We thank the reviewers for the detailed and insightful reviews. The reviewers noted that our work 1) makes "significant contribution" [R1] and "significantly improves the communication efficiency" [R3, R7], 2) makes "clear comparison with the state-of-the-art results" [R1], 3) provides "accurate claims" [R1] and "sound theory" [R5], and 4) "the technique is of independent interest" [R7]. We will answer questions below and incorporate feedback into our final revision.

[R1]: "a much more interesting case is the case of heterogeneous data"

• Our proof framework can generalize to heterogeneous data if we allow an overhead term that depends on the inter-client heterogeneity. For example, if we assume $\sup_w \|\nabla F_m(w) - \nabla F(w)\| \leq \zeta$, we can modify the stability analysis by introducing an additive overhead involving ζ as in the heterogeneous analysis of FEDAVG [Woodworth et al., 2020]. That said, it may be more interesting to design heterogeneity-aware algorithms that avoid this overhead, which was so far not well-understood even without acceleration and is beyond the scope of this paper.

[R1]: "the extra squared root of K (# of local steps) is needed to obtain acceleration in the usual sense"

• The $\sqrt{K}$ originates from the acceleration-stability tradeoff (Line 226). While it remains an open question whether the overhead is necessary for lower bounds, we provided two evidences 1) theoretically standard AGD is not initial-value stable (§F), and 2) empirically direct federation of AGD may indeed perform worse than our principled FEDAC (§A.4).

[R1]: "in lines 97-98 authors do not mention the following works on quantization: ..."

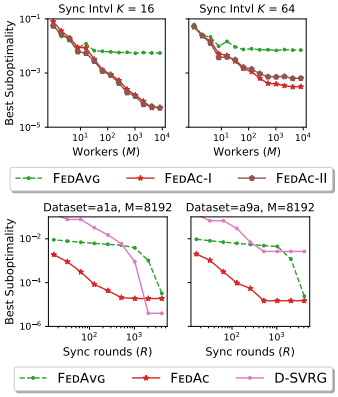
• Thanks for the suggestion of the references. We will add these references regarding quantization in the next revision.

[R1]: "(at Line 39) notice that (Woodworth et al., 2020) analyzed Local-AC-SA for quadratic problems as well."

• We have mentioned Woodworth et al.'s work on quadratic at Line 60. We will reiterate at Line 39 in the next revision.

[R1]: "Why do you test only FEDAC-I in experiment? Have you tried FEDAC-II?"

• We have indeed tested FEDAC-II and stated in §A that "FEDAC-II is qualitatively similar to FEDAC-I empirically so we show FEDAC-I only." We will include more experiments on FEDAC-II in the next revision. Particularly under the same settings of §5, FEDAC-I is slightly better as the condition number is large (see figure on the right).



[R3]: "Can author compare with distributed finite-sum solvers in experiments?"

• We compared with `DSVRG` (Lee et al., 2017), see figure on the right. Under the same settings of §5 (dataset `a9a`, size 33k) FEDAC outperforms `DSVRG`. On a smaller (1.6k) dataset `a1a`, `DSVRG` is better only if the communication is very frequent. In general, one can obtain moderate accuracy with FEDAC in a short parallel time under limited communication, whereas finite-sum solvers may be preferred if high accuracy is required and the dataset is relatively small. We conjecture that FEDAC can be incorporated with variance reduction techniques to attain better performance in finite-sum (ERM) settings.

[R3]: "For $Q = 0$, the term about variance is independent from R. Is it correct?"

• Exactly. When $Q = 0$, the last term will vanish. This gives a smooth interpolation of existing quadratic analysis.

[R3]: The bounds have undesired terms such as large constants and polylog factor.

• Thanks for your suggestions. We will try to reduce these terms in the next revision. For strongly-convex analysis, the polylog factors are the artifacts of constant step-size η and do not emerge until the very end of the analysis. For example, Lemma 6 (Convergence of FEDAC-I for general η) does not involve any polylog factors. As stated in footnote 12 on Page 17, there are standard techniques (*e.g.*, [Lacoste-Julien et al., 2012]) to reduce such polylog factors by decaying η and averaging. However, adopting such techniques will complicate the overall analysis due to the time-variant η.

[R5]: "the insights of bounds are lacked" "compare with more works such as accelerated distributed SGD"

• We will clarify the insights more in next revision. Here is the summary: for general-convex case, under A1, our bound for FEDAC-I is $\tilde{\mathcal{O}}(\frac{LD_0^2}{TR} + \frac{\sigma D_0}{\sqrt{MT}} + \frac{L^{1/3}\sigma^{2/3}D_0^{4/3}}{T^{1/3}R^{2/3}})$. The first term corresponds to the deterministic convergence, which is better than the one for FEDAVG, that is, LD_0^2/T. The second term corresponds to the stochasticity of the problem which is not improvable. The third term corresponds to the overhead of infrequent communication, which is also better than FEDAVG ($\frac{L^{1/3}\sigma^{2/3}D_0^{4/3}}{T^{1/3}R^{1/3}}$) due to acceleration. The intuition is that FEDAC can achieve the same progress with smaller step-size η, which lowers this overhead incurred by the discrepancy of clients (see Line 51-55 for related discussions).

• Ideally we hope the second term to dominate the bound so one can gain by scaling up M. Since the third term of FEDAC has better dependency on R, one only need fewer rounds of communication to keep the third term dominated.

• The bound for accelerated-distributed-SGD is $\mathcal{O}(\frac{LD_0^2}{R^2} + \frac{\sigma D_0}{\sqrt{MT}})$. In comparison to FEDAC, it has worse deterministic convergence rate (*i.e.* first term, since $T \geq R$ trivially holds) but does not incur the third-term overhead.

[R7]: "The authors conjecture FEDAC can be generalized to non-convex problems. Does that mean we can implement accelerated sgd such as (Ghadimi et al., 2013) to Federated setting and gain performance improvement?"

• While FEDAC may lead to empirical improvements on non-convex problems, we do not expect it to gain theoretical improvement over FEDAVG. This is because directly applying convex acceleration may not improve non-convex rates even with a single machine (*e.g.*, [Ghadimi et al., 2013] does not have better theoretical bounds than SGD for non-convex problems). However, it is possible to combine our result with recent non-convex acceleration algorithms (*e.g.*, [Carmon et al., 2018]) that use convex acceleration in a more sophisticated way and are provably faster than GD.