We thank the reviewers for their careful reading and feedback. We now address their comments in turn.

Reviewer 1: *"The main weakness is the lack of a theoretical result . . . when proximal steps are not solved exactly."*
To clarify, our submission does contain more general inexact convergence results. Specifically, Theorem 3 (Appendix A.2.1, pp. 13) contains results when the residuals for the proximal updates are allowed to vary, and are not uniform across iterations. Our revision includes a mention of these more general results following the statement of Theorem 1 in the main text. Additionally, from the proof of Theorem 3, it is easy to see that if the inexactness conditions (23) (respectively, (13)) hold only in expectation (for example, if one uses a stochastic gradient method to implement the proximal updates) then the conclusions (24) (respectively, (14)) also hold in expectation. We will mention this extension following the presentation of Theorem 3. These results should allow a practitioner to translate standard results regarding iteration complexity of (stochastic) gradient methods to this setting.

"*The results are quite easy to obtain . . .*"
While we view the simplicity of the argument as a positive feature, it is worthwhile emphasizing two new technical contributions in terms of the theory of operator splitting methods for distributed convex optimization. The first is the inexact updates (Theorems 1 and 3): we agree that the corresponding results are known for gradient methods, but in the context of Peaceman-Rachford and similar splitting procedures, Theorem 3 (Appendix A.2.1, pp. 13) is to our knowledge new. The second technical contribution to highlight is our analysis of non-strongly convex problems via reduction to weakly convex problems. Although well known for gradient methods, the $\tilde{O}(1/t^2)$ rate for non-strongly convex problems that we give in Theorem 2 is to our knowledge new for Peaceman-Rachford. Our revision will mention this following the statements of Theorems 2 and 3.

Reviewer 2: *"In the proof of Theorem 2, I cannot see where the condition on the step-size is used . . . I am surprised about the condition on the step-size . . . It seems that [vectors other than $x^{(1)}$] can be used. Some explanation needs to be included . . ."*
To clarify, the condition on the stepsize is used in ll. 415 "by squaring the guarantee of Theorem 1. . . " in the proof of Theorem 2. Specifically, after regularizing, the losses become λ-strongly convex and $(L_j + \lambda)$-smooth, and so the correct stepsize scales in the inverse of the geometric mean of these parameters (see Theorem 1). Although one can regularize around other points in the reduction we employ, inequality (34) will be less clean, and the iteration complexity (see ll. 417) will be less clean, in that it would depend on more than the initial conditions (distance of the initializer to opt). Although the inverse dependence on epsilon (note that our step size is $1/\sqrt{\varepsilon^2 + L^*\varepsilon}$) may at first seem counter-intuitive, note that when ε is large, this means the weight of the objective is small as compared to the effect of the regularization term. Moreover, this type of scaling has been noted previously for gradient-based reductions from weak to strong convexity [3].

"*The experiments are not strong . . . [use] bigger models and larger datasets.*"
The extended version of our paper (on arXiv) includes additional experiments on larger datasets, that show our procedure's behavior on easy and difficult problem instances, as measured by problem conditioning. We will include a citation in the experiments section that points to these simulations.

Reviewer 3: *". . . the convergence properties of FedSplit are strictly worse than those of AGD . . . mention transparently in the paper . . . in [the homogeneous setting], FedSplit should strictly outperform AGD.""*
We agree that the iteration complexity of FedSplit and AGD are comparable, assuming exact gradient computations/exact proximal evaluations and in this setting one should prefer AGD. On the other hand, it is known that AGD is sensitive to inexact updates (see [1]). In comparison, our inexact convergence guarantees show that FedSplit is robust to noisy updates (see Theorems 1 and 3). We also agree that if the proximal stepsize is sufficiently small (as compared to the desired accuracy level), then FedSplit dominates AGD in iteration complexity in the homogeneous setting. Our revision mentions these points in the paragraph preceding section 5.

"*In fact, it was already well known for the 'past procedures' to not have the correct fixed points.*"
We thank the reviewer for bringing SCAFFOLD [2] to our attention. We were not familiar with this work prior to this submission. While there is obvious conceptual overlap (and we will of course cite it in the revision), based on our reading, this paper does not explicitly demonstrate that FedProx or FedAvg have incorrect fixed points. To further clarify, we have rephrased this contribution as "We demonstrate that procedures such as FedProx and FedGD do not generally have correct fixed points, even for simple quadratic objectives." We believe this is a more specific and accurate characterization of our results.

Reviewer 4: *"The* `FedSplit` *method is more like a deterministic distributed optimization algorithm. The connections to the multi-device communications and failures mentioned in the paper are weak."*
Although operator splitting has been used successfully in standard distributed optimization settings, we believe that the mild dependence on condition number (which in the context of federated optimization corresponds to reduced rounds of communication) is important in multi-device communication. Additionally, our inexact guarantees as stated in Theorems 1 and 3 are especially important in multiparty communication occurring during federated optimization: device updates may be inexact due to computational constraints or errors may arise during communication. We have made these connections clearer following the statement of Theorem 1 in our revision. Finally, we agree that device failures are not addressed in the present work; we have removed the mention of this topic in the broader impact statement. (Our ongoing work has obtained results in this setting.)

"*The paper could add more details on the derivation of algorithm 1 and give some intuition. . . "*
Thank you for this feedback. We agree that additional explanation is warranted here. Our revision now includes an additional paragraph in section 3.1, explaining the relationship between problem (2), its optimality conditions, and a monotone inclusion problem that results with Algorithm 1. This additional background should help readers less familiar with operator splitting techniques.

[1] O. Devolder, F. Glineur, and Y. Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1–2):37–75, 2014.

[2] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh. SCAFFOLD: stochastic controlled averaging for on-device federated learning. Technical Report arxiv.org/abs/1910.06378, October 2019.

[3] Z. Allen Zhu and E. Hazan. Optimal black-box reductions between optimization objectives. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 1606–1614, 2016.