Quantile Filtered Imitation Learning

David Brandfonbrener **William F. Whitney** **Rajesh Ranganath** **Joan Bruna**
Department of Computer Science, Center for Data Science
New York University
david.brandfonbrener@nyu.edu

Abstract

We introduce quantile filtered imitation learning (QFIL), a novel policy improvement operator designed for offline reinforcement learning. QFIL performs policy improvement by running imitation learning on a filtered version of the offline dataset. The filtering process removes s, a pairs whose estimated Q values fall below a given quantile of the pushforward distribution over values induced by sampling actions from the behavior policy. The definitions of both the pushforward Q distribution and resulting value function quantile are key contributions of our method. We prove that QFIL gives us a safe policy improvement step with function approximation and that the choice of quantile provides a natural hyperparameter to trade off bias and variance of the improvement step. Empirically, we perform a synthetic experiment illustrating how QFIL effectively makes a bias-variance tradeoff and we see that QFIL performs well on the D4RL benchmark.

1 Introduction

Offline RL offers tantalizing promise in diverse applications from robotics to healthcare [Levine et al., 2020]. However, offline RL has some fundamental limitations. In particular, policies cannot in general extrapolate beyond the coverage of the dataset since unknown parts of state-action space may be dangerous. Subject to this safety constraint, algorithms still face a bias-variance tradeoff when attempting to perform a policy improvement step. Explicitly, we can reduce the variance of our learned policy by remaining closer to the behavior policy where we have more data. But this comes at the cost of bias away from the optimal policy and towards the behavior.

In this paper we propose a novel policy improvement operator called quantile filtered imitation learning (QFIL) to safely make this bias-variance tradeoff. Our improvement operator can be coupled with any value estimation technique to get an offline RL algorithm. Simply put, QFIL attempts to imitate actions from the dataset that perform well while ignoring those that perform poorly. To decide which actions to imitate we use the pushforward distribution induced by pushing samples from the behavior policy through the estimated Q function and then only imitate actions whose Q estimates exceed some quantile of the pushforward distribution. Importantly, since the policy learning step is simply imitation learning, we ensure that the policy has no incentive to choose actions outside of the data distribution, which provides safety. Selecting a high quantile allows us to make a less biased and more aggressive update by imitating a smaller subset of the data, while a low quantile provides a lower-variance update.

We provide both theoretical and empirical arguments for the efficacy of QFIL. On the theory side, we prove a safe policy improvement guarantee that illustrates how the quantile τ controls the bias-variance tradeoff. On the empirical side, we first provide a toy experiment that demonstrates how the quantile τ controls the bias-variance tradeoff. Then we demonstrate that QFIL can achieve performance competitive with the state of the art on the D4RL [Fu et al., 2020] benchmark on the MuJoCo tasks using one-step on-policy value estimation and on the Ant Maze tasks using iterative off-policy value estimation.

Offline Reinforcement Learning Workshop at Neural Information Processing Systems, 2021

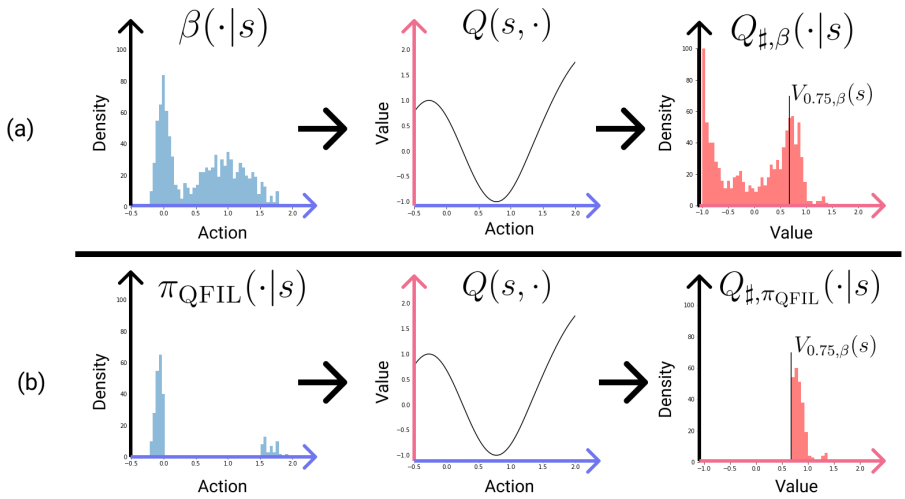

Figure 1: (a) An illustration of the pushforward distribution $Q_{\sharp,\beta}(\cdot|s)$ induced by pushing samples from $\beta(\cdot|s)$ through $Q(s,\cdot)$. $Q_{\sharp,\beta}(\cdot|s)$ is the distribution of predicted Q-values generated by sampling an action from the policy β and evaluating it using Q. The value function quantile $V_{0.75,\beta}(s)$ is the 0.75 quantile of this pushforward distribution. (b) An illustration of how QFIL defines an improved policy π_{QFIL} that only selects actions for which $Q(s,a) \geq V_{0.75,\beta}(s)$.

2 Setup

We will consider an offline RL setup as follows. Let $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, P, R, \gamma\}$ be a discounted infinite-horizon MDP. Define $r(s,a) = \mathbb{E}[R(s,a)]$. Rather than interacting with $\mathcal{M}$, we only have access to a dataset $\mathcal{D}$ of N tuples of (s_i, a_i, r_i) collected by trajectories from some behavior policy β with initial state distribution ρ. Define $d_\pi(s) := (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \mathbb{P}_{\rho,P,\pi}(s_t = s)$ to be the discounted state visitation distribution under a policy π. The objective is to maximize the expected return J of the learned policy $J(\pi) := \mathbb{E}_{\rho,P,\pi}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right]$.

Algorithm 1: OAMPI

input : K, dataset D_N, estimated behavior $\hat{\beta}$
Set $\pi_0 = \hat{\beta}$. Initialize $\widehat{Q}^{\pi_{-1}}$ randomly.
for *k = 1, . . . , K* **do**
  Policy evaluation: $\widehat{Q}^{\pi_{k-1}} = \mathcal{E}(\pi_{k-1}, D_N, \widehat{Q}^{\pi_{k-2}})$
  Policy improvement: $\pi_k = \mathcal{I}(\widehat{Q}^{\pi_{k-1}}, \hat{\beta}, D_N, \pi_{k-1})$
end

Following Brandfonbrener et al. [2021] we consider a generic template for model free offline RL algorithms as offline approximate modified policy iteration (OAMPI). This template, shown in Algorithm 1, alternates between steps of policy evaluation and policy improvement using the fixed offline dataset. This paper focuses on proposing a specific version of the improvement step that can be used in tandem with various evaluation steps. Moreover, as we will show, this improvement operator can be useful both in the one-step ($K = 1$) and iterative ($K >> 1$) regimes. This template is generic enough to capture essentially all related work by just replacing the $\mathcal{E}$ and $\mathcal{I}$ operators as we will discuss in more detail in Section 4 after introducing QFIL.

3 Our algorithm: Quantile Filtered Imitation Learning (QFIL)

In this section we will introduce our main algorithmic contributions. First, we will formally define value function quantiles in terms of pushforward distributions over Q values. Then we define quantile filtered imitation learning (QFIL). Finally, we provide some theoretical analysis of QFIL to demonstrate how the quantile governs a bias-variance tradeoff in the policy improvement step.

3.1 Pushforward Q distributions and value function quantiles

The main mathematical objects that we introduce in this work are the pushforward Q distribution $Q_{\sharp,\pi}(\cdot|s)$ for some action-value function Q and policy π and the resulting value function quantile

$V_{\tau,\pi}(s)$ at quantile τ. These concepts are illustrated in Figure 1. Intuitively, this pushforward distribution $Q_{\sharp,\pi}(\cdot|s)$ is the distribution over $\mathbb{R}$ that is generated when we sample actions from π conditioned on s and then "push" those actions "forward" through the function $Q(s,\cdot)$. Then $V_{\tau,\pi}$ simply takes a quantile of this induced distribution over $\mathbb{R}$.

More formally, we can define the pushforward distribution over values at a state s induced by any π and Q evaluated at any measurable set of values $B \subseteq \mathbb{R}$ as:

$$Q_{\sharp,\pi}(B|s) := \pi(Q^{-1}(B;s)|s) \tag{1}$$

where Q^{-1} gives the pre-image of B under Q so that for any $v \in B$, we have $Q^{-1}(v;s) \subseteq \mathcal{A}$ and for any $a \in Q^{-1}(v;s)$, we have $Q(s,a) = v$.

Equivalenty, since $Q_{\sharp,\pi}(\cdot|s)$ is a real-valued distribution, we can define it in terms of its CDF. Let $X \sim Q_{\sharp,\pi}(\cdot|s)$, then we have that

$$\mathbb{P}(X \leq v) = \mathbb{P}_{a\sim\pi|s}(Q(s,a) \leq v). \tag{2}$$

Now we can directly define the value function quantile $V_{\tau,\pi}(s)$ by taking the τ quantile of the pushforward Q distribution:

$$V_{\tau,\pi}(s) := \sup\left\{v \in \mathbb{R} \;\; s.t. \;\; \mathbb{P}_{X\sim Q_{\sharp,\pi}(\cdot|s)}(X \leq v) \leq \tau\right\} \tag{3}$$

$$= \sup\left\{v \in \mathbb{R} \;\; s.t. \;\; \mathbb{P}_{a\sim\pi|s}(Q(s,a) \leq v) \leq \tau\right\} \tag{4}$$

Comparison to distributional RL. We want to emphasize that the value function quantile is a very different object from the quantile value function considered in work on distributional RL [Dabney et al., 2018]. That work attempts to approximate the full distribution of stochastic returns Z^π for some policy π by computing various quantiles of the distribution over Z^π conditioned on s, a using a bellman backup. These objects consider stochasticity over *entire trajectories*. In contrast, we are using the standard Q functions and the pushforward distribution over Q values $Q_{\sharp,\pi}(\cdot|s)$ at s only considers stochasticity induced by π at the first step in a trajectory.

3.2 The QFIL policy improvement operator

QFIL defines a novel policy improvement operator that can be incorporated into the generic OAMPI framework introduced in Section 2. The operator simply filters out data where the estimated Q value falls below the τ value function quantile. Explicitly, we define the policy update as follows:

$$\mathcal{D}_\tau = \left\{s, a \in \mathcal{D} \;\; s.t. \;\; \widehat{Q}^k(s,a) > \widehat{V}^k_{\tau,\beta}(s)\right\}, \qquad \pi_{k+1} = \arg\max_\pi \sum_{s,a\in\mathcal{D}_\tau} \log \pi(a|s). \tag{5}$$

Note that importantly, we use the pushforward distribution induced by β to compute the value function quantile. This ensures that we are evaluating our estimated value functions on data from the data-generating distribution. Also note that we can combine the QFIL policy improvement operator with any policy evaluation operator to get $\widehat{Q}^k$.

Practical benefits of QFIL. There are three main benefits to the QFIL improvement operator.

1. Safety: the QFIL objective only encourages imitation of actions already in the dataset. This is useful in domains where we know that the behavior policy is safe but that there may be unsafe actions if we attempt to extrapolate beyond what we have seen in the data.
2. Bias-variance tradeoff: the quantile provides an effective and intuitive knob to trade off bias and variance. As we will see in theory and practice below, the quantile essentially allows us to choose how much data we use to estimate a new policy. We can get a small sample of data from a near optimal policy, but at the cost of high variance. Or we can use a large sample, but at the cost of a bias towards the behavior policy and away from an optimal policy. The quantile is intuitive since for example setting $\tau = 0.9$ means we expect to have about 10% of the data from $\mathcal{D}$ present in $\mathcal{D}_\tau$. A practitioner can usually have a decent sense of how much data is needed to reliably solve the supervised policy estimation problem and can use this to propose reasonable settings of the quantile.
3. Optimization: the QFIL policy learning step only uses standard supervised learning. This means we get access to all the tools, tricks, and hyperparameters that have been designed for optimizing supervised learning problems. In contrast, policy improvement operators that attempt to optimize Q directly face a different optimization problem that machine learning tools have not been optimized for.

Implementation-level decisions. The main implementation-level decisions left to instantiate the algorithm are to (1) decide how to estimate Q, and (2) how to estimate $V_{\tau,\beta}$. For Q we experiment with both on-policy estimates of Q^β using SARSA Q estimation and off-policy estimates of Q^{π_k} using uncorrected DDPG-style Q estimation where we sample $a' \sim \pi_k$. To compute $V_{\tau,\beta}$ we use a sampling-based approach. Specifically, we take M samples from an estimated behavior policy $\hat{\beta}$ and then compute the empirical quantile after pushing the M samples through Q. One final note is that when implementing QFIL we can replace the explicit computation of $\mathcal{D}_\tau$ by computing 0-1 weights for each datapoint as $\mathbb{1}[\widehat{Q}^k(s,a) > \widehat{V}^k_{\tau,\beta}(s)]$.

3.3 Theoretical analysis

Now that we have laid out the algorithm we will provide some brief theoretical justification for QFIL. Specifically, we will show that the quantile provides an effective way to trade off the variance of the improvement step caused by finite data and errors in the approximation and estimation of the Q function and policy with bias induced by imitating suboptimal actions. Here we will only prove a guarantee for the one-step variant of the algorithm that uses on-policy value estimation of Q^β, and we leave a treatment of off-policy learning for future work.

To make our analysis we will need a few definitions and assumptions. Namely,

1. Assume value estimation error is bounded $\mathbb{E}_{\substack{s\sim d_\beta \\ a\sim\beta|s}}[(Q^\beta(s,a) - \widehat{Q}^\beta(s,a))^2] \leq \varepsilon_Q(N)$, whp.
2. Assume that for any policy π we can produce an estimate $\hat{\pi}$ based on N samples of $s \sim d, a \sim \pi.|s$ such that: $\mathbb{E}_{s\sim d}[D_{TV}(\hat{\pi}(\cdot|s)\|\pi(\cdot|s))] \leq \varepsilon_\pi(N)$, whp.
3. Assume value function quantile estimation such that $\mathbb{P}_{a\sim\beta|s}(\widehat{Q}^\beta(s,a) \leq \widehat{V}_{\tau,\beta}(s)) = \tau$.[1]
4. Let $Q^\beta_{\max} = \sup_{s,a} Q^\beta(s,a)$ and $A^\beta_{\max} = \sup_{s,a} A^\beta(s,a)$
5. Define the QFIL policy for $\widehat{Q}^\beta$ by $\pi_\tau(a|s) := \beta(a|s)\frac{\mathbb{1}[\widehat{Q}^\beta(s,a)\geq\widehat{V}_{\tau,\beta}(s)]}{1-\tau}$.

Now we are ready to state the main result which is a lower bound on the expected improvement of our learned policy $\hat{\pi}$ over the behavior β. The proof is deferred to Appendix C.

Proposition 1. *Under the above assumptions and letting W_1 denote the Wasserstein-1 distance,* QFIL *returns a policy $\hat{\pi}$ such that whp*

$$(J(\hat{\pi}) - J(\beta))(1-\gamma) \geq \mathop{\mathbb{E}}_{s\sim d_\beta}\left[W_1\left(\widehat{Q}^\beta_{\sharp,\pi_\tau}(\cdot|s),\ \widehat{Q}^\beta_{\sharp,\beta}(\cdot|s)\right)\right] \tag{6}$$

$$- 2\left(\sqrt{\frac{\varepsilon_Q(N)}{1-\tau}} + \frac{\varepsilon_\pi((1-\tau)N)Q^\beta_{\max}}{1-\tau} + \frac{\gamma A^\beta_{\max}}{1-\gamma}\left(\tau + \varepsilon_\pi((1-\tau)N)\right)\right). \tag{7}$$

Let's examine each term in the bound. The first Wasserstein distance term expresses the inverse of the bias: it is large for large values of τ that approach higher value policies and it decreases to 0 for lower values of τ that approach pure imitation. In other words, this term quantifies the improvement of the QFIL policy π_τ over the behavior β according to the estimated Q function $\widehat{Q}^\beta$. Note that because the Wasserstein distance is always at least zero, so is this term. Perhaps this term is easiest to understand in pictures as it measures the distance between the real-valued distributions shown in red in the rightmost panels of Figure 1. Setting a higher quantile τ will move more mass to higher estimated values from the pushforward behavior distribution $\widehat{Q}^\beta_{\sharp,\beta}(\cdot|s)$ to get to the pushforward QFIL distribution $\widehat{Q}^\beta_{\sharp,\pi_\tau}(\cdot|s)$. The precise scaling with τ will be highly dependent on $\widehat{Q}^\beta$ and β.

The second term encapsulates the variance, which is the error due to the finite sample size, approximation error of the Q function, approximation error of our imitation learning, and effects of distribution shift. Importantly, this term becomes more negative for larger values of τ. Explicitly, if we assume a $\frac{1}{\sqrt{N}}$ scaling for the ε term then we can expect the variance bound to scale with $\frac{1}{(1-\tau)^{3/2}} + \tau$. This captures the intuition that staying closer to the behavior by setting a small value of τ will reduce the variance. This bound is likely often much too pessimistic about the impacts of distribution shift, but it is beyond the scope of this paper to provide a more intricate problem-dependent analysis.

[1] This assumption is stronger than the first two. We conjecture that it can be weakened to gracefully allow for approximation of the value function quantile, but leave this to future work.

4 Related work

The most closely related line of work presents various filtered and weighted imitation learning improvement operators. This includes MARWIL [Wang et al., 2018], CRR [Wang et al., 2020], AWR [Peng et al., 2019], AWAC [Nair et al., 2020], BAIL [Chen et al., 2020], and ABM [Siegel et al., 2020]. While all of these algorithms have various differences, most of them perform policy improvement using exponentially weighted imitation learning similar to:

$$\pi_{k+1} = \arg\max_{\pi} \sum_{s,a \in \mathcal{D}} \exp[\alpha(Q(s,a) - V(s))] \log \pi(a|s). \tag{8}$$

The key difference between QFIL and these papers is the introduction of the value function quantile into the advantage calculation. The quantile τ replaces the hyperparameter α from the exponentially weighted family of algorithms and provides a more graceful way to trade off bias and variance. In some sense this is a numerical issue where larger values of α can cause instability due to very large weights. In practice implementations need to clip the weights and then as α increases the weight just approaches the hard threshold $\mathbb{1}[Q(s,a) - V(s) > 0]$ multiplied by some constant. Moreover, the approaches are not mutually exclusive and can be combined by using the value function quantile $V_{\tau,\beta}$ in place of V, but at the cost of now having to tune bothe τ and α.

Another relevant piece of related work is Chen et al. [2021] which in addition to proposing the decision transformer proposes the %BC baseline. This baseline runs filtered imitation learning where the epsiodic return is used in place of the Q function and a constant value function that does not depend on the state, but does try to estimate a quantile of the distribution over returns. In contrast, QFIL uses a learned Q function and state-dependent value function quantile estimate.

The other relevant detail is the use of hard filtering instead of expoential weighting. This is motivated by Byrd and Lipton [2018] which shows that importance weighting is ineffective when training large, flexible neural network models. Some algorithms (CRR, ABM, and BAIL, %BC) also use hard filtering like QFIL. None of them use value function quantiles. CRR proposes a variant called CRR-max that uses the max of m samples to effectively estimate the $\frac{m-1}{m}$ quantile. BAIL uses the "upper envelope" of the data, which relies more on generalization of function approximation and as a result may exclude all datapoints at some states when the estimate of the upper envelope exceeds the 1.0 quantile of the pushforward distribution. QFIL provides a more flexible and consistent mechanism to trade off bias and variance. An extended discussion of other, less closely related offline policy improvement operators can be found in Appendix A.

In concurrent work, Kostrikov et al. [2021a] propose a similar method that uses exponentiated expectile advantage functions to weight the imitation loss. However, instead of first learning a standard Q function and then computing a value function quantile, they modify the Bellman backup directly so that the Q estimates are not estimating the Q values of a particular policy. Moreover, at an implementation level, we use hard filtering and estimate quantiles from samples, while Kostrikov et al. [2021a] use exponential weighting and expectile regression. Finally, because our algorithm does not modify the Bellman backup, it is simpler to provide a more rigorous theoretical analysis.

5 Experiments

5.1 Synthetic experiment

To illustrate the bias-variance tradeoff we discussed in the prior section, we created a simple synthetic problem. Since we are focused on the policy improvement step rather than the evaluation step, we will use a finite horizon environment with horizon equal to 1 (i.e. a contextual bandit). We use real-valued, continuous, 1-dimensional states and actions and learn $\hat{\beta}, \widehat{Q}$, and $\hat{\pi}$ with small neural networks of width 50 and depth 2. The data is generated according to:

$$s \sim U([0,1]), \qquad a|s \sim \frac{s+\epsilon}{2}, \epsilon \sim U([0,1]), \qquad r(s,a) = \begin{cases} 1 - |a - (1-s)| & a \in [\frac{s}{2}, \frac{s+1}{2}] \\ -1 & otherwise \end{cases}$$

Note that the reward function penalizes actions with zero probability under the behavior.

We simulate several experiments with various dataset sizes and quantiles. Full experimental details can be found in Appendix B. We visualize the results in Figure 2 demonstrating that higher quantiles have more variance but less bias. As a result, different quantiles are optimal depending upon the dataset size.

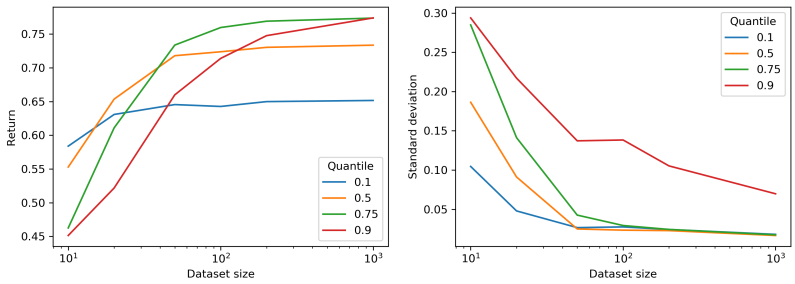

Figure 2: Figures showing the mean (left) and standard deviation (right) across 50 random seeds of the return of the policies learned by QFIL on the synthetic problem with various quantiles and datasets. The plots illustrate how the quantile τ governs a bias-variance tradeoff.

5.2 D4RL experiments

Now we move on to larger scale experiments on the D4RL benchmark [Fu et al., 2020]. Full experimental details can be found in Appendix B.

One-step MuJoCo. Recent work of Brandfonbrener et al. [2021] has showed that simply performing one step of policy improvement ($K = 1$ in OAMPI) yields state of the art performance on most of the MuJoCo tasks in the D4RL benchmark excepting the "random" datasets where iterative algorithms perform better. Our first experiment shows that replacing the exponentiated advantage function used by MARWIL and CRR yields slightly better performance on many of the MuJoCo tasks and can be seen in Table 1. We also compare to a % BC baseline that just runs behavior cloning on the trajectories with the highest returns (where we sweep over different values for the amount of data to keep). By using only the one-step algorithm this experiment attempts to isolate the effect of the QFIL operator for policy improvement from effects of off-policy evaluation.

Table 1: Results for one-step algorithms on MuJoCo tasks from the D4RL benchmark. We run 3 seeds and tune over 4 hyperparameter values. We report the mean and standard deviation across seeds using 100 evaluation episodes for the best hyperparameter. Exp-adv refers to exponentiated advantage as in MARWIL or CRR, here limited to one step.

Dataset	BC	% BC	One-step Exp-adv	One-step QFIL
halfcheetah-med	41.9 ± 0.1	42.3 ± 0.2	48.4 ± 0.1	**50.6 ± 0.0**
walker2d-med	68.6 ± 6.3	73.7 ± 1.5	81.8 ± 2.2	**83.7 ± 0.9**
hopper-med	49.9 ± 3.1	57.5 ± 1.5	59.6 ± 2.5	**62.8 ± 2.0**
halfcheetah-med-rep	34.3 ± 0.7	37.8 ± 0.4	38.1 ± 1.3	**40.4 ± 0.8**
walker2d-med-rep	26.2 ± 2.4	**61.8 ± 1.8**	49.5 ± 12.0	55.0 ± 0.8
hopper-med-rep	23.2 ± 2.8	77.0 ± 5.2	**97.5 ± 0.7**	94.6 ± 2.3
halfcheetah-med-exp	61.1 ± 2.7	92.0 ± 0.3	93.4 ± 1.6	**94.4 ± 0.5**
walker2d-med-exp	78.5 ± 22.4	108.8 ± 0.4	**113.0 ± 0.4**	112.7 ± 0.7
hopper-med-exp	49.1 ± 4.3	**110.1 ± 0.6**	103.3 ± 9.1	107.6 ± 2.2
halfcheetah-rand	2.2 ± 0.0	2.2 ± 0.1	3.2 ± 0.1	**5.6 ± 0.3**
walker2d-rand	0.9 ± 0.1	2.4 ± 0.1	5.6 ± 0.8	**6.0 ± 0.5**
hopper-rand	2.0 ± 0.1	5.1 ± 1.9	**7.5 ± 0.4**	7.0 ± 0.5

One-step antmaze. While the Mujoco tasks see the best performance with one-step methods, this can be attributed to the dense rewards and low-coverage behavior policies. In contrast, the antmaze tasks in the D4RL benchmark have sparse rewards and diverse behavior policies, giving a different sort of challenge to offline RL algorithms. Results are shown in Table 2. We see larger gains from QFIL over the baselines on the harder tasks.

Iterative antmaze Now we go beyond one-step updates and show that QFIL can be effectively combined with off-policy Q estimation. To do this we simply instantiate the OAMPI algorithm using

Table 2: Results for one-step algorithms on antmaze tasks from the D4RL benchmark. Again, we run 3 seeds, tune over 4 hyperparameter values, and use 100 eval episodes.

Dataset	BC	% BC	One-step Exp-Adv	One-step QFIL
umaze	60.3 ± 7.8	57.3 ± 2.1	75.7 ± 5.6	$\mathbf{91.0 \pm 1.4}$
umaze-diverse	54.0 ± 3.7	65.3 ± 6.3	$\mathbf{66.0 \pm 4.9}$	61.7 ± 9.5
medium-play	0.3 ± 0.5	0.7 ± 0.9	0.7 ± 0.9	$\mathbf{13.3 \pm 8.4}$
medium-diverse	0.0 ± 0.0	0.7 ± 0.5	1.0 ± 1.4	$\mathbf{4.0 \pm 4.2}$
large-play	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	$\mathbf{1.3 \pm 1.9}$
large-diverse	0.0 ± 0.0	0.3 ± 0.5	0.0 ± 0.0	$\mathbf{0.7 \pm 0.9}$

Table 3: Results for iterative algorithms on antmaze tasks from the D4RL benchmark. This time we run 3 seeds, tune over 4 hyperparameter values, and use 100 eval episodes.

Dataset	Iterative Exp-Adv	Iterative QFIL	CQL	IQL
umaze	61.3 ± 6.2	$\mathbf{96.7 \pm 1.2}$	74.0	87.5
umaze-diverse	70.0 ± 3.7	67.7 ± 0.9	**84.0**	62.2
medium-play	0.3 ± 0.5	60.0 ± 8.0	61.2	**71.2**
medium-diverse	0.0 ± 0.0	61.3 ± 19.0	53.7	**71.0**
large-play	0.3 ± 0.5	4.0 ± 5.7	15.8	**39.6**
large-diverse	0.3 ± 0.5	12.7 ± 17.9	14.9	**47.5**

QFIL as the improvement operator and using standard SARSA-style fitted Q evaluation with target networks as the evaluation operator. Importantly, since the QFIL policies are derived from imitation, they remain within the data distribution to allow for easier off-policy evaluation. Moreover, since we compute value function quantiles by pushing forward the estimated behavior, those quantiles are also not being queried out of distribution. Results in Table 3 show that QFIL substanitally outperforms the exponentiated advantage algorithm and achieves performance competitive with state of the art algorithms CQL [Kumar et al., 2020] and IQL [Kostrikov et al., 2021a] that modify the Bellman backups (where results for those algorithms are taken directly from those papers). Note that this iterative algorithm is not strictly covered by our theory from the previous section which only considers the one-step approach. It is an interesting direction for future work to get better theoretical guarantees for iterative algorithms.

6 Discussion

Here we have presented QFIL, a novel policy improvement operator for offline RL. QFIL provides a safe policy improvement step that only imitates actions already in the dataset and does so in a numerically stable way that only uses supervised learning while still providing a useful knob in the quantile to trade off bias and variance. Along the way we offered definitions of the pushforward Q distribution and value function quantile that we hope can find uses beyond QFIL. Indeed, anywhere that the advantage function is used, it could be replaced by a quantile advantage that uses a value function quantile instead of the standard value function.

Acknowledgements

We would like to thank Ilya Kostrikov for suggestions about the antmaze datasets and the anonymous reviewers for their suggestions about related work. This work is partially supported by the Alfred P. Sloan Foundation, NSF RI-1816753, NSF CAREER CIF 1845360, NSF CHS-1901091, Samsung Electronics, and the Institute for Advanced Study. DB is supported by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International Conference on Machine Learning*, pages 22–31. PMLR, 2017.

David Brandfonbrener, William F Whitney, Rajesh Ranganath, and Joan Bruna. Offline rl without off-policy evaluation. In *Advances in Neural Information Processing Systems*, 2021.

Jonathon Byrd and Zachary C Lipton. What is the effect of importance weighting in deep learning? *arXiv preprint arXiv:1812.03372*, 2018.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *arXiv preprint arXiv:2106.01345*, 2021.

Xinyue Chen, Zijian Zhou, Zheng Wang, Che Wang, Yanqiu Wu, and Keith Ross. Bail: Best-action imitation learning for batch deep reinforcement learning. *Advances in Neural Information Processing Systems*, 33, 2020.

Will Dabney, Mark Rowland, Marc Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *arXiv preprint arXiv:2106.06860*, 2021.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. *arXiv preprint arXiv:1812.02900*, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021a.

Ilya Kostrikov, Jonathan Tompson, Rob Fergus, and Ofir Nachum. Offline reinforcement learning with fisher divergence critic regularization. *arXiv preprint arXiv:2103.08050*, 2021b.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *arXiv preprint arXiv:2006.04779*, 2020.

Romain Laroche, Paul Trichelair, and Remi Tachet Des Combes. Safe policy improvement with baseline bootstrapping. In *International Conference on Machine Learning*, pages 3652–3661. PMLR, 2019.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Yao Liu, Adith Swaminathan, Alekh Agarwal, and Emma Brunskill. Provably good batch reinforcement learning without great exploration. *arXiv preprint arXiv:2007.08202*, 2020.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

Ashvin Nair, Murtaza Dalal, Abhishek Gupta, and Sergey Levine. Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Noah Siegel, Jost Tobias Springenberg, Felix Berkenkamp, Abbas Abdolmaleki, Michael Neunert, Thomas Lampe, Roland Hafner, Nicolas Heess, and Martin Riedmiller. Keep doing what worked: Behavior modelling priors for offline reinforcement learning. In *International Conference on Learning Representations*, 2020.

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

Qing Wang, Jiechao Xiong, Lei Han, Han Liu, Tong Zhang, et al. Exponentially weighted imitation learning for batched historical data. In *Advances in Neural Information Processing Systems*, pages 6288–6297, 2018.

Ziyu Wang, Alexander Novikov, Konrad Zolna, Josh S Merel, Jost Tobias Springenberg, Scott E Reed, Bobak Shahriari, Noah Siegel, Caglar Gulcehre, Nicolas Heess, et al. Critic regularized regression. *Advances in Neural Information Processing Systems*, 33, 2020.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning, 2019.

A Related work continued

There are many other policy improvement operators in the offline RL literature. BCQ [Fujimoto et al., 2018] and MBS [Liu et al., 2020] maximize the Q value subject to behavior constraints. These algorithms try to maximize Q subject to some minimum probability under the estimated behavior. In contrast QFIL imitates instead of optimizing Q, making it less likely to choose unseen actions while still providing the quantile hyperparameter to allow for more aggressive updates. SPIBB [Laroche et al., 2019] instead constrains the policy based on uncertainty estimates. BRAC [Wu et al., 2019] and TD3+BC [Fujimoto and Gu, 2021] attempt to optimize the estimated Q values subject to some regularization toward the behavior. These approaches are intuitive and effective on some tasks, but by attempting to optimize estimated Q values and without stronger constraints they are potentially more susceptible to selecting out-of-distribution actions. Other work modifies the policy evaluation step to learn conservative [Kumar et al., 2020] or regularized [Kostrikov et al., 2021b] Q values. Here we focus on modifications to the improvement step. An interesting direction for future work would be to combine modifications to the improvement and evaluation steps.

B Experimental details

B.1 Synthetic experiment

As explained in the main text, data is sampled from the following distributions:

$$s \sim U([0,1]), \qquad a|s \sim \frac{s+\epsilon}{2}, \epsilon \sim U([0,1]), \qquad r(s,a) = \begin{cases} 1 - |a - (1-s)| & a \in [\frac{s}{2}, \frac{s+1}{2}] \\ -1 & otherwise \end{cases}.$$

Then the algorithm proceeds in three steps: (1) estimate $\hat{\beta}$, (2) estimate $\widehat{Q}$, and (3) learn $\hat{\pi}$. Note that because we are in a contextual bandit problem, $\widehat{Q}$ does not depend on the behavior policy β. But of course we will only be able to learn $\widehat{Q}$ at actions selected by the behavior. These functions are learned to minimize the following objectives, summed over the dataset:

$$\ell_\beta(s,a,r) = -\log \hat{\beta}(a|s) \tag{9}$$

$$\ell_Q(s,a,r) = (r - \widehat{Q}(s,a))^2 \tag{10}$$

$$\ell_\pi(s,a,r) = -\mathbb{1}[\widehat{Q}(s,a) \geq \widehat{V}_{\tau,\beta}(s)] \log \hat{\pi}(a|s) \tag{11}$$

where we estimate $\widehat{V}_{\tau,\beta}$ from $\widehat{Q}$ and $\hat{\beta}$ at a given state s as follows. First we take 100 samples $a_1, \ldots, a_{100}$ from $\hat{\beta}(\cdot|s)$. Then we compute $\widehat{V}_{\tau,\beta}(s)$ as the empirical τ quantile of the set of real numbers $\{\widehat{Q}(s,a_i) : 1 \leq i \leq 100\}$.

All networks $(\hat{\beta}, \widehat{Q}, \hat{\pi})$ are MLPs with width 50, depth 2, and ReLU activation functions and are implemented in PyTorch Paszke et al. [2019]. The policy networks output truncated normal distributions that are truncated to the bounds of the action space and log standard deviations are bounded in [-5, 0]. We train using the Adam optimizer [Kingma and Ba, 2014] with learning rate 0.001 and batch size 64 for all networks. Each network is trained for 1000 gradient steps. Evaluation for each dataset is conducted by sampling 100 states, then sampling one action $a \sim \hat{\pi}(\cdot|s)$, and then evaluating the reward function for each s, a pair. This gives an estimated reward for one random seed. We report mean and standard deviation over 50 random seeds in Figure 2.

B.2 D4RL experiments

Datasets are all derived from the D4RL benchmark suite [Fu et al., 2020]. We consider two algorithmic outlines: one-step and iterative. We generally follow the hyperparameter choices from Brandfonbrener et al. [2021].

One-step. For the one-step experiments the algorithm proceeds in a similar three step manner as the synthetic experiment: (1) estimate $\hat{\beta}$, (2) estimate $\widehat{Q}^\beta$, and (3) learn $\hat{\pi}$. We use SARSA Q estimation [Sutton and Barto, 2018] with target networks [Mnih et al., 2015] for increased stability. We initialize

$\hat{\pi}$ at $\hat{\beta}$. The loss functions at each sample (s, a, r, s', a') from the replay buffer are now:

$$\ell_\beta(s, a, r, s', a') = -\log \hat{\beta}(a|s) \tag{12}$$

$$\ell_{Q^\beta}(s, a, r, s', a') = (\widehat{Q}^\beta(s, a) - r - \gamma \underline{\widehat{Q}^\beta}(s', a'))^2 \tag{13}$$

$$\ell_\pi(s, a, r, s', a') = -\mathbb{1}[\widehat{Q}^\beta(s, a) \geq \widehat{V}^\beta_{\tau,\beta}(s)] \log \hat{\pi}(a|s) \tag{14}$$

where $\underline{\widehat{Q}^\beta}$ denotes the target network.

All networks $(\hat{\beta}, \widehat{Q}^\beta, \hat{\pi})$ are MLPs with width 1024, depth 2, and ReLU activation functions. The policy networks output truncated normal distributions that are truncated to the bounds of the action space and log standard deviations are bounded in [-5, 0]. We train using the Adam optimizer with learning rate 0.0001 and batch size 512 for all networks. The target network weights are updated every 2 gradient steps using an exponentially weighted moving average with hyperaparameter 0.005. The discount factor is $\gamma = 0.99$.

We train $\hat{\beta}$ for 500k gradient steps, $\widehat{Q}^\beta$ for 2 million gradient steps, and $\hat{\pi}$ for 100k gradient steps.

For QFIL we sweep the quantile τ over $[0.5, 0.75, 0.9, 0.95]$ and use 100 samples to estimate $\widehat{V}^\beta_{\tau,\beta}(s)$. For the exponentially weighted baseline we sweep the temperature parameter α over $[0.3, 1.0, 3.0, 10.0]$, use 10 samples to estimate $\widehat{V}_\beta(s)$, and clip the weights at 100 for numerical stability.

We train the entire procedure with 3 random seeds for each of the 6 hyperparameter values. Then we evaluate 100 episodes using the modal actions from $\hat{\pi}$ and calculate the mean return. We report the mean and standard deviation over seeds for the best hyperparameter value in the tables in the paper.

Iterative. For the iterative antmaze experiments we generally leave all of the hyperparameters the same, but make the following modifications to the learning procedure. Instead of learning $\hat{\pi}$ as step 3, we now initialize $\widehat{Q}^\pi$ at $\widehat{Q}^\beta$ and then iteratively update $\hat{\pi}$ and $\widehat{Q}^\pi$. In each iteration we take 1 gradient step to update π and 2 gradient steps to update $\widehat{Q}^\pi$. The loss functions are now:

$$\ell_{Q^\pi}(s, a, r, s', a') = (\widehat{Q}^\pi(s, a) - r - \gamma \underline{\widehat{Q}^\pi}(s', \underline{a'}))^2, \quad \underline{a'} \sim \hat{\pi}(\cdot|s') \tag{15}$$

$$\ell_\pi(s, a, r, s', a') = -\mathbb{1}[\widehat{Q}^\pi(s, a) \geq \widehat{V}^\pi_{\tau,\beta}(s)] \log \hat{\pi}(a|s). \tag{16}$$

We run this for 300k steps which corresponds to 300k gradient steps on $\hat{\pi}$ and 600k gradient steps on $\widehat{Q}^\pi$. We again sweep hyperparameters sets of $[0.75, 0.9, 0.95, 0.99]$ and $[0.3, 1.0, 3.0, 10.0]$ for QFIL and the exponentially weighted baseline respectively.

C Proof of Proposition 1

The proof first needs two lemmas, one novel and one from Achiam et al. [2017].

Lemma 2 (Advantage)**.** *Under the above assumptions and letting W_1 denote the Wasserstein-1 distance,* QFIL *returns a policy $\hat{\pi}$ such that whp*

$$\mathop{\mathbb{E}}_{\substack{s\sim d_\beta \\ a\sim\hat{\pi}|s}}[A^\beta(s, a)] \geq \mathop{\mathbb{E}}_{s\sim d_\beta}\left[W_1\left(\widehat{Q}^\beta_{\sharp,\pi_\tau}(\cdot|s),\ \widehat{Q}^\beta_{\sharp,\beta}(\cdot|s)\right)\right] \tag{17}$$

$$- 2\sqrt{\frac{\varepsilon_Q(N)}{1-\tau}} - \frac{2Q^\beta_{\max}\varepsilon_\pi((1-\tau)N)}{1-\tau}. \tag{18}$$

Proof. At a high level, the proof decomposes into three parts, one for each term. The first term comes from the expected improvement of the target policy π_τ under the estimated Q function $\widehat{Q}^\beta$. The second term comes from error in value estimation and the last term from error in imitation learning. First we will set up the decomposition, then we will go through each step.

For the rest of the proof, unless noted otherwise, we will assume that $s \sim d_\beta$.

Decomposition. We begin by decomposing the expected advantage:

$$\mathop{\mathbb{E}}_{\substack{s\\a\sim\hat{\pi}|s}}[A^\beta(s,a)] = \mathop{\mathbb{E}}_{\substack{s\\a\sim\hat{\pi}|s}}[Q^\beta(s,a)] - \mathop{\mathbb{E}}_{\substack{s\\a\sim\beta|s}}[Q^\beta(s,a)] \tag{19}$$

$$= \mathop{\mathbb{E}}_{\substack{s\\a\sim\pi_\tau|s}}[Q^\beta(s,a)] - \mathop{\mathbb{E}}_{\substack{s\\a\sim\beta|s}}[Q^\beta(s,a)] + \underbrace{\mathop{\mathbb{E}}_{\substack{s\\a\sim\hat{\pi}|s}}[Q^\beta(s,a)] - \mathop{\mathbb{E}}_{\substack{s\\a\sim\pi_\tau|s}}[Q^\beta(s,a)]}_{T_\pi} \tag{20}$$

$$= \underbrace{\mathop{\mathbb{E}}_{\substack{s\\a\sim\pi_\tau|s}}[\widehat{Q}^\beta(s,a)] - \mathop{\mathbb{E}}_{\substack{s\\a\sim\beta|s}}[\widehat{Q}^\beta(s,a)]}_{T_W} \tag{21}$$

$$+ \underbrace{\mathop{\mathbb{E}}_{\substack{s\\a\sim\pi_\tau|s}}[Q^\beta(s,a) - \widehat{Q}^\beta(s,a)] - \mathop{\mathbb{E}}_{\substack{s\\a\sim\beta|s}}[Q^\beta(s,a) - \widehat{Q}^\beta(s,a)]}_{T_Q} + T_\pi \tag{22}$$

$$= T_W + T_Q + T_\pi \tag{23}$$

Wasserstein term. First, we show that by the definition of the pushforward distribution, we have that

$$T_W = \mathop{\mathbb{E}}_{\substack{s\\a\sim\pi_\tau|s}}[\widehat{Q}^\beta(s,a)] - \mathop{\mathbb{E}}_{\substack{s\\a\sim\beta|s}}[\widehat{Q}^\beta(s,a)] = \mathop{\mathbb{E}}_{\substack{s\\v\sim\widehat{Q}^\beta_{\sharp,\pi_\tau}|s}}[v] - \mathop{\mathbb{E}}_{\substack{s\\v\sim\widehat{Q}^\beta_{\sharp,\beta}|s}}[v] \tag{24}$$

$$= \mathop{\mathbb{E}}_{s}\left[\mathop{\mathbb{E}}_{v\sim\widehat{Q}^\beta_{\sharp,\pi_\tau}|s}[v] - \mathop{\mathbb{E}}_{v\sim\widehat{Q}^\beta_{\sharp,\beta}|s}[v]\right] \tag{25}$$

Now, we need to transform this into the Wasserstein-1 distance. Since the values are 1-dimensional real values, we know that $W_1(P,Q)$ can be written as $\int_0^1 |F_P^{-1}(z) - F_Q^{-1}(z)|dz$ where F_P^{-1} is the inverse CDF of P. We claim that the inverse CDF of $\widehat{Q}^\beta_{\sharp,\pi_\tau}(\cdot|s)$ is strictly greater than the inverse CDF of $\widehat{Q}^\beta_{\sharp,\beta}(\cdot|s)$, allowing us to rewrite the above expression as a Wasserstein-1 distance.

Note that this is equivalent to showing that the CDF of $\widehat{Q}^\beta_{\sharp,\pi_\tau}(\cdot|s)$ is strictly less than the CDF of $\widehat{Q}^\beta_{\sharp,\beta}(\cdot|s)$. To show this, let F_{π_τ} denote the CDF of $\widehat{Q}^\beta_{\sharp,\pi_\tau}(\cdot|s)$ and F_β denote the CDF of the CDF of $\widehat{Q}^\beta_{\sharp,\beta}(\cdot|s)$. By the definition of π_τ we have that

$$F_{\pi_\tau}(v) = \begin{cases} 0 & v < \widehat{V}^\beta_{\tau,\beta}(s) \\ \frac{1}{1-\tau}(F_\beta(v) - \tau) & otherwise \end{cases} \tag{26}$$

We need to show that for all v we have $F_{\pi_\tau}(v) \leq F_\beta(v)$. Break this into two cases. (1) If $v < V^\beta_{\tau,\beta}(s)$ then the inequality clearly holds since $F_{\pi_\tau}(v) = 0$ and $F_\beta(v) \geq 0$. (2) If $v \geq V^\beta_{\tau,\beta}(s)$, we note that $F_{\pi_\tau}(v)$ is an affine transformation of $F_\beta(v)$ with a positive coefficient $\frac{1}{1-\tau} \geq 1$. Note that at $v = V^\beta_{\tau,\beta}(s)$ we have $F_\beta(v) = \tau$ and $F_{\pi_\tau} = 0$ and that the lowest v such that $F_\beta(v) = 1$ also is the lowest v such that $F_{\hat{\pi}}(v) = 1$. Since the CDFs are just an affine transformation of each other and we have that $F_\beta(v) \geq F_{\pi_\tau}(v)$ at the two end points of the segment, we know that $F_{\pi_\tau}(v) \leq F_\beta(v)$ for all v as desired.

Thus, we have that

$$T_W = \mathop{\mathbb{E}}_{s}\left[\mathop{\mathbb{E}}_{v\sim\widehat{Q}^\beta_{\sharp,\pi_\tau}|s}[v] - \mathop{\mathbb{E}}_{v\sim\widehat{Q}^\beta_{\sharp,\beta}|s}[v]\right] \tag{27}$$

$$= \mathop{\mathbb{E}}_{s}\left[\int_0^1 F^{-1}_{\pi_\tau}(z) - F^{-1}_\beta(z)dz\right] = \mathop{\mathbb{E}}_{s}\left[\int_0^1 |F^{-1}_{\pi_\tau}(z) - F^{-1}_\beta(z)|dz\right] \tag{28}$$

$$= \mathop{\mathbb{E}}_{s}\left[W_1\left(\widehat{Q}^\beta_{\sharp,\pi_\tau}(\cdot|s),\ \widehat{Q}^\beta_{\sharp,\beta}(\cdot|s)\right)\right] \tag{29}$$

Imitation learning term. It will be useful to define some random variables. Let S the random variable of the state distributed according to $\mathcal{D}$. Let A be the action according the the behavior policy $\beta(\cdot|S)$. Let E be the event $\widehat{Q}^\beta(S,A) \geq V^\beta_{\tau,\beta}(S)$.

We can define the conditional distribution that generates this data according to Bayes' rule:

$$\mathbb{P}(A=a|S=s,E) = \frac{\mathbb{P}(A=a|S=s)\mathbb{P}(E|A=a,S=s)}{\mathbb{P}(E|S=s)} = \beta(a|s)\frac{\mathbb{1}[\widehat{Q}^\beta(s,a) \geq V^\beta_{\tau,\beta}(s)]}{\mathbb{P}(E|S=s)} \tag{30}$$

After applying the filter, we say data generated from $s|E$ by the policy π_τ defined as

$$\pi_\tau(a|s) = \beta(a|s)\frac{\mathbb{1}[\widehat{Q}^\beta(s,a) \geq V^\beta_{\tau,\beta}(s)]}{\mathbb{P}(E|S=s)} = \beta(a|s)\frac{\mathbb{1}[\widehat{Q}^\beta(s,a) \geq V^\beta_{\tau,\beta}(s)]}{1-\tau} \tag{31}$$

By definition of the filtering process we know that the filtering selects a dataset of expected size $(1-\tau)N$ for the imitation learning step. By assumption our algorithm outputs $\hat{\pi}$ such that

$$\mathop{\mathbb{E}}_{s|E}[D_{TV}(\hat{\pi}(\cdot|s)\|\pi_\tau(\cdot|s))] \leq \varepsilon_\pi((1-\tau)N). \tag{32}$$

Note that states are generated conditioned on E. Then, applying importance weighting, Holder's inequality, and the fact that $p(E|s) = 1-\tau$ we get

$$\begin{aligned}
-T_\pi &= \mathop{\mathbb{E}}_{\substack{s\\a\sim\pi_\tau|s}}[Q^\beta(s,a)] - \mathop{\mathbb{E}}_{\substack{s\\a\sim\hat{\pi}|s}}[Q^\beta(s,a)] && (33)\\
&= \mathop{\mathbb{E}}_{s}\int_a \Big(\pi_\tau(a|s)Q^\beta(s,a) - \hat{\pi}(a|s)Q^\beta(s,a)\Big) && (34)\\
&\leq Q^\beta_{\max}\int_s p(s)\int_a |\pi_\tau(a|s) - \hat{\pi}(a|s)| && (35)\\
&= Q^\beta \max \int_s \frac{p(E)}{p(E|s)}p(s|E)\int_a |\pi_\tau(a|s) - \hat{\pi}(a|s)| && (36)\\
&= Q^\beta_{\max}\mathop{\mathbb{E}}_{s|E}\frac{p(E)}{p(E|s)}\int_a |\pi_\tau(a|s) - \hat{\pi}(a|s)| && (37)\\
&\leq Q^\beta_{\max}\sup_s \frac{1}{p(E|s)}\mathop{\mathbb{E}}_{s|E}\int_a |\pi_\tau(a|s) - \hat{\pi}(a|s)| && (38)\\
&= \frac{2Q^\beta_{\max}}{1-\tau}\mathop{\mathbb{E}}_{s|E}[D_{TV}(\hat{\pi}(\cdot|s)\|\pi_\tau(\cdot|s))] && (39)\\
&\leq \frac{2Q^\beta_{\max}}{1-\tau}\varepsilon_\pi((1-\tau)N). && (40)
\end{aligned}$$

Value estimation term. We bound T_Q in two parts. When actions are generated from β we can just use Jensen's inequality and apply our value estimation assumption to bound

$$\left|\mathop{\mathbb{E}}_{\substack{s\\a\sim\beta|s}}\Big[Q^\beta(s,a) - \widehat{Q}^\beta(s,a)\Big]\right| \leq \sqrt{\mathop{\mathbb{E}}_{\substack{s\\a\sim\beta|s}}\Big[(Q^\beta(s,a) - \widehat{Q}^\beta(s,a))^2\Big]} \leq \sqrt{\varepsilon_Q(N)} \tag{41}$$

When actions are generated from π_τ we need to be more careful about distribution shift. Explicitly we use importance weighting, Cauchy-Schwarz, the fact that $p(E|s) = 1-\tau$, and our value estimation

assumption to get that

$$\left| \mathop{\mathbb{E}}_{\substack{s \\ a\sim\pi_\tau|s}} [Q^\beta(s,a) - \widehat{Q}^\beta(s,a)] \right| = \left| \mathop{\mathbb{E}}_{\substack{s \\ a\sim\beta|s}} \left[\frac{\pi_\tau(a|s)}{\beta(a|s)} (Q^\beta(s,a) - \widehat{Q}^\beta(s,a)) \right] \right| \tag{42}$$

$$\leq \sqrt{\mathop{\mathbb{E}}_{\substack{s \\ a\sim\beta|s}} \left[\frac{\pi_\tau(a|s)^2}{\beta(a|s)^2} \right] \mathop{\mathbb{E}}_{\substack{s \\ a\sim\beta|s}} \left[(Q^\beta(s,a) - \widehat{Q}^\beta(s,a))^2 \right]} \tag{43}$$

$$= \sqrt{\mathop{\mathbb{E}}_{\substack{s \\ a\sim\beta|s}} \left[\frac{\mathbb{1}[\widehat{Q}^\beta(s,a) \geq V^\beta_{\tau,\beta}(s)]^2}{p(E|s)^2} \right] \mathop{\mathbb{E}}_{\substack{s \\ a\sim\beta|s}} \left[(Q^\beta(s,a) - \widehat{Q}^\beta(s,a))^2 \right]} \tag{44}$$

$$= \sqrt{\mathop{\mathbb{E}}_{s} \frac{1}{(1-\tau)^2} \mathop{\mathbb{E}}_{a\sim\beta|s} \left[\mathbb{1}[\widehat{Q}^\beta(s,a) \geq V^\beta_{\tau,\beta}(s)] \right] \mathop{\mathbb{E}}_{\substack{s \\ a\sim\beta|s}} \left[(Q^\beta(s,a) - \widehat{Q}^\beta(s,a))^2 \right]} \tag{45}$$

$$= \sqrt{\mathop{\mathbb{E}}_{s} \frac{1-\tau}{(1-\tau)^2} \mathop{\mathbb{E}}_{\substack{s \\ a\sim\beta|s}} \left[(Q^\beta(s,a) - \widehat{Q}^\beta(s,a))^2 \right]} \tag{46}$$

$$\leq \sqrt{\frac{\varepsilon_Q(N)}{1-\tau}} \tag{47}$$

Since $1 \leq \sqrt{\frac{1}{1-\tau}}$, we can combine the two parts to bound $-T_Q \leq 2\sqrt{\frac{\varepsilon_Q(N)}{1-\tau}}$.
Combining the bounds on T_W, T_π, and T_Q gives us the desired result □

Lemma 3 (Achiam et al. [2017])**.** *Under the same assumptions as above, further define* $A^{\beta,\pi}_\infty = \sup_s |\, \mathbb{E}_{a\sim\pi}[A^\beta(s,a)]|$

$$J(\pi) - J(\beta) \geq \frac{1}{1-\gamma} \left(\mathbb{A}^\beta(\pi) - \frac{2\gamma A^{\beta,\pi}_\infty}{1-\gamma} \mathop{\mathbb{E}}_{s\sim d^\beta} \left[D_{TV}\big(\pi(\cdot|s) \| \beta(\cdot|s)\big) \right] \right). \tag{48}$$

Proposition 4. *Under the above assumptions and letting* W_1 *denote the Wasserstein-1 distance,* QFIL *returns a policy* $\hat{\pi}$ *such that whp*

$$(J(\hat{\pi}) - J(\beta))(1-\gamma) \geq \mathop{\mathbb{E}}_{s\sim d_\beta} \left[W_1 \left(\widehat{Q}^\beta_{\sharp,\pi_\tau}(\cdot|s),\ \widehat{Q}^\beta_{\sharp,\beta}(\cdot|s) \right) \right] \tag{6}$$

$$- 2 \left(\sqrt{\frac{\varepsilon_Q(N)}{1-\tau}} + \frac{\varepsilon_\pi((1-\tau)N) Q^\beta_{\max}}{1-\tau} + \frac{\gamma A^\beta_{\max}}{1-\gamma} \left(\tau + \varepsilon_\pi((1-\tau)N) \right) \right). \tag{7}$$

Proof. The proposition follows directly from plugging Lemma 2 into the advantage term of Lemma 3 and then bounding the TV term, which we will do now.

$$\mathop{\mathbb{E}}_{s}\left[D_{TV}\left(\hat{\pi}(\cdot|s)\|\beta(\cdot|s)\right)\right] \leq \mathop{\mathbb{E}}_{s}\left[D_{TV}\left(\hat{\pi}(\cdot|s)\|\pi_\tau(\cdot|s)\right)\right] + \mathop{\mathbb{E}}_{s}\left[D_{TV}\left(\pi_\tau(\cdot|s)\|\beta(\cdot|s)\right)\right] \tag{49}$$

$$\leq \varepsilon((1-\tau)N) + \frac{1}{2}\mathop{\mathbb{E}}_{s}\left[\int_a |\pi_\tau(a|s) - \beta(a|s)|\right] \tag{50}$$

$$= \varepsilon((1-\tau)N) + \frac{1}{2}\mathop{\mathbb{E}}_{s}\left[\int_a |\pi_\tau(a|s) - \beta(a|s)|\mathbb{1}[\widehat{Q}^\beta(s,a) < V^\beta_{\tau,\beta}(s)]\right. \tag{51}$$

$$\left. + \int_a |\pi_\tau(a|s) - \beta(a|s)|\mathbb{1}[\widehat{Q}^\beta(s,a) \geq V^\beta_{\tau,\beta}(s)]\right] \tag{52}$$

$$= \varepsilon((1-\tau)N) + \frac{1}{2}\mathop{\mathbb{E}}_{s}\left[\int_a \beta(a|s)\mathbb{1}[\widehat{Q}^\beta(s,a) < V^\beta_{\tau,\beta}(s)]\right. \tag{53}$$

$$\left. + \int_a (\frac{\beta(a|s)}{1-\tau} - \beta(a|s))\mathbb{1}[\widehat{Q}^\beta(s,a) \geq V^\beta_{\tau,\beta}(s)]\right] \tag{54}$$

$$= \varepsilon((1-\tau)N) + \frac{1}{2}\mathop{\mathbb{E}}_{s}\left[\tau + \frac{\tau}{1-\tau}\int_a \beta(a|s)\mathbb{1}[\widehat{Q}^\beta(s,a) \geq V^\beta_{\tau,\beta}(s)]\right] \tag{55}$$

$$= \varepsilon((1-\tau)N) + \frac{1}{2}\mathop{\mathbb{E}}_{s}\left[\tau + \frac{\tau}{1-\tau}(1-\tau)\right] \tag{56}$$

$$= \varepsilon((1-\tau)N) + \tau \tag{57}$$

Plugging this in yields the desired bound. □