We thank the reviewers for their insightful comments and encouraging feedback. We hope that the concerns raised are addressed adequately below and that our work will be appropriately re-evaluated.

Speedups (R1) Reviewer 1 raises two concerns about speedups which we believe to be based on a misunderstanding. Firstly, our large reductions in communication (e.g. $10\times$) lead to smaller reductions in wall-clock time (e.g. $2\times$). We think this is expected, as all mentioned wall-clock times *include forward and backward passes* in addition to gradient compression and communication. A $2\times$ reduction in this metric seems significant. We will clarify this in the paper.

Secondly, the reviewer suspects that Tables 6 and 7 show timings for the slower GLOO backend. Let us clarify that all such timings are measured in default conditions: NCCL, all-reduce, 16 GPUs, and end-to-end as described before. The scaling plots in Figure 3 show speedups of 9.3 (PowerSGD) vs 7.1 (SGD) on Cifar over single worker SGD. This is consistent with the 23% savings (9.3 vs. 7.1) reported in Table 6. We include results for an LSTM (Table 7) for completeness. The LSTM's speedups are better due to their higher communication-to-computation ratio.

Failure cases for PowerSGD (R1) We agree with Reviewer 1 that an outline of when PowerSGD works and when it breaks would be helpful. To date, we have not observed any failure cases of the 1-step power iteration in the algorithm. To achieve good accuracy in the same number of steps as SGD, a sufficiently high rank (2 or 4 in practice) is required.

Larger models and clusters (R1, R3, R5) We are currently running additional experiments on a larger cluster (64 GPUs) with larger models (ResNet-50). This should further test the effect of network latency (R5). Extrapolating the scaling plots in Figure 3, we expect PowerSGD to perform favorably in those conditions.

Convergence of Algorithms 1 and 2 (R1, R3) While we do not currently include an end-to-end convergence proof for PowerSGD, each of its core components are well studied. Algorithm 2 (EF-SGD with Momentum) adds momentum to the well-studied EF-SGD algorithm (as in Karimireddy et al. 2019). EF-SGD is guaranteed to converge if the compressor $\mathcal{C}$ satisfies $\|X - \mathcal{C}(X)\|_2^2 \leq (1-\delta)\,\|X\|_2^2$. This condition is satisfied by PowerSGD with SVD for best rank-k approximation (see Appendices A.1 and A.2). The cheaper 1-step power iteration with warm start is akin to the famous Oja's algorithm (Oja, 1982) and is empirically shown to yield the same performance as a full SVD.

Linearity of PowerSGD (R3) We use the term linearity to mean that PowerSGD on a single worker with gradient matrix $\bar{M} := (M_1 + M_2)/2$ is equivalent to PowerSGD with two workers with their own gradients M_1 and M_2 (for any number of workers and any split of M.) To see that this holds, consider that P in line 4 of Algorithm 1 in the two-worker example amounts to $P = \frac{1}{2}(M_1 + M_2)Q = \bar{M}Q$. The matrices M_1 and M_2 are never multiplied with each other. The same is true for Q in line 7. This makes PowerSGD just a function of the average gradient $\bar{M}$.

PowerSGD without feedback (R3) Because PowerSGD's very-low-rank gradient approximations are coarse, it required error feedback to converge in our experiments. We will include the requested comparison in the Appendix.

GradiVeQ (Yu et al. 2018) (R3) We thank Reviewer 3 for pointing us to this interesting work. We will include this method in our discussion.

High Cifar-10 accuracy (R3) We use a ResNet-18 based on `torchvision`. Compared to the ResNet-20 model used for Cifar-10 in the original paper, the layers have more feature maps (are wider), explaining the superior performance. He et al. use this wider architecture for ImageNet.

Global batch size (R5) Reviewer 5 mentions that some related papers scale the number of workers while keeping the global batch size fixed. Most of the work we are aware of instead keep the local batch size fixed (e.g. Goyal et al. (2017)) since it better utilizes the computational power of the workers. Moreover, if we kept a fixed global batch size the computation performed per bit communicated would decrease with the number of workers—only further favoring compressed algorithms such as ours.

End-to-end speedup results (R5) Time-to-accuracy results, as requested by Reviewer 5, can currently be found in Appendix C of the submission. We will consider including these plots in the main paper.

Goyal, P., et al. "Accurate, large minibatch SGD: Training ImageNet in 1 hour." arXiv 2017.
He, K., et al. "Deep residual learning for image recognition." CVPR 2016.
Karimireddy, S.P. et al., "Error feedback fixes SignSGD and other gradient compression schemes." ICML 2019.
Oja, E. "Simplified neuron model as a principal component analyzer." Journal of Mathematical Biology, 1982.