We thank the reviewers for their feedback. Below, we address the main questions/concerns raised.

Reviewer 1. *Confusion on parallelization:* Theorems $5.1, 5.2$, show that OFTPL converges at $O(T^{-1})$ rate after running for T iterations and making $2T$ parallel calls to the optimization oracle in each iteration. This is equivalent to the claim made in lines 16-18, with T replaced by $T^{1/2}$.

Comparison with [18]: we note that the result of [18] was improved by [11], which we compare against (see lines 316-317). [18] show that FTPL converges to a NE at $O(T^{-1/3})$ rate using T calls to the optimization oracle. [11] improve this result and show that FTPL converges at $O(T^{-1/2})$ rate. The OFTPL algorithm achieves these rates, but unlike FTPL which runs for T iterations, OFTPL only requires $T^{1/2}$ iterations and makes $O(T^{1/2})$ parallel calls to the optimization oracle in each iteration.

Line 451: If the predictions of OFTPL are stable, we can actually show that regularizer R is strongly convex (we rely on this property of R in line 454). However, it need not be differentiable. As a result, the traditional Bregmann divergence is not well defined. So we need to do a more careful analysis. As rightly pointed out, our analysis relies on Bregmann divergence like quantities defined in line 451.

Reviewer 2. *OFTPL vs OFTRL.* Although the idea of optimism has been applied to FTRL quite some time ago, its extension to FTPL has not been done so far. The main challenge has been that the proof methodology of FTPL is quite different from that of FTRL, and it was unknown how to incorporate optimism in this proof framework. Our work addresses this challenge and incorporates optimism in the proof methodology of FTPL, by using the duality between perturbation and regularization. Computationally, OFTPL has significant benefits over OFTRL. *In the convex case*, unlike OFTRL, OFTPL only requires access to a linear optimization oracle, which is much easier to implement for a number of problems of interest [8,9,10] (the counterpart of this in offline optimization is projected gradient descent vs. Frank-Wolfe algorithms, which has been widely studied [Jaggi 2013]). Moreover, OFTPL achieves the same $O(T^{-1})$ rates as OFTRL by making T parallel calls to the linear optimization oracle in each iteration. The computational advantages of OFTPL over OFTRL become even more stark in the *nonconvex case*. The well known OFTRL algorithm for nonconvex case is entropic mirror descent, which works in the space of probability distributions over $\mathcal{X}$ (see line 102). In this case, OFTRL recommends playing an entire distribution over $\mathcal{X}$ in each iteration, which is not feasible in practice.

Bounds. By letting the bounds depend on $\mathbf{x}_t, P_t$, we get very tight regret bounds. When instantiated for specific problems such as minimax games, such bounds help us derive fast convergence rates. To be precise, the terms depending on $\mathbf{x}_t, P_t$ are very crucial for deriving the results in Theorems 5.1, 5.2. Without these exact terms, we believe one can't show fast convergence rates. Besides, such terms also appear in the regret bounds of OFTRL [14].

Significance of results. We believe the quadratic improvement is very significant, especially for large scale problems such as training of GANs, adversarial training on ImageNet. Our results show that the performance of FTPL run with $T = 1000$ can be matched by OFTPL run with $T = 50$ iterations with multiple parallel calls to the optimization oracle in each iteration. Given that each iteration of OFTPL/OFTRL can take hours, this is a significant improvement.

Reviewer 3. *Nonconvex Bounds.* We believe the nonconvex bounds are very useful, especially given the prevalence of parallel compute resources. For example, many popular nonconvex games such as adversarial training and GAN training arise in the context of deep learning, where one has access to GPUs which support parallel computation.

Last iterate convergence. We agree that having last-iterate convergence is useful in the convex case. We will pursue this direction in the future.

Reviewer 4. *FTPL vs. Frank-Wolfe.* The updates of FTPL and vanilla online Frank-Wolfe (FW) algorithms look very similar to each other, except for the perturbation term in the former. This suggests there might be a deeper connection between the two. However, it is not immediately clear if FW is as parallelizable as OFTPL. We believe this is an interesting direction to pursue in the future.

Perturbation-regularization analog. The duality between perturbation and regularization also holds for optimistic versions of FTPL and FTRL. This follows from Proposition 3.1 by replacing $\nabla_{1:t-1}$ with $\nabla_{1:t-1} + g_t$. We infact rely on this duality between OFTPL and OFTRL in our proof of Theorem 4.1 (see line 449).

Corollary 4.2. One way to understand the usefulness of the last term in the bound is to study the rates of convergence one would be able to derive for smooth minimax games without this term. For the setting considered in Theorem 5.1, one can only show $O(T^{-3/4})$ rate of convergence without the last term in the bound of Corollary 4.2.

[1] Jaggi, Martin. "Revisiting Frank-Wolfe: Projection-free sparse convex optimization." Proceedings of the 30th international conference on machine learning. No. CONF. 2013.