We would like to thank all reviewers for the constructive feedback. We argue that most of the issues raised are either minor or no issues at all. Before proceeding, we would like to highlight the contribution of this work that we believe was overlooked. Specifically, our work is the first to show the optimality of local methods (=most popular FL optimizers) in the non-iid regime. So far, local methods have been known to be optimal only in certain scenarios with iid data (=not a reasonable assumption for FL); thus we provide an important justification of local methods in the practical regime. See point 7) for more details. We hope that the scores will be raised accordingly.

R1 Thanks a lot for the positive evaluation of our work!
1) Assumption 3.1. Assumption 3.1 was only used for the lower complexity bounds; it defines the algorithm class for which we provide the lower bounds. Note that virtually literally any reasonable FL optimizer satisfies the assumption: FedAvg/Local SGD, FedProx, FedSplit, SCAFFOLD, variants of (accelerated) SGD, or any algorithm that we propose or mention in our paper. Informally, we restrict ourselves to the algorithms whose iterates only lie in the span of the previously observed gradients. From the optimization perspective, it does not make sense to do not have it satisfied for any reasonable algorithm. To assume something along those lines is indeed typical in the lower-bound optimization literature. Note that a recent paper on the lower bounds for the classical FL objective assumes the same thing: `https://arxiv.org/pdf/2005.10675.pdf` (see Sec. 4.1).2) Gap. There is still a rather small gap in a single setting. However, in most scenarios we consider, there is no gap. 3) Rest. Thanks a lot for the useful MAML reference; we will mention it! Typos: thanks, fixed!

R2 Thank you very much for the positive evaluation of our work!
4) line 432. There are 2 typos in the appendix: a typo in the definition of $\mathbf{M}$; the term $1/2$ should be replaced with $1/n$. At the same time, there is a typo in the equation (9): there should be $\frac{n}{\lambda}$ instead of $\frac{1}{\lambda}$. Back to your question – function ψ does not depend on the parity of n; Hessian of ψ is now the first component of the equation defining $\mathbf{M}$, while functions f_i depend on the parity (note that we chose them as quadratics). Thanks a lot for noticing this! We will also add a subscript to highlight the dimensions of identity matrices. 5) AL2SGD+. The rationale behind AL2SGD+ is: take the variance reduced gradient estimator from L2SGD+, and feed it into L-Katyusha. AL2SGD+ looks different than L-Katyusha is since we wrote it from the FL perspective. 6) Different values of p, ρ. Setting those parameters to be identical yields a suboptimal complexity in certain scenarios, this is why we chose them differently. Rest: fixed, thanks!

R3 A few concerns were raised about the contribution of this work. Below we argue that those are not a concern.
7) Not focusing on the "scale with the number of clients". Our paper provides a different result that is much more valuable to the FL community! In particular, we are the first to show that local methods (variants of FedAvg/local SGD or FedProx) are optimal in the heterogeneous (non-iid) data regime both in terms of the communication and local computation complexity. We believe this is of great import in the FL area as i) local methods are known to be suboptimal in the non-iid regime for the classical non-personalized FL (see Woodworth et al. 2020 `https://arxiv.org/pdf/2006.04735.pdf` for example) and ii) local methods are the most popular FL optimizers in practice. By showing the optimality of local methods, we justify the common FL practices (i.e., use of local methods on non-iid data), which we believe is of greater value than the focus on the number of clients. 8) "Results that can be obtained by plugging in the analyses for the centralized algorithm and then showing that things go through in the FL setting". The lower bound we propose can not be obtained by any means from the known results. The key thing is that both upper and lower bounds are new (the upper bound was somewhat easier to get), and they are matching in most cases (see the reply to R1). While it is true that complexities of some methods are not hard to get (which we clearly state in the paper), it was non-trivial to find out the algorithms that match the proposed lower bounds, as well as to set them up properly so that the bound is achieved.

R4 The reviewer raised several concerns. Below, we argue that neither of those are major.
9) Thm 3.1. We missed a comma in the theorem statement: we meant $L \geq \mu$, $\lambda \geq \mu$ instead of the original $L \geq \mu\lambda \geq \mu$. Sorry! Those are the only constraints we have. As you noticed, there are some extra constants (100 and 101) over there for the bound to be meaningful. First of all, our results hold for a finite-dimensional problem; those constants can be improved by blowing up the dimension to infinity, which we chose to avoid. Secondly, even for the infinite-dimensional construction, many distributed lower bounds incur a similar (but smaller) constant (see, for example [20]). Next, requiring $\lambda \geq 100\mu$ is usually fine since the strong convexity constant μ is often tiny (from a learning theory perspective, one should the weight for the Tikhonov's regularization is inversely proportional to the dataset size). Lastly, we shall stress that this (rather mild) parameter restriction only holds for the lower complexity bounds. 10) Notations and tables. We will clarify those, thanks! 11) The proposed methods are trivially extended from existing methods. See the point 8). 12) Additional feedback, Q1. Indeed, it means that the optimal algorithm requires only a constant (=small) number of local oracle queries in between the communications. Note that the computation and communication complexities are not the same for the local gradient oracle once $\lambda < L$; in such a case, one needs to perform more local gradient steps in between of the communications. 13) Additional feedback, Q2. We never assume that $\lambda \geq 1$. We only assume that $\lambda \geq \mu$ for the lower bound; the algorithm complexities hold regardless of the choice of λ.