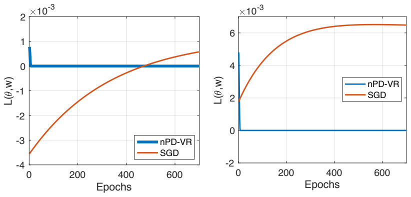
The authors would like to thank all the three reviewers for their useful feedback and the area chair for handling this paper. To address the reviewers' comments, upon acceptance of this paper, we will *(i) include numerical experiment results*, *(ii) provide an explicit bound on α, β*, and *(iii) improve clarity*. Some common concerns are as follows.

●● **Empirical Results**: To illustrate the pratical performance of nPD-VR, we experimented with `MountainCar` dataset w/ $m = 5000$. We ran Sarsa to obtain a good policy, then we generate a trajectory of the state-action pairs. For the nonlinearity, we parametrize V as a two-layer neural network with n hidden neurons. We set $\gamma = 0.95$, $\alpha = 10^{-4}$, $\beta = 10^{-8}$ and constraint as $\boldsymbol{\theta} \in \Theta = [0,1]^n$, $\boldsymbol{w} \in [0,100]^n$. Trajectory of the objective $\mathcal{L}(\boldsymbol{\theta}^{(k)}, \boldsymbol{w}^{(k)})$ is shown on the right. The objective of nPD-VR converges to (close to) zero in 4-5 passes on data, while a single timescale SGD on (16) takes a long time (or fail) to converge. Details of this experiment will be found in final version.



(Left) $n = 50$ neurons (Right) $n = 100$ neurons.

Reviewer 1: We thank the reviewer for providing constructive and supportive comments.

Typos: We apologize for the typos made. **Firstly**, thanks to your suggestions, we have corrected the definition of the projection Π and fixed the constant for the cost function. **Secondly**, we clarify that our algorithm only guarantees approximate *stationary points* to the *saddle point* MSPBE problem. They will be corrected in the final version.

Assumption 1: We acknowledge that it is not obvious to check. An intuition is that as Θ is bounded and the objective is strongly concave in $\boldsymbol{w}$, the dual update (w.r.t. $\boldsymbol{w}$) at the kth iteration pulls the dual variable towards $\boldsymbol{w}^\star(\boldsymbol{\theta}^{(k)})$, the unique maximizer given $\boldsymbol{\theta}^{(k)}$, this suggests $\boldsymbol{w}^{(k)}$ may stay in a bounded set. Details will be provided in the final version.

Complexity: We will include a comparison to single-timescale primal-dual SGD in terms of computation and memory. The per iteration computation complexity for both methods are $\mathcal{O}(d^2)$ (due to Hessian-gradient mult.), and the memory requirement is $\mathcal{O}(d)$ for SGD and $\mathcal{O}(md)$ for nPD-VR – we only need to store $\boldsymbol{\theta}_i^{(k)}, \boldsymbol{w}_i^{(k)}$ as in (20). Convergence speed for nPD-VR is $\mathcal{O}(1/K)$ while the SGD is only anticipated to converge at $\mathcal{O}(1/\sqrt{K})$ (no known result in the literature in this setting). The $\mathcal{O}(d^2)$ complexity may appear impractical, yet we can apply a diagonal approximation.

Reviewer 2: We thank the reviewer for providing constructive and supportive comments.

Contributions w.r.t. Related Work: The suggested references are useful and will be included. We remark that [1]-[4] only considered linear function approximation, while this work focuses on the nonlinear setting. Also, the fast convergence of primal-dual SAGA for *one-sided non-convex* problem is new even to the optimization community.

Existence of β: We checked (a0),(a1),(a2),(a3),(a4) carefully and derived this:

$$\alpha \leq \min\left\{\frac{\mu^2}{8L_{\boldsymbol{w}}^2 m}, \frac{1/m}{(16L_{\boldsymbol{w}}^2+2)}, \frac{1/m}{12\overline{L}^2 + 96L_{\boldsymbol{w}}^2/\mu^2}\right\},\ \beta \leq \min\left\{\frac{\mu^2}{48L_{\boldsymbol{w}}^2}\alpha, \frac{1}{8}\left(8m\overline{L}^2\frac{L_{\boldsymbol{w}}^2}{\mu^2} + \frac{L_\theta}{2} + \frac{\mu^2}{8m}\right)^{-1}\right\}.$$

$\overline{L}^2 = 2L_{\boldsymbol{w}}^2 + L_\theta^2$ and loose constraints are skipped. To get α, β, we first fix α with the first inequality, then obtain β.

Reviewer 3: We emphasize that our contributions are substantial, from both TD learning and optimization perspectives:

First, we disagree that our paper does not provide a *real analysis for TD algorithm*. Our (10) is actually a TD learning problem, and we focus on tackling its **batch/offline version (16)**. While in this setting the randomness in state/action becomes decoupled from the learning process, there are **vast applications** related scenarios with offline available data and experience replay – as studied in "Batch Reinforcement Learning" by Lange et al., "Least squares policy evaluation algorithms with linear function approximation" by Nedić et al., etc. (references will be added). All these works only focused on *linear function approximation*, while we study the **nonlinear** case. Nonlinear TD learning is a challenging problem due to **non-convexity** in the underlying optimization. It has only been studied by a few authors, e.g., [4,7], and there are **no prior finite-time analysis** papers. While we consider the batch/offline setting (16), we developed an algorithm with finite-time analysis and is **efficient**. This result is one of the first in the literature and advances the analytical understanding for TD learning.

We also disagree that our analysis follows from *standard techniques* in optimization. While the use of variance reduction (VR) on finite-sum problems is common, applying and analyzing VR on **one-sided non-convex primal dual problem** (that arises from nonlinear TD) is **new** and **non-trivial**, even to the optimization community. The only comparable results are the recent works [22,24,27] with focuses on two-timescale and batch update methods. Our novelty is also evidenced in the analysis in Appendix D where we developed novel analysis techniques to handle the unique challenges.

An **online nonlinear TD learning** algorithm, that accounts for Markovian randomness in state/action, is an interesting extension. It relates to a stochastic approximation scheme [arXiv:1806.02450] for bilevel programs [arXiv:1802.02246], and a finite time analysis is possible. This, however, belongs to a different setting than our focus.