Dear reviewers, thank you for taking the time to review our paper and your careful remarks. We have addressed all of your questions, remarks and suggested improvements below.

Rev. #2: Correction in Eq. (5). We thank the reviewer for pointing out this inconsistency, it is in fact due to a typo in Eq. (5). When $n \leq L/\mu$ the optimal mini-batch size b^* should be $b^* = \max\{\hat{b}, 1\}$ instead of n, where $\hat{b} := \sqrt{\frac{n}{2}\frac{L_{\max}-L}{nL-L_{\max}}}$. This fixes the inconsistency that the reviewer noted, indeed when $L = L_{\max}$ we have that $b^* = 1$.

Smoothness and rate improvements. In the case where $L = L_{\max}$, such as the extreme case of ridge regression with a diagonal matrix, there is indeed no benefit in mini-batching. But in general $L \leq L_{\max} \leq nL$ (see Lemma A.6), and in particular when n is large $L_{\max}$ can be orders of magnitude greater than L. Indeed, all the benefits of mini-batching come from the gap between L and $L_{\max}$, which is why any effective mini-batch analysis needs to leverage this gap and our analysis is the first to do so in this setting.

Meaning of *"practice of SVRG"* in the title. We propose and analyse algorithms which are as close as possible to what is already used in practice. In particular, we show that the commonly used parameter setting $m = n$ together with not resetting the inner iterates give the best rates one could expect in our setting. We do not claim to establish a new *defacto* practice, but rather to better understand the SVRG that is already implemented and used for generalized linear models[1]

Innovative nature of our results and analysis. Our analysis cracks the open question of showing the benefits for mini-batching for SVRG. We do this by using a new Lyapunov-style analysis. It is this novelty that allows us to freely set the inner loop length m, but also understand mini-batching. Moreover, neither the theory for the non-convex setting [10] nor the accelerated SVRG [1] are able to show the benefits of mini-batching. In particular, it is shown in [1] that the iteration complexity benefits from a larger mini-batch size, but not the total complexity. And [10] determines the optimal importance sampling strategy depending on a batch size fixed *a priori*.

Behaviour of SVRG in Figure 2. The purpose of this experiment was to see which of the settings guaranteed to work in theory for SVRG resulted in the best practical performance. That is, we compared the settings of the previous theory (see Theorem 6.5 of Bubeck, Convex Optimization Algorithms and Complexity, 2015) that suggests to set $m = 20L_{\max}/\mu$ (standard SVRG is the blue curve) and our theory which suggests that any choice for m between $3L_{\max}/\mu$ and n will suffice. Because m is so large for the standard SVRG, we can see the method stalling. Thus the experiment carefully highlights the issues with the current standing theory of SVRG. Though one could always resort to using a grid search to determine parameters such as m, the stepsize or the mini-batch size, the purpose of our work is to exactly avoid the elevated costs related to performing such grid searches.

Usefulness in large-scale setting. The setting of our paper is for strongly convex learning problems. Both CIFAR and ImageNet are not appropriate benchmarks in our setting since convex models are not able to properly fit or make predictions over these data sets, independently of the optimization method used. The main applications of SVRG are medium-scale convex learning problems, such as those that we solved: regularized logistic regression on *ijcnn1* ($n = 141,691$, $d = 22$) and *real-sim* ($n = 72,309$, $d = 20,958$), and ridge regression on *YearPredictionMSD* ($n = 515,345$, $d = 90$) and *slice* ($n = 53,500$, $d = 384$). We will make sure to clarify this in our final submission.

Rev. #4: Benefit of providing optimal b and m. Since our submission, we have performed extensive numerical experiments comparing our optimal mini-batch size b^* against the empirical best mini-batch size over a large grid. We find that the resulting empirical complexity of using our b^* is remarkably close to the best one over the grid search. We will include these new experiments in our final submission. In particular, we are now able to predict the optimal mini-batch size even more accurately due to some minor improvements we have made in our theory (we have shaved off some constants). We will also include this update on the improved theory in our final submission.

Free-SVRG needs μ, not L-SVRG-D. In Algorithm 1, the reference point is reset to an exponentially weighted average of the inner iterates. The weights p_t depend on the strong convexity constant (see Eq. (11) and Algo. 1). This motivated us to present and analyse a single loop version of SVRG, which does not require μ at any point. We will make sure to clarify this point in our discussion.

Comment on Fig. 3. The experiments were run on a multi-core server. The difference between some of the epoch and time plots is due to the mini-batch methods benefiting from shared memory parallelism. Unfortunately, this is always an issue with plotting time, as it depends on the architecture used. We will include information about the architecture in the supplementary material and explain this apparent discrepancy.

Our definition of complexity. We will clarify that the symbol $C_m(b)$ is in fact an upper bound on the total complexity.

Rev. #5: We thank you for pointing out these references. Streaming SVRG (Frostig et al.) presents some interesting ideas: they provide statistical guarantees for a version of SVRG in the infinite samples framework, and a schedule of increasing mini-batch sizes. Yet their assumptions and objectives are different from ours. As for Jain et al. (2018), their analysis uses a similar notion of expected smoothness, but they only analyse quadratics and parallelised SGD, which is a completely different setting.

[1] See for example the lightning package from scikit-learn [20]: http://contrib.scikit-learn.org/lightning/