We thank the reviewers for a careful reading and the feedback of our submission. We were happy to hear the overall positive nature of the reviews and constructive critiques provided. In addition to fixing the typos suggested, we will try to address the concerns that were raised in the review.

Reviewer 1:

- We introduce PIA (Proximal Iterate Average) primarily because it is a natural way of parallelizing the APROX family. The performance guarantees we prove do not apply to this method because PIA does not operate on a model of the minibatch-average function $\bar{F}$, instead averaging conditionally independent updates. Our main results are bounds that depend on the variance of the modeled function at every step. Since PAM and PMA (Proximal Average Model and Proximal Model of the Average) both model the minibatch-average $\bar{F}$, as the minibatch size grows, the modeled function's variance decreases, in turn leading to better upper bounds. We provide a visual/intuitive explanation of the differences in Figure 1, which shows how iterate averaging (PIA) picks a good direction but cannot progress as far as PAM or PMA does. We did not pursue theoretical results for PIA because of its lackluster empirical performance.
- In Line 99, we will change the gradient to subgradient.
- (Now) standard techniques indeed give martingale-based high probability results; we omit such results for lack of space, focusing instead on empirical results to highlight the methods' applicability. If the reviewers feel that such results are more important than empirics, we are happy to change emphasis.
- The definitions of interpolation we use are in [3].
- We call the gains linear as we obtain speedup proportional to minibatch size m (distinct from linear convergence): the iterations $K(\epsilon, m)$ to achieve ϵ accuracy with minibatches of size m approximately satisfies $K(\epsilon, 1)/K(\epsilon, m) = m$.
- We cap the iterations in the simulations at 1000; we will note this in the final version of the paper.

Reviewer 2: The reviewer points out that the paper could better differentiate itself from previous work (particularly Asi & Duchi 2019 [AD]); we will clarify this in the final version. Briefly, [AD] do not consider parallelization or anything beyond single observations per iteration. Their work does not show the gains of proximal methods with minibatches and the advantages over stochastic gradient methods in this setting (e.g. speedup for non-smooth functions).

- Demonstrating provable speedup from minibatching almost always require assumptions like smoothness. The restriction on minibatch sizes to achieve linear speedup is natural: improvements from minibatching saturate once the gradients become sufficiently low noise, as the problem complexity approaches deterministic optimization lower bounds. This saturation behaviour of iteration complexity with minibatch size has been observed (e.g., [3]).
- Regarding speedups for non-smooth functions with restricted strong convexity, we note that the speedup in this setting improves known results for stochastic gradient methods, which obtain no speedups for non-smooth functions without substantial care [2]. Without smoothness, additional assumptions are *necessary*; information theoretic lower bounds [1, Thm. V.1] show worst-case complexity $1/\sqrt{k}$ even for deterministic (0 noise) problems.

Reviewer 3: We agree with the reviewer about the importance of parallel implementation. In our experiments, we report complexity in terms of function value/gradient/proximal point calculations and iterations, as (except for extremely large minibatch sizes m) these are the most expensive part of the calculation—more expensive than broadcast and gather steps. This also reduces the noise in measuring real-time results.

Reviewer 4:

- Please see the first bullet below "Reviewer 1" to address comments on why the proposed variants enjoy speedups.
- In line 143, when $m = \frac{9k\sigma_0^2}{L^2R^2}$, the second term equals the first term; thus, when $m \ll \frac{k\sigma_0^2}{L^2R^2}$, the second term is larger.
- The experiments in Figures 3–5 illustrate each method's sensitivity to stepsizes, which introduce the same scaling as modifying Lipschitz constants. To further illustrate the sensitivity of our methods, we will include experiments with varying condition number in the final version of the paper.

References

[1] G. Braun, C. Guzmán, and S. Pokutta. Lower bounds on the oracle complexity of nonsmooth convex optimization via information theory. *IEEE Transactions on Information Theory*, 63(7), 2017.

[2] J. C. Duchi, P. L. Bartlett, and M. J. Wainwright. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22(2):674–701, 2012.

[3] S. Ma, R. Bassily, and M. Belkin. The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.