline{C}) \mapsto \mathcal{B}(\mathrm{X} \times \mathrm{A}; \overline{C})$ as

$$(\mathbb{T}Q)(x,a) = c(x,a) + \gamma \int_{\mathrm{X}} \min_{a' \in \mathrm{A}} Q(x',a') p(dx'|x,a) \tag{16}$$

The optimal value functions are defined as usual by $C^*(x) = \sup_\pi C^\pi(x)$ and $Q^*(x,a) = \sup_\pi Q^\pi(x,a) \ \ \forall x \in \mathrm{X}, a \in \mathrm{A}$.

For a given policy π, the *Bellman evaluation operator* $\mathbb{T}^\pi : \mathcal{B}(\mathrm{X} \times \mathrm{A}; \overline{C}) \mapsto \mathcal{B}(\mathrm{X} \times \mathrm{A}; \overline{C})$ as

$$(\mathbb{T}^\pi Q)(x,a) = c(x,a) + \gamma \int_{\mathrm{X}} Q(x', \pi(x')) p(dx'|x,a) \tag{17}$$

It is well known that $\mathbb{T}^\pi Q^\pi = Q^\pi$, a fixed point of the $\mathbb{T}^\pi$ operator.

D.2. Data distribution and weighted ℓ_2 norm

Denote the state-action data generating distribution as μ, induced by some data-generating (behavior) policy π_{D}, that is, $(x_i, a_i) \sim \mu$ for $(x_i, a_i, x_i', c_i) \in \mathrm{D}$.

Note that data set D is formed by multiple trajectories generated by π_{D}. For each (x_i, a_i), we have $x_i' \sim p(\cdot|x_i, a_i)$ and $c_i = c(x_i, a_i)$. For any (measurable) function $f : \mathrm{X} \times \mathrm{A} \mapsto \mathbb{R}$, define the μ-weighted ℓ_2 norm of f as $\|f\|_\mu^2 = \int_{\mathrm{X}\times\mathrm{A}} f(x,a)^2 \mu(dx, da) = \int_{\mathrm{X}\times\mathrm{A}} f(x,a)^2 \mu_x(dx) \pi_{\mathrm{D}}(a|dx)$. Similarly for any other state-action distribution ρ, $\|f\|_\rho^2 = \int_{\mathrm{X}\times\mathrm{A}} f(x,a)^2 \rho(dx, da)$

D.3. Inherent Bellman error

FQE and FQI depend on a chosen function class F to approximate $Q(x,a)$. To express how well the Bellman operator $\mathbb{T}g$ can be approximated by a function in the policy class F, when $\mathbb{T}g \notin \mathrm{F}$, a notion of distance, known as inherent Bellman error was first proposed by (Munos, 2003) and used in the analysis of related ADP algorithms (Munos & Szepesvári, 2008; Munos, 2007; Antos et al., 2008a;b; Lazaric et al., 2010; 2012; Lazaric & Restelli, 2011; Maillard et al., 2010).

Definition D.1 (Inherent Bellman Error)**.** Given a function class F and a chosen distribution ρ, the *inherent Bellman error* of F is defined as

$$d_{\mathrm{F}} = d(\mathrm{F}, \mathbb{T}\mathrm{F}) = \sup_{h \in \mathrm{F}} \inf_{f \in \mathrm{F}} \|f - \mathbb{T}h\|_\rho$$

where $\|\cdot\|_\rho$ is the $\rho-$weighted ℓ_2 norm and $\mathbb{T}$ is the Bellman optimality operator defined in (16)

To analyze FQE, we will form a similar definition for the Bellman evaluation operator

Definition D.2 (Inherent Bellman Evaluation Error)**.** Given a function class F and a policy π, the *inherent Bellman*

evaluation error of F is defined as

$$d_{\mathrm{F}}^{\pi} = d(\mathrm{F}, \mathbb{T}^{\pi}\mathrm{F}) = \sup_{h \in \mathrm{F}} \inf_{f \in \mathrm{F}} \|f - \mathbb{T}^{\pi} h\|_{\rho_\pi}$$

where $\|\cdot\|_{\rho_\pi}$ is the ℓ_2 norm weighted by ρ_π. ρ_π is defined as the state-action distribution induced by policy π, and $\mathbb{T}^{\pi}$ is the Bellman operator defined in (17)

D.4. Concentrability coefficients

Let P^{π} denote the operator acting on $f : \mathrm{X} \times \mathrm{A} \mapsto \mathbb{R}$ such that $(P^{\pi} f)(x, a) = \int_{\mathrm{X}} f(x', \pi(x')) p(x'|x, a) dx'$. Acting on f (e.g., approximates Q), P^{π} captures the transition dynamics of taking action a and following π thereafters.

The following definition and assumption are standard in the analysis of related approximate dynamic programming algorithms (Lazaric et al., 2012; Munos & Szepesvári, 2008; Antos et al., 2008a). As approximate value iteration and policy iteration algorithms perform policy update, the new policy at each round will induce a different stationary state-action distribution. One way to quantify the distribution shift is the notion of concentrability coefficient of future state-action distribution, a variant of the notion introduced by (Munos, 2003).

Definition D.3 (Concentrability coefficient of state-action distribution)**.** Given data generating distribution $\mu \sim \pi_{\mathrm{D}}$, initial state distribution χ. For $m \geq 0$, and an arbitrary sequence of stationary policies $\{\pi_m\}_{m \geq 1}$ let

$$\beta_{\mu}(m) = \sup_{\pi_1, \ldots, \pi_m} \left\| \frac{d(\chi P^{\pi_1} P^{\pi_2} \ldots P^{\pi_m})}{d\mu} \right\|_{\infty}$$

($\beta_{\mu}(m) = \infty$ if the future state distribution $\chi P^{\pi_1} P^{\pi_2} \ldots P^{\pi_m}$ is not absolutely continuous w.r.t. μ, i.e, $\chi P^{\pi_1} P^{\pi_2} \ldots P^{\pi_m}(x, a) > 0$ for some $\mu(x, a) = 0$)

Assumption 3. $\beta_{\mu} = (1 - \gamma)^2 \sum_{m \geq 1} m \gamma^{m-1} \beta_{\mu}(m) < \infty$

Combination Lock Example. An example of an MDP that violates Assumption 3 is the "combination lock" example proposed by (Koenig & Simmons, 1996). In this finite MDP, we have N states $\mathrm{X} = \{1, 2, \ldots, N\}$, and 2 actions: going L or R. The initial state is $x_0 = 1$. In any state x, action L takes agent back to initial state x_0, and action R advances the agent to the next state $x + 1$ in a chain fashion. Suppose that the reward is 0 everywhere except for the very last state N. One can see that for an MDP such that any behavior policy π_D that has a bounded from below probability of taking action L from any state x, i.e., $\pi_D(L|x) \geq \nu > 0$, then it takes an exponential number of trajectories to learn or evaluate a policy that always takes action R. In this setting, we can see that the concentration coefficient β_{μ} can be designed to be arbitrarily large.

D.5. Complexity measure of function class F

Definition D.4 (Random L_1 Norm Covers)**.** Let $\epsilon > 0$, let F be a set of functions $\mathrm{X} \mapsto \mathbb{R}$, let $x_1^n = (x_1, \ldots, x_n)$ be n fixed points in X. Then a collection of functions $\mathrm{F}_{\epsilon} = \{f_1, \ldots, f_N\}$ is an ϵ-cover of F on x_1^n if

$$\forall f \in \mathrm{F}, \exists f' \in \mathrm{F}_{\epsilon} : |\frac{1}{n} \sum_{i=1}^{n} f(x_i) - \frac{1}{n} \sum_{i=1}^{n} f'(x_i)| \leq \epsilon$$

The empirical covering number, denote by $\mathcal{N}_1(\epsilon, \mathrm{F}, x_1^n)$, is the size of the smallest ϵ-cover on x_1^n. Take $\mathcal{N}_1(\epsilon, \mathrm{F}, x_1^n) = \infty$ if no finite ϵ-cover exists.

Definition D.5 (Pseudo-Dimension)**.** A real-valued function class F has pseudo-dimension $\mathtt{dim}_{\mathrm{F}}$ defined as the VC dimension of the function class induced by the sub-level set of functions of F. In other words, define function class $\mathrm{H} = \{(x, y) \mapsto \mathrm{sign}(f(x) - y : f \in \mathrm{F}\}$, then

$$\mathtt{dim}_{\mathrm{F}} = \mathtt{VC\text{-}dimension}(\mathrm{H})$$

E. Generalization Analysis of Fitted Q Evaluation

In this section we prove the following statement for Fitted Q Evaluation (FQE).

Theorem E.1 (Guarantee for FQE - General Case (theorem 4.2 in main paper)). *Under Assumption 3, for $\epsilon > 0$ & $\delta \in (0,1)$, after K iterations of Fitted Q Evaluation (Algorithm 3), for $n = O\big(\frac{\overline{C}^4}{\epsilon^2}(\log\frac{K}{\delta} + \dim_{\mathrm{F}} \log\frac{\overline{C}^2}{\epsilon^2} + \log \dim_{\mathrm{F}})\big)$, we have with probability $1-\delta$:*

$$\big|C(\pi) - \widehat{C}(\pi)\big| \leq \frac{\gamma^{1/2}}{(1-\gamma)^{3/2}}\big(\sqrt{\beta_\mu}\,(2d_{\mathrm{F}}^{\pi} + \epsilon) + \frac{2\gamma^{K/2}\overline{C}}{(1-\gamma)^{1/2}}\big).$$

Theorem E.2 (Guarantee for FQE - Bellman Realizable Case). *Under Assumptions 3-4, for any $\epsilon > 0, \delta \in (0,1)$, after K iterations of Fitted Q Evaluation (Algorithm 3), when $n \geq \frac{24\cdot 214\cdot \overline{C}^4}{\epsilon^2}\big(\log\frac{K}{\delta} + \dim_{\mathrm{F}} \log\frac{320\overline{C}^2}{\epsilon^2} + \log(14e(\dim_{\mathrm{F}}+1))\big)$, we have with probability $1-\delta$:*

$$\big|C(\pi) - \widehat{C}(\pi)\big| \leq \frac{\gamma^{1/2}}{(1-\gamma)^{3/2}}\big(\sqrt{\beta_\mu}\epsilon + \frac{2\gamma^{K/2}\overline{C}}{(1-\gamma)^{1/2}}\big)$$

We first focus on theorem E.2, analyzing FQE assuming a sufficiently rich function class F so that the Bellman evaluation update $\mathbb{T}^\pi$ is closed wrt F (thus inherent Bellman evaluation error is 0). We call this the *Bellman evaluation realizability assumption*. This assumption simplifies the presentation of our bounds and also simplifies the final error analysis of Algo. 2.

After analyzing FQE under this Bellman realizable setting, we will turn to error bound for general, non-realizable setting in theorem E.1 (also theorem 4.2 in the main paper). The main difference in the non-realizable setting is the appearance of an extra term d_{F}^{π} our final bound.

E.1. Error bound for single iteration - Bellman realizable case

Assumption 4 (Bellman evaluation realizability). *We consider function classes* F *sufficiently rich so that* $\forall f, \mathbb{T}^\pi f \in \mathrm{F}$.

We begin with the following result bounding the error for a single iteration of FQE, under "training" distribution $\mu \sim \pi_D$

Proposition E.3 (Error bound for single iteration). *Let the functions in* F *also be bounded by $\overline{C}$, and let $\dim_{\mathrm{F}}$ denote the pseudo-dimension of the function class* F. *We have with probability at least $1-\delta$:*

$$\|Q_k - \mathbb{T}^\pi Q_{k-1}\|_\mu < \epsilon$$

when $n \geq \frac{24\cdot 214\cdot \overline{C}^4}{\epsilon^2}\left(\log\frac{1}{\delta} + \dim_{\mathrm{F}} \log\frac{320\overline{C}^2}{\epsilon^2} + \log(14e(\dim_{\mathrm{F}}+1))\right)$

Remark E.4. *Note from proposition E.3 that the dependence of sample complexity n here on ϵ is $\widetilde{O}(\frac{1}{\epsilon^2})$, which is better than previously known analysis for Fitted Value Iteration (Munos & Szepesvári, 2008) and FittedPolicyQ (continuous version of Fitted Q Iteration (Antos et al., 2008a)) dependence of $\widetilde{O}(\frac{1}{\epsilon^4})$. The finite sample analysis of LSTD (Lazaric et al., 2010) showed an $\widetilde{O}(\frac{1}{\epsilon^2})$ dependence using linear function approximation. Here we prove similar convergence rate for general non-linear (bounded) function approximators.*

Proof of Proposition E.3. Recall the training target in round k is $y_i = c_i + \gamma Q_{k-1}(x_i', \pi(x_i'))$ for $i = 1, 2, \ldots, n$, and $Q_k \in \mathrm{F}$ is the solution to the following regression problem:

$$Q_k = \arg\min_{f \in \mathrm{F}} \frac{1}{n}\sum_{i=1}^{n}(f(x_i, a_i) - y_i)^2$$

Consider random variables $(x, a) \sim \mu$ and $y = c(x,a) + \gamma Q_{k-1}(x', \pi(x'))$ where $x' \sim p(\cdot|x,a)$. By this definition, $\mathbb{T}^\pi Q_{k-1}$ is the *regression function* that minimizes square loss $\min_{h:\mathbb{R}^{X\times A}\mapsto\mathbb{R}} \mathbb{E}|h(x,a) - y|^2$ out of all functions h (not necessarily in F). This is due to $(\mathbb{T}^\pi Q_{k-1})(\tilde{x}, \tilde{a}) = \mathbb{E}\left[y|x = \tilde{x}, a = \tilde{a}\right]$ by definition of the Bellman operator. Consider Q_{k-1} fixed and we now want to relate the learned function Q_k over finite set of n samples with the regression function over the whole data distribution via uniform deviation bound. We use the following lemma:

Lemma E.5 ((Györfi et al., 2006), theorem 11.4. Original version (Lee et al., 1996), theorem 3). *Consider random vector (X, Y) and n i.i.d samples (X_i, Y_i). Let $m(x)$ be the (optimal) regression function under square loss $m(x) = \mathbb{E}[Y|X = x]$. Assume $|Y| \leq B$ a.s. and $B \leq 1$. Let* F *be a set of function $f : \mathbb{R}^d \mapsto \mathbb{R}$ and let $|f(x)| \leq B$. Then for each $n \geq 1$*

$$\mathbf{P}\Bigg\{\exists f \in \mathrm{F} : \mathbb{E}|f(X) - Y|^2 - \mathbb{E}|m(X) - Y|^2 - \frac{1}{n}\sum_{i=1}^{n}\left(|f(X_i) - Y_i|^2 - |m(X_i) - Y_i|^2\right) \geq$$

$$\left. \epsilon \cdot \left(\alpha + \beta + \mathbb{E}|f(X) - Y|^2 - \mathbb{E}|m(X) - Y|^2\right) \right\}$$
$$\leq 14 \sup_{x_1^n} \mathcal{N}_1\left(\frac{\beta\epsilon}{20B}, \mathrm{F}, x_1^n\right) \exp\left(-\frac{\epsilon^2(1-\epsilon)\alpha n}{214(1+\epsilon)B^4}\right)$$

where $\alpha, \beta > 0$ *and* $0 < \epsilon < 1/2$

To apply this lemma, first note that since $\mathbb{T}^\pi Q_{k-1}$ is the optimal regression function[6], we have

$$\mathbb{E}_\mu\left[(Q_k(x,a) - y)^2\right] = \mathbb{E}_\mu\left[\left(Q_k(x,a) - \mathbb{T}^\pi Q_{k-1}(x,a) + \mathbb{T}^\pi Q_{k-1}(x,a) - y\right)^2\right]$$
$$= \mathbb{E}_\mu\left[\left(Q_k(x,a) - \mathbb{T}^\pi Q_{k-1}(x,a)\right)^2\right] + \mathbb{E}_\mu[\left(\mathbb{T}^\pi Q_{k-1}(x,a) - y\right)^2]$$

thus

$$\|Q_k - \mathbb{T}^\pi Q_{k-1}\|_\mu^2 = \mathbb{E}\left[\left(Q_k(x,a) - \mathbb{T}^\pi Q_{k-1}(x,a)\right)^2\right] = \mathbb{E}\left[(Q_k(x,a) - y)^2\right] - \mathbb{E}\left[(\mathbb{T}^\pi Q_{k-1}(x,a) - y)^2\right]$$

where by definition

$$\mathbb{E}\left[\left(Q_k(x,a) - \mathbb{T}^\pi Q_{k-1}(x,a)\right)^2\right] = \int \left(Q_k(x,a) - \mathbb{T}^\pi Q_{k-1}(x,a)\right)^2 \mu(dx, da)$$
$$= \int (Q_k(x,a) - \mathbb{T}^\pi(x,a))^2 \mu_x(dx)\pi_{\mathrm{D}}(a|dx)$$

Next, given a fixed data set $\widetilde{D}_k \sim \mu$

$$\mathbf{P}\left\{\|Q_k - \mathbb{T}^\pi Q_{k-1}\|_\mu^2 > \epsilon\right\} = \mathbf{P}\left\{\mathbb{E}\left[(Q_k(x,a) - y)^2\right] - \mathbb{E}\left[(\mathbb{T}^\pi Q_{k-1}(x,a) - y)^2\right] > \epsilon\right\}$$
$$\leq \mathbf{P}\Bigg\{\mathbb{E}\left[(Q_k(x,a) - y)^2\right] - \mathbb{E}\left[(\mathbb{T}^\pi Q_{k-1}(x,a) - y)^2\right]$$
$$- 2 \cdot \left(\frac{1}{n}\sum_{i=1}^n (Q_k(x_i,a_i) - y_i)^2 - \frac{1}{n}\sum_{i=1}^n (\mathbb{T}^\pi Q_{k-1}(x_i,a_i) - y_i)^2\right) > \epsilon\Bigg\} \tag{18}$$
$$= \mathbf{P}\Bigg\{\mathbb{E}\left[(Q_k(x,a) - y)^2\right] - \mathbb{E}\left[(\mathbb{T}^\pi Q_{k-1}(x,a) - y)^2\right]$$
$$- \frac{1}{n}\sum_{i=1}^n \left[(Q_k(x_i,a_i) - y_i)^2 - (\mathbb{T}^\pi Q_{k-1}(x_i,a_i) - y_i)^2\right]$$
$$> \frac{1}{2}(\epsilon + \mathbb{E}\left[(Q_k(x,a) - y)^2\right] - \mathbb{E}\left[(\mathbb{T}^\pi Q_{k-1}(x,a) - y)^2\right])\Bigg\} \tag{19}$$
$$\leq \mathbf{P}\Bigg\{\exists f \in \mathrm{F} : \mathbb{E}\left[(f(x,a) - y)^2\right] - \mathbb{E}\left[(\mathbb{T}^\pi Q_{k-1}(x,a) - y)^2\right]$$
$$- \frac{1}{n}\sum_{i=1}^n \left[(f(x_i,a_i) - y_i)^2 - (\mathbb{T}^\pi Q_{k-1}(x_i,a_i) - y_i)^2\right]$$
$$\geq \frac{1}{2}(\frac{\epsilon}{2} + \frac{\epsilon}{2} + \mathbb{E}\left[(f(x,a) - y)^2\right] - \mathbb{E}\left[(\mathbb{T}^\pi Q_{k-1}(x,a) - y)^2\right])\Bigg\}$$
$$\leq 14 \sup_{x_1^n} \mathcal{N}_1\left(\frac{\epsilon}{80\overline{C}}, \mathrm{F}, x_1^n\right) \cdot \exp\left(-\frac{n\epsilon}{24 \cdot 214\overline{C}^4}\right) \tag{20}$$

Equation (18) uses the definition of $Q_k = \arg\min_{f \in \mathrm{F}} \frac{1}{n}\sum_{i=1}^n (f(x_i,a_i) - y_i)^2$ and the fact that $\mathbb{T}^\pi Q_{k-1} \in \mathrm{F}$, thus making the extra term a positive addition. Equation (19) is due to rearranging the terms. Equation (20) is an application of lemma E.5. We can further bound the empirical covering number by invoking the following lemma due to Haussler (Haussler, 1995):

Lemma E.6 ((Haussler, 1995), Corollary 3)**.** *For any set* X*, any points* $x^{1:n} \in \mathrm{X}^n$*, any class* F *of functions on* X *taking*

[6]It is easy to see that if $m(x) = \mathbb{E}[y|x]$ is the regression function then for any function $f(x)$, we have $\mathbb{E}\left[(f(x) - m(x))(m(x) - y) = 0\right]$

values in $[0, \overline{C}]$ *with pseudo-dimension* $\dim_{\mathrm{F}} < \infty$*, and any* $\epsilon > 0$

$$\mathcal{N}_1(\epsilon, \mathrm{F}, x_1^n) \leq e(\dim_{\mathrm{F}} + 1)\left(\frac{2e\overline{C}}{\epsilon}\right)^{\dim_{\mathrm{F}}}$$

Applying lemma E.6 to equation (20), we have the inequality

$$\mathbf{P}\left\{ \|Q_k - \mathbb{T}^\pi Q_{k-1}\|_\mu^2 > \epsilon \right\} \leq 14 \cdot e \cdot (\dim_{\mathrm{F}} + 1)\left(\frac{320\overline{C}^2}{\epsilon}\right)^{\dim_{\mathrm{F}}} \cdot \exp\left(-\frac{n\epsilon}{24 \cdot 214\overline{C}^4}\right) \tag{21}$$

We thus have that when $n \geq \frac{24 \cdot 214 \cdot \overline{C}^4}{\epsilon^2}\left(\log\frac{1}{\delta} + \dim_{\mathrm{F}} \log\frac{320\overline{C}^2}{\epsilon^2} + \log(14e(\dim_{\mathrm{F}} + 1))\right)$:

$$\|Q_k - \mathbb{T}^\pi Q_{k-1}\|_\rho < \epsilon$$

with probability at least $1 - \delta$. Notice that the dependence of sample complexity n here on ϵ is $\widetilde{O}(\frac{1}{\epsilon^2})$, which is better than previously known analyses for other approximate dynamic programming algorithms such as Fitted Value Iteration (Munos & Szepesvári, 2008), FittedPolicyQ (Antos et al., 2008b;a) with dependence of $O(\frac{1}{\epsilon^4})$.

E.2. Error bound for single iteration - Bellman non-realizable case

We now give similar error bound for the general case, where Assumption 4 does not hold. Consider the decomposition

$$\begin{aligned}
\|Q_k - \mathbb{T}^\pi Q_{k-1}\|_\mu^2 &= \mathbb{E}\left[(Q_k(x,a) - y)^2\right] - \mathbb{E}\left[(\mathbb{T}^\pi Q_{k-1}(x,a) - y)^2\right] \\
&= \Bigg\{ \mathbb{E}\left[(Q_k(x,a) - y)^2\right] - \mathbb{E}\left[(\mathbb{T}^\pi Q_{k-1}(x,a) - y)^2\right] \\
&\qquad - 2 \cdot \left(\frac{1}{n}\sum_{i=1}^n (Q_k(x_i,a_i) - y_i)^2 - \frac{1}{n}\sum_{i=1}^n (\mathbb{T}^\pi Q_{k-1}(x_i,a_i) - y_i)^2 \right) \Bigg\} \\
&\qquad + \left\{ 2 \cdot \left(\frac{1}{n}\sum_{i=1}^n (Q_k(x_i,a_i) - y_i)^2 - \frac{1}{n}\sum_{i=1}^n (\mathbb{T}^\pi Q_{k-1}(x_i,a_i) - y_i)^2 \right) \right\} \\
&= \texttt{component_1} + \texttt{component_2}
\end{aligned}$$

Splitting the probability of error into two separate bounds. We saw from the previous section (equation (21)) that

$$\mathbf{P}(\texttt{component\_1} > \epsilon/2) \leq 14 \cdot e \cdot (\dim_{\mathrm{F}} + 1)\left(\frac{640\overline{C}^2}{\epsilon}\right)^{\dim_{\mathrm{F}}} \cdot \exp\left(-\frac{n\epsilon}{48 \cdot 214\overline{C}^4}\right) \tag{22}$$

We no longer have $\texttt{component\_2} \leq 0$ since $\mathbb{T}^\pi Q_{k-1} \notin \mathrm{F}$. Let $f^* = \inf_{f \in \mathrm{F}} \|f - \mathbb{T}^\pi Q_{k-1}\|_\mu^2$. Since $Q_k = \arg\min_{f \in \mathrm{F}} \frac{1}{n}\sum_{i=1}^n (f(x_i,a_i) - y_i)^2$, we can upper-bound $\texttt{component\_2}$ by

$$\texttt{component\_2} \leq 2 \cdot \left(\frac{1}{n}\sum_{i=1}^n (f^*(x_i,a_i) - y_i)^2 - \frac{1}{n}\sum_{i=1}^n (\mathbb{T}^\pi Q_{k-1}(x_i,a_i) - y_i)^2 \right)$$

We can treat f^* as a fixed function, unlike random function Q_k, and use standard concentration inequalities to bound the empirical average from the expectation. Let random variable $z = ((x,a), y)$, $z_i = ((x_i,a_i), y_i), i = 1, \ldots, n$ and let

$$h(z) = (f^*(x,a) - y)^2 - (\mathbb{T}^\pi Q_{k-1}(x,a) - y)^2$$

We have $|h(z)| \leq 4\overline{C}^2$. We will derive a bound for

$$\mathbf{P}\left(\frac{1}{n}\sum_{i=1}^n h(z_i) - \mathbb{E}h(z) > \frac{\epsilon}{4} + \mathbb{E}h(z) \right)$$

using Bernstein inequality(Mohri et al., 2012). First, using the relationship $h(z) = (f^*(x,a) + \mathbb{T}^\pi Q_{k-1}(x,a) - 2y)(f^*(x,a) - \mathbb{T}^\pi Q_{k-1}(x,a))$, the variance of $h(z)$ can be bounded by a constant factor of $\mathbb{E}h(z)$, since

$$\begin{aligned}
\mathbf{Var}(h(z)) &\leq \mathbb{E}h(z)^2 \leq 16\overline{C}^2 \mathbb{E}\left[(f^*(x,a) - \mathbb{T}^\pi Q_{k-1}(x,a))^2\right] \\
&= 16\overline{C}^2\left(\mathbb{E}\left[(f^*(x,a) - y)^2\right] - \mathbb{E}\left[(\mathbb{T}^\pi Q_{k-1}(x,a) - y)^2\right]\right) \qquad (23) \\
&= 16\overline{C}^2 \mathbb{E}h(z) \qquad (24)
\end{aligned}$$

Equation (23) stems from $\mathbb{T}^\pi Q_{k-1}$ being the optimal regression function. Now we can apply equation (24) and Bernstein inequality to obtain

$$\mathbf{P}\left(\frac{1}{n}\sum_{i=1}^{n}h(z_i)-\mathbb{E}h(z)>\frac{\epsilon}{4}+\mathbb{E}h(z)\right)\leq\mathbf{P}\left(\frac{1}{n}\sum_{i=1}^{n}h(z_i)-\mathbb{E}h(z)>\frac{\epsilon}{4}+\frac{\mathbf{Var}(h(z))}{16\overline{C}^2}\right)\leq\dots$$
$$\leq\exp\left(-\frac{n\left(\frac{\epsilon}{4}+\frac{\mathbf{Var}}{16\overline{C}^2}\right)^2}{2\mathbf{Var}+2\frac{4\overline{C}^2}{3}\left(\frac{\epsilon}{4}+\frac{\mathbf{Var}}{16\overline{C}^2}\right)}\right)$$
$$\leq\exp\left(-\frac{n\left(\frac{\epsilon}{4}+\frac{\mathbf{Var}}{16\overline{C}^2}\right)^2}{\left(32\overline{C}^2+\frac{8\overline{C}^2}{3}\right)\left(\frac{\epsilon}{4}+\frac{\mathbf{Var}}{16\overline{C}^2}\right)}\right)=\exp\left(-\frac{n\left(\frac{\epsilon}{4}+\frac{\mathbf{Var}}{16\overline{C}^2}\right)}{32\overline{C}^2+\frac{8\overline{C}^2}{3}}\right)\leq\exp\left(-\frac{1}{128+\frac{32}{3}}\cdot\frac{n\epsilon}{\overline{C}^2}\right)$$

Thus

$$\mathbf{P}\left(2\cdot\left[\frac{1}{n}\sum_{i=1}^{n}h(z_i)-2\mathbb{E}h(z)\right]>\frac{\epsilon}{2}\right)\leq\exp\left(-\frac{3}{416}\cdot\frac{n\epsilon}{\overline{C}^2}\right)\tag{25}$$

Now we have

$$\texttt{component\_2}\leq 2\cdot\frac{1}{n}\sum_{i=1}^{n}h(z_i)=2\cdot\left[\frac{1}{n}\sum_{i=1}^{n}h(z_i)-2\mathbb{E}h(z)\right]+4\mathbb{E}h(z)$$

Using again the fact that $\mathbb{T}^\pi Q_{k-1}$ is the optimal regression function

$$\mathbb{E}h(z)=\mathbb{E}_D\left[(f^*(x,a)-y)^2\right]-\mathbb{E}_D\left[(\mathbb{T}^\pi Q_{k-1}(x,a)-y)^2\right]=\mathbb{E}_D\left[(f^*(x,a)-\mathbb{T}^\pi Q_{k-1}(x,a))^2\right]$$
$$=\inf_{f\in\mathrm{F}}\|f-\mathbb{T}^\pi Q_{k-1}\|_\mu^2\tag{26}$$

Combining equations (22), (25) and (26), we can conclude that

$$\mathbf{P}\left\{\|Q_k-\mathbb{T}^\pi Q_{k-1}\|_\mu^2-4\inf_{f\in\mathrm{F}}\|f-\mathbb{T}^\pi Q_{k-1}\|_\mu^2>\epsilon\right\}\leq 14\cdot e\cdot(\texttt{dim}_\mathrm{F}+1)\left(\frac{640\overline{C}^2}{\epsilon}\right)^{\texttt{dim}_\mathrm{F}}\cdot\exp\left(-\frac{n\epsilon}{48\cdot 214\overline{C}^4}\right)$$
$$+\exp\left(-\frac{3}{416}\cdot\frac{n\epsilon}{\overline{C}^2}\right)$$

thus implying

$$\mathbf{P}\left\{\|Q_k-\mathbb{T}^\pi Q_{k-1}\|_\mu-2\inf_{f\in\mathrm{F}}\|f-\mathbb{T}^\pi Q_{k-1}\|_\mu>\epsilon\right\}\leq 14\cdot e\cdot(\texttt{dim}_\mathrm{F}+1)\left(\frac{640\overline{C}^2}{\epsilon^2}\right)^{\texttt{dim}_\mathrm{F}}\cdot\exp\left(-\frac{n\epsilon^2}{48\cdot 214\overline{C}^4}\right)$$
$$+\exp\left(-\frac{3}{416}\cdot\frac{n\epsilon^2}{\overline{C}^2}\right)\tag{27}$$

We now can further upper-bound the term $2\inf_{f\in\mathrm{F}}\|f-\mathbb{T}^\pi Q_{k-1}\|_\mu\leq 2\sup_{f'\in\mathrm{F}}\inf_{f\in\mathrm{F}}\|f-\mathbb{T}^\pi f'\|_\mu=2d_\mathrm{F}^\pi$ (the worst-case *inherent Bellman evaluation error*), leading to the final bound for the Bellman non-realizable case.

One may wish to further remove the inherent Bellman evaluation error from our error bound. However, counter-examples exist where the inherent Bellman error cannot generally be estimated using function approximation (see section 11.6 of (Sutton & Barto, 2018)). Fortunately, inherent Bellman error can be driven to be small by choosing rich function class F (low bias), at the expense of more samples requirement (higher variance, through higher pseudo-dimension $\texttt{dim}_\mathrm{F}$).

While the bound in (27) looks more complicated than the Bellman realizable case in equation 21, note that the convergence rate will still be $O(\frac{1}{n^2})$.

E.3. Bounding the error across iterations

Previous sub-sections E.2 and E.2 have analyzed the error of FQE for a single iteration in Bellman realizable and non-realizable case. We now analyze how errors from different iterations flow through the FQE algorithm. The proof borrows the idea from lemma 3 and 4 of (Munos & Szepesvári, 2008) for fitted value iteration (for value function V instead of Q), with appropriate modifications for our off-policy evaluation context.

Recall that C^π, Q^π denote the true value function and action-value function, respectively, under the evaluation policy π.

And $C_K = \mathbb{E}[Q_K(x, \pi(x))]$ denote the value function associated with the returned function Q_K from algorithm 3. Our goal is to bound the difference $C^\pi - C_K$ between the true value function and the estimated value of the returned function Q_K.

Let the unknown state-action distribution induced by the evaluation policy π be ρ. We first bound the loss $\|Q^\pi - Q_K\|_\rho$ under the "test-time "distribution ρ of (x, a), which differs from the state-action μ induced by data-generating policy π_{D}. We will then lift the loss bound from Q_K to C_K.

Step 1: Upper-bound the value estimation error

Let $\epsilon_{k-1} = Q_k - \mathbb{T}^\pi Q_{k-1} \in \mathrm{X} \times \mathrm{A}, \bar{\mathcal{C}}$. We have for every k that

$$\begin{aligned} Q^\pi - Q_k &= \mathbb{T}^\pi Q^\pi - \mathbb{T}^\pi Q_{k-1} + \epsilon_{k-1} \quad (Q^\pi \text{ is fixed point of } T^\pi) \\ &= \gamma P^\pi (Q^\pi - Q_{k-1}) + \epsilon_{k-1} \end{aligned}$$

Thus by simple recursion

$$\begin{aligned} Q^\pi - Q_K &= \sum_{k=0}^{K-1} \gamma^{K-k-1} (P^\pi)^{K-k-1} \epsilon_k + \gamma^K (P^\pi)^K (Q^\pi - Q_0) \\ &= \frac{1-\gamma^{K+1}}{1-\gamma} \left[\sum_{k=0}^{K-1} \frac{(1-\gamma)\gamma^{K-k-1}}{1-\gamma^{K+1}} (P^\pi)^{K-k-1} \epsilon_k + \frac{(1-\gamma)\gamma^K}{1-\gamma^{K+1}} (P^\pi)^K (Q^\pi - Q_0) \right] \\ &= \frac{1-\gamma^{K+1}}{1-\gamma} \left[\sum_{k=0}^{K-1} \alpha_k A_k \epsilon_k + \alpha_K A_K (Q^\pi - Q_0) \right] \end{aligned} \tag{28}$$

where for simplicity of notations, we denote

$$\begin{aligned} \alpha_k &= \frac{(1-\gamma)\gamma^{K-k-1}}{1-\gamma^{K+1}} \text{ for } k < K, \alpha_K = \frac{(1-\gamma)\gamma^K}{1-\gamma^{K+1}} \\ A_k &= (P^\pi)^{K-k-1}, A_K = (P^\pi)^K \end{aligned}$$

Note that A_k's are probability kernels and α_k's are deliberately chosen such that $\sum_k \alpha_k = 1$.

We can apply point-wise absolute value on both sides of (28) with $|f|$ being the short-hand notation for $|f(x, a)|$ and inequality holds point-wise. By triangle inequalities:

$$|Q^\pi - Q_K| \leq \frac{1-\gamma^{K+1}}{1-\gamma} \left[\sum_{k=0}^{K-1} \alpha_k A_k |\epsilon_k| + \alpha_K A_K |Q^\pi - Q_0| \right] \tag{29}$$

Step 2: Bounding $\|Q^\pi - Q_K\|_\rho$ for any unknown distribution ρ. To handle distribution shift from μ to ρ, we decompose the loss as follows:

$$\begin{aligned} \|Q^\pi - Q_K\|_\rho^2 &= \int \rho(dx, da) \left(Q^\pi(x, a) - Q_K(x, a)\right)^2 \\ &\leq \left[\frac{1-\gamma^{K+1}}{1-\gamma}\right]^2 \int \rho(dx, da) \left[\left(\sum_{k=0}^{K-1} \alpha_k A_k |\epsilon_k| + \alpha_K A_K |Q^\pi - Q_0| \right)(x, a) \right]^2 \text{ (from(29))} \\ &\leq \left[\frac{1-\gamma^{K+1}}{1-\gamma}\right]^2 \int \rho(dx, da) \left[\sum_{k=0}^{K-1} \alpha_k (A_k \epsilon_k)^2 + \alpha_K (A_K (Q^* - Q_0))^2 \right](x, a) \text{ (Jensen)} \\ &\leq \left[\frac{1-\gamma^{K+1}}{1-\gamma}\right]^2 \int \rho(dx, da) \left[\sum_{k=0}^{K-1} \alpha_k A_k \epsilon_k^2 + \alpha_K A_K (Q^* - Q_0)^2 \right](x, a) \text{ (Jensen)} \end{aligned}$$

Using assumption 3 (assumption 1 of the main paper), we can bound each term ρA_k as

$$\rho A_k = \rho (P^\pi)^{K-k-1} \leq \mu \beta_\mu (K-k-1) \text{ (definition D.3)}$$

Thus

$$\|Q^\pi - Q_K\|_\rho^2 \leq \left[\frac{1-\gamma^{K+1}}{1-\gamma}\right]^2 \left[\frac{1}{1-\gamma^{K+1}} \sum_{k=0}^{K-1} (1-\gamma)\gamma^{K-k-1} \beta_\mu (K-k-1) \|\epsilon_k\|_\mu^2 + \alpha_K (2\overline{C})^2 \right]$$

Assumption 3 (stronger than necessary for proof of FQE) can be used to upper-bound the first order concentration coefficient:

$$(1-\gamma)\sum_{m\geq 0}\gamma^m\beta_\mu(m) \leq \frac{\gamma}{1-\gamma}\left[(1-\gamma)^2\sum_{m\geq 1}m\gamma^{m-1}\beta_\mu(m)\right] = \frac{\gamma}{1-\gamma}\beta_\mu$$

This gives the upper-bound for $\|Q^\pi - Q_K\|_\rho^2$ as

$$\begin{aligned}\|Q^\pi - Q_K\|_\rho^2 &\leq \left[\frac{1-\gamma^{K+1}}{1-\gamma}\right]^2\left[\frac{\gamma}{(1-\gamma)(1-\gamma^{K+1})}\beta_\mu\max_k\|\epsilon_k\|_\mu^2 + \frac{(1-\gamma)\gamma^K}{1-\gamma^{K+1}}(2\overline{C})^2\right]\\ &\leq \frac{1-\gamma^{K+1}}{(1-\gamma)^2}\left[\frac{\gamma}{1-\gamma}\beta_\mu\max_k\|\epsilon_k\|_\mu^2 + (1-\gamma)\gamma^K(2\overline{C})^2\right]\\ &\leq \frac{\gamma}{(1-\gamma)^3}\beta_\mu\max_k\|\epsilon_k\|_\mu^2 + \frac{\gamma^K}{1-\gamma}(2\overline{C})^2\end{aligned}$$

Using $a^2+b^2 \leq (a+b)^2$ for nonnegative a, b, we conclude that

$$\|Q^\pi - Q_K\|_\rho \leq \frac{\gamma^{1/2}}{(1-\gamma)^{3/2}}\left(\sqrt{\beta_\mu}\max_k\|\epsilon_k\|_\mu + \frac{\gamma^{K/2}}{(1-\gamma)^{1/2}}2\overline{C}\right) \tag{30}$$

Step 3: Turning error bound from Q to $|C^\pi - C_K|$ Now we can choose ρ to be the state-action distribution by the evaluation policy π. The error bound on the value function C follows simply by integrating inequality (30) over state-action pairs induced by π. The final error across iterations can be related to individual iteration error by

$$|C^\pi - C_K| \leq \frac{\gamma^{1/2}}{(1-\gamma)^{3/2}}\left(\sqrt{\beta_\mu}\max_k\|\epsilon_k\|_\mu + \frac{\gamma^{K/2}}{(1-\gamma)^{1/2}}2\overline{C}\right) \tag{31}$$

E.4. Finite-sample guarantees for Fitted Q Evaluation

Combining results from (21), (27) and (31), we have the final guarantees for FQE under both realizable and general cases.

Realizable Case - Proof of theorem E.2. From (21), when $n \geq \frac{24\cdot 214\cdot\overline{C}^4}{\epsilon^2}\left(\log\frac{K}{\delta} + \mathtt{dim}_\mathrm{F}\log\frac{320\overline{C}^2}{\epsilon^2} + \log(14e(\mathtt{dim}_\mathrm{F}+1))\right)$, we have $\|\epsilon_k\|_\mu < \epsilon$ with probability at least $1-\delta/K$ for any $0 \leq k < K$. Thus we conclude that for any $\epsilon > 0, 0 < \delta < 1$, after K iterations of Fitted Q Evaluation, the value estimate returned by Q_K satisfies:

$$|C^\pi - C_K| \leq \frac{\gamma^{1/2}}{(1-\gamma)^{3/2}}\left(\sqrt{\beta_\mu}\epsilon + \frac{\gamma^{K/2}}{(1-\gamma)^{1/2}}2\overline{C}\right)$$

holds with probability $1-\delta$ when $n \geq \frac{24\cdot 214\cdot\overline{C}^4}{\epsilon^2}\left(\log\frac{K}{\delta} + \mathtt{dim}_\mathrm{F}\log\frac{320\overline{C}^2}{\epsilon^2} + \log(14e(\mathtt{dim}_\mathrm{F}+1))\right)$. This concludes the proof of theorem E.2.

Non-realizable Case - Proof of theorem E.1 and theorem 4.2 of main paper. Similarly, from (27) we have

$$\begin{aligned}\mathbf{P}\left\{\|Q_k - \mathbb{T}^\pi Q_{k-1}\|_\mu - 2\inf_{f\in\mathrm{F}}\|f - \mathbb{T}^\pi Q_{k-1}\|_\mu > \epsilon\right\} \leq 14\cdot e\cdot(\mathtt{dim}_\mathrm{F}+1)\left(\frac{640\overline{C}^2}{\epsilon^2}\right)^{\mathtt{dim}_\mathrm{F}}\cdot\exp\left(-\frac{n\epsilon^2}{48\cdot 214\overline{C}^4}\right)\\ + \exp\left(-\frac{3}{416}\cdot\frac{n\epsilon^2}{\overline{C}^2}\right)\end{aligned}$$

Since $\inf_{f\in\mathrm{F}}\|f - \mathbb{T}^\pi Q_{k-1}\|_\mu \leq \sup_{h\in\mathrm{F}}\inf_{f\in\mathrm{F}}\|f - \mathbb{T}^\pi h\|_\mu = d_\mathrm{F}^\pi$ (the *inherent Bellman evaluation error*), similar arguments to the realizable case lead to the conclusion that for any $\epsilon > 0, 0 < \delta < 1$, after K iterations of FQE:

$$|C^\pi - C_K| \leq \frac{\gamma^{1/2}}{(1-\gamma)^{3/2}}\left(\sqrt{\beta_\mu}(2d_\mathrm{F}^\pi + \epsilon) + \frac{\gamma^{K/2}}{(1-\gamma)^{1/2}}2\overline{C}\right)$$

holds with probability $1-\delta$ when $n = O\big(\frac{\overline{C}^4}{\epsilon^2}(\log\frac{K}{\delta} + \mathtt{dim}_\mathrm{F}\log\frac{\overline{C}^2}{\epsilon^2} + \log\mathtt{dim}_\mathrm{F})\big)$, thus finishes the proof of theorem E.1.

Note that in both cases, the $\widetilde{O}(\frac{1}{\epsilon^2})$ dependency of n is significant improvement over previous finite-sample analysis of related approximate dynamic programming algorithms (Munos & Szepesvári, 2008; Antos et al., 2008b;a). This dependency matches that of previous analysis using linear function approximators from (Lazaric et al., 2012; 2010) for LSTD and LSPI algorithms. Here our analysis, using similar assumptions, is applicable for general non-linear, bounded function classes. , which is an improvement over convergence rate of $O(\frac{1}{n^4})$ in related approximate dynamic programming algorithms (Antos et al., 2008a;b; Munos & Szepesvári, 2008).

F. Finite-Sample Analysis of Fitted Q Iteration (FQI)

F.1. Algorithm and Discussion

Algorithm 4 Fitted Q Iteration with Function Approximation: $\texttt{FQI}(c)$ (Ernst et al., 2005)

Input: Collected data set $\mathrm{D} = \{x_i, a_i, x'_i, c_i\}_{i=1}^n$. Function class F
1: Initialize $Q_0 \in \mathrm{F}$ randomly
2: **for** $k = 1, 2, \ldots, K$ **do**
3: Compute target $y_i = c_i + \gamma \min_a Q_{k-1}(x'_i, a) \;\; \forall i$
4: Build training set $\widetilde{\mathrm{D}}_k = \{(x_i, a_i), y_i\}_{i=1}^n$
5: Solve a supervised learning problem:
$Q_k = \arg\min_{f \in \mathrm{F}} \frac{1}{n} \sum_{i=1}^n (f(x_i, a_i) - y_i)^2$
6: **end for**
Output: $\pi_K(\cdot) = \arg\min_a Q_K(\cdot, a)$ (greedy policy with respect to the returned function Q_K)

The analysis of FQI (algorithm 4) follows analogously from the analysis of FQE from the previous section (Appendix E). For brevity, we skip certain detailed derivations, especially those that are largely identical to FQE's analysis.

To the best of our knowledge, a finite-sample analysis of FQI with general non-linear function approximation has not been published (Continuous FQI from (Antos et al., 2008a) is in fact a Fitted Policy Iteration algorithm and is different from algo 4). In principle, one can adapt existing analysis of fitted value iteration (Munos & Szepesvári, 2008) and FittedPolicyQ (Antos et al., 2008b;a) to show that under similar assumptions, among policies greedy w.r.t. functions in F, FQI will find $\epsilon-$ optimal policy using $n = \widetilde{O}(\frac{1}{\epsilon^4})$ samples. We derive an improved analysis of FQI with general non-linear function approximations, with better sample complexity of $n = \widetilde{O}(\frac{1}{\epsilon^2})$. We note that the appendix of (Lazaric & Restelli, 2011) contains an analysis of LinearFQI showing similar rate to ours, albeit with linear function approximators.

In this section, we prove the following statement:

Theorem F.1 (Guarantee for FQI - General Case (theorem 4.3 in main paper))**.** *Under Assumption 3, for any $\epsilon > 0, \delta \in (0, 1)$, after K iterations of Fitted Q Iteration (algorithm 4), for $n = O\big(\frac{\overline{C}^4}{\epsilon^2}(\log \frac{K}{\delta} + \texttt{dim}_\mathrm{F} \log \frac{\overline{C}^2}{\epsilon^2} + \log \texttt{dim}_\mathrm{F})\big)$, we have with probability $1 - \delta$:*

$$C^* - C(\pi_K) \leq \frac{2\gamma}{(1-\gamma)^3}\left(\sqrt{\beta_\mu}\,(2d_\mathrm{F} + \epsilon) + 2\gamma^{K/2}\overline{C}\right)$$

where π_K is the policy greedy with respect to the returned function Q_K, and C^* is the value of the optimal policy.

The key steps to the proof follow similar scheme to the proof of FQE. We first bound the error for each iteration, and then analyze how the errors flow through the algorithm.

F.2. Single iteration error bound $\|Q_k - \mathbb{T}Q_{k-1}\|_\mu$

Here μ is the state-action distribution induced by the data-generating policy π_D.

We begin with the decomposition:

$$\|Q_k - \mathbb{T}Q_{k-1}\|_\mu^2 = \mathbb{E}\left[(Q_k(x,a) - y)^2\right] - \mathbb{E}\left[(\mathbb{T}Q_{k-1}(x,a) - y)^2\right]$$

$$= \left\{\mathbb{E}\left[(Q_k(x,a) - y)^2\right] - \mathbb{E}\left[(\mathbb{T}Q_{k-1}(x,a) - y)^2\right] - 2 \cdot \left(\frac{1}{n}\sum_{i=1}^n (Q_k(x_i,a_i) - y_i)^2 - \frac{1}{n}\sum_{i=1}^n (\mathbb{T}Q_{k-1}(x_i,a_i) - y_i)^2\right)\right\}$$

$$+ \left\{2 \cdot \left(\frac{1}{n}\sum_{i=1}^n (Q_k(x_i,a_i) - y_i)^2 - \frac{1}{n}\sum_{i=1}^n (\mathbb{T}Q_{k-1}(x_i,a_i) - y_i)^2\right)\right\}$$

$$= \texttt{component\_1} + \texttt{component\_2}$$

For $\mathbb{T}$ the Bellman (optimality) operator (equation 16), $\mathbb{T}Q_{k-1}$ is the *regression function* that minimizes square loss $\min_{h:\mathbb{R}^{X\times A}\mapsto\mathbb{R}} \mathbb{E}|h(x,a) - y|^2$, with the random variables $(x,a) \sim \mu$ and $y = c(x,a) + \gamma \min_{a'} Q_{k-1}(x', a')$ where $x' \sim p(x'|x,a)$. Invoking lemma E.5 and following the steps similar to equations (18),(19),(20) and (21) from appendix E, we

can bound the first component as

$$\mathbf{P}(\texttt{component\_1} > \epsilon/2) \leq 14 \cdot e \cdot (\texttt{dim}_{\mathrm{F}} + 1) \left(\frac{640\overline{C}^2}{\epsilon}\right)^{\texttt{dim}_{\mathrm{F}}} \cdot \exp\left(-\frac{n\epsilon}{48 \cdot 214\overline{C}^4}\right) \tag{32}$$

Let $f^* = \arg\inf_{f \in \mathrm{F}} \|f - \mathbb{T}Q_{k-1}\|_\mu^2$. Since $Q_k = \arg\min_{f \in \mathrm{F}} \frac{1}{n}\sum_{i=1}^n (f(x_i, a_i) - y_i)^2$, we can upper-bound `component_2` by

$$\texttt{component\_2} \leq 2 \cdot \left(\frac{1}{n}\sum_{i=1}^n (f^*(x_i, a_i) - y_i)^2 - \frac{1}{n}\sum_{i=1}^n (\mathbb{T}Q_{k-1}(x_i, a_i) - y_i)^2\right)$$

Let random variable $z = ((x, a), y)$, $z_i = ((x_i, a_i), y_i), i = 1, \ldots, n$ and let

$$h(z) = (f^*(x, a) - y)^2 - (\mathbb{T}Q_{k-1}(x, a) - y)^2$$

We have $|h(z)| \leq 4\overline{C}^2$. We can derive a bound for $\mathbf{P}\left(\frac{1}{n}\sum_{i=1}^n h(z_i) - \mathbb{E}h(z) > \frac{\epsilon}{4} + \mathbb{E}h(z)\right)$ using Bernstein inequality, similar to equations (23) and (24) from appendix E to obtain:

$$\mathbf{P}\left(2 \cdot \left[\frac{1}{n}\sum_{i=1}^n h(z_i) - 2\mathbb{E}h(z)\right] > \frac{\epsilon}{2}\right) \leq \exp\left(-\frac{3}{416} \cdot \frac{n\epsilon}{\overline{C}^2}\right) \tag{33}$$

Now we have

$$\texttt{component\_2} \leq 2 \cdot \frac{1}{n}\sum_{i=1}^n h(z_i) = 2 \cdot \left[\frac{1}{n}\sum_{i=1}^n h(z_i) - 2\mathbb{E}h(z)\right] + 4\mathbb{E}h(z)$$

Since

$$\begin{aligned}\mathbb{E}h(z) &= \mathbb{E}_{\widetilde{D}_k}\left[(f^*(x, a) - y)^2\right] - \mathbb{E}_{\widetilde{D}_k}\left[(\mathbb{T}Q_{k-1}(x, a) - y)^2\right] = \mathbb{E}_{\widetilde{D}_k}\left[(f^*(x, a) - \mathbb{T}Q_{k-1}(x, a))^2\right] \\ &= \inf_{f \in \mathrm{F}} \|f - \mathbb{T}Q_{k-1}\|_\mu^2 \end{aligned} \tag{34}$$

Combining equations (32), (33) and (34), we obtain that

$$\begin{aligned}\mathbf{P}\left\{\|Q_k - \mathbb{T}Q_{k-1}\|_\mu^2 - 4\inf_{f \in \mathrm{F}} \|f - \mathbb{T}Q_{k-1}\|_\mu^2 > \epsilon\right\} &\leq 14 \cdot e \cdot (\texttt{dim}_{\mathrm{F}} + 1)\left(\frac{640\overline{C}^2}{\epsilon}\right)^{\texttt{dim}_{\mathrm{F}}} \cdot \exp\left(-\frac{n\epsilon}{48 \cdot 214\overline{C}^4}\right) \\ &\quad + \exp\left(-\frac{3}{416} \cdot \frac{n\epsilon}{\overline{C}^2}\right)\end{aligned} \tag{35}$$

F.3. Propagation of error bound for $\|Q^* - Q^{\pi_K}\|_\rho$

The analysis of error propagation for FQI is more involved than that of FQE, but the proof largely follows the error propagation analysis in lemma 3 and 4 of (Munos & Szepesvári, 2008) in the fitted value iteration context (for V function). We include the Q function's (slighly more complicated) derivation here for completeness.

Recall that π_K is greedy wrt the learned function Q_K returned by FQI. We aim to bound the difference $C^* - C^{\pi_K}$ between the optimal value function and that π_K. For a (to-be-specified) distribution ρ of state-action pairs (different from the data distribution μ), we bound the generalization loss $\|Q^* - Q^{\pi_K}\|_\rho$

Step 1: Upper-bound the propagation error (value). Let $\epsilon_{k-1} = Q_k - \mathbb{T}Q_{k-1}$. We have that

$$\begin{aligned}Q^* - Q_k = \mathbb{T}^{\pi^*}Q^* - \mathbb{T}^{\pi^*}Q_{k-1} + \mathbb{T}^{\pi^*}Q_{k-1} - \mathbb{T}Q_{k-1} + \epsilon_{k-1} &\leq \mathbb{T}^{\pi^*}Q^* - \mathbb{T}^{\pi^*}Q_{k-1} + \epsilon_{k-1} \text{ (b/c } \mathbb{T}Q_{k-1} \geq \mathbb{T}^{\pi^*}Q_{k-1}) \\ &= \gamma P^{\pi^*}(Q^* - Q_{k-1}) + \epsilon_{k-1}\end{aligned}$$

Thus by recursion $Q^* - Q_K \leq \sum_{k=0}^{K-1} \gamma^{K-k-1}(P^{\pi^*})^{K-k-1}\epsilon_k + \gamma^K(P^{\pi^*})^K(Q^* - Q_0)$

Step 2: Lower-bound the propagation error (value). Similarly

$$\begin{aligned}Q^* - Q_k = \mathbb{T}Q^* - \mathbb{T}^{\pi_{k-1}}Q^* + \mathbb{T}^{\pi_{k-1}}Q^* - \mathbb{T}Q_{k-1} + \epsilon_{k-1} &\geq \mathbb{T}^{\pi_{k-1}}Q^* - \mathbb{T}Q_{k-1} + \epsilon_{k-1} \text{ (as } \mathbb{T}Q^* \geq \mathbb{T}^{\pi_{k-1}}Q^*) \\ &\geq \mathbb{T}^{\pi_{k-1}}Q^* - \mathbb{T}^{\pi_{k-1}}Q_{k-1} + \epsilon_{k-1} \text{ (b/c } \pi_{k-1} \textit{ greedy wrt } Q_{k-1}) \\ &= \gamma P^{\pi_{k-1}}(Q^* - Q_{k-1}) + \epsilon_{k-1}\end{aligned}$$

And by recursion $Q^* - Q_K \geq \sum_{k=0}^{K-1} \gamma^{K-k-1}(P^{\pi_{K-1}}P^{\pi_{K-2}}\ldots P^{\pi_{k+1}})\epsilon_k + \gamma^K(P^{\pi_{K-1}}P^{\pi_{K-2}}\ldots P^{\pi_0})(Q^* - Q_0)$

Step 3: Upper-bound the propagation error (policy). Beginning with a decomposition of value wrt to policy π_K

$$\begin{aligned}
Q^* - Q^{\pi_K} &= \mathbb{T}^{\pi^*}Q^* - \mathbb{T}^{\pi^*}Q_K + \mathbb{T}^{\pi^*}Q_K - \mathbb{T}^{\pi_K}Q_K + \mathbb{T}^{\pi_K}Q_K - \mathbb{T}^{\pi_K}Q^{\pi_K} \\
&\leq (\mathbb{T}^{\pi^*}Q^* - \mathbb{T}^{\pi^*}Q_K) + (\mathbb{T}^{\pi_K}Q_K - \mathbb{T}^{\pi_K}Q^{\pi_K}) \qquad (\text{ since } \mathbb{T}^{\pi^*}Q_K \leq \mathbb{T}Q_K = \mathbb{T}^{\pi_K}Q_K) \\
&= \gamma P^{\pi^*}(Q^* - Q_K) + \gamma P^{\pi_K}(Q_K - Q^{\pi_K}) \\
&= \gamma P^{\pi^*}(Q^* - Q_K) + \gamma P^{\pi_K}(Q_K - Q^* + Q^* - Q^{\pi_K})
\end{aligned}$$

Thus leading to $(I - \gamma P^{\pi_K})(Q^* - Q^{\pi_K}) \leq \gamma(P^{\pi^*} - P^{\pi_K})(Q^* - Q_K)$ The operator $(I - \gamma P^{\pi_K})$ is invertible and $(I - \gamma P^{\pi_K})^{-1} = \sum_{m\geq 0}\gamma^m (P^{\pi_K})^m$ is monotonic. Thus

$$\begin{aligned}
Q^* - Q^{\pi_K} &\leq \gamma(I - \gamma P^{\pi_K})^{-1}(P^{\pi^*} - P^{\pi_K})(Q^* - Q_K) \\
&= \gamma(I - \gamma P^{\pi_K})^{-1}P^{\pi^*}(Q^* - Q_K) - \gamma(I - \gamma P^{\pi_K})^{-1}P^{\pi_K}(Q^* - Q_K) \qquad (36)
\end{aligned}$$

Applying inequalities from Step 1 and Step 2 to the RHS of (36), we have

$$\begin{aligned}
Q^* - Q^{\pi_K} \leq (I - \gamma P^{\pi_K})^{-1}\Bigg[&\sum_{k=0}^{K-1}\gamma^{K-k}\left((P^{\pi^*})^{K-k} - P^{\pi_K}P^{\pi_{K-1}}\dots P^{\pi_{k+1}}\right)\epsilon_k \\
&+ \gamma^{K+1}\left((P^{\pi^*})^{K+1} - (P^{\pi_K}P^{\pi_{K-1}}\dots P^{\pi_0})\right)(Q^* - Q_0)\Bigg] \qquad (37)
\end{aligned}$$

Next we apply point-wise absolute value on RHS of (37), with $|\epsilon_k|$ being the short-hand notation for $|\epsilon_k(x,a)|$ point-wise. Using triangle inequalities and rewriting (37) in a more compact form ((Munos & Szepesvári, 2008)):

$$Q^* - Q^{\pi_K} \leq \frac{2\gamma(1-\gamma^{K+1})}{(1-\gamma)^2}\left[\sum_{k=0}^{K-1}\alpha_k A_k|\epsilon_k| + \alpha_K A_K|Q^* - Q_0|\right]$$

where $\alpha_k = \frac{(1-\gamma)\gamma^{K-k-1}}{1-\gamma^{K+1}}$ for $k < K, \alpha_K = \frac{(1-\gamma)\gamma^K}{1-\gamma^{K+1}}$ and

$$\begin{aligned}
A_k &= \frac{1-\gamma}{2}(I - \gamma P^{\pi_K})^{-1}\left[(P^{\pi^*})^{K-k} + P^{\pi_K}P^{\pi_{K-1}}\dots P^{\pi_{k+1}}\right] \text{ for } k < K \\
A_K &= \frac{1-\gamma}{2}(I - \gamma P^{\pi_K})^{-1}\left[(P^{\pi^*})^{K+1} + P^{\pi_K}P^{\pi_{K-1}}\dots P^{\pi_0}\right]
\end{aligned}$$

Note that A_k's are probability kernels that combine the P^{π_i} terms and α_k's are chosen such that $\sum_k \alpha_k = 1$.

Step 4: Bounding $\|Q^* - Q^{\pi_K}\|_\rho^2$ for any test distribution ρ.

This step handles distribution shift from μ to ρ (similar to Step 2 from sub-section E.3 of appendix E)

$$\|Q^* - Q^{\pi_K}\|_\rho^2 \leq \left[\frac{2\gamma(1-\gamma^{K+1})}{(1-\gamma)^2}\right]^2 \int \rho(dx,da)\left[\sum_{k=0}^{K-1}\alpha_k A_k \epsilon_k^2 + \alpha_K A_K (Q^* - Q_0)^2\right](x,a) \text{ (twice Jensen)}$$

Using assumption 3 (assumption 1 in the main paper), each term ρA_k is bounded as

$$\begin{aligned}
\rho A_k &= \frac{1-\gamma}{2}\rho(I - \gamma P^{\pi_K})^{-1}\left[(P^{\pi^*})^{K-k} + P^{\pi_K}P^{\pi_{K-1}}\dots P^{\pi_{k+1}}\right] \\
&= \frac{1-\gamma}{2}\sum_{m\geq 0}\gamma^m \rho(P^{\pi_K})^m\left[(P^{\pi^*})^{K-k} + P^{\pi_K}P^{\pi_{K-1}}\dots P^{\pi_{k+1}}\right] \leq (1-\gamma)\sum_{m\geq 0}\gamma^m\beta_\mu(m+K-k)\mu \qquad (\text{def D.3})
\end{aligned}$$

Thus

$$\begin{aligned}
\|Q^* - Q^{\pi_K}\|_\rho^2 &\leq \left[\frac{2\gamma(1-\gamma^{K+1})}{(1-\gamma)^2}\right]^2\left[\frac{1}{1-\gamma^{K+1}}\sum_{k=0}^{K-1}(1-\gamma)^2\sum_{m\geq 0}\gamma^{m+K-k-1}\beta_\mu(m+K-k)\|\epsilon_k\|_\mu^2 + \alpha_K(2\overline{C})^2\right] \\
&\leq \left[\frac{2\gamma(1-\gamma^{K+1})}{(1-\gamma)^2}\right]^2\left[\frac{1}{1-\gamma^{K+1}}\beta_\mu\max_k\|\epsilon_k\|_\mu^2 + \frac{(1-\gamma)\gamma^K}{1-\gamma^{K+1}}(2\overline{C})^2\right] \qquad (\text{assumption 3}) \\
&\leq \left[\frac{2\gamma(1-\gamma^{K+1})}{(1-\gamma)^2}\right]^2\left[\frac{1}{1-\gamma^{K+1}}\beta_\mu\max_k\|\epsilon_k\|_\mu^2 + \frac{\gamma^K}{1-\gamma^{K+1}}(2\overline{C})^2\right] \\
&\leq \left[\frac{2\gamma}{(1-\gamma)^2}\right]^2\left[\beta_\mu\max_k\|\epsilon_k\|_\mu^2 + \gamma^K(2\overline{C})^2\right]
\end{aligned}$$

Using $a^2 + b^2 \leq (a+b)^2$ for nonnegative a, b, we thus conclude that

$$\|Q^* - Q^{\pi_K}\|_\rho \leq \frac{2\gamma}{(1-\gamma)^2}\left(\sqrt{\beta_\mu}\max_k \|\epsilon_k\|_\mu + 2\gamma^{K/2}\overline{C}\right) \tag{38}$$

Step 5: Bounding $C^* - C^{\pi_K}$ Using the performance difference lemma (lemma 6.1 of (Kakade & Langford, 2002), which states that $C^* - C^{\pi_K} = -\frac{1}{1-\gamma}\mathbb{E}_{\substack{x\sim d_{\pi_K}\\ a\sim\pi_K}} A^*[x,a]$. We can upper-bound the performance difference of value function as

$$\begin{aligned}
C^* - C^{\pi_K} &= \frac{1}{1-\gamma}\mathbb{E}_{\substack{x\sim d_{\pi_K}\\ a\sim\pi_K}}[C^*(x) - Q^*(x,a)] = \frac{1}{1-\gamma}\mathbb{E}_{x\sim d_{\pi_K}}[C^*(x) - Q^*(x,\pi_K(x))] \\
&\leq \frac{1}{1-\gamma}\mathbb{E}_{x\sim d_{\pi_K}}[Q^*(x,\pi^*(x)) - Q_K(x,\pi^*(x)) + Q_K(x,\pi_K(x) - Q^*(x,\pi_K(x))] \text{ (greedy)} \\
&\leq \frac{1}{1-\gamma}\mathbb{E}_{x\sim d_{\pi_K}}|Q^*(x,\pi^*(x)) - Q_K(x,\pi^*(x))| + |Q_K(x,\pi_K(x) - Q^*(x,\pi_K(x))| \\
&\leq \frac{1}{1-\gamma}\left(\|Q^* - Q^{\pi_K}\|_{d_{\pi_K}\times\pi^*} + \|Q^* - Q^{\pi_K}\|_{d_{\pi_K}\times\pi_K}\right) \text{ (upper-bound 1-norm by 2-norm)} \\
&\leq \frac{2\gamma}{(1-\gamma)^3}\left(\sqrt{\beta_\mu}\max_k \|\epsilon_k\|_\mu + 2\gamma^{K/2}\overline{C}\right)
\end{aligned} \tag{39}$$

Note that inequality (39) follows from (38) by specifying $\rho = \chi P^{\pi_K} P^{\pi^*}$ and $\rho = \chi P^{\pi_K} P^{\pi_K}$, respectively ($\chi$ is the initial state distribution).

F.4. Finite-sample guarantees for Fitted Q Iteration

From (35) we have:

$$\begin{aligned}
\mathbf{P}\left\{\|Q_k - \mathbb{T}Q_{k-1}\|_\mu - 2\inf_{f\in\mathrm{F}}\|f - \mathbb{T}Q_{k-1}\|_\mu > \epsilon\right\} \leq 14\cdot e\cdot(\mathtt{dim}_\mathrm{F}+1)\left(\frac{640\overline{C}^2}{\epsilon^2}\right)^{\mathtt{dim}_\mathrm{F}}\cdot\exp\left(-\frac{n\epsilon^2}{48\cdot 214\overline{C}^4}\right) \\
+\exp\left(-\frac{3}{416}\cdot\frac{n\epsilon^2}{\overline{C}^2}\right)
\end{aligned}$$

Note that $\inf_{f\in\mathrm{F}}\|f - \mathbb{T}Q_{k-1}\|_\mu \leq \sup_{h\in\mathrm{F}}\inf_{f\in\mathrm{F}}\|f - \mathbb{T}h\|_\mu = d_\mathrm{F}$ (the *inherent Bellman error* from equation 16). Combining with equation (39), we have the conclusion that for any $\epsilon > 0, 0 < \delta < 1$, after K iterations of Fitted Q Iteration, and for π_K the greedy policy wrt Q_K:

$$C^* - C_K^\pi \leq \frac{2\gamma}{(1-\gamma)^3}\left(\sqrt{\beta_\mu}(2d_\mathrm{F}+\epsilon) + 2\gamma^{K/2}\overline{C}\right)$$

holds with probability $1-\delta$ when $n = O\big(\frac{\overline{C}^4}{\epsilon^2}(\log\frac{K}{\delta} + \mathtt{dim}_\mathrm{F}\log\frac{\overline{C}^2}{\epsilon^2} + \log\mathtt{dim}_\mathrm{F})\big)$.

Note that compared to the Fitted Value Iteration analysis of (Munos & Szepesvári, 2008), our error includes an extra factor 2 for d_F.

F.5. Statement for the Bellman-realizable Case

To facilitate the end-to-end generalization analysis of theorem 4.4 in the main paper, we include a version of FQI analysis under Bellman-realizable assumption in this section. The theorem is a consequence of previous analysis in this section.

Assumption 5 (Bellman evaluation realizability)**.** *We consider function classes* F *sufficiently rich so that* $\forall f, \mathbb{T}f \in \mathrm{F}$.

Theorem F.2 (Guarantee for FQI - Bellman-realizable Case)**.** *Under Assumption 3 and 5, for any* $\epsilon > 0, \delta \in (0,1)$*, after* K *iterations of Fitted Q Iteration, for* $n \geq \frac{24\cdot 214\cdot\overline{C}^4}{\epsilon^2}\big(\log\frac{K}{\delta} + \mathtt{dim}_\mathrm{F}\log\frac{320\overline{C}^2}{\epsilon^2} + \log(14e(\mathtt{dim}_\mathrm{F}+1))\big)$*, we have with probability* $1-\delta$*:*

$$C^* - C(\pi_K) \leq \frac{2\gamma}{(1-\gamma)^3}\big(\sqrt{\beta_\mu}\epsilon + 2\gamma^{K/2}\overline{C}\big)$$

where π_K *is the policy greedy with respect to the returned function* Q_K*, and* C^* *is the value of the optimal policy.*

G. Additional Instantiation of Meta-Algorithm (algorithm 1)

We provide an additional instantiation of the meta-algorithm described in the main paper, with Online Gradient Descent (OGD) (Zinkevich, 2003) and Least-Squares Policy Iteration (LSPI) (Lagoudakis & Parr, 2003) as subroutines. Using LSPI requires a feature map ϕ such that any state-action pair can be represented by k features. The value function is linear in parameters represented by ϕ. Policy representation is simplified to a weight vector $w \in \mathbb{R}^k$.

Similar to our main algorithm 2, OGD updates require bounded parameters λ. We thus introduce hyper-parameter B as the bound of λ in ℓ_2 norm. The gradient update is projected to the ℓ_2 ball when the norm of λ exceeds B (line 15 of algo 5).

Algorithm 5 Batch Learning under Constraints using Online Gradient Descent and Least-Squares Policy Iteration

Input: Dataset $\mathrm{D} = \{x_i, a_i, x'_i, c_i, g_i\}_{i=1}^n \sim \pi_{\mathrm{D}}$. Online algorithm parameters: ℓ_2 norm bound B, learning rate η
Input: Number of basis function k. Basis function ϕ (feature map for state-action pairs)
1: Initialize $\lambda_1 = (0, \dots, 0) \in \mathbb{R}^m$
2: **for** each round t **do**
3: Learn $w_t \leftarrow \texttt{LSPI}(c + \lambda_t^\top g)$ *// LSPI with cost $c + \lambda_t^\top g$*
4: Evaluate $\widehat{C}(w_t) \leftarrow \texttt{LSTDQ}(w_t, c)$ *// Algo 7 with π_t, cost c*
5: Evaluate $\widehat{G}(w_t) \leftarrow \texttt{LSTDQ}(w_t, g)$ *// Algo 7 with π_t,* cost g
6: $\widehat{w}_t \leftarrow \frac{1}{t}\sum_{t'=1}^t w_{t'}$
7: $\widehat{C}(\widehat{w}_t) \leftarrow \frac{1}{t}\sum_{t'=1}^t \widehat{C}(w_{t'}), \widehat{G}(\widehat{w}_t) \leftarrow \frac{1}{t}\sum_{t'=1}^t \widehat{G}(w_{t'})$
8: $\widehat{\lambda}_t \leftarrow \frac{1}{t}\sum_{t'=1}^t \lambda_{t'}$
9: Learn $\widetilde{w} \leftarrow \texttt{LSPI}(c + \widehat{\lambda}_t^\top g)$ *// LSPI with cost $c + \widehat{\lambda}_t^\top g$*
10: Evaluate $\widehat{C}(\widetilde{w}) \leftarrow \texttt{LSTDQ}(\widetilde{w}, c), \widehat{G}(\widetilde{w}) \leftarrow \texttt{LSTDQ}(\widetilde{w}, g)$
11: $\widehat{\mathrm{L}}_{\max} = \max_{\lambda, \|\lambda\|_2 \le B} \left(\widehat{C}(\widehat{w}_t) + \lambda^\top (\widehat{G}(\widehat{w}_t) - \tau)\right)$
12: $\widehat{\mathrm{L}}_{\min} = \widehat{C}(\widetilde{w}) + \widehat{\lambda}_t^\top (\widehat{G}(\widetilde{w}) - \tau)$
13: **if** $\widehat{\mathrm{L}}_{\max} - \widehat{\mathrm{L}}_{\min} \le \omega$ **then**
14: Return $\widehat{\pi}_t$ greedy w.r.t $\widehat{w}_t$ $\left(\text{i.e., } \widehat{\pi}_t(x) = \arg\min_{a \in \mathrm{A}} \widehat{w}_t^\top \phi(x, a) \ \ \forall x\right)$
15: **end if**
16: $\lambda_{t+1} = \mathcal{P}(\lambda_t - \eta(\widehat{G}(\pi_t) - \tau))$ where projection $\mathcal{P}(\lambda) = B\frac{\lambda}{\max\{B, \|\lambda\|_2\}}$
17: **end for**

Algorithm 6 Least-Squares Policy Iteration: $\texttt{LSPI}(c)$ (Lagoudakis & Parr, 2003)

Input: Stopping criterion ϵ
1: Initialize $w' \leftarrow w_0$
2: **repeat**
3: $w \leftarrow w'$
4: $w' \leftarrow \texttt{LSTDQ}(w, c)$
5: **until** $\|w - w'\| \le \epsilon$
Output: Policy weight w $\left(\text{i.e., } \pi(x) = \arg\min_{a \in \mathrm{A}} w^\top \phi(x, a) \ \ \forall x\right)$

Algorithm 7 $\texttt{LSTDQ}(w, c)$ (Lagoudakis & Parr, 2003)

1: Initialize $\widetilde{\mathbf{A}} \leftarrow \mathbf{0}$ *// $k \times k$ matrix*
2: Initialize $\widetilde{b} \leftarrow \mathbf{0}$ *// $k \times 1$ vector*
3: **for** each $(x, a, x', c) \in \mathrm{D}$ **do**
4: $a' = \arg\min_{\widetilde{a} \in \mathrm{A}} w^\top \phi(x', \widetilde{a})$
5: $\widetilde{\mathbf{A}} \leftarrow \widetilde{\mathbf{A}} + \phi(x, a)\left(\phi(x, a) - \gamma\phi(x', a')\right)^\top$
6: $\widetilde{b} \leftarrow \widetilde{b} + \phi(x, a)c$
7: **end for**
8: $\widetilde{w} \leftarrow \widetilde{\mathbf{A}}^{-1}\widetilde{b}$
Output: $\widetilde{w}$

H. Additional Experimental Details

H.1. Environment Descriptions and Procedures

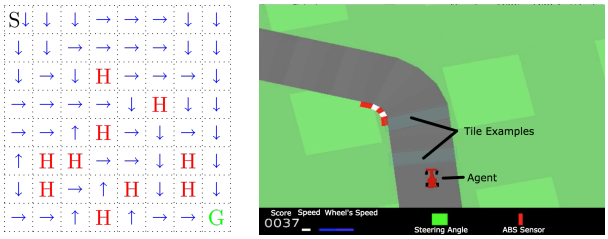

Figure 3. Depicting the *FrozenLake* and *CarRacing* environments.

Frozen Lake. The environment is a 8x8 grid as seen in Figure 3 (left), based on OpenAi's FrozenLake-v0. In each episode, the agent starts from S and traverse to goal G. While traversing the grid, the agent must avoid the pre-determined holes denoted by H. If the agent steps off of the grid, the agent returns to the same grid location. The episode terminates when the agent reaches the goal or falls into a hole. The arrows in Figure 3 (left) is an example policy returned by our algorithm, showing an optimal route.

Denote X_{holes} as the set of all holes in the grid and $\mathrm{X}_{goal} = \{x_{goal}\}$ is a singleton set representing the goal in the grid. The contrained batch policy learning problem is:

$$\begin{aligned} \min_{\pi\in\Pi} \quad & C(\pi) = \mathbb{E}[\mathbb{I}(x' \notin \mathrm{X}_{goals})] = \mathrm{P}(x' \notin \{x_{goal}\}) \\ \text{s.t.} \quad & G(\pi) = \mathbb{E}[\mathbb{I}(x' \in \mathrm{X}_{holes})] = \mathrm{P}(x' \in \mathrm{X}_{holes}) \leq \tau \end{aligned} \tag{40}$$

We collect 5000 trajectories by selecting an action randomly with probability .95 and an action from a DDQN-trained model with probability .05.Furthermore we set $B = 30$ and $\eta = 50$, the hyperparameters of our Exponentiated Gradient subroutine. We set the threshold for the constraint $\tau = .1$.

Car Racing. The environment is a racetrack as seen in Figure 3 (right), modified from OpenAi's CarRacing-v0. In each state, given by the raw pixels, the agent has 12 actions: $a \in A = \{(i,j,k) | i \in \{-1,0,1\}, j \in \{0,1\}, k \in \{0,.2\}\}$. The action tuple (i,j,k) cooresponds to steering angle, amount of gas applied and amount of brake applied, respectively. In each episode, the agent starts at the same point on the track and must traverse over 95% of the track, given by a discretization of 281 tiles. The agent recieves a reward of $+\frac{1000}{281}$ for each unique tile over which the agent drives. The agent receives a penalty of $-.1$ per-time step. Our collected dataset takes the form: $\mathrm{D} = \{(x_{t-6}, x_{t-3}, x_t), a_t, (x_{t-3}, x_t, x_{t+3}), c_t, g_{0,t}, g_{1,t}\}$ where x_i denotes the image at timestep i and a_t is applied 3 times between x_t and x_{t+3}. This frame-stacking option is common practice in online RL for Atari and video games.In our collected dataset D, the maximum horizon is 469 time steps.

The first constraint concerns accumulated number of brakes, a proxy for smooth driving or acceleration. The second constraint concerns how far the agent travels away from the center of the track, given by the Euclidean distance between the agent and the closest point on the center of the track. Let N_t be the number of tiles that is collected by the agent in time t. The constrained batch policy learning problem is:

$$\begin{aligned} \min_{\pi\in\Pi} \quad & \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t (-\frac{1000}{281} N_t + .1)] \\ \text{s.t.} \quad & G_0(\pi) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \mathbb{I}(a_t \in \mathrm{A}_{braking})] \leq \tau_0 \\ & G_1(\pi) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t d(u_t, v_t)] \leq \tau_1 \end{aligned} \tag{41}$$

We instatiate our subroutines, FQE and FQI, with multi-layered CNNs. Furthermore we set $B = 10$ and $\eta = .01$, the hyperparameters of our Exponentiated Gradient subroutine. We set the threshold for the constraint to be about 75% of the value exhibited by online RL agent trained by DDQN (Van Hasselt et al., 2016).

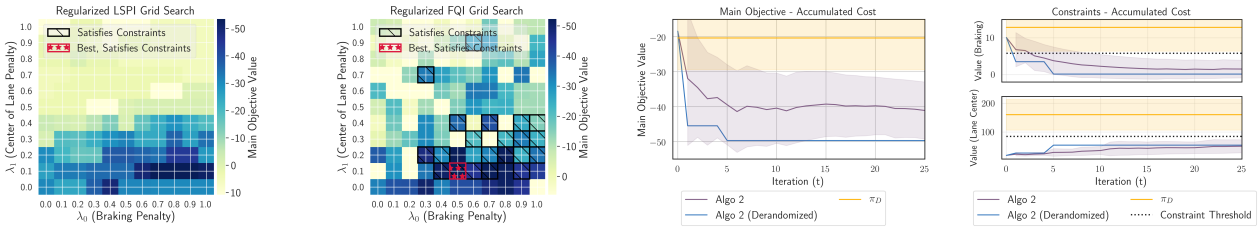

Figure 4. (First and Second figures) Result of 2-D grid-search for one-shot, regularized policy learning for *LSPI* (left) and *FQI* (right). (Third and Fourth figures) value range of individual policies in our mixtured policy and data generating policy π_D for main objective (left) and cost constraint (right)

H.2. Additional Discussion for the Car Racing Experiment

Regularized policy learning and grid-search. We perform grid search over a range of regularization parameters λ for both Least-Squares Policy Iteration - LSPI ((Lagoudakis & Parr, 2003)) and Fitted Q Iteration - FQI ((Ernst et al., 2005)). The results, seen from the the first and second plot of Figure 4, show that one-shot regularized learning has difficulty learning a policy that satisfies both constraints. We augment LSPI with non-linear feature mapping from one of our best performing FQI model (using CNNs representation). While both regularized LSPI and regularized FQI can achieve low main objective cost, the constraint cost values tend to be sensitive with the λ step. Overall for the whole grid search, about 10% of regularized policies satisfy both constraints, while none of the regularized LSPI policy satisfies both constraints.

Mixture policy and de-randomization. As our algorithm returned a mixture policy, it is natural to analyze the performance of individual policies in the mixture. The third and fourth plot from Figure 4 show the range of performance of individual policy in our mixture (purple band). We compare individual policy return with the stochastic behavior of the data generation policy. Note that our policies satisfy constraints almost always, while the individual policy returned in the mixture also tends to outperform π_D with respect to the main objective cost.

Off-policy evaluation standalone comparison. Typically, inverse propensity scoring based methods call for stochastic behavior and evaluation policies (Precup et al., 2000; Swaminathan & Joachims, 2015). However in this domain, the evaluation policy and environment are both deterministic, with long horizon (the max horizon is D is 469). Consequently Per-Decision Importance Sampling typically evaluates the policy as 0. In general, off-policy policy evaluation in long-horizon domains is known to be challenging (Liu et al., 2018; Guo et al., 2017). We augment PDIS by approximating the evaluation policy with a stochastic policy, using a softmin temperature parameter. However, PDIS still largely shows significant errors. For Doubly Robust and Weighted Doubly Robust methods, we train a model of the environment as follows:

- a 32-dimensional representation of state input is learned using variational autoencoder. Dimensionality reduction is necessary to aid accuracy, as original state dimension is $96 \times 96 \times 3$
- an LSTM is used to learn the transition dynamics $P(z(x')|z(x), a)$, where $z(x)$ is the low-dimensional representation learned from previous step. Technically, using a recurrent neural networks is an augmentation to the dynamical modeling, as true MDPs typically do not require long-term memory
- the model is trained separately on a different dataset, collected indendently from the dataset D used for evaluation

The architecture of our dynamics model is inspired by recent work in model-based online policy learning (Ha & Schmidhuber, 2018). However, despite our best effort, learning the dynamics model accurately proves highly challenging, as the horizon and dimensionality of this domain are much larger than popular benchmarks in the OPE literature (Jiang & Li, 2016; Thomas & Brunskill, 2016; Farajtabar et al., 2018). The dynamics model has difficulty predicting the future state several time steps away. Thus we find that the long-horizon, model-based estimation component of DR and WDR in this high-dimensional setting is not sufficiently accurate. For future work, a thorough benchmarking of off-policy evaluation methods in high-dimensional domains would be a valuable contribution.