We thank all reviewers for appreciating our technical contributions: a novel algorithm and a sound theoretical analysis. Please find our response to each reviewer below.

To Reviewer #1:

-*Definition 3.1*: We will state in Definition 3.1 explicitly that we assume $\gamma > 0$ to avoid confusion. Thanks for pointing out. We will also state that we assume $\gamma > 0$ again before Theorem 6.1 to emphasize this assumption.

-*Assuming $\gamma > 0$ is restrictive:* Assuming γ strictly positive is not a restrictive assumption for the *finite action setting* considered in this paper. First, in the contextual linear bandit literature, this assumption is widely discussed. See, e.g., [1,2]. The notion, context, in the bandit literature is essentially $\phi(s)$ in our paper and the number of contexts can also be infinite. Second, there are many natural environments in RL which satisfy this assumption. For example, in many environments, states can be classified as good states and bad states. In these environments, an agent can obtain a reward only if it is in a good state. There are also two kinds of actions: good actions and bad actions. If the agent is in a good state and chooses a good action, the agent will transit to a good state. If the agent chooses a bad action, the agent will transit to a bad state. If the agent is in a bad state, whatever action the agent chooses, the agent will transit to a bad state. Note that for this kind of environments, there is a strictly positive gap between good actions and bad actions when the agent is in good states and there is no difference between good actions and bad actions when the agent is in bad states. In this case, γ is strictly positive, since by Definition 3.1, we take the minimum over all state-action pairs with *strictly* positive gap. These environments are natural generalizations of the combination lock environment [3]. Some Atari games, e.g. Freeway, have a similar flavor as these environments.

-*What if γ is unknown:* While our algorithm requires γ as an input for deciding hyper-parameters, it is easy to see from the proof that as long as we choose some $\gamma' \leq \gamma$ and choose hyper-parameters according to γ', the sample complexity bound still holds. Therefore, we can just do a grid search over $[0, 1]$ to find such γ'. We will add more discussion.

-*Section 5:* We will put an overview of our algorithm at the beginning of Section 5. The design of our algorithm is actually fairly natural as we explain below. Since we know Q^* belongs to a function class, the natural idea is to use regression to recover it (Line 9 of Algorithm 2), for which we need unbiased samples. To obtain unbiased samples for level h, we need to make sure that we execute the optimal policy for level $h' = h + 1, \ldots, H$ (this accounts for our definition in Eq. 1). Therefore, in Line 1-2 of Algorithm 1, we learn Q^* from the last level to the first level (learning at the last level is equivalent to the stochastic bandit problem, where we have unbiased samples). The tricky part is that the Q-function we learned, say at level $h' = h + 1, \ldots, H$, is only guaranteed to be optimal with respect to the roll-in policy we used to gather samples. When we learn Q^* at level h, the roll-in policy for level $h' = h + 1, \ldots, H$ can change. To address this problem, we need to check that the learned Q-function is still optimal for the new roll-in policy (Algorithm 3 and the DSEC oracle). We have explained the intuition of the DSEC oracle in Section 1.

-*Abstract / introduction:* We follow one of the common writing styles that first states our precise result and gives a description of our main techniques in the abstract, and then puts an informal main theorem and a technical overview in the introduction. We are happy to adjust our writing style.

-*Regularizer in Oracle 4.1 / minor concerns / typos*: We will fix them accordingly. Thanks for pointing out.

To Reviewer #2:

-*Q-function is not exactly linear:* This is a great question. Our algorithm can still be applied if the best linear predictor has approximation error smaller than $\frac{\gamma}{\mathrm{poly}(H,d)}$. The proof is essentially the same. We will add more discussion.

-*Most state-action pairs have a large gap:* This is an interesting question. We do believe the sample complexity can be improved with proper quantification on "most state-action pairs". We will discuss this in the final version.

-*Need to collect on-the-go reward*: We agree this incurs higher computation complexity. We will list improving the computation complexity as an open problem.

-*Proof of Theorem 6.1:* In Line 401 and 402 we use the induction hypothesis of later levels. We will add more details.

-*Notation γ:* Thanks for pointing out. We will change the notation to $\triangle$.

To Reviewer #3:

-*More general setting / generalization of Lemma 4:* Thanks for the question! We have discussed going beyond linear function approximation in Section 7. A natural extension of Lemma 4 is to study the Fisher information matrix, though this generalization is not trivial. Once we know the number of calls to the DSEC oracle is bounded (via a generalization of Lemma 4), we can then reuse the current proof framework.

-*Optimality / connections with stochastic bandits:* We will discuss more about connections with stochastic bandits. Thanks for the suggestion. While Line 6-8 is not optimal for the stochastic bandit case, it not obvious how to improve that part since Algorithm 3 also depends on the data collection policy. We will add more discussion.

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits.

[2] Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback.

[3] Sham M Kakade. On the sample complexity of reinforcement learning.