Reviewer #1 We thank the reviewer for the valuable comments.

1. Large constants in Ψ and κ:
   The constant 400 in Ψ comes from Lemma B.5, which shows how to construct a normal barrier in the lifted domain from an arbitrary self-concordant barrier in the original decision set. However, as we mention in Footnote 2, our algorithm works with *any* normal barrier, not just this particular one (since our proofs only use those general properties of normal barriers from Section B.2). Therefore, the constant could be much smaller as long as a normal barrier with a smaller constant exists, which is indeed true for several canonical examples (see Nesterov and Nemirovskii, 1994). Similarly, the constant in κ is determined by the constant in Ψ (see the proof of Lemma B.12), and thus could be smaller as well.
   As we discuss in the paper, our algorithm is essentially SCRiBLe with a new sampling scheme and a new adaptive learning rate schedule. We thus believe that our algorithm is at least as empirically viable as SCRiBLe.
2. Other minor comments and missing related works:
   Thanks for pointing these out! We will revise the paper accordingly.

Reviewer #2 We thank the reviewer for the valuable comments.

Reviewer #3 We thank the reviewer for the valuable comments.

1. Efficiency of OMD with log-barrier regularizer for MDPs:
   In general, since the OMD step is a convex optimization problem with poly$(|S|, |A|)$ linear constraints, one could apply any standard convex solver to implement the algorithm efficiently. Jin et al. 2020 were able to reduce the problem to another optimization problem with only positivity constraints due to the special structure of the entropy regularizer (which does not hold for log-barrier), but in the end they still require applying a convex solver to implement the OMD step.
2. How to sample $\boldsymbol{s}_t$ efficiently:
   Yes, the way you mention is correct and efficient (except that in the end one also needs to normalize the norm to 1). Another efficient way to sample $\boldsymbol{s}_t$ is to first uniformly randomly sample a point s from a d-dimensional unit sphere, then let $\boldsymbol{s}_t = \boldsymbol{H}_t^{\frac{1}{2}} \begin{bmatrix} s \\ 0 \end{bmatrix}$, and finally normalize $\boldsymbol{s}_t$. We will add these discussions to the final version.
3. Typos and clarification:
   Thanks for pointing out the typos on Page 13 and Page 16. We will revise the paper accordingly. As for the question regarding line 587, note that in the description of Algorithm 2, we restrict the choice of $\boldsymbol{w}_t$ in $\boldsymbol{\Omega}' = \left\{\boldsymbol{w} = (w, 1) : w \in \Omega, \pi_{w_1}(w) \leq 1 - \frac{1}{T}\right\}$, therefore, we have $1 - \pi_{\boldsymbol{w}_1}(\boldsymbol{w}_t) = 1 - \pi_{w_1}(w_t) \geq \frac{1}{T}$.

Reviewer #4 We thank the reviewer for the valuable comments and suggestions, and will take them into account when revising the paper.