IS OFFLINE DECISION MAKING POSSIBLE WITH ONLY FEW SAMPLES? RELIABLE DECISIONS IN DATA-STARVED BANDITS VIA TRUST REGION ENHANCEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

What can an agent learn in a stochastic Multi-Armed Bandit (MAB) problem from a dataset that contains just a single sample for each arm? Surprisingly, in this work, we demonstrate that even in such a data-starved setting it may still be possible to find a policy competitive with the optimal one. This paves the way to reliable decision-making in settings where critical decisions must be made by relying only on a handful of samples. Our analysis reveals that *stochastic policies can be substantially better* than deterministic ones for offline decision-making. Focusing on offline multi-armed bandits, we design an algorithm called Trust Region of Uncertainty for Stochastic policy enhancemenT (TRUST) which is quite different from the predominant value-based lower confidence bound approach. Its design is enabled by localization laws, critical radii, and relative pessimism. We prove that its sample complexity is comparable to that of LCB on minimax problems while being substantially lower on problems with very few samples. Finally, we consider an application to offline reinforcement learning in the special case where the logging policies are known.

1 INTRODUCTION

In several important problems, critical decisions must be made with just very few samples of pre-collected experience. For example, collecting samples in robotic manipulation may be slow and costly, and the ability to learn from very few interactions is highly desirable (Hester & Stone, 2013; Liu et al., 2021). Likewise, in clinical trials and in personalized medical decisions, reliable decisions must be made by relying on very small datasets (Liu et al., 2017). Sample efficiency is also key in personalized education (Bassen et al., 2020; Ruan et al., 2023). However, to achieve good performance, the state-of-the-art algorithms may require millions of samples (Fu et al., 2020). These empirical findings seem to be supported by the existing theories: the sample complexity bounds, even minimax optimal ones, can be large in practice due to the large constants and the warmup factors (Ménard et al., 2021; Li et al., 2022; Azar et al., 2017; Zanette et al., 2019).

In this work, we study whether it is possible to make reliable decisions with only a few samples. We focus on an offline Multi-Armed Bandit (MAB) problem, which is a foundation model for decision-making (Lattimore & Szepesvári, 2020). In online MAB, an agent repeatedly chooses an arm from a set of arms, each providing a stochastic reward. Offline MAB is a variant where the agent cannot interact with the environment to gather new information and instead, it must make decisions based on a pre-collected dataset without playing additional exploratory actions, aiming at identifying the arm with the highest expected reward (Audibert et al., 2010; Garivier & Kaufmann, 2016; Russo, 2016; Ameko et al., 2020).

The standard approach to the problem is the Lower Confidence Bound (LCB) algorithm (Rashidinejad et al., 2021), a pessimistic variant of UCB (Auer et al., 2002) that involves selecting the arm with the highest lower bound on its performance. LCB encodes a principle called *pessimism under uncertainty*, which is the foundation principle for most algorithms for offline bandits and reinforcement learning (RL) (Jin et al., 2020; Zanette et al., 2020; Xie et al., 2021; Yin & Wang, 2021; Kumar et al.,

2020; Kostrikov et al., 2021). Unfortunately, the available methods that implement the principle of pessimism under uncertainty can fail in a data-starved regime because they rely on confidence intervals that are too loose when just a few samples are available. For example, even on a simple MAB instance with ten thousand arms, the best-known (Rashidinejad et al., 2021) performance bound for the LCB algorithm requires 24 samples per arm in order to provide meaningful guarantees, see Section 2. In more complex situations, such as in the sequential setting with function approximation, such a problem can become more severe due to the higher metric entropy of the function approximation class and the compounding of errors through time steps.

These considerations suggest that there is a "barrier of entry" to decision-making, both theoretically and practically: one needs to have a substantial number of samples in order to make reliable decisions even for settings as simple as offline MAB where the guarantees are tighter. Given the above technical reasons, and the lack of good algorithms and guarantees for data-starved decision problems, it is unclear whether it is even possible to find good decision rules with just a handful of samples.

In this paper, we make a substantial contribution towards lowering such barriers of entry. We discover that a carefully-designed algorithm tied to an advanced statistical analysis can substantially improve the sample complexity, both theoretically and practically, and enable reliable decision-making with just a handful of samples. More precisely, we focus on the offline MAB setting where we show that even if the dataset contains just a *single sample* in every arm, it may still be possible to compete with the optimal policy. This is remarkable, because with just one sample per arm—for example from a Bernoulli distribution—it is impossible to estimate the expected payoff of any of the arms! Our discovery is enabled by several key insights:

- We search over *stochastic* policies, which can yield better performance for offline-decision making;
- We use a *localized* notion of metric entropy to carefully control the size of the stochastic policy class that we search over;
- We implement a concept called *relative pessimism* to obtain sharper guarantees.

These considerations lead us to design a trust region policy optimization algorithm called Trust Region of Uncertainty for Stochastic policy enhancemenT (TRUST), one that offers superior theoretical as well as empirical performance compared to LCB in a data-scarce situation.

Moreover, we apply the algorithm to selected reinforcement learning problems from (Fu et al., 2020) in the special case where information about the logging policies is available. We do so by a simple reduction from reinforcement learning to bandits, by mapping policies and returns in the former to actions and rewards in the latter, thereby disregarding the sequential aspect of the problem. Although we rely on the information of the logging policies being available, the empirical study shows that our algorithm compares well with a strong deep reinforcement learning baseline (i.e, CQL from (Kumar et al., 2020)), without being sensitive to partial observability, sparse rewards, and hyper-parameters.

2 Data-Starved Multi-Armed Bandits

In this section, we describe the MAB setting and give an example of a "data-starved" MAB instance where prior methods (such as LCB) can fail. We informally say that an offline MAB is "data-starved" if its dataset contains only very few samples in each arm.

Notation. We let $[n] = \{1, 2, ..., n\}$ for a positive integer n. We let $\|\cdot\|_2$ denote the Euclidean norm for vectors and the operator norm for matrices. We hide constants and logarithmic factors in the $\widetilde{O}(\cdot)$ notation. We let $\mathbb{B}_p^d(s) = \{x \in \mathbb{R}^d : \|x\|_p \leq s\}$ for any $s \geq 0$ and $p \geq 1$. $a \lesssim b$ ($a \gtrsim b$) means $a \leq Cb$ ($a \geq Cb$) for some numerical constant C. $a \simeq b$ means that both $a \lesssim b$ and $b \lesssim a$ hold.

Multi-armed bandits. We consider the case where there are d arms in a set $\mathcal{A} = \{a_1, ..., a_d\}$ with expected reward $r(a_i), i \in [d]$. We assume access to an offline dataset $\mathcal{D} = \{(x_i, r_i)\}_{i \in [N]}$ of action-reward tuples, where the experienced actions $\{x_i\}_{i \in [N]}$ are i.i.d. from a distribution μ. Each experienced reward is a random variable with expectation $\mathbb{E}[r_i] = r(x_i)$ and independent Gaussian noises $\zeta_i := r(x_i) - \mathbb{E}[r_i]$. For $i \in [d]$, we denote the number of pulls to arm a_i in $\mathcal{D}$ by $N(a_i)$ or N_i, while the variance of the noise for arm a_i is denoted by σ_i^2. We denote the optimal arm

as $a^* \in \arg\max_{a \in \mathcal{A}}[r(a)]$ and the single policy concentrability as $C^* = 1/\mu(a^*)$ where μ is the distribution that generated the dataset. Without loss of generality, we assume the optimal arm is unique. We also write $r = (r_1, r_2, ..., r_d)^\top$. Without loss of generality, we assume there is at least one sample for each arm (such arm can otherwise be removed).

Lower confidence bound algorithm. One simple but effective method for the offline MAB problem is the Lower Confidence Bound (LCB) algorithm, which is inspired by its online counterpart (UCB) (Auer et al., 2002). Like UCB, LCB computes the empirical mean $\widehat{r}_i$ associated to the reward of each arm i along with its half confidence width b_i. They are defined as

$$\widehat{r}_i := \frac{1}{N(a_i)} \sum_{k:x_k=a_i} x_k,\ b_i := \sqrt{\frac{2\sigma_i^2}{N(a_i)} \log\left(\frac{2d}{\delta}\right)}. \tag{1}$$

This definition ensures that each confidence interval brackets the corresponding expected reward with probability $1-\delta$:

$$\widehat{r}_i - b_i \leq r\left(a_i\right) \leq \widehat{r}_i + b_i \quad \forall i \in [d]. \tag{2}$$

The width of the confidence level depends on the noise level σ_i, which can be exploited by variance-aware methods (Zhang et al., 2021; Min et al., 2021; Yin et al., 2022; Dai et al., 2022). When the true noise level is not accessible, we can replace it with the empirical standard deviation or with a high-probability upper bound. For example, when the reward for each arm is restricted to be within $[0, 1]$, a simpler upper bound is $\sigma_i^2 \leq 1/4$.

Unlike UCB, the half-width of the confidence intervals for LCB is not added, but subtracted, from the empirical mean, resulting in the lower bound $l_i = \widehat{r}_i - b_i$. The action identified by LCB is then the one that maximizes the resulting lower bound, thereby incorporating the principle of pessimism under uncertainty (Jin et al., 2020; Kumar et al., 2020). Specifically, given the dataset $\mathcal{D}$, LCB selects the arm using the following rule:

$$\widehat{a}_{\mathsf{LCB}} := \arg\max_{a_i \in \mathcal{A}}\ l_i, \tag{3}$$

(Rashidinejad et al., 2021) analyzed the LCB strategy. Below we provide a modified version of their theorem.

Theorem 2.1 (LCB Performance)**.** *Suppose the noise of arm a_i is sub-Gaussian with proxy variance σ_i^2. Let $\delta \in (0, 1/2)$. Then, we have*

1. *(Comparison with any arm) With probability at least $1-\delta$, for any comparator policy $a_i \in \mathcal{A}$, it holds that $r(a_i) - r(\widehat{a}_{\mathsf{LCB}}) \leq \sqrt{8\sigma_i^2 \log(2d/\delta)/N(a_i)}$.*
2. *(Comparison with the optimal arm) Assume $\sigma_i = 1$ for any $i \in [d]$ and $N \geq 8C^* \log(1/\delta)$. Then, with probability at least $1-2\delta$, one has $r(a^*) - r(\widehat{a}_{\mathsf{LCB}}) \leq \sqrt{4C^* \log(2d/\delta)/N}$.*

The statement of this theorem is slightly different from that in (Rashidinejad et al., 2021), in the sense that their suboptimality is over $\mathbb{E}_{\mathcal{D}}[r\left(a^*\right) - r\left(\widehat{a}_{\mathsf{LCB}}\right)]$ instead of a high-probability one. (Rashidinejad et al., 2021) proved the minimax optimality of the algorithm when the single policy concentrability $C^* \geq 2$ and the sample size $N \geq \widetilde{O}(C^*)$.

A data-starved MAB problem and failure of LCB. In order to highlight the limitation of a strategy such as LCB, let us describe a specific data-starved MAB instance, specifically one with $d = 10000$ arms, equally partitioned into a set of *good arms* (i.e., $\mathcal{A}_g$) and a set of *bad arms* (i.e., $\mathcal{A}_b$). Each good arm returns a reward following the uniform distribution over $[0.5, 1.5]$, while each bad arm returns a reward which follows $\mathcal{N}(0, 1/4)$.

Assume that we are given a dataset that contains *only one sample per each arm*. Instantiating the LCB confidence interval in equation 2 with $\sigma_i \leq 1/2$ and $\delta = 0.1$, one obtains $\widehat{r}_i - 2.5 \leq r(a_i) \leq \widehat{r}_i + 2.5$. Such bound is uninformative, because the lower bound for the true reward mean is less than the reward value of the worst arm. The performance bound for LCB confirms this intuition, because Theorem 2.1 requires at least $N(a_i) \geq \lceil 8 * \log(1/0.05) \rceil = 24$ samples in each arm to provide any guarantee with probability at least 0.9 (here $C^* = d$).

Can stochastic policies help? At a first glance, extracting a good decision-making strategy for the problem discussed in Section 2 seems like a hopeless endeavor, because it is information-theoretically impossible to reliably estimate the expected payoff of any of the arms with just a single sample on each. In order to proceed, the key idea is to enlarge the search space to contain *stochastic policies*.

Definition 2.2 (Stochastic Policies)**.** A stochastic policy over a MAB is a probability distribution $w \in \mathbb{R}^d, w_i \geq 0, \sum_{i=1}^d w_i = 1$.

To exemplify how stochastic policies can help, consider the *behavioral cloning* policy, which mimics the policy that generated the dataset for the offline MAB in Section 2. Such policy is stochastic, and it plays all arms uniformly at random, thereby achieving a score around 0.5 with high probability. The value of the behavioral cloning policy can be readily estimated using the Hoeffding bound (e.g., Proposition 2.5 in (Wainwright, 2019)): with probability at least $1 - \delta = 0.9$, (here $d = 10000$ is the number of arms and $\sigma = 1/2$ is the true standard deviation), the value of behavioral cloning policy is greater or equal than $1/2 - \sqrt{2\sigma^2 \log(2/\delta)/d} \approx 0.488$. Such value is higher than the one guaranteed for LCB by Theorem 2.1. Intuitively, a stochastic policy that selects multiple arms can be evaluated more accurately because it averages the rewards experienced over different arms. This consideration suggests optimizing over stochastic policies.

By optimizing a lower bound on the performance of the stochastic policies, it should be possible to find one with a provably high return. Such an idea leads to solving an offline *linear bandit* problem, as follows

$$\max_{w \in \mathbb{R}^d, w_i \geq 0, \sum_{i=1}^d w_i = 1} \sum_{i=1}^d w_i \widehat{r}_i - c(w) \tag{4}$$

where $c(w)$ is a suitable confidence interval for the policy w and $\widehat{r}_i$ is the empirical reward for the i-th arm defined in equation 1. While this approach is appealing, enlarging the search space to include all stochastic policies brings an increase in the metric entropy of the function class, and concretely, a $\sqrt{d}$ factor (Abbasi-Yadkori et al., 2011; Rusmevichientong & Tsitsiklis, 2010; Hazan & Karnin, 2016; Jun et al., 2017; Kim et al., 2022) in the confidence intervals $c(w)$ (in equation 4), which negates all gains that arise from considering stochastic policies. In the next section, we propose an algorithm that bypasses the need for such $\sqrt{d}$ factor by relying on a more careful analysis and optimization procedure.

3 Trust Region of Uncertainty for Stochastic policy enhancemenT (TRUST)

In this section, we introduce our algorithm, called Trust Region of Uncertainty for Stochastic policy enhancemenT (TRUST). At a high level, the algorithm is a policy optimization algorithm based on a trust region centered around a reference policy. The size of the trust region determines the degree of pessimism, and its optimal problem-dependent size can be determined by analyzing the supremum of a problem-dependent empirical process. In the sequel, we describe 1) the decision variables, 2) the trust region optimization program, and 3) some techniques for its practical implementation.

3.1 Decision variables

The algorithm searches over the class of stochastic policies given by the weight vector $w = (w_1, w_2, ..., w_d)^\top$ of Definition 2.2. Instead of directly optimizing over the weights of the stochastic policy, it is convenient to center w around a *reference stochastic policy* $\widehat{\mu}$ which is either known to perform well or is easy to estimate. In our theory and experiments, we consider a simple setup and use the behavioral cloning policy weighted by the noise levels $\{\sigma_i\}$ if they are known. Namely, we consider

$$\widehat{\mu}_i = \frac{N_i/\sigma_i^2}{\sum_{j=1}^d N_j/\sigma_j^2} \quad \forall i \in [d]. \tag{5}$$

When the size of the noise σ_i is constant across all arms, the policy $\widehat{\mu}$ is the behavioral cloning policy; when σ_i differs across arms, $\widehat{\mu}$ minimizes the variance of the empirical reward $\widehat{\mu} = \arg\min_{w \in \mathbb{R}^d, w_i \geq 0, \sum_i w_i = 1} \mathrm{Var}\left(w^\top \cdot \widehat{r}\right)$, where $\widehat{r} = (\widehat{r}_1, ..., \widehat{r}_d)^\top$ is defined in equation 1. Using such definition, we define as *decision variable* the *policy improvement* vector $\Delta := w - \widehat{\mu}$. This preparatory step is key: it allows us to implement **relative pessimism**, namely pessimism on the improvement—represented by Δ—rather than on the absolute value of the policy w. Moreover, by restricting the search space to a ball around $\widehat{\mu}$, one can efficiently reduce the metric entropy of the policy class and obtain tighter confidence intervals.

3.2 TRUST REGION OPTIMIZATION

Trust region. TRUST (Algorithm 1) returns the stochastic policy $\pi_{TRUST} = \widehat{\Delta} + \widehat{\mu} \in \mathbb{R}^d$, where $\widehat{\mu}$ is the reference policy defined in equation 5 and $\widehat{\Delta}$ is the policy improvement vector. In order to accurately quantify the effect of the improvement vector Δ, we constrain it to a trust region $\mathsf{C}(\varepsilon)$ centered around $\widehat{\mu}$ where $\varepsilon > 0$ is the radius of the trust region. More concretely, for a given radius $\varepsilon > 0$, the trust region is defined as

$$\mathsf{C}(\varepsilon) := \Bigg\{ \Delta : \Delta_i + \widehat{\mu}_i \geq 0, \quad \|\Delta + \widehat{\mu}\|_1 = 1, \quad \sum_{i=1}^{d} \Delta_i^2 \frac{\sigma_i^2}{N_i} \leq \varepsilon^2 \Bigg\}. \tag{6}$$

The trust region above serves two purposes: it ensures that the policy $\widehat{\Delta} + \widehat{\mu}$ still represents a valid stochastic policy, and it regularizes the policy around the reference policy $\widehat{\mu}$. We then search for the best policy within $\mathsf{C}(\varepsilon)$ by solving the optimization program

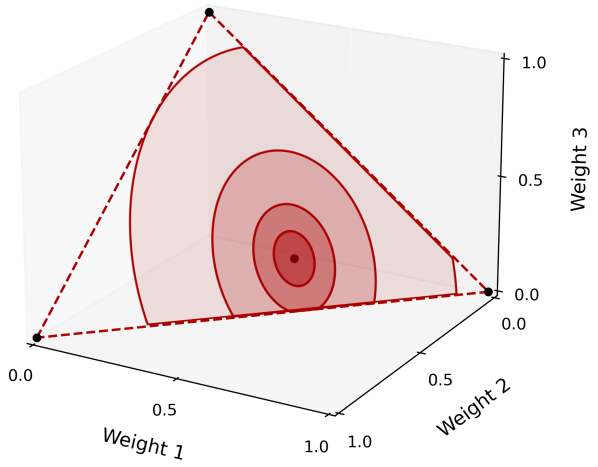

Figure 1: A simple diagram for the trust regions on a 3-dim simplex. The central point is the reference (stochastic) policy, while red ellipses are trust regions around this reference policy.

$$\widehat{\Delta}_\varepsilon := \underset{\Delta \in \mathsf{C}(\varepsilon)}{\arg\max}\, \Delta^\top \widehat{r}. \tag{7}$$

Computationally, the program equation 7 is a second-order cone program (Alizadeh & Goldfarb, 2003; Boyd & Vandenberghe, 2004), which can be solved efficiently with standard off-the shelf libraries (Diamond & Boyd, 2016).

When $\varepsilon = 0$, the trust region only includes the vector $\Delta = 0$, and the reference policy $\widehat{\mu}$ is the only feasible solution. When $\varepsilon \to \infty$, the search space includes all stochastic policies. In this latter case, the solution identified by the algorithm coincides with the greedy algorithm which chooses the arm with the highest empirical return. Rather than leaving ε as a hyper-parameter, in the following we highlight a selection strategy for ε based on localized Gaussian complexities.

Critical radius. The choice of ε is crucial to the performance of our algorithm because it balances optimization with regularization. Such consideration suggests that there is an optimal choice for the radius ε which balances searching over a larger space with keeping the metric entropy of such space under control. The optimal problem-dependent choice $\widehat{\varepsilon}_*$ can be found as a solution of a certain equation involving a problem-dependent *supremum of an empirical process*. More concretely, let E be the feasible set of ε (e.g., $E = \mathbb{R}^+$). We define the critical radius as

Definition 3.1 (Critical Radius)**.** The critical radius $\widehat{\varepsilon}_*$ of the trust region is the solution to the program

$$\widehat{\varepsilon}_* = \underset{\varepsilon \in E}{\arg\max} \left[\widehat{\Delta}_\varepsilon^\top \cdot \widehat{r} - \mathcal{G}(\varepsilon) \right]. \tag{8}$$

Such equation involves a quantile of the localized gaussian complexity $\mathcal{G}(\varepsilon)$ of the stochastic policies identified by the trust region. Mathematically, this is defined as

Definition 3.2 (Quantile of the supremum of Gaussian process)**.** We denote the noise vector as $\eta = \widehat{r} - r$, which by our assumption is coordinate-wise independent and satisfies $\eta_i \sim \mathcal{N}\left(0, \sigma_i^2 / N(a_i)\right)$. We define $\mathcal{G}(\varepsilon)$ as the smallest quantity such that with probability at least $1 - \delta$, for any $\varepsilon \in E$, it holds that $\sup_{\Delta \in \mathsf{C}(\varepsilon)} \Delta^\top \eta \leq \mathcal{G}(\varepsilon)$.

In essence, $\mathcal{G}(\varepsilon)$ is an upper quantile of the supremum of the Gaussian process $\sup_{\Delta \in \mathsf{C}(\varepsilon)} \Delta^\top \eta$ which holds uniformly for every $\varepsilon \in E$. We also remark that this quantity depends on the feasible set E and the trust region $\mathsf{C}(\varepsilon)$, and hence, is highly problem-dependent.

The critical radius plays a crucial role: it is the radius of the trust region that *optimally* balances optimization with uncertainty. Enlarging ε enlarges the search space for Δ, enabling the discovery of policies with potentially higher return. However, this also brings an increase in the metric entropy of the policy class encoded by $\mathcal{G}(\varepsilon)$, which means that each policy can be estimated less accurately. The critical radius represents the optimal tradeoff between these two forces. The final improvement vector that TRUST returns, which we denote as $\widehat{\Delta}_*$, is determined by solving equation 7 with the critical radius $\widehat{\varepsilon}_*$ defined in equation 8. In mathematical terms, we express this as

$$\widehat{\Delta}_* := \underset{\Delta \in \mathsf{C}(\widehat{\varepsilon}_*)}{\arg\max}\, \Delta^\top \widehat{r}. \tag{9}$$

Implementation details. Since it can be difficult to solve equation 8 for a continuous value of $\varepsilon \in E = \mathbb{R}^+$, we use a discretization argument by considering the following candidate subset:

$$E = \left\{ \varepsilon_0, \frac{\varepsilon_0}{\alpha}, ..., \frac{\varepsilon_0}{\alpha^{|E|-1}} \right\}, \tag{10}$$

where $\alpha > 1$ is the decaying rate and ε_0 is the largest possible radius, which is the maximal weighted distance from the reference policy to any vertex. Mathematically, this is defined as $\varepsilon_0 = \max_{i \in [d]} \sqrt{\sum_{j \neq i} \widehat{\mu}_j^2 \sigma_j^2 / N_j + (1 - \widehat{\mu}_i)^2 \sigma_i^2 / N_i}$. Our analysis that leads to Theorem 4.1 takes into account such discretization argument.

In line 2 of Algorithm 1, the algorithm works by estimating the quantile of the supremum of the localized Gaussian complexity $\mathcal{G}(\varepsilon)$ that appears in Definition 3.2, and then choose the ε that maximizes the objective function in equation 8. Although $\mathcal{G}(\varepsilon)$ can be upper bounded analytically, in our experiments we aim to obtain tighter guarantees and so we estimate it via Monte-Carlo. This can be achieved by 1) sampling independent noise vectors η, 2) solving $\sup_{\Delta \in \mathsf{C}(\varepsilon)} \Delta^\top \eta$ and 3) estimating the quantile via order statistics. More details can be found in Appendix D.

In summary, our practical algorithm can be seen as solving the optimization problem

$$(\widehat{\varepsilon}_*, \widehat{\Delta}_*) = \underset{\varepsilon \in E, \Delta \in \mathsf{C}(\varepsilon)}{\arg\max} \left\{ \Delta^\top \widehat{r} - \widehat{\mathcal{G}}(\varepsilon) \right\}$$

where $\widehat{r} \in \mathbb{R}^d$ is the empirical reward vector with $\widehat{r}_i$ defined in equation 1. Here, $\widehat{\mathcal{G}}(\varepsilon)$ is computed according to the Monte-Carlo method defined in Algorithm 2 in Appendix D and E is the candidate subset for radius defined in equation 10. This indicates a balance between the empirical reward of a stochastic policy and the local entropy metric it induces.

Algorithm 1 Trust Region of Uncertainty for Stochastic policy enhancemenT (TRUST)

Input: Offline dataset $\mathcal{D}$, failure probability δ, the candidate set for the trust region widths E (in practice, this is chosen as equation 10).

1. For $\varepsilon \in E$, compute $\widehat{\Delta}_\varepsilon$ from equation 7.
2. For $\varepsilon \in E$, estimate $\mathcal{G}(\varepsilon)$ via Monte-Carlo method (see Algorithm 2 in Appendix D).
3. Solve equation 8 to obtain the critical radius $\widehat{\varepsilon}_*$.
4. Compute the optimal improvement vector in $\mathsf{C}(\widehat{\varepsilon}_*)$ via equation 9, denoted as $\widehat{\Delta}_*$.
5. Return the stochastic policy $\pi_{TRUST} = \widehat{\mu} + \widehat{\Delta}_*$.

4 THEORETICAL GUARANTEES

Problem-dependent analysis In this section, we provide some theoretical guarantees for the policy π_{TRUST} returned by TRUST. We present 1) an improvement over the reference policy $\widehat{\mu}$, 2) a sub-optimality gap with respect to any comparator policy π and 3) an actionable lower bound on the performance of the output policy. Given a stochastic policy π, we let $V^\pi = \mathbb{E}_{a \sim \pi}[r(a)]$ denote its value function. Furthermore, we denote a comparator policy π by a triple $(\varepsilon, \Delta, \pi)$ such that $\varepsilon > 0, \Delta \in \mathsf{C}(\varepsilon), \pi = \widehat{\mu} + \Delta$.

Theorem 4.1 (Main theorem)**.** *TRUST has the following properties.*

1. *With probability at least* $1-\delta$*, the improvement over the behavioral policy is at least*

$$V^{\pi_{TRUST}} - V^{\widehat{\mu}} \geq \sup_{\varepsilon \leq \varepsilon_0, \Delta \in \mathsf{C}(\varepsilon)} \left[\Delta^\top r - 2\mathcal{G}\left(\lceil \varepsilon \rceil\right)\right], \quad \textit{where} \quad \lceil \varepsilon \rceil = \inf\{\varepsilon' \in E, \varepsilon' \geq \varepsilon\}. \tag{11}$$

2. *With probability at least* $1-\delta$*, for any stochastic comparator policy* $(\varepsilon, \Delta, \pi)$*, the sub-optimality of the output policy can be upper bounded as*

$$V^{\pi} - V^{\pi_{TRUST}} \leq 2\mathcal{G}\left(\lceil \varepsilon \rceil\right). \tag{12}$$

3. *With probability at least* $1-2\delta$*, the data-dependent lower bound on* $V^{\pi_{TRUST}}$ *satisfies*

$$V^{\pi_{TRUST}} \geq \pi_{TRUST}^\top \widehat{r} - \mathcal{G}\left(\lceil \widehat{\varepsilon}_* \rceil\right) - \sqrt{\frac{2\log(1/\delta)}{\sum_{j=1}^d N_j/\sigma_j^2}}, \tag{13}$$

where $\pi_{TRUST} = \widehat{\mu} + \widehat{\Delta}_*$ *is the policy output by Algorithm 1.*

The proof of Theorem 4.1 is deferred to Appendix B. A fine-grained analysis for the suboptimality is contained in Appendix E. Our guarantees are problem-dependent as a function of the Gaussian process $\mathcal{G}(\cdot)$; in Section 5 we show how these can be instantiated on an actual problem, highlighting the tightness of the analysis. Equation (11) highlights the improvement with respect to the behavioral policy. It is expressed as a trade-off between maximizing the improvement $\Delta^\top r$ and minimizing its uncertainty $\mathcal{G}\left(\lceil \varepsilon \rceil\right)$. The presence of the $\sup_\varepsilon$ indicates that TRUST achieves an *optimal* balance between these two factors. The state of the art guarantees that we are aware of highlight a trade-off between value and variance (Jin et al., 2021; Min et al., 2021). The novelty of our result lies in the fact that TRUST optimally balances the uncertainty implicitly as a function of the 'coverage' as well as the metric entropy of the search space. That is, TRUST selects the most appropriate search space by trading off its metric entropy with the quality of the policies that it contains. The right-hand side in Equation (13) gives actionable statistical guarantees on the quality of the final policy and it can be fully computed from the available dataset; we give an example of the tightness of the analysis in Section 5.

Localized Gaussian complexity $\mathcal{G}(\varepsilon)$**.** In Theorem 4.1, we upper bound the suboptimality $V^\pi - V^{\pi_{TRUST}}$ via a notion of localized metric entropy $\mathcal{G}(\cdot)$. It is the quantile of the supremum of a Gaussian process, which can hardly be calculated analytically but can be efficiently estimated via Monte Carlo method (which does not collect additional samples, e.g., see Appendix D). It can also be concentrated around its expectation, which is also *localized Gaussian width*, a concept well-established in statistical learning theory (Bellec, 2019; Wei et al., 2020; Wainwright, 2019). More concretely, this is the localized Gaussian width for an affine simplex: $\mathbb{E}[\sup_{\Delta \in \mathsf{C}(\varepsilon)} \Delta^\top \eta] = \mathbb{E}[\sup_{\mathbb{S}^{d-1} \cap \{\Delta : \|\Delta\|_\Sigma \leq \varepsilon\}} \Delta^\top \eta]$, where $\mathbb{S}^{d-1}$ denotes the simplex in $\mathbb{R}^d$ and $\Sigma := \mathrm{diag}\left(\frac{\sigma_1^2}{N_1}, \frac{\sigma_2^2}{N_2}, ..., \frac{\sigma_d^2}{N_d}\right)$ is the weighted matrix. Moreover, this localized Gaussian width can be upper bound via

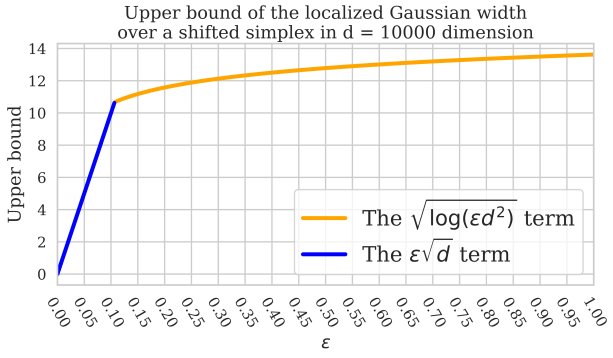

Figure 2: The upper bound for the localized Gaussian width over a shifted simplex on $d = 10000$ dimension. The shifted simplex is $\{\Delta \in \mathbb{R}^d : \sum_{i=1}^d \Delta_i = 0\}$. The two-staged upper bound is based on Theorem 1 in (Bellec, 2019)

$$\mathbb{E}\left[\sup_{\Delta \in \mathsf{C}(\varepsilon)} \Delta^\top \eta\right] \lesssim \min\left\{\sqrt{\log\left(d\varepsilon^2\right)}, \varepsilon\sqrt{d}\right\}. \tag{14}$$

To make it clearer, we plot this upper bound for localized Gaussian width in Figure 2. In equation 14, the rate matches the minimax lower bound up to universal constant (Gordon et al., 2007; Lecué & Mendelson, 2013; Bellec, 2019). To see the implication of the upper bound equation 14, let's consider a simple example where the logging policy is uniform over all arms. We denote the optimal arm as a^* and define $C^* := 1/\mu(a^*)$ as the concentrability coefficient. By applying equation 14 and some concentration techniques (see Wainwright, 2019), we can perform a fine-grained analysis for the suboptimality induced by π_{TRUST}. Specifically, with probability at least $1-\delta$, one has

$$V^{\pi_*} - V^{\pi_{TRUST}} \lesssim \sqrt{C^* \log(2d|E|/\delta)/N}. \tag{15}$$

Note that, the high-probability upper bound here is minimax optimal up to constant and logarithmic factor (Rashidinejad et al., 2021) when $C^* \geq 2$. Moreover, this example of uniform logging policy is an instance where LCB achieves minimax sub-optimality (up to constant and log factors) (see the proof of Theorem 2 in Rashidinejad et al., 2021). In this case, TRUST will achieve the same level of guarantees for the suboptimality of the output policy. We also empirically show the effectiveness of TRUST in Section 5. The full theorem for a fine-grained analysis for the suboptimality and its proof are deferred to Appendix E.

Augmentation with LCB. Compared to classical LCB, Algorithm 1 considers a much larger searching space, which encompasses not only the vertices of the simplex but the inner points as well. This enlargement of searching space shows great advantage, but this also comes with the price of larger uncertainty, especially when the width ε is large. In LCB, one considers the uncertainty by upper bound the noise at each vertex uniformly, while in our case, the uniform upper bound for a sub-region of the shifted simplex must be considered. When ε is large, the trust region method will induce larger uncertainty and tend to select a more stochastic policy than LCB and hence, can achieve worse performance. Moreover, when each arm has sufficiently many data samples to roughly estimate its mean return to reasonable accuracy, LCB works well because it chooses the arm with a tight lower bound. However the current results for LCB do not cover the important case where only few samples (e.g., less than 24 as described in Section 2) are available. Encouragingly our work shows strong results in such settings. To determine the most effective final policy, one can always combine TRUST (Algorithm 1) with LCB and select the better one between them based on the lower bound induced by two algorithms. By comparing the lower bounds of LCB and TRUST, the value of the finally output policy is guaranteed to outperform the lower bound for either LCB or TRUST with high probability. We defer the detailed algorithm and its theoretical guarantees to Appendix G.

5 Experiments

We present simulated experiments where we show the failure of LCB and the strong performance of TRUST. Moreover, we also present an application of TRUST to offline reinforcement learning.

Simulated experiments: A data-starved MAB. We consider a data-starved MAB problem with $d = 10000$ arms denoted by $a_i, i \in [d]$. The reward distributions are

$$r(a_i) \sim \mathsf{Uniform}(0.5, 1.5) \text{ for } i \leq 5000; \quad r(a_i) \sim \mathcal{N}(0, 1/4) \text{ for } i > 5000. \tag{16}$$

Namely, the set of good arms have reward random variables from a uniform distribution over $[0.5, 1.5]$ with unit mean, while the bad arms return a Gaussian reward with zero mean. We consider a dataset that contains a single sample for each of these arms.

We test Algorithm 1 on this MAB instance with fixed variance level $\sigma_i = 1/2$. We set the reference policy $\widehat{\mu}$ to be the behavioral cloning policy, which coincides with the uniform policy. We also test LCB and the greedy method which simply chooses the policy with the highest empirical reward.

In this example, the greedy algorithm fails because it erroneously selects an arm with a reward > 1.5, but such reward can only originate from a normal distribution with mean zero. Despite LCB incorporates the principle of pessimism under uncertainty, it selects an arm with average return equal to zero; its performance lower bound given by the confidence intervals is -1.5, which is almost vacuous and very uninformative. The behavioral cloning policy performs better, because it selects an arm uniformly at random, achieving the score 0.5.

Behavior Policy	Greedy	LCB	LCB Lower Bound	Improvement by TRUST	TRUST	TRUST Lower Bound
0.5	0	0	-1.5	0.42	0.92	0.6

Table 1: Results of simulated experiments in a 10000-arm bandit. The reward distribution is described in equation 16. The offline dataset includes one sample for each arm. The greedy method chooses the arm with the highest empirical reward. LCB selects an arm based on equation 3. The lower bound for LCB and TRUST follow equation 2 and equation 13, respectively.

Algorithm 1 achieves the best performance: the value of the policy that it identifies is 0.92, which *almost matches the optimal policy*. The lower bound on its performance computed by instantiating the RHS in equation 13 is around 0.6, a guarantee much tighter than that for LCB.

In order to gain intuition on the learning mechanics of TRUST, in Figure 3 we progressively enlarge the radius of the trust region from zero to the largest possible radius (on the x axis) and plot the value of the policy that maximizes the linear objective $\Delta^\top \widehat{r},\ \Delta \in \mathsf{C}(\varepsilon)$ for each value of the radius ε. Note that we rescale the range of ε to make the largest possible ε be one. In the same figure we also plot the lower bound computed with the help of equation equation 13.

Initially, the value of the policy increases because the optimization in equation 7 is performed over a larger set of stochastic policies. However, when ε approaches the maximal possible radius, all stochastic policies are included in the optimization program. In this case, TRUST greedily selects the arm with the highest empirical reward, which is from a normal distribution with a mean zero. The optimal balance between the size of the policy search space and its metric entropy is given by the critical radius $\varepsilon = 0.0116\varepsilon_0$, which is the point where the lower bound is the highest.

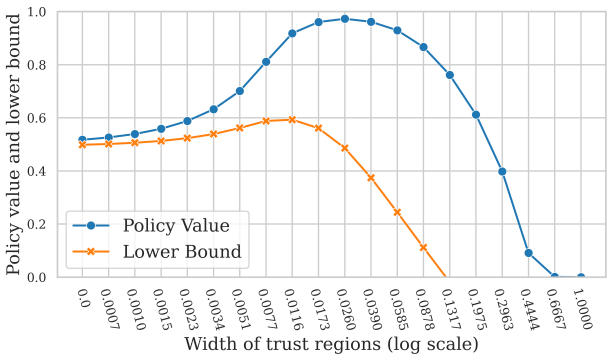

Figure 3: Policy values and their lower bounds for a data-starved MAB instance with 10000 arms whose reward distribution is described in equation 16.

A more general data-starved MAB. Besides the data-starved MAB we constructed, we also show that in general MABs, the performance of TRUST is on par with LCB, but TRUST will have a much tighter statistical guarantee, i.e., a larger lower bound for the value of the returned policy. We did experiments on a $d = 1000$-arm MAB where the reward distribution is $r(a_i) \sim \mathcal{N}(i/1000, 1/4), \quad \forall i \in [d]$. We ran TRUST Algorithm 1 and LCB over 8 different random seeds. When we have a single sample for each arm, TRUST will get a similar score as LCB. However, TRUST give a much tighter statistical guarantee than LCB, in the sense that the lower bound output by TRUST is much higher than that output by LCB so that TRUST can output a policy that is guaranteed to achieved a higher value. Moreover, we found the policies output from TRUST are much more stable than those from LCB. In all runs, while the lowest value of the arm chosen by LCB is around 0.24, all policies returned by TRUST have values above 0.65 with a much smaller variance, as shown in Table 2.

Offline reinforcement learning. In this section, we apply Algorithm 1 to the offline reinforcement learning (RL) setting under the assumption that the logging policies which generated the dataset are accessible. To be clear, our goal is not to exceed the performance of the state of the art deep RL algorithms—our algorithm is designed for bandit problems—but rather to illustrate the usefulness of our algorithm and theory.

Since our algorithm is designed for bandit problems, in order to apply it to the sequential setting, we map MDPs to MABs. Each policy in the MDP maps to an action in the MAB, and each trajectory

return in the MDP maps to an experienced return in the MAB setting. Notice that this reduction disregards the sequential aspect of the problem and thus our algorithm cannot perform 'trajectory stitching' (Levine et al., 2020; Kumar et al., 2020; Kostrikov et al., 2021). Furthermore, it can only be applied under the assumption that the logging policies are known.

Specifically we consider a setting where there are multiple known logging policies, each generating few trajectories. We test Algorithm 1 on some selected environments from the D4RL dataset (Fu et al., 2020) and compare its performance to the (CQL) algorithm (Kumar et al., 2020), a popular and strong baseline for offline RL algorithms. Since the D4RL dataset does not directly include the logging policies, we generate new datasets by running Soft Actor Critic (SAC) (Haarnoja et al., 2018) for 1000 episodes. We store 100 intermediate policies generated by SAC, and roll out 1 trajectory from each policy.

	LCB	TRUST
mean reward	0.718	0.725
mean lower bound	0.156	0.544
variance	0.265	0.038
minimal reward	0.239	0.658

Table 2: Comparison between LCB and TRUST (Algorithm 1) on a data-starved MAB with 1000 arms whose reward distribution follows $r(a_i) \sim \mathcal{N}(i/1000, 1/4)$. Both methods are repeated on 8 random seeds.

We use some default hyper-parameters for CQL.[1] We report the unnormalized scores in Table 3, each averaged over 4 random seeds. Algorithm 1 achieves a score on par with or higher than that of CQL, especially when the offline dataset is of poor quality and when there are very few—or just one—trajectory generated from each logging policy. Notice that while CQL is not guaranteed to outperform the behavioral policy, TRUST is backed by Theorem 4.1.

Additionally, while CQL took around 16-24 hours on one NVIDIA GeForce RTX 2080 Ti, TRUST only took 0.5-1 hours on 10 CPUs. The experimental details are included in Appendix H. Moreover, while the performance of CQL is highly reliant on the choice of hyper-parameters, TRUST is essentially hyper-parameters free.

		CQL	TRUST
Hopper	1-traj-low	499	999
	1-traj-high	2606	3437
Ant	1-traj-low	748	763
	1-traj-high	4115	4488
Walker2d	1-traj-low	311	346
	1-traj-high	4093	4097
HalfCheetah	1-traj-low	5775	5473
	1-traj-high	9067	10380

Table 3: Unnormalized score of CQL and TRUST in 4 environments from D4RL. In 1-traj-low case, we take the first 100 policies in the running of SAC. In 1-traj-high case, we take the $(10x + 1)$-th policy for $x \in [100]$. We sample one trajectory from each policy we take in all experiments.

6 Conclusion

In this paper we make a substantial contribution towards sample efficient decision making, by designing a data-efficient policy optimization algorithm that leverages offline data for the MAB setting. The key intuition of this work is to search over stochastic policies, which can be estimated more easily than deterministic ones. The design of our algorithm is enabled by a number of key insights, such as the use of the localized gaussian complexity which leads to the definition of the critical radius for the trust region. We believe that these concepts can be used more broadly to help design truly sample efficient algorithms, which can in turn enable the application of decision making to new settings where a high sample efficiency is critical.

References

Yasin Abbasi-Yadkori, David Pal, and Csaba Szepesvari. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[1] We use the codebase and default hyper-parameters in `https://github.com/young-geng/CQL`.

Farid Alizadeh and Donald Goldfarb. Second-order cone programming. *Mathematical programming*, 95(1):3–51, 2003.

Mawulolo K Ameko, Miranda L Beltzer, Lihua Cai, Mehdi Boukhechba, Bethany A Teachman, and Laura E Barnes. Offline contextual multi-armed bandits for mobile health interventions: A case study on emotion regulation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pp. 249–258, 2020.

Jean-Yves Audibert, Sébastien Bubeck, et al. Minimax policies for adversarial and stochastic bandits. In *COLT*, volume 7, pp. 1–122, 2009.

Jean-Yves Audibert, Sébastien Bubeck, and Rémi Munos. Best arm identification in multi-armed bandits. In *COLT*, pp. 41–53, 2010.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47:235–256, 2002.

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pp. 263–272. PMLR, 2017.

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Peter L Bartlett, Olivier Bousquet, and Shahar Mendelson. Local rademacher complexities. *The Annals of Statistics*, 33(4):1497–1537, 2005.

Jonathan Bassen, Bharathan Balaji, Michael Schaarschmidt, Candace Thille, Jay Painter, Dawn Zimmaro, Alex Games, Ethan Fast, and John C Mitchell. Reinforcement learning for the adaptive scheduling of educational activities. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–12, 2020.

Pierre C Bellec. Localized gaussian width of m-convex hulls with applications to lasso and convex aggregation. 2019.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

Wei Chen, Yajun Wang, Yang Yuan, and Qinshi Wang. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *The Journal of Machine Learning Research*, 17(1): 1746–1778, 2016.

Ching-An Cheng, Tengyang Xie, Nan Jiang, and Alekh Agarwal. Adversarially trained actor critic for offline reinforcement learning. *arXiv preprint arXiv:2202.02446*, 2022.

Richard Combes, Mohammad Sadegh Talebi Mazraeh Shahi, Alexandre Proutiere, et al. Combinatorial bandits revisited. *Advances in neural information processing systems*, 28, 2015.

Yan Dai, Ruosong Wang, and Simon S Du. Variance-aware sparse linear bandits. *arXiv preprint arXiv:2205.13450*, 2022.

Rémy Degenne and Vianney Perchet. Anytime optimal algorithms in stochastic multi-armed bandits. In *International Conference on Machine Learning*, pp. 1587–1595. PMLR, 2016.

Steven Diamond and Stephen Boyd. Cvxpy: A python-embedded modeling language for convex optimization. *The Journal of Machine Learning Research*, 17(1):2909–2913, 2016.

Yaqi Duan and Martin J Wainwright. Policy evaluation from a single path: Multi-step methods, mixing and mis-specification. *arXiv preprint arXiv:2211.03899*, 2022.

Yaqi Duan and Martin J Wainwright. A finite-sample analysis of multi-step temporal difference estimates. In *Learning for Dynamics and Control Conference*, pp. 612–624. PMLR, 2023.

Yaqi Duan, Mengdi Wang, and Martin J Wainwright. Optimal policy evaluation using kernel-based temporal difference methods. *arXiv preprint arXiv:2109.12002*, 2021.

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, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pp. 2052–2062. PMLR, 2019.

Aurélien Garivier and Emilie Kaufmann. Optimal best arm identification with fixed confidence. In *Conference on Learning Theory*, pp. 998–1027. PMLR, 2016.

Sara A Geer. *Empirical Processes in M-estimation*, volume 6. Cambridge university press, 2000.

Yehoram Gordon, Alexander E Litvak, Shahar Mendelson, and Alain Pajor. Gaussian averages of interpolated bodies and applications to approximate reconstruction. *Journal of Approximation Theory*, 149(1):59–73, 2007.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Botao Hao, Xiang Ji, Yaqi Duan, Hao Lu, Csaba Szepesvári, and Mengdi Wang. Bootstrapping statistical inference for off-policy evaluation. *arXiv preprint arXiv:2102.03607*, 2021.

Elad Hazan and Zohar Karnin. Volumetric spanners: an efficient exploration basis for learning. *Journal of Machine Learning Research*, 2016.

Todd Hester and Peter Stone. Texplore: real-time sample-efficient reinforcement learning for robots. *Machine learning*, 90:385–429, 2013.

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

Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, pp. 5084–5096. PMLR, 2021.

Kwang-Sung Jun, Aniruddha Bhargava, Robert Nowak, and Rebecca Willett. Scalable generalized linear bandits: Online computation and hashing. *Advances in Neural Information Processing Systems*, 30, 2017.

Yeoneung Kim, Insoon Yang, and Kwang-Sung Jun. Improved regret analysis for variance-adaptive linear bandits and horizon-free linear mixture mdps. *Advances in Neural Information Processing Systems*, 35:1060–1072, 2022.

Vladimir Koltchinskii. Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, 47(5):1902–1914, 2001.

Vladimir Koltchinskii. Local rademacher complexities and oracle inequalities in risk minimization. 2006.

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

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

Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. In *Artificial Intelligence and Statistics*, pp. 535–543. PMLR, 2015.

Tze Leung Lai. Adaptive treatment allocation and the multi-armed bandit problem. *The annals of statistics*, pp. 1091–1114, 1987.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. *Advances in neural information processing systems*, 20, 2007.

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

Guillaume Lecué and Shahar Mendelson. Learning subgaussian classes: Upper and minimax bounds. *arXiv preprint arXiv:1305.4825*, 2013.

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, Laixi Shi, Yuxin Chen, Yuejie Chi, and Yuting Wei. Settling the sample complexity of model-based offline reinforcement learning. *arXiv preprint arXiv:2204.05275*, 2022.

Zihao Li, Zhuoran Yang, and Mengdi Wang. Reinforcement learning with human feedback: Learning dynamic choices via pessimism. *arXiv preprint arXiv:2305.18438*, 2023.

Rongrong Liu, Florent Nageotte, Philippe Zanne, Michel de Mathelin, and Birgitta Dresp-Langley. Deep reinforcement learning for the control of robotic manipulation: a focussed mini-review. *Robotics*, 10(1):22, 2021.

Yi Liu and Veronika Ročková. Variable selection via thompson sampling. *Journal of the American Statistical Association*, 118(541):287–304, 2023.

Ying Liu, Brent Logan, Ning Liu, Zhiyuan Xu, Jian Tang, and Yangzhi Wang. Deep reinforcement learning for dynamic treatment regimes on medical registry data. In *2017 IEEE international conference on healthcare informatics (ICHI)*, pp. 380–385. IEEE, 2017.

Pierre Ménard, Omar Darwiche Domingues, Anders Jonsson, Emilie Kaufmann, Edouard Leurent, and Michal Valko. Fast active learning for pure exploration in reinforcement learning. In *International Conference on Machine Learning*, pp. 7599–7608. PMLR, 2021.

Yifei Min, Tianhao Wang, Dongruo Zhou, and Quanquan Gu. Variance-aware off-policy evaluation with linear function approximation. *Advances in neural information processing systems*, 34: 7598–7610, 2021.

Wenlong Mou, Martin J Wainwright, and Peter L Bartlett. Off-policy estimation of linear functionals: Non-asymptotic theory for semi-parametric efficiency. *arXiv preprint arXiv:2209.13075*, 2022.

Ofir Nachum, Bo Dai, Ilya Kostrikov, Yinlam Chow, Lihong Li, and Dale Schuurmans. Algaedice: Policy gradient from arbitrary experience. *arXiv preprint arXiv:1912.02074*, 2019.

Preetum Nakkiran, Behnam Neyshabur, and Hanie Sedghi. The deep bootstrap framework: Good online learners are good offline generalizers. *arXiv preprint arXiv:2010.08127*, 2020.

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.

Paria Rashidinejad, Banghua Zhu, Cong Ma, Jiantao Jiao, and Stuart Russell. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *arXiv preprint arXiv:2103.12021*, 2021.

Sherry Ruan, Allen Nie, William Steenbergen, Jiayu He, JQ Zhang, Meng Guo, Yao Liu, Kyle Dang Nguyen, Catherine Y Wang, Rui Ying, et al. Reinforcement learning tutor better supported lower performers in a math task. *arXiv preprint arXiv:2304.04933*, 2023.

Paat Rusmevichientong and John N Tsitsiklis. Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411, 2010.

Daniel Russo. Simple bayesian algorithms for best arm identification. In *Conference on Learning Theory*, pp. 1417–1418. PMLR, 2016.

Nian Si, Fan Zhang, Zhengyuan Zhou, and Jose Blanchet. Distributionally robust policy evaluation and learning in offline contextual bandits. In *International Conference on Machine Learning*, pp. 8884–8894. PMLR, 2020.

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

Philip Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High confidence policy improvement. In *International Conference on Machine Learning*, pp. 2380–2388. PMLR, 2015.

Roman Vershynin. High-dimensional probability. *University of California, Irvine*, 2020.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019.

Kerong Wang, Hanye Zhao, Xufang Luo, Kan Ren, Weinan Zhang, and Dongsheng Li. Bootstrapped transformer for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35:34748–34761, 2022.

Siwei Wang and Wei Chen. Thompson sampling for combinatorial semi-bandits. In *International Conference on Machine Learning*, pp. 5114–5122. PMLR, 2018.

Yuting Wei, Billy Fang, and Martin J. Wainwright. From gauss to kolmogorov: Localized measures of complexity for ellipses. 2020.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

Tengyang Xie and Nan Jiang. Batch value-function approximation with only realizability. *arXiv preprint arXiv:2008.04990*, 2020.

Tengyang Xie, Ching-An Cheng, Nan Jiang, Paul Mineiro, and Alekh Agarwal. Bellman-consistent pessimism for offline reinforcement learning. *arXiv preprint arXiv:2106.06926*, 2021.

Wei Xiong, Han Zhong, Chengshuai Shi, Cong Shen, Liwei Wang, and Tong Zhang. Nearly minimax optimal offline reinforcement learning with linear function approximation: Single-agent mdp and markov game. *arXiv preprint arXiv:2205.15512*, 2022.

Ming Yin and Yu-Xiang Wang. Towards instance-optimal offline reinforcement learning with pessimism. *arXiv preprint arXiv:2110.08695*, 2021.

Ming Yin, Yaqi Duan, Mengdi Wang, and Yu-Xiang Wang. Near-optimal offline reinforcement learning with linear representation: Leveraging variance information with pessimism. *arXiv preprint arXiv:2203.05804*, 2022.

Andrea Zanette, Emma Brunskill, and Mykel J. Kochenderfer. Almost horizon-free structure-aware best policy identification with a generative model. In *Advances in Neural Information Processing Systems*, 2019.

Andrea Zanette, Alessandro Lazaric, Mykel J Kochenderfer, and Emma Brunskill. Provably efficient reward-agnostic navigation with linear value iteration. In *Advances in Neural Information Processing Systems*, 2020.

Andrea Zanette, Martin J Wainwright, and Emma Brunskill. Provable benefits of actor-critic methods for offline reinforcement learning. *arXiv preprint arXiv:2108.08812*, 2021.

Ruiqi Zhang, Xuezhou Zhang, Chengzhuo Ni, and Mengdi Wang. Off-policy fitted q-evaluation with differentiable function approximators: Z-estimation and inference theory. In *International Conference on Machine Learning*, pp. 26713–26749. PMLR, 2022.

Zihan Zhang, Jiaqi Yang, Xiangyang Ji, and Simon S Du. Improved variance-aware confidence sets for linear bandits and linear mixture mdp. *Advances in Neural Information Processing Systems*, 34:4342–4355, 2021.

A ADDITIONAL RELATED WORK

Multi-armed bandit (MAB) is a classical decision-making framework (Lattimore & Szepesvári, 2020; Lai & Robbins, 1985; Lai, 1987; Langford & Zhang, 2007; Auer, 2002; Bubeck et al., 2012; Audibert et al., 2009; Degenne & Perchet, 2016). The natural approach in offline MABs is the LCB algorithm (Ameko et al., 2020; Si et al., 2020), an offline variant of the classical UCB method (Auer et al., 2002) which is minimax optimal (Rashidinejad et al., 2021). The optimization over stochastic policies is also considered in combinatorial multi-armed bandits (CMAB) (Combes et al., 2015). Most works on CMAB focus on variants of the UCB algorithm (Kveton et al., 2015; Combes et al., 2015; Chen et al., 2016) or of Thompson sampling (Wang & Chen, 2018; Liu & Ročková, 2023), and they are generally online. Our framework can also be applied to offline reinforcement learning (RL) (Sutton & Barto, 2018) whenever the logging policies are accessible. There exist a lot of practical algorithms for offline RL (Fujimoto et al., 2019; Peng et al., 2019; Wu et al., 2019; Kumar et al., 2020; Kostrikov et al., 2021). Theory has also been investigated extensively in tabular domain and function approximation setting (Nachum et al., 2019; Xie & Jiang, 2020; Zanette et al., 2021; Xie et al., 2021; Yin et al., 2022; Xiong et al., 2022). Some works also tried to establish general guarantees for deep RL algorithms via sophisticated statistical tools, such as bootstrapping (Thomas et al., 2015; Nakkiran et al., 2020; Hao et al., 2021; Wang et al., 2022; Zhang et al., 2022).

We rely on the notion of pessimism, which is a key concept in offline bandits and RL. While most prior works focused on the so-called absolute pessimism (Jin et al., 2020; Xie et al., 2021; Yin et al., 2022; Rashidinejad et al., 2021; Li et al., 2023), the work of (Cheng et al., 2022) applied pessimism not on the policy value but on the difference (or improvement) between policies. However, their framework is very different from ours. We make extensive use of two key concepts, namely localization laws and critical radii (Wainwright, 2019), which control the relative scale of the signal and uncertainty. The idea of localization plays a critical role in the theory of empirical process (Geer, 2000) and statistical learning theory (Koltchinskii, 2001; 2006; Bartlett & Mendelson, 2002; Bartlett et al., 2005). The concept of critical radius or critical inequality is used in non-parametric regression (Wainwright, 2019) and in off-policy evaluation (Duan et al., 2021; Duan & Wainwright, 2022; 2023; Mou et al., 2022).

B PROOF OF THEOREM 4.1

To prove Lemma E.3, we first define the following event

$$\mathcal{E} := \left\{ \sup_{\Delta \in \mathsf{C}(\varepsilon)} \Delta^\top \eta \leq \mathcal{G}(\varepsilon) \quad \forall \varepsilon \in E \right\}. \tag{17}$$

When $\mathcal{E}$ happens, the quantity $\mathcal{G}(\varepsilon)$ can upper bound the supremum of the Gaussian process we care about, and hence, we can effectively upper bound the uncertainty for any stochastic policy using $\mathcal{G}(\cdot)$. It follows from the Definition 3.2 that the event $\mathcal{E}$ happens with probability a leas $1-\delta$.

We can now prove all the claims in the theorem, starting from the first and the second. A comparator policy π identifies a weight vector w, an improvement vector Δ and a radius ε such that $w = \widehat{\mu} + \Delta$ and $\Delta \in \mathsf{C}(\varepsilon)$. In fact, we can always take ε to be the minimal value such that $\Delta \in \mathsf{C}(\varepsilon)$. The first claim in Equation (11) can be proved by establishing that with probability at least $1-\delta$

$$w^\top r - \pi_{TRUST}^\top r = \Delta^\top r - \widehat{\Delta}_*^\top r \leq 2\mathcal{G}(\lceil \varepsilon \rceil), \tag{18}$$

where π_{TRUST} is the policy weight returned by Algorithm 1. In order to show Equation (18), we can decompose $\widehat{\Delta}_*^\top r$ using the fact that $\widehat{\varepsilon}_* \in E$ and $\widehat{\Delta}_* \in \mathsf{C}(\widehat{\varepsilon}_*)$ to obtain

$$\widehat{\Delta}_*^\top r = \widehat{\Delta}_*^\top \widehat{r} - \widehat{\Delta}_*^\top \eta \geq \widehat{\Delta}_*^\top \widehat{r} - \mathcal{G}(\widehat{\varepsilon}_*) = \widehat{\Delta}_*^\top \widehat{r} - \mathcal{G}(\lceil \widehat{\varepsilon}_* \rceil). \tag{19}$$

To further lower bound the RHS above, we have the following lemma, which shows that Algorithm 1 can be written in an equivalent way.

Lemma B.1. *The output of Algorithm 1 satisfies*

$$\left(\widehat{\varepsilon}_*, \widehat{\Delta}_*\right) = \underset{\varepsilon \leq \varepsilon_0, \Delta \in \mathsf{C}(\varepsilon)}{\arg\max} \left[\Delta^\top \widehat{r} - \mathcal{G}(\lceil \varepsilon \rceil)\right]. \tag{20}$$

This shows that Algorithm 1 optimizes over an objective function which consists of a signal term (i.e., $\Delta^\top \widehat{r}$) minus a noise term (i.e., $\mathcal{G}(\lceil \varepsilon \rceil)$). Applying this lemma to equation 19, we know

$$\widehat{\Delta}_*^\top r \geq \Delta^\top \widehat{r} - \mathcal{G}(\lceil \varepsilon \rceil) = \Delta^\top r + \Delta^\top \eta - \mathcal{G}(\lceil \varepsilon \rceil). \tag{21}$$

After recalling that under $\mathcal{E}$

$$\Delta^\top \eta \leq \sup_{\Delta \in \mathsf{C}(\varepsilon)} \Delta^\top \eta \leq \sup_{\Delta \in \mathsf{C}(\lceil \varepsilon \rceil)} \Delta^\top \eta \leq \mathcal{G}(\lceil \varepsilon \rceil), \tag{22}$$

plugging the equation 22 back into equation 21 concludes the bound in Equation (18), which also proves our first claim. Rearranging the terms in Equation (18) and taking supremum over all comparator policies, we obtain

$$\widehat{\Delta}_*^\top r \geq \sup_{\varepsilon \leq \varepsilon_0, \Delta \in \mathsf{C}(\varepsilon)} \left[\Delta^\top r - 2\mathcal{G}(\lceil \varepsilon \rceil)\right], \tag{23}$$

which proves the first claim since $V^{\pi_{TRUST}} - V^{\widehat{\mu}} = \widehat{\Delta}_*^\top r$.

In order to prove the last claim, it suffices to lower bound the policy value of the reference policy $\widehat{\mu}$. From equation 5, we have $\widehat{\mu}(\widehat{r} - r) \sim \mathcal{N}(0, 1/[\sum_{i=1}^d N_i/\sigma_i^2])$, which implies with probability at least $1-\delta$,

$$\widehat{\mu}(\widehat{r} - r) \leq \sqrt{\frac{2\log(1/\delta)}{\sum_{i=1}^d N_i/\sigma_i^2}} \tag{24}$$

from the standard Hoeffding inequality (e.g., Prop 2.5 in (Wainwright, 2019)). Combining equation 19 and equation 24, we obtain

$$\begin{aligned}
\pi_{TRUST}^\top r = \widehat{\mu}^\top r + \widehat{\Delta}_*^\top r &\geq \widehat{\mu}^\top \widehat{r} + \widehat{\mu}^\top (r - \widehat{r}) + \widehat{\Delta}_*^\top \widehat{r} - \mathcal{G}(\widehat{\varepsilon}_*) && \text{(From equation 19)} \\
&\geq \pi_{TRUST}^\top \widehat{r} - \mathcal{G}(\widehat{\varepsilon}_*) - \sqrt{\frac{2\log(1/\delta)}{\sum_{i=1}^d N_i/\sigma_i^2}} && \text{(From equation 24)}
\end{aligned}$$

with probability at least $1-2\delta$. Therefore, we conclude.

C One-sample case with strong signals

In this section, we give a simple example of one-sample-per-arm case. This can be view as a special case of data-starved MAB and Theorem 4.1 can be applied to get a non-trivial guarantees. Specifically, consider an MAB with $2d$ arms. Assume the true mean reward vector is $r = (1, 1, ..., 1, 0, 0, ..., 0)^\top$ and the noise vector is $\eta \sim \mathcal{N}(0, \sigma^2 I_{2d})$ That is, the first d arms have rewards independently sampled from $\mathcal{N}(1, 1)$ and the rewards for other d arms are independently sampled from $\mathcal{N}(0, 0)$. The stochastic reference policy is set to the uniform one, i.e., $\widehat{\mu} = (\frac{1}{d}, \frac{1}{d}, ...\frac{1}{d})^\top$.

We apply Algorithm 1 to this MAB instance. In the next theorem, we will show that for a specific ε, the optimal improvement in $\mathsf{C}(\varepsilon)$ (denoted as $\widehat{\Delta}_\varepsilon$ in equation 7) can achieve an improved reward value of constant level.

Proposition C.1. *Assume $r = (1, 1, ..., 1, 0, 0, ..., 0)^\top$ and noise $\eta \sim \mathcal{N}(0, I_{2d})$. For any $0 \leq \varepsilon \leq \frac{1}{\sqrt{d}}$, with probability at least $1-\delta$, the improvement of policy value can be lower bounded by*

$$\widehat{\Delta}_\varepsilon^\top r \geq \varepsilon\sqrt{d}\left[\frac{1}{2} - \sigma\left(1 + \sqrt{\frac{8\log(2/\delta)}{d}}\right)\right],$$

where the improvement vector in $\mathsf{C}(\varepsilon)$ is defined in equation 7. Therefore, for $\varepsilon = \frac{1}{\sqrt{d}}$ and $d \geq 8\log(2/\delta)$, with probability at least $1-\delta$, we can get a constant policy improvement

$$\widehat{\Delta}_\varepsilon^\top r \geq \frac{1}{2} - 2\sigma.$$

Proof. We define the optimal improvement vector as

$$\Delta_\varepsilon^* := \arg\max_{\Delta \in \mathsf{C}(\varepsilon)} \Delta^\top r.$$

Then, from the definition of $\widehat{\Delta}_\varepsilon$, we have

$$\widehat{\Delta}_\varepsilon^\top r = \widehat{\Delta}_\varepsilon^\top \widehat{r} - \widehat{\Delta}_\varepsilon^\top \eta \geq \left(\Delta_\varepsilon^*\right)^\top \widehat{r} - \widehat{\Delta}_\varepsilon^\top \eta = \left(\Delta_\varepsilon^*\right)^\top r + \left(\Delta_\varepsilon^*\right)^\top \eta - \widehat{\Delta}_\varepsilon^\top \eta \geq \underbrace{\left(\Delta_\varepsilon^*\right)^\top r}_{\text{signal}} - \underbrace{\left[\sup_{\Delta \in \mathsf{C}(\varepsilon)} \Delta^\top \eta - \left(\Delta_\varepsilon^*\right)^\top \eta\right]}_{\text{noise}}. \tag{25}$$

In order to lower bound the policy value improvement, it suffices to lower bound the signal part and upper bound the noise. We denote $\mathcal{H} = \{x \in \mathbb{R}^d : \sum_{i=1}^d x_i = 0\}$ as a hyperplane in $\mathbb{R}^d$. To deal with the signal part, it suffices to notice that

$$\mathsf{C}(\varepsilon) \subset \mathcal{H} \cap \mathbb{B}_2^d(\varepsilon).$$

We denote $r_\parallel$ as the orthogonal projection of r on the $\mathcal{H}$ and $r_\perp = r - r_\parallel$. In the strong signal case, we have

$$r_\parallel = \left(\frac{1}{2}, \frac{1}{2}, ..., \frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}, ..., -\frac{1}{2}\right)^\top, \quad r_\perp = \left(\frac{1}{2}, \frac{1}{2}, ..., \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, ..., \frac{1}{2}\right)^\top.$$

Then, the signal part satisfies

$$\sup_{\Delta \in \mathsf{C}(\varepsilon)} \Delta^\top r = \sup_{\Delta \in \mathsf{C}(\varepsilon)} \Delta^\top r_\parallel \leq \sup_{\Delta \in \mathcal{H} \cap \mathbb{B}_2^d(\varepsilon)} \Delta^\top r_\parallel = \left(\varepsilon \cdot \frac{r_\parallel}{\left\|r_\parallel\right\|_2}\right)^\top r_\parallel = \varepsilon \left\|r_\parallel\right\|_2 = \frac{\varepsilon\sqrt{d}}{2}. \tag{26}$$

On the other hand, we notice that when $\varepsilon \leq \frac{1}{\sqrt{d}}$,

$$\varepsilon \cdot \frac{r_\parallel}{\left\|r_\parallel\right\|_2} = \left(\frac{\varepsilon}{\sqrt{d}}, \frac{\varepsilon}{\sqrt{d}}, ..., \frac{\varepsilon}{\sqrt{d}}, -\frac{\varepsilon}{\sqrt{d}}, -\frac{\varepsilon}{\sqrt{d}}, ..., -\frac{\varepsilon}{\sqrt{d}}\right)^\top \in \mathsf{C}(\varepsilon).$$

So actually the inequality in the equation 26 should be an equation, which implies

$$\sup_{\Delta \in \mathsf{C}(\varepsilon)} \Delta^\top = \frac{\varepsilon\sqrt{d}}{2}. \tag{27}$$

For the noise part, we decompose the noise as $\eta = \eta_\perp + \eta_\parallel$, where $\eta_\parallel$ is the orthogonal projection of η on $\mathcal{H}$. Then, from $\mathsf{C}(\varepsilon) \subset \mathcal{H} \cap \mathbb{B}_2^d(\varepsilon)$, one has

$$\begin{aligned}\sup_{\Delta \in \mathsf{C}(\varepsilon)} \Delta^\top \eta &= \sup_{\Delta \in \mathsf{C}(\varepsilon)} \Delta^\top \left(\eta_\parallel + \eta_\perp\right) = \sup_{\Delta \in \mathsf{C}(\varepsilon)} \Delta^\top \eta_\parallel \leq \sup_{\Delta \in \mathcal{H} \cap \mathbb{B}_2^d(\varepsilon)} \Delta^\top \eta_\parallel \\ &= \left(\varepsilon \cdot \frac{\eta_\parallel}{\left\|\eta_\parallel\right\|_2}\right)^\top \eta_\parallel = \varepsilon \left\|\eta_\parallel\right\|_2 \leq \varepsilon \left\|\eta\right\|_2.\end{aligned}$$

This implies $\widehat{\Delta}_\varepsilon^\top r \geq \left(\Delta_\varepsilon^*\right)^\top r - [\varepsilon \left\|\eta\right\|_2 - \left(\Delta_\varepsilon^*\right)^\top \eta]$. From our assumption, $\frac{1}{\sigma^2}\left\|\eta\right\|_2^2$ is a chi-square random variable with degree d, so from the Example 2.11 in (Wainwright, 2019), we know with probability at least $1 - \delta/2$, one has

$$\frac{\left\|\eta\right\|_2^2}{d\sigma^2} \leq 1 + \sqrt{\frac{8\log\left(2/\delta\right)}{d}}.$$

This implies

$$\left\|\eta\right\|_2 \leq \sqrt{d\sigma^2\left(1 + \sqrt{\frac{8\log\left(2/\delta\right)}{d}}\right)} \leq \sqrt{d}\sigma\left(1 + \sqrt{\frac{2\log\left(2/\delta\right)}{d}}\right).$$

The last inequality comes from $\sqrt{1+u} \leq 1 + \frac{u}{2}$ for positive u. Moreover, since Δ_ε^* is a fixed vector, we know $\left(\Delta_\varepsilon^*\right)^\top \eta \sim \mathcal{N}\left(0, \sigma^2\left\|\Delta_\varepsilon^*\right\|_2^2\right)$. So with probability at least $1 - \delta/2$, one has

$$\left(\Delta_\varepsilon^*\right)^\top \eta \geq -\sigma\left\|\Delta_\varepsilon^*\right\|_2\sqrt{2\log\left(\frac{2}{\delta}\right)} \geq -\sigma\varepsilon\sqrt{2\log\left(\frac{2}{\delta}\right)}$$

Combining the two terms above, one has with probability at least $1-\delta$, it holds

$$\varepsilon \|\eta\|_2 - (\Delta_\varepsilon^*)^\top \eta \leq \varepsilon\sqrt{d}\sigma\left(1+\sqrt{\frac{2\log(2/\delta)}{d}}\right)+\sigma\varepsilon\sqrt{2\log\left(\frac{2}{\delta}\right)} = \varepsilon\sqrt{d}\sigma\left(1+\sqrt{\frac{8\log(2/\delta)}{d}}\right). \tag{28}$$

Combining equation 25, equation 27 and equation 28, we finish the proof. □

D MONTE CARLO COMPUTATION

Algorithm 2 Monte-Carlo method for computing $\mathcal{G}(\varepsilon)$

Input: Offline dataset $\mathcal{D}$, the radius value $\varepsilon \in E$, the total sample size M and threshold M_0.
1. Independently sample M noise vectors, denoted as η_i for $i \in [M]$, where $\eta_i \sim \mathcal{N}0, \sigma_i^2/N(a_i), \sigma_i^2$ is the noise variance for the i-th arm and $N(a_i)$ is the sample size for a_i in $\mathcal{D}$.
2. Solve $X_i := \sup_{\Delta\in\mathsf{C}(\varepsilon)} \Delta^\top \eta_i$ for $i \in [M]$ and order them as $X_{(1)} \leq X_{(2)} \leq ... \leq X_{(M)}$.
3. **Return** $X_{(M-M_0+1)}$ as an estimate of $\mathcal{G}(\varepsilon)$ defined in Definition 3.2.

As discussed in Section 3, we can estimate $\mathcal{G}(\varepsilon)$ using classical Monte Carlo method. In this section, we illustrate the detailed implementation. We first sample M i.i.d. noise and then solve $\sup_{\Delta\in\mathsf{C}(\varepsilon)} \Delta^\top\eta$ for each to get M suprema. We eventually select the M_0-th largest values of all suprema as our estimate for the bonus function, where M_0 is a pre-computed integer dependent on M and the pre-determined failure probability $\delta > 0$. Here, the program $\sup_{\Delta\in\mathsf{C}(\varepsilon)} \Delta^\top\eta$ is a second-order cone program and can be efficiently solved via standard off-the shelf libraries (Alizadeh & Goldfarb, 2003; Boyd & Vandenberghe, 2004; Diamond & Boyd, 2016). The pseudocode for the Monte-Carlo sampling is in Algorithm 2.

To determine M_0, we denote η_i as the i.i.d. noise vector for $i \in [M]$ and $X_i = \sup_{\Delta\in\mathsf{C}(\varepsilon)} \Delta^\top\eta$. We denote the order statistics of X_i-s as $X_{(1)} \leq X_{(2)} \leq ... \leq X_{(M)}$. Suppose the cumulative distribution function of X_i is $F(x)$, then from the property of the order statistics, we know the cumulative distribution function of $X_{(M-M_0+1)}$ is

$$F_{X_{(M-M_0+1)}}(x) = \sum_{j=M-M_0+1}^{M} C_M^j \left(F(x)\right)^j \left(1-F(x)\right)^{M-j}.$$

We denote $q_{1-\delta}$ as the $(1-\delta)$-lower quantile of the random variable X, then we have $F_{X_{(M-M_0+1)}}(q_{1-\delta}) = \sum_{j=M-M_0+1}^{M} C_M^j (1-\delta)^j(\delta)^{M-j}$. For integer M and $\delta > 0$, we define $Q(M,\delta)$ as the maximal integer M_0 such that $\sum_{j=M-M_0+1}^{M} C_M^j(1-\delta)^j(\delta)^{M-j} \leq \delta$. With this definition, we take a fixed M and a total failure tolerance δ for all $\varepsilon \in E$, then we take

$$M_0 = Q\left(M, \frac{\delta}{2|E|}\right)$$

as the threshold number. Under this choice, for any $\varepsilon \in E$, with probability at least $1-\delta/2|E|$, it holds $X_{(M-M_0+1)} > q_{1-\delta/2|E|}$. On the other hand, with probability $1-\delta/2|E|$, it holds that $\sup_{\Delta\in\mathrm{C}(\varepsilon)} \Delta^\top\eta \leq q_{1-\delta/2|E|}$ This implies

$$\sup_{\Delta\in\mathrm{C}(\varepsilon)} \Delta^\top\eta \leq q_{1-\delta/2|E|} < X_{(M-M_0+1)}$$

with probability at least $1-\delta/|E|$. From a union bound, we know with probability at least $1-\delta$, the bound above holds for any $\varepsilon \in E$.

E A FINE-GRAINED ANALYSIS TO THE SUBOPTIMALITY

We have shown a problem-dependent upper bound for the suboptimality in equation 12. In this section, we will give a further upper bound for $\mathcal{G}(\varepsilon)$ and hence, for the suboptimality. We have the following theorem. The proof is deferred to Appendix E.1.

Theorem E.1. *For a policy π (deterministic or stochastic), we denote its reward value as V^{π}. TRUST has the following properties.*

1. *We denote a comparator policy as a triple $(\varepsilon, \Delta, \pi)$ such that $\varepsilon = \sum_{i=1}^{d} \frac{\sigma_i^2 \Delta_i^2}{N_i}, \pi = \widehat{\mu} + \Delta$. We take the discrete candidate set E defined in equation 10. With probability at least $1-\delta$, for any stochastic comparator policy $(\varepsilon, \Delta, \pi)$, the sub-optimality of the output policy of Algorithm 1 can be upper bounded as*

$$V^{\pi} - V^{\pi_{TRUST}} \leq 2\sqrt{2\sum_{i=1}^{d} \frac{\alpha \Delta_i^2 \sigma_i^2}{N_i} \log\left(\frac{2|E|}{\delta}\right)} + 2\min\left\{\sqrt{\sum_{i=1}^{d} \frac{\alpha \Delta_i^2 \sigma_i^2}{N_i}}, 4D\sqrt{\log_+\left(\frac{4ed\sum_{i=1}^{d} \frac{\alpha \Delta_i^2 \sigma_i^2}{N_i}}{D^2}\right)}\right\} \tag{29}$$

   where D is defined as any quantity satisfying

$$D \geq \sqrt{\max_{i\in[d]}\left[\frac{\sigma_i^2}{N_i} - \frac{2\sigma_i^2}{N}\right] + \frac{\sum_{j=1}^{d} N_j \sigma_j^2}{N^2}}. \tag{30}$$

   α is the decaying rate defined in equation 10, $\log_+(a) = \max(1, \log(a))$.

2. *(Comparison with the optimal policy) We further assume $\sigma_i = 1$ for $i \in [d]$ and assume the offline dataset is generated from the policy $\mu(\cdot)$ with $\min_{i\in[d]} \mu(a_i) > 0$. Without loss of generality we assume a_1 is the optimal arm and denote the optimal policy as π_*. We write*

$$C^* := \frac{1}{\mu(a_1)}, \quad C_{\min} := \frac{1}{\min_{i\in[d]} \mu(a_i)}. \tag{31}$$

   When $N \geq 8C_{\min}\log(d/\delta)$, with probability at least $1-2\delta$, one has

$$V^{\pi_*} - V^{\pi_{TRUST}} \lesssim \sqrt{\frac{C_{\min}}{N} \log_+\left(\frac{dC^*}{C_{\min}}\right)} + \sqrt{\frac{C^*}{N}\log\left(\frac{2|E|}{\delta}\right)}. \tag{32}$$

   Specially, when $C_{\min} \simeq C^*$, we have with probability at least $1-2\delta$,

$$V^{\pi_*} - V^{\pi_{TRUST}} \lesssim \sqrt{\frac{C^*}{N}\log\left(\frac{2d|E|}{\delta}\right)}. \tag{33}$$

We remark that equation 29 is problem-dependent, and it gives an explicit upper bound for $\mathcal{G}(\lceil\varepsilon\rceil)$ in equation 12. This is derived by first concentrating $\mathcal{G}(\varepsilon)$ around $\mathbb{E}\sup_{\Delta\in\mathsf{C}(\varepsilon)} \Delta^{\top}\eta$, which is well-defined as localized Gaussian width or local Gaussian complexity (Koltchinskii, 2006), and then upper bounding the localized Gaussian width of a convex hull via tools in convex analysis (Bellec, 2019). Different from Theorem 2.1, when $\pi = a_i$ represents a single arm, equation 29 relies not only on σ_i^2/N_i, but on σ_j^2/N_j for $j \neq i$ as well, since the size of trust regions depend on σ_i^2/N_i for all $i \in [d]$.

Notably, equation 33 gives an analogous upper bound depending on $\mu(\cdot)$ and N, which is comparable to the bound for LCB in Theorem 2.1 up to constant and logarithmic factors. This indicates that, when behavioral cloning policy is not too imbalanced, TRUST is guaranteed to achieve the same level of performance as LCB. In fact, this improvement is remarkable since TRUST is exploring a much larger policy searching space than LCB, which encompasses all stochastic policies (the whole simplex) rather than the set of all single arms only. We also remark that both the bound in Theorem 2.1 and in equation 46 are worst-case upper bound, and in practice, we will show in Section 5 that in some settings, TRUST can achieve good performance while LCB fails completely.

Is TRUST minimax-optimal? We consider the hard cases in MAB (Rashidinejad et al., 2021) where LCB achieves the minimax-optimal upper bound and we show for these hard cases, TRUST will achieve the same sample complexity as LCB up to log and constant factors. More specifically, we

consider a two-arm MAB $\mathcal{A} = \{1, 2\}$ and the uniform behavioral cloning policy $\mu(1) = \mu(2) = 1/2$. For $\delta\ in[0, 1/4]$, we define $\mathcal{M}_1$ and $\mathcal{M}_2$ are two MDPs whose reward distributions are as follows.

$$\mathcal{M}_1 : r(1) \sim \mathsf{Bernoulli}\left(\frac{1}{2}\right),\ r(2) \sim \mathsf{Bernoulli}\left(\frac{1}{2} + \delta\right)$$
$$\mathcal{M}_2 : r(1) \sim \mathsf{Bernoulli}\left(\frac{1}{2}\right),\ r(2) \sim \mathsf{Bernoulli}\left(\frac{1}{2} - \delta\right),$$

where $\mathsf{Bernoulli}\,(p)$ is the Bernoulli distribution with probability p. The next result is a corollary from Theorem E.1.

Corollary E.2. *We define $\mathcal{M}_1, mdp_2$ as above for $\delta \in [0, 1/4]$. Assume $N \geq \widetilde{O}(1)$. Then, we have*

1. *The minimax optimal lower bound for the suboptimality of LCB is*

$$\inf_{\widehat{a}_{\mathsf{LCB}} \in \mathcal{A}} \sup_{\mathcal{M} \in \{\mathcal{M}_1, \mathcal{M}_2\}} \mathbb{E}_{\mathcal{D}}\left[r(a^*) - r(\widehat{a}_{\mathsf{LCB}})\right] \gtrsim \sqrt{\frac{C^*}{N}}, \tag{34}$$

   where $\mathbb{E}_{\mathcal{D}}[\cdot]$ is the expectation over the offline dataset $\mathcal{D}$.

2. *The upper bound for suboptimality of TRUST mathces the lower bound above up to log factor. Namely, for any $\mathcal{M} \in \{\mathcal{M}_1, \mathcal{M}_2\}$, one has*

$$\mathbb{E}_{\mathcal{D}}\left[r(a^*) - V^{\pi_{TRUST}}\right] \lesssim \sqrt{\frac{C^* \log(dN)}{N}}. \tag{35}$$

The first claim comes from Theorem 2 of (Rashidinejad et al., 2021), while the second claim is a direct corollary to Theorem E.1.

E.1 Proof of Theorem E.1

Proof. Recall from Theorem 4.1 that for any comparator policy $(\varepsilon, \Delta, \pi)$ defined above, one has

$$V^{\pi} - V^{\pi_{TRUST}} \leq 2\mathcal{G}\left(\lceil\varepsilon\rceil\right),$$

where $\lceil\varepsilon\rceil := \inf\{\varepsilon' \in E : \varepsilon \leq \varepsilon'\}$. The following lemma upper bounds the quantile of Gaussian suprema $\mathcal{G}(\varepsilon)$ for each $\varepsilon \in E$. The proof is deferred to Appendix E.2.

Lemma E.3. *For $\varepsilon \in E$, one can upper bound $\mathcal{G}(\varepsilon)$ as follows.*

$$\mathcal{G}(\varepsilon) \leq \min\left\{\varepsilon \cdot \sqrt{d},\ 4D\sqrt{\log_+\left(\frac{4ed\varepsilon^2}{D^2}\right)}\right\} + \sqrt{2\varepsilon^2 \log\left(\frac{2|E|}{\delta}\right)} \tag{36}$$

where $\log_+(a) = \max(1, \log(a))$ and D is a quantity satisfying

$$D \geq \sqrt{\max_{i \in [d]}\left[\frac{\sigma_i^2}{N_i} - \frac{2\sigma_i^2}{N}\right] + \frac{\sum_{j=1}^d N_j \sigma_j^2}{N^2}}. \tag{37}$$

Applying Lemma E.3 to $\lceil\varepsilon\rceil \in E$, we obtain

$$V^{\pi} - V^{\pi_{TRUST}} \leq 2\min\left\{\lceil\varepsilon\rceil \cdot \sqrt{d},\ 4D\sqrt{\log_+\left(\frac{4ed\lceil\varepsilon\rceil^2}{D^2}\right)}\right\} + 2\sqrt{2\lceil\varepsilon\rceil^2 \log\left(\frac{2|E|}{\delta}\right)}. \tag{38}$$

Since $\varepsilon = \sum_{i=1}^d \frac{\sigma_i^2 \Delta_i^2}{N_i}$, we know from our discretization scheme in equation 10

$$\lceil\varepsilon\rceil \leq \alpha \cdot \sum_{i=1}^d \frac{\sigma_i^2 \Delta_i^2}{N_i}. \tag{39}$$

Bridging equation 39 into equation 38, we obtain our first claim. In order to get the second claim, we take $\sigma_i = 1$ for $i \in [d]$ and $\Delta = \pi_* - \widehat{\mu}$, which is the vector pointing at the vertex corresponding to the optimal arm from the uniform reference policy $\widehat{\mu}$ defined in equation 5. Then, we have

$$\sum_{i=1}^{d} \frac{\Delta_i^2 \sigma_i^2}{N_i} = \frac{1}{N_1} - \frac{2}{N} + \frac{1}{N} = \frac{1}{N_1} - \frac{1}{N} \leq \frac{1}{N_1},$$

where N_1 is the sample size for the optimal arm a_1. Therefore, we can further bound equation 38 as

$$V^{\pi_*} - V^{\pi_{TRUST}} \leq 4D\sqrt{\log_+\left(\frac{4\alpha e d}{N_1 D^2}\right)} + 2\sqrt{\frac{2\alpha}{N_1}\log\left(\frac{2|E|}{\delta}\right)}. \tag{40}$$

Finally, we take a specific value of D and lower bound N_1 via Chernoff bound in Lemma E.7. From Lemma E.7, we know that when $N \geq 8C_{\min}\log(d/\delta)$, with probability at least $1-\delta$, we have

$$N_i \geq \frac{1}{2} N\mu(a_i) \tag{41}$$

for any $i \in [d]$. Recall the definition of D in equation 37, we know that D can be arbitrary value greater than $\sqrt{\max_{i\in[d]}\left[\frac{\sigma_i^2}{N_i} - \frac{2\sigma_i^2}{N}\right] + \frac{\sum_{j=1}^d N_j\sigma_j^2}{N}}$. Then, when $\sigma_i = 1$, one has

$$\sqrt{\max_{i\in[d]}\left[\frac{\sigma_i^2}{N_i} - \frac{2\sigma_i^2}{N}\right] + \frac{\sum_{j=1}^d N_j\sigma_j^2}{N^2}} \leq \sqrt{\frac{1}{\min_{i\in[d]} N_i}}.$$

We denote $N_j = \min_{i\in[d]} N_i$ (when there are multiple minimizers, we arbitrarily pick one). Then, we have

$$\sqrt{\max_{i\in[d]}\left[\frac{\sigma_i^2}{N_i} - \frac{2\sigma_i^2}{N}\right] + \frac{\sum_{j=1}^d N_j\sigma_j^2}{N^2}} \leq \sqrt{\frac{1}{N_j}} \leq \sqrt{\frac{2}{N\mu(a_j)}} \leq \sqrt{\frac{2}{N\cdot\min_{i\in[d]}\mu(a_i)}} = \sqrt{\frac{2C_{\min}}{N}}.$$

Therefore, we take $D = \sqrt{\frac{2C_{\min}}{N}}$ in equation 40 and apply $N_1 \geq \frac{1}{2}N\mu(a_i)$ to obtain

$$V^{\pi_*} - V^{\pi_{TRUST}} \leq 4\sqrt{\frac{2C_{\min}}{N}\log_+\left(\frac{4\alpha e d C^*}{C_{\min}}\right)} + 4\sqrt{\frac{\alpha C^*}{N}\log\left(\frac{2|E|}{\delta}\right)}, \tag{42}$$

which proves equation 32. Finally, when $C^* \simeq C_{\min}$, one has

$$V^{\pi_*} - V^{\pi_{TRUST}} \lesssim \sqrt{\frac{C^*}{N}\log\left(\frac{2d|E|}{\delta}\right)}.$$

Therefore, we conclude. □

E.2 Proof of Lemma E.3

Proof. Recall that $\Delta = (\Delta_1, \Delta_2, ..., \Delta_d)^\top$ is the improvement vector, $\eta = (\eta_1, \eta_2, ..., \eta_d)^\top$ is the noise vector, where entries are independent and $\eta_i \sim \mathcal{N}(0, \sigma_i^2/N_i)$ and N_i is the sample size of arm a_i in the offline dataset. To proceed with the proof, let's further define

$$\widetilde{\eta} = (\widetilde{\eta}_1, \widetilde{\eta}_2, ..., \widetilde{\eta}_d)^\top, \quad \widetilde{\Delta} = \left(\widetilde{\Delta}_1, \widetilde{\Delta}_2, ..., \widetilde{\Delta}_d\right)^\top, \quad \text{where} \quad \widetilde{\eta}_i = \eta_i\frac{\sqrt{N_i}}{\sigma_i},\ \widetilde{\Delta}_i = \frac{\Delta_i\sigma_i}{\sqrt{N_i}}. \tag{43}$$

With this notation, one has

$$\widetilde{\eta} \sim \mathcal{N}(0, I_d), \quad \eta^\top\Delta = \widetilde{\eta}^\top\widetilde{\Delta}.$$

We also write the equivalent trust region (for $\widetilde{\Delta}$) as

$$\widetilde{\mathsf{C}}(\varepsilon) = \left\{\widetilde{\Delta} \in \mathbb{R}^d : \frac{\sqrt{N_i}}{\sigma_i}\widetilde{\Delta}_i + \widehat{\mu}_i \geq 0, \quad \sum_{i=1}^{d}\left[\frac{\sqrt{N_i}}{\sigma_i}\widetilde{\Delta}_i + \widehat{\mu}_i\right] = 1, \quad \left\|\widetilde{\Delta}\right\|_2 \leq \varepsilon\right\}, \tag{44}$$

where $\widehat{\mu} = (\widehat{\mu}_1, \widehat{\mu}_2, ..., \widehat{\mu}_d)^\top$ is the policy weight for the reference policy. From the definition above, one has for any $\varepsilon > 0$,

$$\Delta \in \mathsf{C}(\varepsilon) \;\Leftrightarrow\; \widetilde{\Delta} \in \widetilde{\mathsf{C}}(\varepsilon).$$

Then, we apply Lemma E.4 to $\sup_{\Delta \in \mathsf{C}(\varepsilon)} \Delta^\top \eta$ for a $\varepsilon \in E$. One has with probability at least $1 - \frac{\delta}{|E|}$,

$$\left| \sup_{\Delta \in \mathsf{C}(\varepsilon)} \Delta^\top \eta - \mathbb{E} \sup_{\Delta \in \mathsf{C}(\varepsilon)} \Delta^\top \eta \right| \leq \sqrt{2\varepsilon^2 \log\left(\frac{2|E|}{\delta}\right)}.$$

From a union bound, one immediately has with probability at least $1 - \delta$, for any $\varepsilon \in E$, it holds that

$$\sup_{\Delta \in \mathsf{C}(\varepsilon)} \Delta^\top \eta \leq \mathbb{E} \sup_{\Delta \in \mathsf{C}(\varepsilon)} \Delta^\top \eta + \sqrt{2\varepsilon^2 \log\left(\frac{2|E|}{\delta}\right)}. \tag{45}$$

From the definition of $\mathcal{G}(\varepsilon)$ in equation 3.2, we know that $\mathcal{G}(\varepsilon)$ is the minimal quantity that satisfy equation 45 with probability at least $1 - \delta$. Therefore, one has

$$\mathcal{G}(\varepsilon) \leq \mathbb{E} \sup_{\Delta \in \mathsf{C}(\varepsilon)} \Delta^\top \eta + \sqrt{2\varepsilon^2 \log\left(\frac{2|E|}{\delta}\right)} = \mathbb{E}_{\widetilde{\eta} \sim \mathcal{N}(0, I_d)} \left[\sup_{\widetilde{\Delta} \in \widetilde{\mathsf{C}}(\varepsilon)} \widetilde{\Delta}^\top \widetilde{\eta} \right] + \sqrt{2\varepsilon^2 \log\left(\frac{2|E|}{\delta}\right)} \quad \forall \varepsilon \in E. \tag{46}$$

Note that, the first term in the RHS of equation 46 is well-defined as localized Gaussian width over the convex hull defined by the trust region $\mathsf{C}(\varepsilon)$ (or equivalently, $\widetilde{\mathsf{C}}(\varepsilon)$). We denote

$$T := \left\{ \widetilde{\Delta} \in \mathbb{R}^d : \frac{\sqrt{N_i}}{\sigma_i} \widetilde{\Delta}_i + \widehat{\mu}_i \geq 0, \quad \sum_{i=1}^d \left[\frac{\sqrt{N_i}}{\sigma_i} \widetilde{\Delta}_i + \widehat{\mu}_i \right] = 1 \right\}. \tag{47}$$

We immediately have that T is a convex hull of d points in $\mathbb{R}^d$ and the vertices of this convex hull are the vertices of the simplex in $\mathbb{R}^d$ shifted by the reference policy $\widehat{\mu}$. In what follows, we plan to apply Lemma E.5 to the localized Gaussian width of $T \cap \varepsilon \mathbb{B}_2$. However, T is not subsumed by the unit ball in $\mathbb{R}^d$, so we need to do some additional scaling. Note that, the zero vector is included in T. Let's compute the farthest distance for the vertices of T. We denote the i-th vertex of T as

$$\widetilde{\Delta} = \left(-\frac{\sigma_1}{\sqrt{N_1}} \widehat{\mu}_1, ..., -\frac{\sigma_{i-1}}{\sqrt{N_{i-1}}} \widehat{\mu}_{i-1}, \frac{\sigma_i}{\sqrt{N_i}} (1 - \widehat{\mu}_i), -\frac{\sigma_{i+1}}{\sqrt{N_{i+1}}} \widehat{\mu}_{i+1}, ..., -\frac{\sigma_d}{\sqrt{N_d}} \widehat{\mu}_d \right). \tag{48}$$

The ℓ_2-norm of this improvement vector is

$$\left\| \widetilde{\Delta} \right\|_2 = \sqrt{\frac{\sigma_i^2}{N_i} - \frac{2\sigma_i^2}{N} + \frac{\sum_{i=1}^d N_i \sigma_i^2}{N^2}},$$

where N is the total sample size of the offline dataset. Therefore, the maximal radius of T can be upper bounded by D, where D is any quantity that satisfies

$$D \geq \sqrt{\max_{i \in [d]} \left[\frac{\sigma_i^2}{N_i} - \frac{2\sigma_i^2}{N} \right] + \frac{\sum_{j=1}^d N_j \sigma_j^2}{N^2}}. \tag{49}$$

We denote $S = \frac{1}{D} \cdot T := \left\{ \frac{1}{D} \cdot x : x \in T \right\}$. Then, from Lemma E.5, one has

$$\begin{aligned}
\mathbb{E}_{\widetilde{\eta} \sim \mathcal{N}(0, I_d)} \left[\sup_{\widetilde{\Delta} \in \widetilde{\mathsf{C}}(\varepsilon)} \widetilde{\Delta}^\top \widetilde{\eta} \right] &= \mathbb{E}_{\widetilde{\eta} \sim \mathcal{N}(0, I_d)} \left[\sup_{\widetilde{\Delta} \in T \cap \varepsilon \mathbb{B}_2} \widetilde{\Delta}^\top \widetilde{\eta} \right] \\
&= D \cdot \mathbb{E}_{\widetilde{\eta} \sim \mathcal{N}(0, I_d)} \left[\sup_{\widetilde{\Delta} \in S \cap \frac{\varepsilon}{D} \cdot \mathbb{B}_2} \widetilde{\Delta}^\top \widetilde{\eta} \right] \\
&\qquad \left(S \cap \tfrac{\varepsilon}{D} \cdot \mathbb{B}_2 \text{ can be got by scaling } T \cap \varepsilon \mathbb{B}_2 \text{ bt } \tfrac{1}{D}\right) \\
&\leq D \cdot \left[\left(4 \sqrt{\log_+ \left(4ed \left(\frac{\varepsilon^2}{D^2} \wedge 1 \right) \right)} \right) \wedge \left(\frac{\varepsilon}{D} \sqrt{d} \right) \right] \\
&\qquad \left(\text{Take } s = \tfrac{\varepsilon}{D} \text{ and } M = d \text{ in Lemma E.5}\right) \\
&= D \cdot \left[\left(4 \sqrt{\log_+ \left(4ed \left(\frac{\varepsilon^2}{D^2} \right) \right)} \right) \wedge \left(\frac{\varepsilon}{D} \sqrt{d} \right) \right]. \\
&\qquad \left(\varepsilon \leq D \text{ for any } \varepsilon \in E\right)
\end{aligned}$$

This finishes the proof. □

E.3 AUXILIARY LEMMAS

Lemma E.4 (Concentration of Gaussian suprema, Exercise 5.10 in (Wainwright, 2019)). *Let $\{X_\theta, \theta \in \mathbb{T}\}$ be a zero-mean Gaussian process, and define $Z = \sup_{\theta\in\mathbb{T}} X_\theta$. Then, we have*

$$\mathbb{P}[|Z - \mathbb{E}[Z]| \geq \delta] \leq 2\exp\left(-\frac{\delta^2}{2\sigma^2}\right),$$

where $\sigma^2 := \sup_{\theta\in\mathbb{T}} \operatorname{var}(X_\theta)$ is the maximal variance of the process.

Lemma E.5 (Localized Gaussian Width of a Convex Hull, Proposition 1 in (Bellec, 2019)). *Let $d \geq 1, M \geq 2$ and T be the convex hull of M points in $\mathbb{R}^d$. We write $\mathbb{B}_2 = \left\{x \in \mathbb{R}^d : \|x\|_2 \leq 1\right\}$ and $s\mathbb{B}_2 = \left\{s \cdot x : x \in \mathbb{R}^d, \|x\|_2 \leq 1\right\}$. Assume $T \subset \mathbb{B}_2^d(1)$. Let $g \in \mathbb{R}^d$ be a standard Gaussian vector. Then, for all $s > 0$, one has*

$$\mathbb{E}\left[\sup_{x\in T\cap s\mathbb{B}_2} x^\top g\right] \leq \left(4\sqrt{\log_+\left(4eM\left(s^2 \wedge 1\right)\right)}\right) \wedge \left(s\sqrt{d \wedge M}\right), \tag{50}$$

where $\log_+(a) = \max(1, \log(a)), a \wedge b = \min\{a, b\}$.

Lemma E.6 (Chernoff bound for binomial random variables, Theorem 2.3.1 in (Vershynin, 2020)). *Let X_i be independent Bernoulli random variables with parameters p_i. Consider their sum $S_N = \sum_{i=1}^N X_i$ and denote its mean by $\mu = \mathbb{E}S_N$. Then, for any $t > \mu$, we have*

$$\mathbb{P}\{S_N \geq t\} \leq e^{-\mu}\left(\frac{e\mu}{t}\right)^t.$$

Lemma E.7 (Chernoff bound for offline MAB). *Under the setting in Theorem E.1, we have*

$$\mathbb{P}\left(N_i \geq \frac{1}{2}N\mu(a_i) \quad \forall i \in [d]\right) \leq 1 - d\exp\left(-\frac{N \cdot \min_{j\in[d]} \mu(a_j)}{8}\right),$$

Proof. For arm $i \in [d]$, we take $\mu = N\mu(a_i)$ and $t = \frac{1}{2}M\mu(a_i)$ in Lemma E.6 and obtain

$$\mathbb{P}\left(N_i \geq \frac{1}{2}N\mu(a_i)\right) \leq \exp\left(-N\mu(a_i)\right)\cdot\left(\frac{eN\mu(a_i)}{\frac{1}{2}N\mu(a_i)}\right)^{\frac{1}{2}N\mu(a_i)} = \exp\left(N\mu(a_i)\left[-1 + \frac{1}{2}\log(2e)\right]\right) \leq \exp\left(-\frac{N\mu(a_i)}{8}\right).$$

We finish the proof by a union bound for all arms. □

F PROOF OF LEMMA B.1

Proof. Recall the definition of $\lceil\varepsilon\rceil$:

$$\lceil\varepsilon\rceil := \inf\{\varepsilon' \in E : \varepsilon' \geq \varepsilon\}. \tag{51}$$

We additionally define

$$\lfloor\varepsilon\rfloor := \sup\{\varepsilon' \in E : \varepsilon' < \varepsilon\}. \tag{52}$$

Specially, if there is no $\varepsilon' \in E$ such that $\varepsilon' < \varepsilon$, then we define $\lfloor\varepsilon\rfloor = 0$. Then we know for any $\varepsilon \leq \varepsilon_0 \in E$ (ε_0 is the largest possible radius) and a finite set E, it holds that

$$\lfloor\varepsilon\rfloor < \varepsilon \leq \lceil\varepsilon\rceil, \quad \text{and} \quad \varepsilon = \lceil\varepsilon\rceil \text{ if and only if } \varepsilon \in E. \tag{53}$$

For any $\varepsilon \in E$, recall $\widehat{\Delta}_\varepsilon$ is the optimal improvement vector within $\mathsf{C}(\varepsilon)$ defined in equation 7. It holds that

$$\begin{aligned}\widehat{\Delta}_\varepsilon :&= \arg\max_{\Delta \in \mathsf{C}(\varepsilon)} \Delta^\top \widehat{r} = \arg\max_{\Delta \in \mathsf{C}(\varepsilon)} \left[\Delta^\top \widehat{r} - \mathcal{G}(\varepsilon)\right] && \text{(since } \mathcal{G}(\varepsilon) \text{ does not depend on } \Delta\text{)} \\ &= \arg\max_{\Delta \in \mathsf{C}(\varepsilon)} \left[\Delta^\top \widehat{r} - \mathcal{G}(\lceil \varepsilon \rceil)\right] && (\varepsilon \in E, \text{ so } \lceil \varepsilon \rceil = \varepsilon) \\ &\leq \arg\max_{\varepsilon' \in (\lfloor \varepsilon \rfloor, \lceil \varepsilon \rceil], \Delta \in \mathsf{C}(\varepsilon')} \left[\Delta^\top \widehat{r} - \mathcal{G}(\lceil \varepsilon' \rceil)\right].\end{aligned}$$

On the other hand, when $\varepsilon \in E$ and $\varepsilon' \in (\lfloor \varepsilon \rfloor, \lceil \varepsilon \rceil]$, one has $\lceil \varepsilon' \rceil = \lceil \varepsilon \rceil = \varepsilon$, so

$$\arg\max_{\varepsilon' \in (\lfloor \varepsilon \rfloor, \lceil \varepsilon \rceil], \Delta \in \mathsf{C}(\varepsilon')} \left[\Delta^\top \widehat{r} - \mathcal{G}(\lceil \varepsilon' \rceil)\right] = \arg\max_{\varepsilon' \in (\lfloor \varepsilon \rfloor, \lceil \varepsilon \rceil], \Delta \in \mathsf{C}(\varepsilon')} \left[\Delta^\top \widehat{r} - \mathcal{G}(\lceil \varepsilon \rceil)\right] \leq \arg\max_{\Delta \in \mathsf{C}(\varepsilon)} \left[\Delta^\top \widehat{r} - \mathcal{G}(\lceil \varepsilon \rceil)\right],$$

where the last inequality comes from the fact that $\mathsf{C}(\varepsilon') \subset \mathsf{C}(\varepsilon)$ when $\varepsilon' \leq \lceil \varepsilon \rceil = \varepsilon$ by definition of the trust region in equation 6. Combining two inequalities above, we have for any $\varepsilon \in E$,

$$\left(\varepsilon, \widehat{\Delta}_\varepsilon\right) = \arg\max_{\varepsilon' \in (\lfloor \varepsilon \rfloor, \lceil \varepsilon \rceil], \Delta \in \mathsf{C}(\varepsilon')} \left[\Delta^\top \widehat{r} - \mathcal{G}(\lceil \varepsilon' \rceil)\right], \tag{54}$$

where the variables in RHS above are ε' and Δ, and Therefore, from the definition of we have

$$\left(\widehat{\varepsilon}_*, \widehat{\Delta}_*\right) = \arg\max_{\varepsilon \in E} \arg\max_{\varepsilon' \in (\lfloor \varepsilon \rfloor, \lceil \varepsilon \rceil], \Delta \in \mathsf{C}(\varepsilon')} \left[\Delta^\top \widehat{r} - \mathcal{G}(\lceil \varepsilon' \rceil)\right] = \arg\max_{\varepsilon \leq \varepsilon_0, \Delta \in \mathsf{C}(\varepsilon)} \left[\Delta^\top \widehat{r} - \mathcal{G}(\lceil \varepsilon \rceil)\right].$$

This finishes the proof. □

G AUGMENTATION WITH LCB

To determine the most effective final policy, we can compare the outputs of the LCB and Algorithm 1 and combine both policies, based on the relative magnitude of their corresponding lower bounds. Specifically, the combined policy is

$$\pi_{\mathsf{combined}} = \begin{cases} \widehat{a}_{\mathsf{LCB}} & \text{If } \max_{a_i \in \mathcal{A}} l_i \geq w_{\mathsf{TR}}^\top \widehat{r} - \mathcal{G}(\lceil \widehat{\varepsilon}_* \rceil) - \sqrt{\dfrac{2\log(1/\delta)}{\sum_{j=1}^d N_j/\sigma_j^2}}, \\ w_{\mathsf{TR}} & \text{If } \max_{a_i \in \mathcal{A}} l_i < w_{\mathsf{TR}}^\top \widehat{r} - \mathcal{G}(\lceil \widehat{\varepsilon}_* \rceil) - \sqrt{\dfrac{2\log(1/\delta)}{\sum_{j=1}^d N_j/\sigma_j^2}}, \end{cases} \tag{55}$$

where $l_i = \widehat{r}_i - b_i$ is defined in equation 1 and $\mathcal{G}(\varepsilon)$ is defined in Definition 3.2. This combined policy will perform at least as well as LCB with high probability. More specifically, we have

Corollary G.1. *We denote the arm chosen by LCB as $\widehat{a}_{\mathsf{LCB}}$. We also denote $r(\cdot)$ as the true reward of a policy (deterministic or stochastic). With probability at least $1 - 3\delta$, one has*

$$V^{\pi_{\mathsf{combined}}} \geq \max_{a_i \in \mathcal{A}} l_i. \tag{56}$$

Proof. We denote $\widehat{r}(\widehat{a}_{\mathsf{LCB}}) = r_{\widehat{a}_{\mathsf{LCB}}}$ and $\widehat{r}(w_{\mathsf{TR}})$ as the empirical reward of the policy returned by LCB and Algorithm 1, respectively. Recall the uncertainty term of LCB in equation 1 and of Algorithm 1 in equation 55, we write $b(\widehat{a}_{\mathsf{LCB}}) = b_{\widehat{a}_{\mathsf{LCB}}}$ and $b(w_{\mathsf{TR}}) = \mathcal{G}(\lceil \widehat{\varepsilon}_* \rceil) + \sqrt{2\log(1/\delta)/[\sum_{j=1}^d N_j/\sigma_j^2]}$. Then, from Theorem 4.1, equation 2 and a union bound, we know with probability at least $1 - 3\delta$, it holds that

$$r(\widehat{a}_{\mathsf{LCB}}) \geq \widehat{r}(\widehat{a}_{\mathsf{LCB}}) - b(\widehat{a}_{\mathsf{LCB}}),\ r(w_{\mathsf{TR}}) \geq \widehat{r}(w_{\mathsf{TR}}) - b(w_{\mathsf{TR}}),$$

which implies

$$\begin{aligned} V^{\pi_{\mathsf{combined}}} &\geq \widehat{r}(\pi_{\mathsf{combined}}) - b(\pi_{\mathsf{combined}}) \\ &\geq \widehat{r}(\widehat{a}_{\mathsf{LCB}}) - b(\widehat{a}_{\mathsf{LCB}}) && \text{(By equation 55)} \\ &= \max_{a_i \in \mathcal{A}} l_i. && \text{(By the definition of } \widehat{a}_{\mathsf{LCB}} \text{ in equation 3)} \end{aligned}$$

Therefore, we conclude. □

H EXPERIMENT DETAILS

We did experiments on Mujoco environment in the D4RL dataset (Fu et al., 2020). All environments we test on are v3. Since the original D4RL dataset does not include the exact form of logging policies, we retrain SAC (Haarnoja et al., 2018) on these environment for 1000 episodes and keep record of the policy in each episode. We test 4 environments in two settings, denoted as '1-traj-low' and '1-traj-high'. In either setting, the offline dataset is generated from 100 policies with one trajectory from each. In the '1-traj-low' setting, the data is generated from the first 100 policies in the training process of SAC, while in the '1-traj-high' setting, it is generated from the policy in $(10x + 1)$-th episodes in the training process.

For all experiments on Mujoco, we average the results over 4 random seeds (from 2023 to 2026), and to run CQL, we use default hyper-parameters in `https://github.com/young-geng/CQL` to run 2000 episodes. For TRUST, we run it using a fixed standard deviation level $\sigma_i = 150$ for all experiments.