We thank all the reviewers for their valuable feedback. We first address some common concerns.

Assumption 1. We will emphasize this assumption at the beginning of the paper. We will mention that although this assumption does not make explicit assumption about the model, it makes implicit assumption about the MDP.

Computational efficiency. We have provided an algorithm in Appendix C to compute the λ-sensitivity. This algorithm requires only an oracle to test whether a given state-action pair is ε-independent with a sequence of state-action pairs, which is again equivalent to evaluating the width function (defined in Line 140). To evaluate the width function, it suffices to have access to a regression oracle by invoking known reductions in [1] and [2], and having access to a regression oracle is indeed a weak assumption in practice. We will add more discussion on the computational efficiency.

[1] Practical Contextual Bandits with Regression Oracles [2] Active Learning for Cost-Sensitive Classification

—— To Reviewer #1 ——

The difference between the sensitive sampling in this paper and the prior work. Sensitivity sampling was proposed and applied in a different context (e.g., in [20, 21, 32] for clustering) to compress datasets. Our definition of sensitivity is similar to previous results, and the main technical novelty here is that we can show the sum of the sensitivity can be upper bounded in terms of the eluder dimension of the function class (Lemma 1). Such a result is crucial for obtaining an upper bound on the complexity of the bonus function. We will add more comparison in the next version.

How sensitivity sampling helps address the problem. To account for the dependency structure in the data sequence, we need to construct a bonus function with bounded complexity, and thus we subsample the dataset to reduce its size. As mentioned in Line 232-233, sensitivity measures the importance of each data point z in a dataset. In our analysis (Proposition 1), we show that by importance sampling according to the sensitivity, the subsampled dataset has bounded size while the confidence region is approximately preserved, and thus the bonus function has bounded complexity.

The connection between Lemma 9 and 10 and Proposition 3 and Lemma 2 in [44]. Lemma 9 and 10 are indeed adapted from Proposition 3 and Lemma 2 in [44], and the main difference is that our confidence region is defined using the subsampled dataset. We have discussed this in Line 331-336, and we will make this clearer in the next version.

Line 599. There is a typo here in the definition of $\mathbb{F}_h^\tau$, which should also includes (s_h^τ, a_h^τ). Conditioned on $\mathbb{F}_h^\tau$ ((s_h^τ, a_h^τ) is fixed and s_{h+1}^τ is random), $\mathbb{E}[V(s_{h+1}^\tau)] = \sum_{s' \in \mathcal{S}} P(s'|s_h^\tau, a_h^\tau)V(s')$, and thus the conditional expectation is 0.

—— To Reviewer #2 ——

Computational complexity / doubling trick. The running time of the current algorithm is polynomial in T, which we will emphasize more in the next version. We do believe the running time can be further reduced by using the doubling trick or online sampling algorithms, and we leave it as a future work to further optimize the running time.

The bonus is related to the properties of the function class. Lemma 9 is adapted from prior work [44] (to handle confidence regions defined using the subsampled dataset), and is analogous to the elliptical potential lemma in the linear case. Note that for the linear case, the summation of the bonus function can be upper bounded by $\widetilde{O}(d)$, and the feature dimension d is also a property of the function class.

—— To Reviewer #3 ——

Assumption. We agree with the reviewer that our assumption allows us to add bonuses and still be able to represent the value function, and we will make this more explicit. However, since we work with *general value function* approximation in this paper, we do need to make some assumptions to make the problem tractable. We would like to remind the reviewer that even for the case of linear functions, assumptions weaker than linear MDP either result in computationally inefficient algorithms (as in [62]) or require the transition to be (nearly) deterministic (as in [18, 19]).

Confidence intervals. Note that for the class of linear functions, the width function defined in Line 140 recovers the usual confidence interval ($\|\phi\|_{\Sigma^{-1}}$) for linear functions. It is not clear to us why the reviewer believes that we "just assume that the confidence intervals are available". Moreover, our definition of the bonus function requires non-trivial effort by extending techniques from sensitivity sampling to make sure the complexity of the bonus function is low. We believe this method itself could be of interest to the machine learning community in general.

Two claims that sends the wrong message. (1) By "by a more refined analysis specialized to the tabular setting, ...", we mean the sample complexity can be improved from $\widetilde{O}(\sqrt{|\mathcal{S}|^3|\mathcal{A}|^3H^2T})$ to $\widetilde{O}(\sqrt{|\mathcal{S}|^2|\mathcal{A}|^2H^2T})$. We will remove that sentence from Remark 1 in the next version. (2) We will remove [62] from that list to avoid possible confusion.

—— To Reviewer #4 ——

Typos. Thanks for pointing out the typos, we will polish the paper and fix these typos in the next version.

Constructing $\mathcal{Z}_j^\alpha$. For each data z, we should also add z into $\mathcal{Z}_{j(z)}^\alpha$ if $j(z) \le N_\alpha$. Sorry for missing this step.

The confidence region looks very pessimistic. Our confidence interval recovers the usual confidence interval ($\|\phi\|_{\Sigma^{-1}}$) for linear functions and thus correctly balances exploration and exploitation in that case. Also, as we show in Lemma 10, the summation of the bonus function can be upper bounded in terms of the eluder dimension of the function class, and thus provides a finite regret bound. We will make this clear in the final version.

Empirical results. We will consider adding empirical results in the next version. Thanks for the suggestion.

Comparison with Ayoub et al. For linear functions, our assumption is equivalent to the linear MDP assumption, where the assumption in Ayoub et al. assumes that the true model is a linear combination of some known models. Therefore, these two assumptions are already incomparable for linear functions, and we will make this clear in the next version.