Rebuttal for "Communication-Efficient Distributed Learning via Lazily Aggregated Quantized Gradients"

Reviewer 1. My main concern on the theoretical analysis. Provide a more fine-grained analysis. The linear rate constant can be tightened/simplified, and will be presented it in the final version. With this tighter analysis, the bound explicitly establishes the dependence of convergence rate on the condition number κ, namely $\sigma_2 = (1 - \frac{a(\xi)+b(\tau)}{\kappa} + \tau^2 c)^{1/\bar{t}}$, where constants $a(\xi)$ and $b(\tau)$ increase as ξ and τ decrease, and all $a(\xi), b(\tau), c$ do not depend on κ, L, μ. Clearly, as quantized values become precise enough ($\tau^2 \to 0$), LAQ approaches the convergence rate of LAG [6], namely, $\sigma_2 = (1 - \frac{a(\xi)}{\kappa})^{1/\bar{t}}$. If there is no quantization or skipping of communication rounds, $\tau^2 \to 0$, $\xi = 0, \bar{t} = 1$, LAQ converges with $\sigma_2 = (1 - \frac{a(0)}{\kappa})$, same as GD. We hope the reviewer will appreciate the merits of this analysis.

Reviewer 2. 1. The relation of Lyapunov function to the simple risk. Analysis does not apply to neural networks. Linear convergence of the Lyapunov function also implies that $f(\boldsymbol{\theta}^k) - f(\boldsymbol{\theta}^*)$, $\|\nabla f(\boldsymbol{\theta}^k)\|_2^2$, and $\|\boldsymbol{\theta}^k - \boldsymbol{\theta}^*\|_2^2$, all converge with a linear rate. From the definition of a Lyapunov function, it is clear that $f(\boldsymbol{\theta}^k) - f(\boldsymbol{\theta}^*) \leq \mathbb{V}(\boldsymbol{\theta}^k) = f(\boldsymbol{\theta}^k) - f(\boldsymbol{\theta}^*) + \sum_{d=1}^{D} \sum_{j=d}^{D} \frac{\xi_j}{\alpha} \|\boldsymbol{\theta}^{k+1-d} - \boldsymbol{\theta}^{k-d}\|_2^2 \leq \sigma_2^k \mathbb{V}^0$, meaning the risk error $f(\boldsymbol{\theta}^k) - f(\boldsymbol{\theta}^*)$ converges linearly. The L-smoothness results in $\|\nabla f(\boldsymbol{\theta}^k)\|_2^2 \leq 2L[f(\boldsymbol{\theta}^k - f(\boldsymbol{\theta}^*)] \leq 2L\sigma_2^k \mathbb{V}^0$; hence, the gradient norm $\|\nabla f(\boldsymbol{\theta}^k)\|_2^2$ also converges linearly. Similarly, the μ-strong convexity implies $\|\boldsymbol{\theta}^k - \boldsymbol{\theta}^*\|_2^2 \leq \frac{2}{\mu}[f(\boldsymbol{\theta}^k - f(\boldsymbol{\theta}^*)] \leq \frac{2}{\mu}\sigma_2^k \mathbb{V}^0$ — $\|\boldsymbol{\theta}^k - \boldsymbol{\theta}^*\|_2^2$ also converges linearly. Convergence analysis of nonconvex nonsmooth objectives is important, and is included in our future research agenda.
2. My only concern is that the experiments have been only conducted on MNIST. If accepted, the final version will report experiments on SUSY, IJCNN1 and COVTYPE, for which the results are also promising.
3. Connection with [2], [3], [4]. Compared with the innovation-agnostic random selection [2], LAQ explicitly leverages the gradient innovation in both worker selection and gradient quantization, which will result in more effective communication reduction. LAQ also differs from the dynamic averaging [3] in their design principles, as [3] skips communication when the local model does not differ too much from the global model, while LAQ skips communication when the fresh gradient does not differ too much from the stale one. Lazy aggregation and dynamic averaging can be jointly leveraged to further reduce communication. Developing lock-free LAQ or asynchronous quantized method based on [4] is interesting. Due to asynchrony, the resultant algorithm may require re-deriving the communication conditions (based on a counterpart of Lemma 2), which likely will complicate analysis of the new Lyapunov function. Since it needs careful investigation, we will tackle it in our future work. A discussion with these references will be added.
4. The proof of them 1, ..., Lyapunov-functions ... not intuitive. The design of Lyapunov function $\mathbb{V}(\boldsymbol{\theta})$ is coupled with the communication rule (7a) that contains a parameter difference term. Intuitively, if no communication is skipped at the current iteration, LAQ behaves as GD that decreases the objective residual in $\mathbb{V}(\boldsymbol{\theta})$; if certain uploads are skipped, LAQ's rule (7a) guarantees that the error of using stale gradients is comparable to the parameter difference in $\mathbb{V}(\boldsymbol{\theta})$ to ensure its descent. Thus, Lyapunov function always decreases. We will add more intuition in the proof. We incorrectly marked the reproducibility, but will make the code publicly available. Thank you for the favorable recommendation.

Reviewer 3. 1. The scheme requires additional memory. The extra memory is low. The server stores the last aggregated gradient (dimension p), and each worker stores the last gradient (dimension p) and D model change norms (D *scalars*).
2. Many critical tuning parameters, such as the step-size α and the Lyapunov parameters ξ are quite complex to tune. With the smoothness $L = 19$ in the simulation setting, our parameters used in simulations slightly violate (17). During the rebuttal period, we conducted a simple experiment with $\xi_d = \xi = 1/160, D = 10$ and $\alpha = 0.01 (\rho = 0.01, \rho_2 = 0.5)$ satisfying (17). It turns out that the result is comparable with that presented in our submission: test accuracy 0.9082, iteration # 2530, communication # 530, and bit # 1.66×10^7. To assess the sensitivity of parameters, we tested under variable α (with $D = 10$) and variable D (with $\alpha = 0.02$) values, as summarized below. Here, $\xi = 0.8/D$ and $\epsilon = 10^{-6}$.

	$\alpha = 0.01$	$\alpha = 0.015$	$\alpha = 0.02$	$\alpha = 0.04$	$D = 2$	$D = 5$	$D = 10$	$D = 15$
Iter #	5219	3459	2663	1410	2448	2503	2663	2749
Comm #	825	626	618	1908	534	512	618	678
Bit # $(\times 10^7)$	2.59	1.96	1.94	5.99	2.67	1.61	1.94	2.13
Accuracy	0.9082	0.9082	0.9082	0.9082	0.9082	0.9082	0.9082	0.9082

3. The increase in iteration counts vs the decrease in communication load. With the updated finer-grained analysis, we can obtain the linear rate constant $\sigma_2 = (1 - \frac{a(\xi)+b(\tau)}{\kappa} + \tau^2 c)^{1/\bar{t}}$, where constants $a(\xi)$ and $b(\tau)$ increase as ξ and τ decrease, and all $a(\xi), b(\tau), c$ do not depend on κ, L, μ. Thus, $\bar{t} \frac{\kappa}{a(\xi)+b(\tau)-\kappa\tau^2 c} \log(1/\epsilon)$ iterations, or $bp\bar{t} \frac{\kappa}{a(\xi)+b(\tau)-\kappa\tau^2 c} \log(1/\epsilon)$ bits, are needed to reach ϵ-accuracy. The compressed GD by Khirirat ("Distributed learning with compressed gradients") in a centralized setup requires the same order of iterations $c_1 \frac{(\mu+\bar{L})^2}{4\mu\bar{L}} \log(1/\epsilon)$, or $(\log_2 p + bp)c_2 c_1 \frac{(\mu+\bar{L})^2}{4\mu\bar{L}} \log(1/\epsilon)$ bits; in the distributed setup, this approach yields a near-optimal solution. Uncompressed methods, e.g., GD, entail $\frac{\kappa+1}{2} \log(1/\epsilon)$ iterations (fewer than LAQ), but more bits (usually encode a float using 32 bits while b bits in LAQ).
4. In Them 1, V decays, what about f or gradient norm? Non-convex case? See our reply to Comment 1 of Reviewer 2.
5. Error compensated schemes. The error compensation schemes skip communicating certain *entries* of the gradient, but communicate with *all workers*. LAQ skips communicating with certain *workers*, but communicates *all (quantized) entries*. The two are not mutually exclusive, and can be used jointly. We will try to empirically compare with [Alistarh etal'18] in the final version. Thanks for recognizing the novelty of our work.