We sincerely appreciate all the reviewers for their valuable feedback and nicely summarizing the highlights of our paper. In this rebuttal, we will answer the questions and provide some additional experiments as suggested. We hope our response can well address the concerns. Thank you.

Dear Reviewer #1: **Literature:** Thanks for referring us to the series of interesting works. Yet, they are essentially different from ours. In [Belilovsky et. al, 2019], the model itself is trained in a layer-by-layer fashion by stacking shallow 1-layer networks, and [Ek et. al, 2020] and [Mo et. al, 2021] are applications of the above training strategy in different scenarios. However, in our approach, the model is trained in the standard federated learning framework (all layers at the same time), with adaptive AMSGrad as the local optimizer, and layer-wise adaptive learning rates. To the best of our knowledge, applying local layer-wise learning rate adjustment to FL has not been studied before. We will cite the mentioned papers and clarify the differences as suggested. Thank you.

Experiments: adf asdfasfasdfafasdfasfd as f

Insight: The insight of the layer-wise adaptive learning rate is that in NN, the magnitude of layer weights and gradients might differ a lot. If the gradient (of a layer) is too small, the weight will not move sufficiently far which slows down the convergence. Thus, besides the coordinate-wise adaptivity brought by local AMSGrad, we further incorporate the layer-wise adaptivity to adjust the learning rates layer-by-layer in every training iteration. The results show that our method can achieve faster convergence, also with possibly improved accuracy, in both IID and non-IID settings. The improved convergence also implies reduced communication cost, since the model needs fewer rounds to converge.

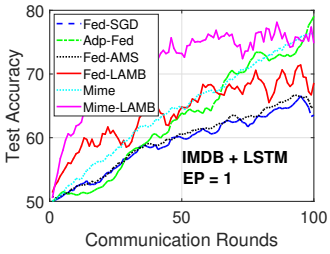

Figure 1. IMDB + LSTM, non-IID.

Extension: Yes, we believe our idea of adopting layer-wise learning rates in FL can be extended to many other methods, such as local SGD and SGD with momentum. Meanwhile, it may also be applied to the adaptive central server updates in FL (e.g., the Adp-Fed method, Algorithm 2 in Appendix A). These are all possible future directions.

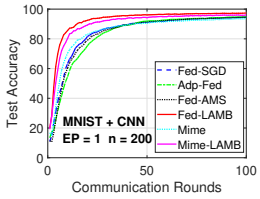

Figure 2. IMDB + LSTM, non-IID.

Dear Reviewer #4: **Novelty of algorithm and theory:** Thanks for mentioning [Charles et. al, NeurIPS 2021], which will be discussed in the paper. Note that, Charles et. al apply layer-wise adaptivity at the central server with local updates still using SGD. This is exactly the Adp-Fed method (Alg. 2 in Appendix A) + LAMB at central node. Our approach is essentially different, since it is built upon local adaptive methods, where local updates follow AMSGrad and layer-wise adaptivity is applied at local models. Also, Charles et. al only provides empirical results, while we also give the convergence analysis. Regarding the convergence rate, note that while in [You et. al, 2020] LAMB shows considerable empirical improvement over Adam, theoretically, the convergence rate of LAMB is the same as that of Adam under common assumptions. In our paper, we show that the convergence rate of Fed-LAMB can also match that of Fed-AMS, in the FL setting. To our knowledge, this is the first convergence analysis of layer-wise adaptive strategy in FL. Therefore, we believe that our algorithm design and theoretical analysis could provide novel contribution to the community.

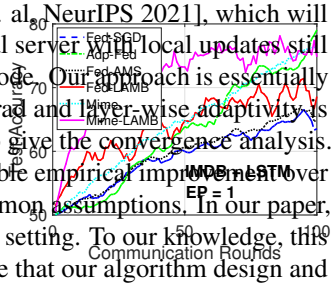

Figure 3. IMDB + LSTM, non-IID.

Dear Reviewer #4: **Second moment aggregation:** In the original paper of [Li et. al, 2020] and [Karimireddy et. al, 2020], the authors have shown that the performance of Fed-AMS and Mime would be much worse without sharing v. Since v controls the implicit learning rate for every dimension, synchronizing v makes all the clients "on the same pace" which helps convergence. Thus, at this stage, the synchronization of v seems a necessary part in the design of federated adaptive methods. One possible trick in practice is to reduce the frequency of communicating v. This can reduce the extra communication cost of v, without affecting the theoretical convergence rate.

Maximum operator: Yes, in practice, we found that ignoring the maximum operation at the central server gives similar results. Since it is known that the theoretical convergence of Adam (without the max) could be problematic, we use AMSGrad (with the max) as the backbone algorithm for soundness, which is also implemented in our experiments for consistency.