Non-adaptive Online Finetuning for Offline Reinforcement Learning

Audrey Huang
audreyh5@illinois.edu
University of Illinois Urbana-Champaign

Mohammad Ghavamzadeh
ghavamza@amazon.com
Amazon

Nan Jiang
nanjiang@illinois.edu
University of Illinois Urbana-Champaign

Marek Petrik
mpetrik@cs.unh.edu
University of New Hampshire

Abstract

Offline reinforcement learning (RL) has emerged as an important framework for applying RL to real-life applications. However, the complete lack of online interactions causes technical difficulties. The *online finetuning* setting which incorporates a limited form of online interactions, often available in practice, has been developed to address these challenges. Unfortunately, existing theoretical frameworks for online finetuning either assume high online sample complexity or require deploying fully adaptive algorithms (i.e., unlimited policy changes), which restrict their application to real-world settings where online interactions and policy updates are expensive and limited. In this paper, we develop a new theoretical framework for online finetuning. Instead of competing with the optimal policy (which inherits the high sample complexity and adaptivity requirements of online RL), we aim to learn a policy that improves as much as possible over an existing *reference* policy using a *pre-specified* number of online samples and a *non-adaptive* data-collection strategy. Our formulation reveals surprising nuances and suggests novel principles that distinguish finetuning from purely online and offline RL.

1 Introduction

Reinforcement Learning (RL) is a form of learning via trial and error in which the agent interacts with the environment and improves its decision-making strategy (or policy) on the fly. Despite numerous successes in simulated domains, such an *online* and *adaptive* protocol has seen difficulties in real-world applications, such as healthcare, finance, and recommendation systems, where deploying unverified and/or ever-changing policies can have undesirable consequences. As a response to this challenge, *offline RL*, in which learning is solely from a pre-collected dataset without online interactions, has received significant attention as a promising framework for deploying RL in real-world tasks (Levine et al., 2020). However, its purely offline nature also gives rise to a host of new challenges, such as difficulties in policy selection (Paine et al., 2020; Zhang & Jiang, 2021) and high sensitivity to hyperparameters (Fujimoto & Gu, 2021; Cheng et al., 2022).

To tackle learning from a pre-collected dataset, researchers have started investigating a more hybrid approach that combines offline and online RL, noting that many applications of interest do allow a limited amount of online interaction in addition to the offline dataset. For example, in recommendation systems, it is often possible to run a fixed policy (upon approval) on a small portion of user traffic to collect more data for validation and further improvement; or in certain medical applications, one can recruit a small group of patients to perform clinical trials. In these cases, the online interactions are often limited in sample size and/or adaptivity (e.g., each new policy needs approval before being

deployed and one cannot change it on the fly (Koenecke et al., 2020)). The hope is that we can use these limited online interactions as a scarce resource to mitigate the caveats of offline RL and to improve upon (or to *finetune* (Xie et al., 2021b)) it.

Unfortunately, attempts at establishing a theoretical framework for this *online finetuning* setting have mostly yielded results that violate the aforementioned practical limitations (Xie et al., 2021b; Song et al., 2022; Wagenmaker et al., 2022; Wagenmaker & Pacchiano, 2023; Li et al., 2023; Zhang & Zanette, 2023):

***Adaptivity*:** Many works run (variants of) standard online RL algorithms in the finetuning phase, requiring full adaptivity, which is undesirable in practical applications where policy changes are costly to implement.

***High sample complexity & structural assumptions*:** Most existing works require a high sample complexity in the online phase that scales with certain structural quantities, such as the number of states/actions in the tabular setting or certain rank/dimension parameter in the function-approximation setting. In the latter case, the low-rankness itself is often an assumption on the environment dynamics which restricts the application scope of the methods.[1]

The above violations are clear signs that the existing theoretical frameworks do not adequately capture the essence of the practical settings. More concretely, all existing works inherit the standard goal of online RL, namely, competing with the **optimal** policy (in either PAC or regret formulation), and this ambitious goal (optimality seeking) comes at the cost of impractical assumptions (adaptivity and/or high complexity). Consequently, the results and methodologies in these works are much closer to those in the online RL literature than in offline RL.

In this paper, we take a different approach to the hybrid offline-online RL problem by removing the impractical assumptions and pursuing a more humble and reachable goal of **improvement maximization** (instead of competing with the optimal). More concretely, we consider the *non-adaptive* setting,[2] where the online policy is decided based on the offline data and is not allowed to be updated during the online phase. Then, with a given online budget, we ask the following question:

How to design an online data-collection strategy from the offline data, such that the policy learned from all the data (offline and online) improves as much as possible over the one learned purely from offline data?

Contributions:

1. We begin by defining a concrete and representative problem setting (Section 2). We then propose a model-based information-theoretic objective for choosing the online data-collection strategy (Eq. (4)), which hallucinates online data from plausible models and simulates the offline algorithm after data collection. Since we do not know the true model that would generate the data, worst-case reasoning (i.e., pessimism (Jin et al., 2020; Xie et al., 2021a)) is employed to guarantee that the objective value is a valid lower-bound of the improvement of interest (Theorem 2).

2. Perhaps surprisingly, we show that in certain cases the objective value—which represents the guaranteed amount of improvement—can be approximately zero across all online policies (Theorem 3), implying that positive improvement may not be obtainable in the worst-case scenario, leading to degenerate behaviors. To address this issue, we show that pessimism plays two different roles in our formulation: in data hallucination and when running the offline algorithm on the combined dataset. By choosing the offline algorithm to provide guard against degenerate policies (Bhardwaj et al., 2022), pessimism in data hallucination can be relaxed to strike a trade-off between the

[1] In contrast, offline RL can enjoy strong guarantees in general settings without making structural assumptions on the environment dynamics (Xie et al., 2021a).

[2] This is also called the *single-deployment* setting. In some practical scenarios, this process can iterate for a small number of times (Matsushima et al., 2020; Huang et al., 2021). While we would eventually like to understand such a multiple-deployment setting, we consider the single-deployment one as a building block which has already proven to be a challenging problem on its own.

magnitude of improvement and the chance that improvement occurs (Eq. (7)), a new principle we refer to as *opportunistic pessimism*.

3. Throughout the development we use multi-armed bandits (MABs) as a running example to provide further intuitions. We report preliminary empirical investigations in MABs in Section 4.

2 Setup

Markov Decision Process Our problem considers decision-making in finite-horizon Markov Decision Processes (MDPs). An MDP is specified by the tuple $M = \{\mathcal{S}, \mathcal{A}, P, R, H\}$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, H is the horizon, $P = \{P_0, \ldots, P_{H-1}\}$ with $P_h : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition dynamics, and $R = \{R_0, \ldots, R_{H-1}\}$ is a (possibly stochastic) reward function with $\mathbb{E}_{r \sim R_h(s,a)}[r] \in [0, 1]$ for all h and $s \in \mathcal{S}, a \in \mathcal{A}$. A policy π maps states or histories to a distribution over actions. For a given MDP model M, let $J_M(\pi) = \mathbb{E}_M[\sum_{h=0}^{H-1} r_h | \pi]$ denote the expected return of a policy π in M. We denote the true model underlying the environment as M^*.

Offline learning and (non-adaptive) online finetuning In our learning setting, we are given a policy class Π, a model class $\mathcal{M}$, an offline dataset $\mathcal{D}^{\text{off}}$ drawn from the environment $M^* \in \mathcal{M}$ (we assume *realizability*) following certain behavior policies, and a policy $\pi^{\text{ref}} \in \Pi$ computed using $\mathcal{D}^{\text{off}}$, Π, and $\mathcal{M}$, before the online data is collected. Our framework is flexible and agnostic to the choice of the algorithm that computes π^{ref}. Our task is to **1)** choose a policy $\mu^{\text{on}} \in \Pi$ [3] to collect an online dataset of size n^{on}, and **2)** run an offline algorithm $\mathfrak{A}_{\text{off}}$ over the combined offline and online dataset to produce a final policy $\hat{\pi}$, with the goal of maximizing the improvement over π^{ref}. The protocol is summarized below:

Non-adaptive Online Finetuning (NOF) Problem

Input: policy class Π, model class $\mathcal{M}$, offline dataset $\mathcal{D}^{\text{off}}$, reference policy $\pi^{\text{ref}} \in \Pi$.

1. Pick online data-collection policy $\mu^{\text{on}} \in \Pi$.
2. Execute μ^{on} in M^* to collect n^{on} samples, denoted by $\mathcal{D}^{\text{on}}$.
3. Run $\mathfrak{A}_{\text{off}}$ over $\mathcal{D}^{\text{off}} \cup \mathcal{D}^{\text{on}}$ and compute $\hat{\pi} \in \Pi$.

Goal: maximize $J_{M^*}(\hat{\pi}) - J_{M^*}(\pi^{\text{ref}})$.

Multi-armed bandits While Section 3 will discuss our problem formulation for the general RL setting, to improve intuition we will interweave examples in the setting of multi-armed bandits (MABs). MABs are a simplified and special case of MDPs that consist of a single state and a set of actions (arms) $\mathcal{A}$. In an MAB model M, each arm $a \in \mathcal{A}$ has a reward distribution $R_M(a)$ with average reward $r_M(a)$, thus $J_M(\pi) = \sum_{a \in \mathcal{A}} \pi(a) r_M(a)$. An MAB dataset $\mathcal{D}$ consists of tuples $\{(a, r)\}$, where a is drawn from a policy over $\mathcal{A}$ and $r \sim R_{M^*}(a)$. Given a dataset $\mathcal{D}$, $n_{\mathcal{D}}(a)$ denotes the number of times $a \in \mathcal{A}$ was pulled in $\mathcal{D}$, and $\hat{r}_{\mathcal{D}}(a)$ denotes the empirical estimate of $r_{M^*}(a)$, i.e., $\hat{r}_{\mathcal{D}}(a) = \frac{1}{n_{\mathcal{D}}(a)} \sum_{(a', r) \in \mathcal{D}} r \mathbf{1}[a' = a]$. For simplicity, our examples throughout the paper will utilize Bernoulli bandits, for which the reward distribution of each arm a is given by $R_{M^*}(a) = \text{Bernoulli}(r_M(a))$, and can be modeled by a single parameter $r_{M^*}(a)$, namely, its expected reward. These examples are designed to elucidate the core challenges of NOF.

3 An Information-Theoretic Objective for NOF

We provide a mathematical formulation and theoretically sound algorithm for the NOF problem described in Section 2, with its core being an information-theoretic objective that guides the choice of μ^{on}. Note that to specify the algorithm we also need to specify $\mathfrak{A}_{\text{off}}$, which we start with.

[3] For simplicity we assume that π^{ref}, μ^{on}, and $\hat{\pi}$ are all chosen from the same policy class Π; it is straightforward to allow for separate policy classes.

3.1 Choosing the Offline Algorithm $\mathfrak{A}_{\text{off}}$

Once $\mathcal{D}^{\text{on}}$ was collected, what we face in Step 3 of NOF (i.e., computing $\hat{\pi}$) is a standard offline RL problem. While we could employ any offline RL algorithm, there are a number of desirable properties:

1. While our goal is to improve over π^{ref}, a careless choice of $\mathfrak{A}_{\text{off}}$ may result in worse performance than π^{ref}, i.e., *negative* improvement. It is desired to have safety assurance that $\hat{\pi}$ is guaranteed to be no worse than π^{ref} under mild conditions.

2. The offline algorithm $\mathfrak{A}_{\text{off}}$ should also enjoy the state-of-the-art offline RL guarantees that the improvement is positive under favorable conditions (otherwise we can satisfy the point above by trivially setting $\hat{\pi} = \pi^{\text{ref}}$, which will never improve over π^{ref}).

The ARMOR algorithm (Bhardwaj et al., 2022) satisfies both the above considerations. It is based on the concept of *version space*, which will also be of vital importance for our later discussions.

Definition 1 (Version space)**.** *Given a model class $\mathcal{M}$, a dataset $\mathcal{D}$, and a confidence parameter δ, the construction of a version space is a procedure that outputs $\mathcal{M}_\delta(\mathcal{D}) \subseteq \mathcal{M}$, satisfying the following: if $\mathcal{D}$ is drawn from $M^* \in \mathcal{M}$, possibly in an adaptive (or non-i.i.d.) manner, then $\mathbb{P}_{\mathcal{D}}[M^* \in \mathcal{M}_\delta(\mathcal{D})] \geq 1 - \delta$.*

Roughly speaking, a version space uses data in $\mathcal{D}$ to rule out unlikely models. There are many ways to implement it depending on the setting: for example, in "Bernoulli" multi-armed bandits (MABs), the version space can be defined using the confidence intervals of the arms (see Example 1). A more general approach is to filter out models with poor data likelihood compared to the MLE (Bhardwaj et al., 2022).[4] Our algorithm design does not depend on the specific form of version space, and we will keep it abstract except for the standard condition of monotonicity in δ, i.e.,

Assumption 1. *We assume that $\mathcal{M}_{\delta_2}(\mathcal{D}) \subseteq \mathcal{M}_{\delta_1}(\mathcal{D})$ for any fixed $\mathcal{D}$ and $0 < \delta_1 \leq \delta_2 \leq 1$.*

Assumption 1 implies that smaller δ's result in larger version spaces, as they indicate higher probability of retaining M^*. With the concept of version space, we can now state the ARMOR algorithm as

$$\hat{\pi} = \operatorname*{argmax}_{\pi \in \Pi} \min_{M \in \mathcal{M}_\delta(\mathcal{D}^{\text{off}} \cup \mathcal{D}^{\text{on}})} J_M(\pi) - J_M(\pi^{\text{ref}}) \tag{1}$$

Intuitively, if we replace the minimum over M with $M = M^*$ in (1), the algorithm exactly maximizes the improvement over π^{ref}, which is our goal. Of course, M^* is unknown in practice, and generally cannot be identified especially if the given dataset $(\mathcal{D}^{\text{off}} \cup \mathcal{D}^{\text{on}})$ does not provide full coverage over the environment. However, we can still make the best effort in eliminating unlikely models and reducing the uncertainty of M^* by forming the version space $\mathcal{M}_\delta(\mathcal{D}^{\text{off}} \cup \mathcal{D}^{\text{on}})$, and then performing worst-case reasoning over the version space. Such a design immediately yields the desired safety guarantee, that $\hat{\pi}$ is no worse than π^{ref} with high probability.

Proposition 1 (Theorem 2 of Bhardwaj et al. (2022))**.** *We have $J_{M^*}(\hat{\pi}) \geq J_{M^*}(\pi^{\text{ref}})$ w.p. $1 - \delta$.*

As for the second consideration listed above, Bhardwaj et al. (2022) show that ARMOR also has strong optimality guarantees and competes with the best policy covered by the data. The MAB example below provides more intuition on ARMOR and its version space.

Example 1 (ARMOR in MABs)**.** *The version space in Bernoulli MABs can be defined as the set of Bernoulli distributions whose parameters lie within the rectangular set of the arms' confidence intervals,*

$$\mathcal{M}_\delta(\mathcal{D}) = \big\{M \in \mathcal{M} : R_M(a) = \text{Bernoulli}(r_M(a)), r_M(a) \in [\text{LCB}_{\mathcal{D}}(a), \text{UCB}_{\mathcal{D}}(a)],\ \forall a \in \mathcal{A}\big\},$$

[4] Although most offline analyses of version spaces assume i.i.d. data, they can often be straightforwardly extended to handle adaptively generated data via martingale concentration inequalities (Jin et al., 2021).

where $\mathrm{UCB}_{\mathcal{D}}(a) = \hat{r}_{\mathcal{D}}(a) + b(a)$ *and* $\mathrm{LCB}_{\mathcal{D}}(a) = \hat{r}_{\mathcal{D}}(a) - b(a)$ *are the upper and lower confidence bounds for arm* $a \in \mathcal{A}$*. For an i.i.d. dataset*[5] $\mathcal{D}$ *and any* $\delta \in [0, 1)$*, the confidence radius* $b(a)$ *can be defined using, e.g., Hoeffding's inequality as* $b(a) := \sqrt{\log(2|\mathcal{A}|/\delta)/2n_{\mathcal{D}}(a)}$*.*

If π^{ref} *is deterministic, i.e.,* $\pi^{\mathrm{ref}}(a) = \mathbf{1}[a = a^{\mathrm{ref}}], \forall a \in \mathcal{A}$ *with* $a^{\mathrm{ref}} \in \mathcal{A}$ *being a fixed arm, which is the case when it is learned using a typical offline RL algorithm such as LCB (Lattimore & Szepesvári, 2020), ARMOR (Eq. (1)) will also output a deterministic policy* $\hat{\pi}(a) = \mathbf{1}[a = \hat{a}]$*, where*

$$\hat{a} = \begin{cases} a^{\mathrm{ref}}, & \text{if } \mathrm{LCB}_{\mathcal{D}}(a) < \mathrm{UCB}_{\mathcal{D}}(a^{\mathrm{ref}}),\ \forall a \neq a^{\mathrm{ref}}, \\ \mathrm{argmax}_{a \neq a^{\mathrm{ref}}} \left\{ \mathrm{LCB}_{\mathcal{D}}(a) - \mathrm{UCB}_{\mathcal{D}}(a^{\mathrm{ref}}) \right\}, & \text{otherwise.} \end{cases} \tag{2}$$

In other words, ARMOR switches from a^{ref} *to another arm* a*, only if* $\mathcal{D}^{\mathrm{off}} \cup \mathcal{D}^{\mathrm{on}}$ *is such that the UCB of* a^{ref} *is smaller than the LCB of* a*.*

3.2 Information-theoretic Objective for μ^{on}

Now that $\mathfrak{A}_{\mathrm{off}}$ is fixed, we turn to the design of the online data-collection policy μ^{on}. As a starting point, suppose that we had access to M^* when choosing μ^{on}, but once it is selected we have to run ARMOR on the combined dataset without access to M^*. In this case, we compute μ^{on} by solving

$$\begin{aligned} \mu^{\mathrm{on}} &= \operatorname*{argmax}_{\mu \in \Pi} \ \mathbb{E}_{\mathcal{D}^{\mu}_{M^*}}[J_{M^*}(\hat{\pi}_{\mathcal{D}^{\mu}_{M^*}}) - J_{M^*}(\pi^{\mathrm{ref}})], \\ \text{where } \ \hat{\pi}_{\mathcal{D}^{\mu}_{M^*}} &= \operatorname*{argmax}_{\pi \in \Pi} \min_{M \in \mathcal{M}_{\delta}(\mathcal{D}^{\mathrm{off}} \cup \mathcal{D}^{\mu}_{M^*})} J_M(\pi) - J_M(\pi^{\mathrm{ref}}). \end{aligned} \tag{3}$$

Here $\mathcal{D}^{\mu}_{M^*}$ is the set of n^{on} samples collected by executing μ^{on} in M^* and $\mathbb{E}_{\mathcal{D}^{\mu}_{M^*}}$ is the expectation w.r.t. the random draws of $\mathcal{D}^{\mu}_{M^*}$. The subscript in $\hat{\pi}_{\mathcal{D}^{\mu}_{M^*}}$ is to distinguish it from the final output policy $\hat{\pi}$ in Eq. (1). These policies are the outputs of ARMOR with different version spaces. For $\hat{\pi}_{\mathcal{D}^{\mu}_{M^*}}$, the version space is defined on the union of $\mathcal{D}^{\mathrm{off}}$ and the dataset $\mathcal{D}^{\mu}_{M^*}$ "hallucinated" in the process of optimizing μ^{on}, while for $\hat{\pi}$ it is defined on the union of $\mathcal{D}^{\mathrm{off}}$ and the actual $\mathcal{D}^{\mathrm{on}}$. Since $\mathcal{D}^{\mu}_{M^*}$ is identically distributed as $\mathcal{D}^{\mathrm{on}}$ when we choose $\mu^{\mathrm{on}} = \mu$, the objective is *exactly* the expected improvement we can obtain in M^* by selecting $\mu^{\mathrm{on}} = \mu$ to collect the online data.[6]

In reality when we do not have access to M^*, we follow a design choice similar to ARMOR and construct a version space to quantify the uncertainty over M^*, and then employ worst-case reasoning (pessimism) to form our objective as

$$\mu^{\mathrm{on}} \in \operatorname*{argmax}_{\mu \in \Pi} \mathrm{OBJ}(\mu, \mathcal{M}') := \min_{M' \in \mathcal{M}'} \ \mathbb{E}_{\mathcal{D}^{\mu}_{M'}} \left[J_{M'}(\hat{\pi}_{\mathcal{D}^{\mu}_{M'}}) - J_{M'}(\pi^{\mathrm{ref}}) \right], \tag{4}$$

where $\mathcal{M}'$ is a version space that we hope can capture M^*. We will consider different design choices for $\mathcal{M}'$ in the rest of this section. For starters, we can set $\mathcal{M}' = \mathcal{M}_{\delta'}(\mathcal{D}^{\mathrm{off}})$, the version space constructed on $\mathcal{D}^{\mathrm{off}}$ with confidence δ'. We also abuse notation and write $\mathrm{OBJ}(\mu, \delta') := \mathrm{OBJ}(\mu, \mathcal{M}_{\delta'}(\mathcal{D}^{\mathrm{off}}))$.

Thanks to the worst-case reasoning (pessimism), if $M^* \in \mathcal{M}'$, the objective value on the RHS of Eq. (4) will give us a lower-bound on the improvement obtained in the real environment M^* by deploying the learned policy μ^{on}. Such a lower-bounding property makes the optimization problem "what you see is what you get", i.e., if we see a large objective value in Eq. (4), it is guaranteed that the real improvement can only be higher (all proofs can be found in Appendix B):

Proposition 2. *Let* $\mathcal{D}^{\mathrm{on}}$ *be the dataset collected using* μ^{on} *in Eq. (4). Then, if* $M^* \in \mathcal{M}'$*, we have*

$$\mathbb{E}_{\mathcal{D}^{\mathrm{on}}} \left[J_{M^*}(\hat{\pi}_{\mathcal{D}^{\mathrm{on}}}) - J_{M^*}(\pi^{\mathrm{ref}}) \right] \geq \mathrm{OBJ}(\mu^{\mathrm{on}}, \mathcal{M}') = \max_{\mu} \ \mathrm{OBJ}(\mu, \mathcal{M}').$$

Moreover, for $\mathcal{M}' = \mathcal{M}_{\delta'}(\mathcal{D}^{\mathrm{off}})$*, the above equation holds w.p.* $\geq 1 - \delta'$*. In addition,* $\mathrm{OBJ}(\mu, \delta')$ *is monotonically non-decreasing in* δ'*.*

[5] One can handle an adaptive dataset by union bounding over time and paying an additional $\log |\mathcal{D}|$ factor.

[6] One design choice we make here is to use the expectation $\mathbb{E}_{\mathcal{D}^{\mu}_{M^*}}$ to convert the distribution of improvement into a scalar metric. Alternatively, we can consider other functionals such as risk-sensitive measures.

Trade-off in the choice of δ' The hyperparameter δ' defines a trade off between how greedy we would like to be with maximizing our objective and the probability of the improvement being realized when we deploy μ^{on} in the true environment M^*. More precisely, for $\delta'_1 \geq \delta'_2$ we have $\mathcal{M}'_{\delta'_1} \subseteq \mathcal{M}'_{\delta'_2}$, and thus, $\max_\mu \text{OBJ}(\mu, \delta'_1) \geq \max_\mu \text{OBJ}(\mu, \delta'_2)$. This is because for δ'_1, the $\min_{M'}$ in Eq. (4) searches over a smaller set of models and as a result is less adversarial/conservative. This means that as δ' increases in magnitude, we will see a larger objective value and hence more significant guaranteed improvement, but the chance that the improvement actually occurs $(1-\delta')$ will become smaller.

An Alternative Objective We conclude by discussing an alternative objective that is a relaxed version of Eq. (4). Instead of using the improvement $J_{M'}(\hat{\pi}) - J_{M'}(\pi^{\text{ref}})$ in the expectation of Eq. (4), one alternative design choice is to directly use the ARMOR objective from Eq. (1), i.e.,

$$\mu^{\text{on}} = \operatorname*{argmax}_{\mu} \text{OBJ}'(\mu, \delta') := \min_{M' \in \mathcal{M}'} \mathbb{E}_{\mathcal{D}^\mu_{M'}} \left[\max_{\pi \in \Pi} \min_{M \in \mathcal{M}_\delta(\mathcal{D}^{\text{off}} \cup \mathcal{D}^\mu_{M'})} J_M(\pi) - J_M(\pi^{\text{ref}}) \right]. \tag{5}$$

This is a relaxation of Eq. (4), as the ARMOR objective itself lower bounds the improvement in M':

$$J_{M'}(\hat{\pi}) - J_{M'}(\pi^{\text{ref}}) \geq \max_{\pi \in \Pi} \min_{M \in \mathcal{M}_\delta(\mathcal{D}^{\text{off}} \cup \mathcal{D}^\mu_{M'})} J_M(\pi) - J_M(\pi^{\text{ref}}),$$

under the event that $M' \in \mathcal{M}_\delta(\mathcal{D}^{\text{off}} \cup \mathcal{D}^\mu_{M'})$.[7] While being looser than Eq. (4), Eq. (5) is also simpler and avoids a few nested computations in Eq. (4) (e.g., computing the argmax policy from the ARMOR objective and plugging it back to M' for evaluation). Another notable difference is that Eq. (5) is monotone in δ, i.e., it is no smaller for larger δ, which is not necessarily the case for Eq. (4) since it incorporates the tradeoff that δ induces (Assumption 1). We empirically investigate the performance of Eq. (4) vs. Eq. (5) in Section 4.

3.3 Degeneracy in Optimization and Opportunistic Pessimism

In Section 3.2, we showed that the optimization objective in Eq. (4) has "what you see is what you get" property, and a good improvement is guaranteed as long as the value of $\max_\mu \text{OBJ}(\mu, \delta')$ is large. Here we first show that unfortunately, under fairly reasonable assumptions, the objective value $\max_\mu \text{OBJ}(\mu, \delta')$ is *guaranteed* to be close to 0. This leads to a degenerate behavior for the algorithm and implies that our formulation is overly conservative.

Proposition 3 ($\text{OBJ}(\mu, \delta') \approx 0$ under mild assumptions)**.** *If there exists a model $M'_0 \in \mathcal{M}_{\delta'}(\mathcal{D}^{\text{off}})$ such that $\pi^{\text{ref}} \in \operatorname{argmax}_{\pi \in \Pi} J_{M'_0}(\pi)$, then*

$$|\text{OBJ}(\mu, \delta')| \;\leq\; \mathbb{P}_{\mathcal{D}^\mu_{M'_0}}[M'_0 \notin \mathcal{M}_\delta(\mathcal{D}^{\text{off}} \cup \mathcal{D}^\mu_{M'_0})], \quad \forall \mu \in \Pi.$$

Two remarks are in order: First, the proposition holds under the condition that π^{ref} is optimal for one of the models in $\mathcal{M}_{\delta'}(\mathcal{D}^{\text{off}})$, which is not a very strong assumption. For example, in Bernoulli MABs, if π^{ref} is computed using the LCB algorithm (Lattimore & Szepesvári, 2020; Rashidinejad et al., 2021) based on $\mathcal{D}^{\text{off}}$ with confidence parameter δ', the condition is *always* satisfied as π^{ref} is optimal for the model in which the mean rewards of the arms are equal to their lower confidence bounds. Second, the RHS of the bound should be treated as a small quantity close to δ for reasons discussed in Footnote 7. This non-zero residual corresponds to the low-probability event that the version space fails to capture the model M'_0 used to hallucinate data, and is a technical artifact due to the mismatch between the expectation in $\mathbb{E}_{\mathcal{D}^\mu_{M'}}$ and the high-probability guarantee of ARMOR.[8]

[7] The slight complication here is that $\mathcal{D}^{\text{off}}$ comes from M^* but $\mathcal{D}^\mu_{M'}$ comes from M', so it is difficult to quantify the likelihood of $M' \in \mathcal{D}^{\text{off}} \cup \mathcal{D}^\mu_{M'}$ using Definition Definition 1 which is stated very abstractly. However, note that $M' \in \mathcal{M}_{\delta'}(\mathcal{D}^{\text{off}})$, meaning that M' is a model consistent with $\mathcal{D}^{\text{off}}$. Therefore, it is very natural to assume that when $\mathcal{D}^{\text{off}}$ is augmented with data sampled from M', then M' should not be eliminated (at least with high probability), since the new observations from $\mathcal{D}^\mu_{M'}$ should favor M' even more.

[8] We refer the readers to Eq. (3) and Footnote 6 for a discussion on the choice of the distribution functional. If we replace the expected improvement with the $(1-\delta)$-quantile, we would obtain an exact 0 on the RHS.

In summary, we show that under reasonable assumptions $\mathrm{OBJ}(\mu, \delta') \approx 0$ for all μ, which makes the optimization meaningless: the objective implies that we can only gain an improvement of ≈ 0, but we can achieve that by simply outputting π^{ref} as the final policy! Moreover, since $\mathrm{OBJ}(\mu, \delta')$ is roughly the same for all μ, **the optimization over μ becomes arbitrary tie-breaking**, which is the last thing we want as doing *anything* else would not be any worse.

Opportunistic pessimism We first note that the above issue is not due to our objective being loose: if $M^* = M'_0$—which is completely possible given the information we have in $\mathcal{D}^{\mathrm{off}}$—then the possible improvement is in fact 0, so $\mathrm{OBJ}(\mu, \delta')$ is a tight lower-bound on the worst-case possible improvement. That said, $M^* = M'_0$ is an uninteresting case as π^{ref} is already optimal, so we should exclude it from consideration when selecting μ^{on}, and a smaller $\mathcal{M}'$ implies less pessimism and an increase in the objective value in general. This leads to the following definition, where $\Delta \in [0, 1]$ is a user-specified hyperparameter,

$$\mathcal{M}_{\delta'}(\mathcal{D}^{\mathrm{off}}, \Delta) := \left\{ M' \in \mathcal{M}_{\delta'}(\mathcal{D}^{\mathrm{off}}) : \Delta(M') \geq \Delta \right\}, \text{ where } \Delta(M') := \max_{\pi \in \Pi} J_{M'}(\pi) - J_{M'}(\pi^{\mathrm{ref}}), \quad (6)$$

which filters out models for which π^{ref} is already near-optimal (up to a gap of Δ). Plugging this into Eq. (4) (i.e., letting $\mathcal{M}' = \mathcal{M}_{\delta'}(\mathcal{D}^{\mathrm{off}}, \Delta)$), our final objective for selecting μ^{on} is:

$$\mu^{\mathrm{on}} = \operatorname*{argmax}_{\mu \in \Pi} \mathrm{OBJ}(\mu, \delta'; \Delta) := \min_{M' \in \mathcal{M}_{\delta'}(\mathcal{D}^{\mathrm{off}}, \Delta)} \mathbb{E}_{\mathcal{D}^{\mu}_{M'}} \left[J_{M'}(\hat{\pi}_{\mathcal{D}^{\mu}_{M'}}) - J_{M'}(\pi^{\mathrm{ref}}) \right]. \quad (7)$$

Crucially, we only filter out the uninteresting models in the version space for M'. It is important to note that both the ARMOR in Eq. (7) (i.e., $\hat{\pi}_{\mathcal{D}^{\mu}_{M'}}$) and the final ARMOR (i.e., $\hat{\pi}$ which uses the real online data $\mathcal{D}^{\mathrm{on}}$) must still use the unfiltered version spaces $\mathcal{M}_{\delta}(\mathcal{D}^{\mathrm{off}} \cup \mathcal{D}^{\mathrm{on}})$ and $\mathcal{M}_{\delta}(\mathcal{D}^{\mathrm{off}} \cup \mathcal{D}^{\mu}_{M^*})$, respectively, in order to retain the guarantee that $\hat{\pi}$ is never worse than π^{ref} (from Theorem 1).

This reveals a more general principle, which we call ***opportunistic pessimism***: our original objective in Eq. (4) employs pessimism in two places: (i) $\mathcal{M}_{\delta'}(\mathcal{D}^{\mathrm{off}})$ (for data hallucination), and (ii) $\mathcal{M}_{\delta}(\mathcal{D}^{\mathrm{off}} \cup \mathcal{D}^{\mu}_{M'})$ and $\mathcal{M}_{\delta}(\mathcal{D}^{\mathrm{off}} \cup \mathcal{D}^{\mathrm{on}})$ (for ARMOR). Theorem 1 shows that as long as ARMOR is used as the offline algorithm, $\hat{\pi}$ will always be competitive with π^{ref} **regardless of the choice of μ^{on}, and hence the choice of $\mathcal{M}'$, the version space for data hallucination**. This provides a strong guardrail for the optimization of μ^{on}, allowing for great flexibility in the design of $\mathcal{M}'$ to trade-off between the objective value and the scope within which the improvement is guaranteed.

Tradeoff in Δ In Eq. (7), the hyperparameter Δ plays a crucial role in designing $\mathcal{M}'$ and exhibits the aforementioned tradeoff. Similar to the monotonicity in δ' discussed earlier, for $\Delta_1 \geq \Delta_2$ we have $\mathcal{M}_{\delta'}(\mathcal{D}^{\mathrm{off}}, \Delta_1) \subseteq \mathcal{M}_{\delta'}(\mathcal{D}^{\mathrm{off}}, \Delta_2)$, thus for larger choices of Δ the objective will search over a smaller set of models and act less conservatively.[9] In fact, an extreme value of Δ would imply *optimism* in data hallucination, where μ^{on} is selected according to the best-case model in the version space. More concretely, setting $\Delta = \Delta_{\max} = \max\{\Delta : |\mathcal{M}_{\delta'}(\mathcal{D}^{\mathrm{off}}, \Delta)| > 0\}$, i.e., the largest possible gap, is approximately equivalent to choosing μ^{on} according to $\max_{M' \in \mathcal{M}_{\delta'}(\mathcal{D}^{\mathrm{off}})}$, instead of the worst case.[10] When Δ is too large relative to $\Delta(M^*)$, however, it will exclude M^* from $\mathcal{M}_{\delta'}(\mathcal{D}^{\mathrm{off}}, \Delta)$, which means the RHS of Eq. (7) will no longer lower bound the true improvement in M^*. This is formalized in the following guarantee for Eq. (7):

Theorem 4. *For $\mathcal{D}^{\mathrm{on}}$ collected using μ^{on} in Eq. (7) and any $\Delta \in [0, 1]$, we have*

(1) w.p. $\geq 1 - \delta$ (w.r.t. the randomness of $\mathcal{D}^{\mathrm{off}} \cup \mathcal{D}^{\mathrm{on}}$), if $\Delta(M^*) < \Delta$, then

$$J_{M^*}(\hat{\pi}) \geq \max_{\pi \in \Pi} J_{M^*}(\pi) - \Delta.$$

[9] If Δ is chosen poorly and large enough such that $\mathcal{M}_{\delta'}(\mathcal{D}^{\mathrm{off}}, \Delta) = \emptyset$, then implemention-wise, one may simply reduce it until $\mathcal{M}_{\delta'}(\mathcal{D}^{\mathrm{off}}, \Delta)$ is nonempty.

[10] One caveat is that $\Delta_{\max}$ is a random variable (depending on π^{ref}), and we also need $|\mathcal{M}_{\delta'}(\mathcal{D}^{\mathrm{off}}, \Delta_{\max})| = 1$ for this equivalence to hold, otherwise n^{on} will be diluted over multiple policies.

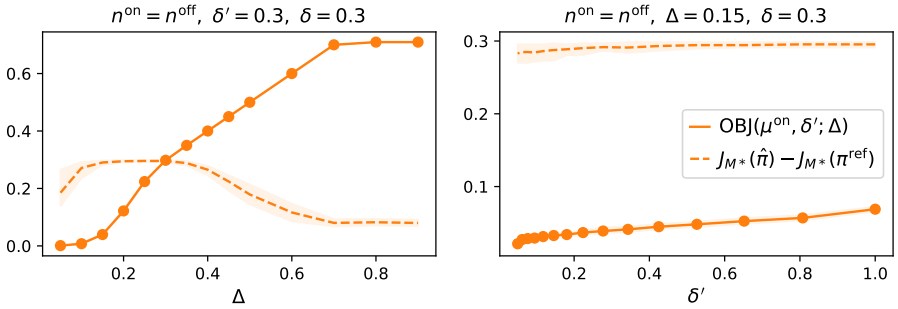

Figure 1: The magnitude-scope trade-off in choosing the **(top)** Δ and **(bottom)** δ'. Y-axis shows the value of Eq. (7) (solid lines) and the actual improvement (dashed lines). Top-left corner is desired as it implies a high objective value and more scenarios where improvement occurs. Shaded region represents 90% quantile over 100 random approximations of $\mathbb{E}_{\mathcal{D}^{\mu}_{M'}}$.

(2) w.p. $\geq 1 - \delta'$ *(w.r.t. the randomness of* $\mathcal{D}^{\text{off}}$*), if* $\Delta(M^*) \geq \Delta$*, then*

$$\mathbb{E}_{\mathcal{D}^{\text{on}}}\left[J_{M^*}(\hat{\pi}) - J_{M^*}(\pi^{\text{ref}})\right] \geq \text{OBJ}(\mu^{\text{on}}, \delta'; \Delta) = \max_{\mu}\ \text{OBJ}(\mu, \delta'; \Delta).$$

Moreover, $\text{OBJ}(\mu, \delta'; \Delta)$ *is monotonically non-decreasing in both* δ' *and* Δ*, and* $\text{OBJ}(\mu, \delta') = \text{OBJ}(\mu, \delta'; 0)$.

The guarantee reflects the trade-off in the choice of Δ, the hyperparameter that we choose: if there is room for improvement at least Δ in M^*, then claim (2) is active and we are guaranteed an improvement of $\text{OBJ}(\mu, \delta'; \Delta)$, which increases with Δ. On the other hand, if there is not enough room for improvement of at least Δ in M^*, then M^* will be excluded from $\mathcal{M}_{\delta'}(\mathcal{D}^{\text{off}}, \Delta)$, nullifying all the guarantees for the optimization of μ^{on}. Fortunately, Proposition 1 is still valid since we still keep M^* in the version space used by ARMOR (Eq. (1)), leading to a $\hat{\pi}$ that is no worse than π^{ref} with high probability, and hence inherits the Δ-optimality of π^{ref}, as per claim (1) in Theorem 4. We remark that claim (2) holds for any choice of offline algorithm used to learn $\hat{\pi}$ in Eq. (5), but claim (1) only holds for ARMOR.

Theorem 4 claim (2) is a "what you see is what you get" type of guarantee, where the improvement lower-bound is computed by the value of the objective itself. Speaking generally, this lower-bound is non-decreasing in n^{on} (more samples means higher probability of improvement), and non-increasing with model complexity (e.g., a bandit with more arms dilutes available exploration samples). We leave for future work the problem of deriving a more "conventional" sample complexity bound for policy improvement, i.e., one that depends on a small number of interpretable parameters, such as the sample size. The missing key is a "complexity" parameter that summarizes the difficulty of the problem instance. One key challenge of NOF is the non-adaptive nature of the online samples, which prohibits full exploration of the environment and application of online complexity parameters. Instead, the NOF parameter must express "difficulty of finding a better policy with limited interactions". As a result, even novel uses of the standard frameworks and complexities from offline RL, e.g., data coverage (Chen & Jiang, 2019; Xie et al., 2021a), and online RL, e.g., structural quantities such as the size/rank of state-action spaces (Jiang et al., 2016; Jin et al., 2018), do not apply. It is unclear what this parameter is, or if it exists at all; identifying it will require significant further work, for which our careful formulation provides a solid foundation.

Example: Bernoulli MAB Lastly, to improve intuition, we instantiate the behavior of Eq. (7) and its guarantee in Bernoulli bandits.

Example 2 (Mechanism of Eq. (7) in Bernoulli MABs)**.** *Let* $\mathcal{A}' = \{a \in \mathcal{A} : \exists M \in \mathcal{M}_{\delta'}(\mathcal{D}^{\text{off}}, \Delta)$ *s.t.* $r_M(a) \geq r_M(a^{\text{ref}}) - \Delta\}$ *denote the set of "candidate arms" for improvement, i.e., arms that have not been eliminated by offline data and have potential for at least* Δ *improvement in* M^**. Because Eq. (7) takes the minimum over all models, it must then strive to separate* all *candidate arms in* $\mathcal{A}'$ equally *(i.e., increase* $\text{LCB}_{\mathcal{D}^{\text{off}} \cup \mathcal{D}^{\mu}_{M'}}(a')$ *over* $\text{UCB}_{\mathcal{D}^{\text{off}} \cup \mathcal{D}^{\mu}_{M'}}(a^{\text{ref}})$ *for all* $a' \in \mathcal{A}'$*, per Eq. (2)), at a gap of* Δ*. Thus for a single deployment, the objective in Eq. (7) cannot "simply*

improve by Δ" in one arm, without trying to improve in all candidate arms by Δ. This represents a fundamental difficulty of single-deployment NOF: because we receive no feedback, in order to guarantee we cover a single better policy we actually need to cover all *candidates. However, as we explore further in Fig. 4 of Appendix C, "simply improving" is possible in a case-by-case basis given favorable conditions (e.g., if pulling any candidate arm results in improvement, see Instance 3 in Fig. 2).*

The theoretical insights of Theorem 4 are validated in Fig. 1 **(top)**, that displays the performance of Eq. (7) on $\mathcal{D}^{\text{off}}$ as depicted in Instance 1 of Fig. 2, which is a Bernoulli bandit with $\Delta(M^*) = 0.3$ also used in our later experiments (Section 4). The actual improvement (dashed line) is close to the maximum possible $\Delta(M^*)$ for a large range of $\Delta \in [0.1, 0.4]$, and is always non-negative, echoing the opportunistic pessimism principle that $\hat{\pi}$ will not decay compared to π^{ref} regardless of the choice of the version space. It is also larger than the objective value with high probability when $\Delta \leq \Delta(M^*)$ (Theorem 4, claim (2)), but can be lower when the version space excludes the true model ($\Delta > \Delta(M^*)$, e.g., when $\Delta > 0.3$ in Fig. 1 **(top)**).

This simulation also displays the dependence of Eq. (7) on the choice of the hyper-parameter Δ. The improvement lower-bound in claim (2) of Theorem 4 is maximized by setting $\Delta = \Delta(M^*)$, but since this is an unknown quantity (and may not necessarily be optimal for a given problem instance), the choice of Δ in general represents a trade-off that might be refined with pre-existing knowledge. When Δ is too small, it is more difficult for any μ to cause the inner ARMOR to switch arms, while Δ too large excludes M^* from the version space, and any simulated improvement may not transfer to the true model. Besides Δ, a similar trade-off can be made by tuning δ' as a hyperparameter:[11] the greater δ', the higher $\text{OBJ}(\mu^{\text{on}}, \delta'; \Delta)$, but the probability that the actual improvement will be at least $\text{OBJ}(\mu^{\text{on}}, \delta'; \Delta)$ (Theorem 4, claim (2)) will be smaller than $(1 - \delta')$. See Fig. 1 **(bottom)** for a visualization of such trade-off curves in the same MAB instance. More generally, one can imagine striking similar trade-offs in other ways, such as defining domain-specific subsets of models that reflect situations where improvement is more important.

4 Simulation Studies

We use simulations in three different instances of 3-armed Bernoulli bandits to corroborate our theoretical intuitions from the preceding sections, and compare the behavior of our method (Eq. (7)) against other baselines of interest in Fig. 3. We emphasize that the experiments are not intended to demonstrate the superiority of our algorithm, but rather to examine how different methods succeed or fail in three representative instances, that each highlights the advantages or potential disadvantages of using pessimistic reasoning over $\mathcal{M}'$, as in Eq. (7), in NOF. Representative draws of offline data from each scenario are displayed in Fig. 2. Due to space constraints, comprehensive experiment details and results are included in Appendix C.

Baselines In Fig. 3, we first plot the improvement when $\mathcal{M}' = \{M^*\}$ from Eq. (3), that represents an improvement "ceiling" when the underlying model is known, which we do not expect to outperform. We also plot our method's improvement Eq. (7) against that of the alternative objective in Eq. (5) using the same version space $\mathcal{M}' = \mathcal{M}_{\delta'}(\mathcal{D}^{\text{off}}, \Delta)$. We compare their behavior against other candidate algorithms for NOF: A) the UCB policy, that deterministically plays the arm $\text{argmax}_{a \in \mathcal{A}} \text{UCB}_{\mathcal{D}^{\text{off}}}(a)$; B) setting $\mathcal{M}' = \{\hat{M}\}$ where $r_{\hat{M}}(a) = \hat{r}_{\mathcal{D}^{\text{off}}}(a)$ for all $a \in \mathcal{A}$, that simply explores according to the point estimate of rewards, which is common in empirical offline-online papers such as in Matsushima et al. (2020); C) continuing to collect data from the offline distribution, i.e., $\mu^{\text{on}} = \mu^{\text{off}}$; and D) setting $\mu^{\text{on}} = \text{unif}(\mathcal{A})$ to uniformly sample actions.

Discussion Both Eq. (7) and its alternative version from Eq. (5) obtain significantly more improvement than either continuing to collect data via μ^{off} for different values of n^{on} (pink), or uniformly collecting online data (brown); we expect this difference to grow as the MDP complexity (e.g., number

[11] Since δ' is often used in the concentration inequalities for constructing the version spaces, we cannot directly tune δ' based on $\text{OBJ}(\mu^{\text{on}}, \delta'; \Delta)$: the latter depends on the randomness in $\mathcal{D}^{\text{off}}$, which invalidates the concentration guarantees. To circumvent the issue one can consider techniques such as sample splitting.

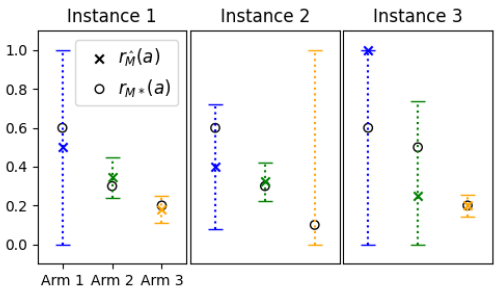

Figure 2: Representative draws of $\mathcal{D}^{\text{off}}$ in our three Bernoulli MAB case studies, with parameters in Table 1. The dashed lines display the rectangular version space $\mathcal{M}_{\delta'}(\mathcal{D}^{\text{off}})$ with $\delta' = 0.05$.

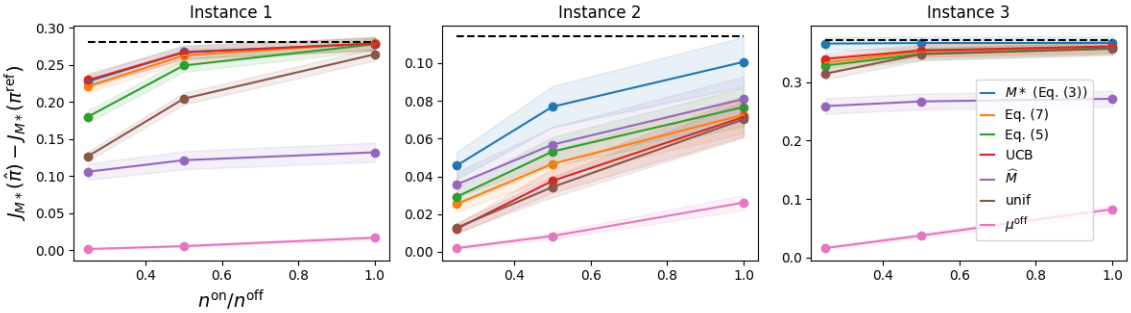

Figure 3: Comparison of methods in Eq. (7) and Eq. (5) against baselines, with fixed $\Delta = 0.3$ and $\delta = \delta' = 0.05$. Shaded region shows ± 1 standard error over 100 random draws of $\mathcal{D}^{\text{off}}$. Dashed black line is maximum possible improvement (averaged over draws of $\mathcal{D}^{\text{off}}$).

of arms in MABs) increases. They also reach the "ceilings" of $\mathcal{M} = \{M^*\}$ (blue) and the maximum possible improvement (black dashed) as the available n^{on} increases. While UCB (red) can perform competitively when the arm it pulls happens to be better than π^{ref} (e.g., Instance 3), it can just as easily underperform significantly when that arm is worse than π^{ref}, which is the case in Instance 2. The version space pessimism in Eq. (5) is crucial for ensuring that any policy improvement predicted from $\mathcal{D}^{\text{off}}$ translates to real improvement in M^* when samples are collected from μ^{on}. This can lead to conservative behavior in specific problem instances, but it also ensures improvement in the worst-case problem instance; by this metric our method outperforms UCB.

Another case in point can be in seen Instance 1 and Instance 3, where μ^{on} chosen according to $\hat{M}$ (purple) is highly suboptimal. Because μ^{off} rarely pulls the arms that are better than a^{ref}, their estimated means in $\hat{M}$ can deviate significantly from in M^*, which is a pitfall that Eq. (7) is robust to since it considers the worst-case over the version space. Conversely, Instance 2 shows that there are scenarios where using pessimism can be disadvantageous. Our method has slightly worse improvement than choosing μ^{on} via $\hat{M}$ because the worst arm a_3 is never pulled in offline data (and by design choice $r_{\hat{M}}(a_3) = 0$ as default). Thus, exploring via $\hat{M}$ will just use online samples to differentiate a_1 from a_2, while Eq. (7) uses them to explore all arms.

In summary, our method may have small reductions in improvement for some instances where less conservative methods may opportunistically do better. However, these less conservative methods perform highly suboptimally in other scenarios, where both Eq. (7) and Eq. (5)'s use of pessimism guarantees they will improve over π^{ref} with high probability. This guarantee will in fact hold for *any* instance (see Theorem 4 Claim (2)).

Conclusion & Future work We have defined a concrete and representative problem setup for non-adaptive online finetuning (NOF), whose goal is to output a policy that improves as much as possible over a (purely offline) reference policy given a single online deployment. We designed and analyzed an information-theoretic algorithm (Eq. (7)) for improvement maximization. As the current implementation of Eq. (7) iterates over all candidate online distributions and models, one important direction of future work involves developing a computationally efficient algorithm. Another involves deriving lower-bounds on Eq. (7)'s expected improvement, which will require novel proof techniques

as the direction of the bound (improvement $\geq \ldots$) is reversed from the typical RL learning guarantee (suboptimality $\leq \ldots$). Lastly, we plan to extend our results to the multiple-deployment setting, for which our single-deployment results form an important building block.

Broader Impact Statement

As this work is largely theoretical in nature, the potential negative impacts are limited. Rather, our paper aims to direct attention of the RL community towards developing guarantees and analysis for offline-online RL within a practically relevant framework.

References

Mohak Bhardwaj, Tengyang Xie, Byron Boots, Nan Jiang, and Ching-An Cheng. Adversarial model for offline reinforcement learning. *arXiv preprint arXiv:2211.04538*, 2022.

Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning*, 2019.

Ching-An Cheng, Tengyang Xie, Nan Jiang, and Alekh Agarwal. Adversarially trained actor critic for offline reinforcement learning. *International Conference on Machine Learning*, 2022.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

Jiawei Huang, Jinglin Chen, Li Zhao, Tao Qin, Nan Jiang, and Tie-Yan Liu. Towards deployment-efficient reinforcement learning: Lower bound and optimality. In *International Conference on Learning Representations*, 2021.

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low bellman rank are pac-learnable. *arXiv preprint arXiv:1610.09512*, 2016.

Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is Q-learning provably efficient? In *Advances in Neural Information Processing Systems*, 2018.

Chi Jin, Qinghua Liu, and Sobhan Miryoosefi. Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. In *Advances in Neural Information Processing Systems*, 2021.

Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline rl? *arXiv preprint arXiv:2012.15085*, 2020.

Allison Koenecke, Andrew Nam, Emily Lake, Joe Nudell, Minnie Quartey, Zion Mengesha, Connor Toups, John R Rickford, Dan Jurafsky, and Sharad Goel. Racial disparities in automated speech recognition. 2020.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

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.

Gen Li, Wenhao Zhan, Jason D Lee, Yuejie Chi, and Yuxin Chen. Reward-agnostic fine-tuning: Provable statistical benefits of hybrid reinforcement learning. *arXiv preprint arXiv:2305.10282*, 2023.

Tatsuya Matsushima, Hiroki Furuta, Yutaka Matsuo, Ofir Nachum, and Shixiang Gu. Deployment-efficient reinforcement learning via model-based offline optimization. *arXiv preprint arXiv:2006.03647*, 2020.

Tom Le Paine, Cosmin Paduraru, Andrea Michi, Caglar Gulcehre, Konrad Zolna, Alexander Novikov, Ziyu Wang, and Nando de Freitas. Hyperparameter selection for offline reinforcement learning. *arXiv preprint arXiv:2007.09055*, 2020.

Vianney Perchet, Philippe Rigollet, Sylvain Chassang, and Erik Snowberg. Batched bandit problems. 2016.

Paria Rashidinejad, Banghua Zhu, Cong Ma, Jiantao Jiao, and Stuart Russell. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Advances in Neural Information Processing Systems*, 34:11702–11716, 2021.

Yuda Song, Yifei Zhou, Ayush Sekhari, J Andrew Bagnell, Akshay Krishnamurthy, and Wen Sun. Hybrid rl: Using both offline and online data can make rl efficient. *arXiv preprint arXiv:2210.06718*, 2022.

Andrew Wagenmaker and Aldo Pacchiano. Leveraging offline data in online reinforcement learning. In *International Conference on Machine Learning*, pp. 35300–35338. PMLR, 2023.

Andrew Wagenmaker, Yifang Chen, Max Simchowitz, Simon S Du, and Kevin Jamieson. Reward-free RL is no harder than reward-aware RL in linear markov decision processes. *arXiv:2201.11206*, 2022.

Tengyang Xie, Ching-An Cheng, Nan Jiang, Paul Mineiro, and Alekh Agarwal. Bellman-consistent pessimism for offline reinforcement learning. *Advances in neural information processing systems*, 34, 2021a.

Tengyang Xie, Nan Jiang, Huan Wang, Caiming Xiong, and Yu Bai. Policy finetuning: Bridging sample-efficient offline and online reinforcement learning. *Advances in neural information processing systems*, 34:27395–27407, 2021b.

Ruiqi Zhang and Andrea Zanette. Policy finetuning in reinforcement learning via design of experiments using offline data. *arXiv preprint arXiv:2307.04354*, 2023.

Siyuan Zhang and Nan Jiang. Towards hyperparameter-free policy selection for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 34:12864–12875, 2021.

A Related Work

Xie et al. (2021b); Song et al. (2022); Wagenmaker & Pacchiano (2023); Li et al. (2023); Zhang & Zanette (2023) all consider with variants of the offline-online RL setting, where offline data is available as well as online interaction.

Of these papers, all but Zhang & Zanette (2023) allow unlimited deployments, which separates them from our problem setting. Specifically, Song et al. (2022) assumes a bilinear MDP, and that π^* is covered by offline data. The online phase of their algorithm learns policies via Fitted Q iteration, which is warm-started with the offline dataset. They demonstrate the learned policy is near-optimal, and their main contribution is a computationally efficient algorithm. Wagenmaker et al. (2022) works with linear MDPs, and develops a notion of offline-online complexity for learning π^* when warm-starting with offline data. In contrast, Xie et al. (2021b) and Li et al. (2023) work with tabular MDPs, but only assume that the offline dataset satisfies a notion of partial π^* coverage. The former considers the finite-horizon MDP setting where π^* is covered up until a specific timestep, whereas the latter defines partial coverage on a per-state-action basis. They propose algorithms that compete with the optimal policy, with Zhang & Zanette (2023) obtaining better sample complexity than either offline or online RL alone.

Zhang & Zanette (2023) also consider the single-deployment setting, but they require a large number of online samples and seek to learn π^* via reward-free exploration. In contrast, our setting focuses on a fixed number of online samples, and we seek only to find a better policy, not the optimal one. This is the other major difference between our setting, and previous related works (beyond the issue of deployments).

Lastly, we note that the single-deployment offline-online RL problem is also related to deployment-efficient RL (Huang et al., 2021) as well as batched bandits (Perchet et al., 2016) (in MABs, our problem setting corresponds to a single batch, but with additional information from logged data).

B Proofs

Proof of Theorem 2. The inequality in the proposition statement follows directly from the inclusion of $M^* \in \mathcal{M}'$ and the definition of OBJ in Eq. (4):

$$\begin{aligned}\mathbb{E}_{\mathcal{D}^{\text{on}}}\left[J_{M^*}(\hat{\pi}_{\mathcal{D}^{\text{on}}}) - J_{M^*}(\pi^{\text{ref}})\right] &= \mathbb{E}_{\mathcal{D}^{\mu^{\text{on}}}_{M^*}}\left[J_{M^*}(\hat{\pi}_{\mathcal{D}^{\mu^{\text{on}}}_{M^*}}) - J_{M^*}(\pi^{\text{ref}})\right] \\ &\geq \min_{M' \in \mathcal{M}'} \mathbb{E}_{\mathcal{D}^{\mu^{\text{on}}}_{M'}}\left[J_{M^*}(\hat{\pi}_{\mathcal{D}^{\mu^{\text{on}}}_{M'}}) - J_{M^*}(\pi^{\text{ref}})\right] = \text{OBJ}(\mu^{\text{on}}, \mathcal{M}')\end{aligned}$$

The equality is from the definition of $\mu^{\text{on}} \in \text{argmax}_{\mu \in \Pi} \text{OBJ}(\mu, \mathcal{M}')$ from Eq. (4).

□

Proof of Theorem 3. For any draw of $\mathcal{D}^{\mu}_{M'_0}$ we have $J_{M'_0}(\hat{\pi}_{\mathcal{D}^{\mu}_{M'_0}}) - J_{M'_0}(\pi^{\text{ref}}) \leq 0$ since $\pi^{\text{ref}} \in \text{argmax}_{\pi \in \Pi} J_{M'_0}(\pi)$. On the event that $M'_0 \in \mathcal{M}_\delta(\mathcal{D}^{\text{off}} \cup \mathcal{D}^{\mu}_{M'_0})$, we additionally have $J_{M'_0}(\hat{\pi}_{\mathcal{D}^{\mu}_{M'_0}}) - J_{M'_0}(\pi^{\text{ref}}) = 0$ since

$$\begin{aligned}J_{M'_0}(\hat{\pi}_{\mathcal{D}^{\mu}_{M'_0}}) - J_{M'_0}(\pi^{\text{ref}}) &\geq \min_{M \in \mathcal{M}_\delta(\mathcal{D}^{\text{off}} \cup \mathcal{D}^{\mu}_{M})} J_M(\hat{\pi}_{\mathcal{D}^{\mu}_{M}}) - J_M(\pi^{\text{ref}}) \\ &= \max_{\pi \in \Pi} \min_{M \in \mathcal{M}_\delta(\mathcal{D}^{\text{off}} \cup \mathcal{D}^{\mu}_{M})} J_M(\pi) - J_M(\pi^{\text{ref}}) \geq 0,\end{aligned}$$

where the last inequality is because $\pi^{\text{ref}} \in \Pi$. Then since $M_0' \in \mathcal{M}_{\delta'}(\mathcal{D}^{\text{off}})$ and using the law of total expectation,

$$
\begin{aligned}
&|\text{OBJ}(\mu, \delta')| \\
&\leq \left| \mathbb{E}_{\mathcal{D}^{\mu}_{M_0'}} \left[J_{M_0'}(\hat{\pi}_{\mathcal{D}^{\mu}_{M_0'}}) - J_{M_0'}(\pi^{\text{ref}}) \right] \right| \\
&\leq \left| \mathbb{E}_{\mathcal{D}^{\mu}_{M_0'}} \left[J_{M_0'}(\hat{\pi}_{\mathcal{D}^{\mu}_{M_0'}}) - J_{M_0'}(\pi^{\text{ref}}) \mid M_0' \in \mathcal{M}_{\delta}(\mathcal{D}^{\text{off}} \cup \mathcal{D}^{\mu}_{M_0'}) \right] \right| \mathbb{P}_{\mathcal{D}^{\mu}_{M_0'}}[M_0' \in \mathcal{M}_{\delta}(\mathcal{D}^{\text{off}} \cup \mathcal{D}^{\mu}_{M_0'})] \\
&+ \left| \mathbb{E}_{\mathcal{D}^{\mu}_{M_0'}} \left[J_{M_0'}(\hat{\pi}_{\mathcal{D}^{\mu}_{M_0'}}) - J_{M_0'}(\pi^{\text{ref}}) \mid M_0' \notin \mathcal{M}_{\delta}(\mathcal{D}^{\text{off}} \cup \mathcal{D}^{\mu}_{M_0'}) \right] \right| \mathbb{P}_{\mathcal{D}^{\mu}_{M_0'}}[M_0' \notin \mathcal{M}_{\delta}(\mathcal{D}^{\text{off}} \cup \mathcal{D}^{\mu}_{M_0'})] \\
&= \left| \mathbb{E}_{\mathcal{D}^{\mu}_{M_0'}} \left[J_{M_0'}(\hat{\pi}_{\mathcal{D}^{\mu}_{M_0'}}) - J_{M_0'}(\pi^{\text{ref}}) \mid M_0' \notin \mathcal{M}_{\delta}(\mathcal{D}^{\text{off}} \cup \mathcal{D}^{\mu}_{M_0'}) \right] \right| \mathbb{P}_{\mathcal{D}^{\mu}_{M_0'}}[M_0' \notin \mathcal{M}_{\delta}(\mathcal{D}^{\text{off}} \cup \mathcal{D}^{\mu}_{M_0'})] \\
&\leq \mathbb{P}_{\mathcal{D}^{\mu}_{M_0'}}[M_0' \notin \mathcal{M}_{\delta}(\mathcal{D}^{\text{off}} \cup \mathcal{D}^{\mu}_{M_0'})]
\end{aligned}
$$

since $J_{M_0'}(\pi) \in [0, 1]$ for any π. □

Proof of Theorem 4. First, we prove the statement in (1). Since $\hat{\pi}$ is learned from $\mathcal{D}^{\text{off}} \cup \mathcal{D}^{\text{on}}$, from the ARMOR guarantee in Theorem 1 we have that $J_{M^*}(\hat{\pi}) \geq J_{M^*}(\pi^{\text{ref}})$ with probability $\geq 1 - \delta$ with respect to the randomness of $\mathcal{D}^{\text{off}} \cup \mathcal{D}^{\text{on}}$. Then if $\Delta(M^*) = \max_{\pi \in \Pi} J_{M^*}(\pi) - J_{M^*}(\pi^{\text{ref}}) < \Delta$, we have

$$
J_{M^*}(\hat{\pi}) \geq J_{M^*}(\pi^{\text{ref}}) > \max_{\pi \in \Pi} J_{M^*}(\pi) - \Delta.
$$

Next, we prove (2). Fix $\mathcal{D}^{\text{off}}$. When $M^* \in \mathcal{M}_{\delta'}(\mathcal{D}^{\text{off}}, \Delta)$,

$$
\begin{aligned}
\mathbb{E}_{\mathcal{D}^{\text{on}}}[J_{M^*}(\hat{\pi}) - J_{M^*}(\pi^{\text{ref}})] &= \mathbb{E}_{\mathcal{D}^{\text{on}}_{M^*}}[J_{M^*}(\hat{\pi}_{\mathcal{D}^{\text{on}}_{M^*}}) - J_{M^*}(\pi^{\text{ref}})] \\
&\geq \min_{M' \in \mathcal{M}_{\delta'}(\mathcal{D}^{\text{off}}, \Delta)} \mathbb{E}_{\mathcal{D}^{\text{on}}_{M'}}[J_{M'}(\hat{\pi}_{\mathcal{D}^{\text{on}}_{M'}}) - J_{M'}(\pi^{\text{ref}})] \\
&= \text{OBJ}(\mu^{\text{on}}, \delta'; \Delta).
\end{aligned}
$$

The theorem statement follows from the fact that $M^* \in \mathcal{M}_{\delta'}(\mathcal{D}^{\text{off}}, \Delta)$ with probability $\geq 1 - \delta'$ (with respect to the randomness of $\mathcal{D}^{\text{off}}$) from Definition 1 and (6) if $\Delta(M^*) \geq \Delta$. □

C Implementation Details and Additional Results for Section 4

	M^*	n^{off}	μ^{off}
Instance 1	$(0.6, 0.3, 0.2)$	200	$[0.01, 0.495, 0.495]$
Instance 2	$(0.6, 0.3, 0.1)$	100	$[0.1, 0.9, 0.0]$
Instance 3	$(0.6, 0.5, 0.2)$	200	$[0.01, 0.01, 0.98]$

Table 1: Parameters for Bernoulli MAB case study instances.

MAB Instances We analyze the behavior of our method in three Bernoulli MAB case studies, with parameters displayed in Table 1. Representative draws of the offline dataset and the corresponding version spaces $\mathcal{M}_{\delta'}(\mathcal{D}^{\text{off}})$ are shown in Fig. 2. Briefly, Instance 1 is the easiest problem instance because μ^{off} eliminates the worse arm, and the goal of NOF is to explore the remaining arm (that is optimal).

Instance 2 and Instance 3 are designed to express the tradeoff between conservatism/pessimism and the potential for improvement in M^*. Instance 2 is an instance where pessimism is crucial for

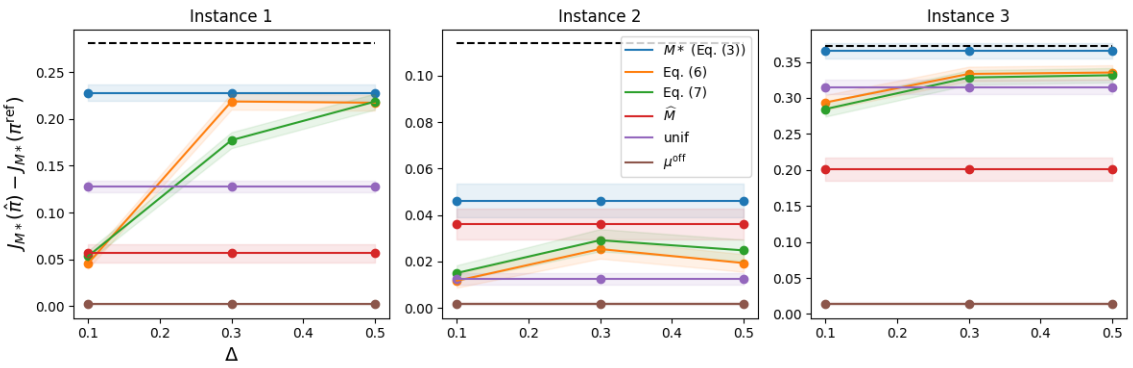

Figure 4: Improvement in in MAB instances for different values of Δ, with $\delta = \delta' = 0.05$ and $n^{\text{on}} = 0.25n^{\text{off}}$ fixed. As before, confidence bands show ± 1 standard error.

improvement, and an algorithm must explore all candidate arms (recall Example 2) in order to guarantee improvement in M^*. The offline data primarily covers the middle arm, but the UCB for the worst arm tends to be larger than the UCB for the better arm. As a result, a less-conservative algorithm (e.g., larger Δ) runs the risk of exploring only the worst arm, which will lead to no improvement. Finally, we note that, as can be seen in Fig. 3 **(middle)**, Instance 2 has larger confidence bands over draws of $\mathcal{D}^{\text{off}}$ because π^{ref} chooses the optimal arm a larger portion of the time from $\mathcal{D}^{\text{off}}$. This is a consequence of the bandit instance design and not our algorithm quality (in fact Instance 2 has the highest probability out of all three-armed bandit instances to exhibit the desired quality of having $\text{UCB}(a_1) > \text{UCB}(a_3)$ and $a^{\text{ref}} = a_2$).

In comparison, the offline data in Instance 3 largely pulls the worst arm, leaving the two better arms as candidates, and the optimal arm generally has the largest UCB. An algorithm for NOF can improve by pulling either of the two better arms. Here, an extreme choice of Δ can expect to do well, while acting conservatively may be empirically less effective because it will unnecessarily explore both arms.

Implementation Details We build the version spaces as specified in Example 1, except that for tighter practical confidence bounds for a given dataset $\mathcal{D}$ and δ we set $\text{LCB}_{\mathcal{D}}(a) = \Phi^{-1}(1-\delta/2)\cdot\text{SE}_{\mathcal{D}}(a)$ and $\text{UCB}_{\mathcal{D}}(a) = \Phi^{-1}(\delta/2)\cdot\text{SE}_{\mathcal{D}}(a)$, where Φ^{-1} is the inverse CDF of the standard Guassian, and $\text{SE}_{\mathcal{D}}(a)$ is the standard error of the rewards observed for a given a. For a fixed $\mathcal{D}^{\text{off}}$, π^{ref} is a deterministic policy learned via LCB, i.e., $\pi^{\text{ref}}(a) = \mathbf{1}[a = a']$ where $a' = \text{argmax}_{a\in\mathcal{A}} \text{LCB}_{\mathcal{D}^{\text{off}}}(a)$. We set Π to be the set of all valid distributions over the arms $\mathcal{A}$, and $\mathcal{M} = [0,1]^{\mathcal{A}}$ to be the set of all models with rewards bounded on the unit interval. Because both Π and $\mathcal{M}$ classes with infinite cardinality, in our implementation we discretize the sets to a grid of 0.05 and search over the resulting set, which results in negligible approximation error as realizability is still satisfied. We approximate $\mathbb{E}_{\mathcal{D}^{\mu}_{M'}}$ in the inner loop of Eq. (7) using 200 random draws of data. All simulations were run on a personal laptop. Generating the results for Fig. 1 took roughly 1-2 hours, while generating the results for Fig. 3, Fig. 4, and Fig. 5 took roughly 1-2 days combined. Results, code, and instructions for running are included in the supplementary material.

Additional Results We also discuss additional results related to the effect of Δ and δ' on performance, in a similar vein to Fig. 3. Aligned with our predictions regarding pessimism from the design of the MAB problem instances, Fig. 4 demonstrates that Instance 1 and Instance 3 do not degrade in performance and even improve slightly with larger Δ (that excludes $M*$ from $\mathcal{M}_{\delta'}(\mathcal{D}^{\text{off}}, \Delta)$), but Instance 2 does. Thus, while a less conservative choice of Δ may lead to good practical performance on a case-by-case basis, a value of Δ that preserves realizability is required in order to guarantee improvement in any given instance.

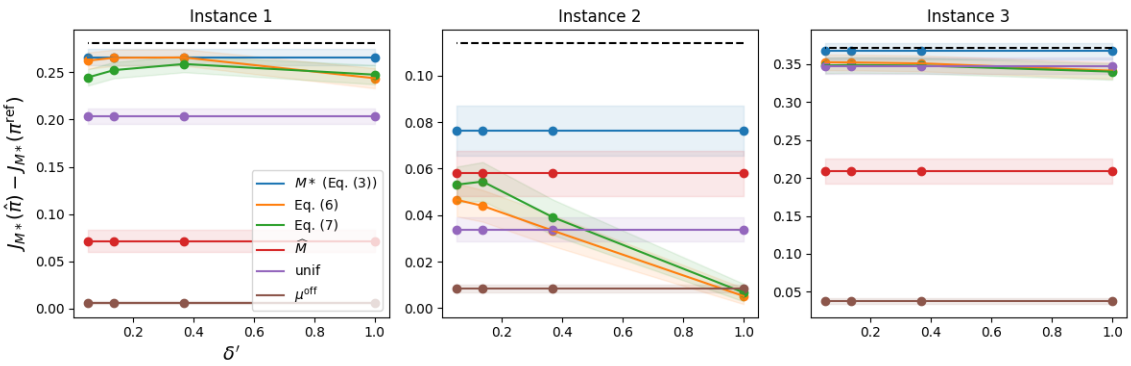

Figure 5: Improvement in in MAB instances for different values of δ', with $\delta = 0.05$, $\Delta = 0.3$, and $n^{\text{on}} = 0.5n^{\text{off}}$ fixed. As before, confidence bands show ± 1 standard error.

Fig. 5 displays the sensitivity of improvement to choice of δ'. While Instance 1 and especially Instance 3 are relatively robust to different values of δ', the algorithm degrades significantly in performance for Instance 2 as δ' increases (and the probability of $M^* \in \mathcal{M}_{\delta'}(\mathcal{D}^{\text{off}}, \Delta)$ being satisfied decreases). The reason for this is related to the above; that Instance 1 and Instance 3 are problems where less conservative behavior can be rewarding, but Instance 2 is an instance where it is extremely punishing.