Reviewer uGfq (4;4):

On the positive side, the submission addresses an interesting topic, on which we still know fairly little. To the extent that I was able to check them (see below), I believe the results check out.

Unfortunately, there are a few significant shorcomings of the current version, which prevent me from advocating for acceptance. I list them in arbitrary order:

The novelty claims are somewhat tenuous, as there is indeed concurrent work which addresses the problem of compression for adaptive optimization, in particular https://arxiv.org/abs/2102.02888 (in ICML) and [13]. The writing of the paper, specifically with respect to the technical parts, is below the standard I would expect for a paper that could be accepted to NeurIPS. To give just one example, the parameter epsilon has multiple uses throughout the paper, but is never formally introduced (e.g. is the eps in algorithm 1 the same as the one in algorithm 2, and the same as the one in Theorem 1?). This may seem like a trifle, but notice that this renders the paper's main result (Theorem 1) difficult to interpret (by the way, eps is not defined in this Theorem, which is unfortunate since one might expect the main result to be self-contained). The theoretical analysis should be improved for this to be a top-tier paper. In particular, the authors provide the 'standard' analysis only for the unrealistic single-node case (why would we run gradient compression on a single node?), and require additional assumptions on the LR sequence (which are non-standard for adaptive algorithms) in the multi-node case. The experiments are conducted on a tiny setup, on which no-one would actually need distributed learning. While the experimental data does corroborate the analysis to some extent, it is executed on tiny datasets and models, on which adaptive optimizers don't actually give state-of-the-art results (relative to e.g. SGD). This is unfortunate, since there actually are state-of-the-art settings (in particular, Transformer models) for which adaptive optimizers are state-of-the-art, and for which distributed training is the norm. I believe the experiments have to be significantly improved in this respect. General comments / Typos

single-machine -> single machine 'distributed learning framework has been' the claim that 'gradient averaging has not been considered for adaptive optimization' seems a bit too strong due to e.g. https://arxiv.org/abs/2102.02888 and [13] 1/sqrt(n) is technically not linear speedup, since it's not 1/n. This is the best achievable is this setting, but the formulation could be strange for someone who is not familiar with this. Please revise element-wisely -> elementwise Algorithm 1: what is epsilon? a small constant? lines 166-167: formulation is weird, please rephrase Theorem 1: why is epsilon not defined here?! Notice you already have ane epsilon defined in line 16 of Algorithm 2? l255: 'irrelevant quantities' -> one could disagree that all the quantities you choose to ignore are irrelevant. In particular, it would be good to have a discussion of the quantities you ignore here. moreover, your discussion of Corrollary 2 appears a bit 'optimistic' to me. In particular, you have the $n * sigma^2$ terms there, which are going to affect you fairly significantly in the regime where n is very large. I don't think this is the case for momentum SGD, which is what you claim.

Our reply:

Reviewer YLji (6;4):

The paper is generally well-written and structured clearly. As the authors mentioned, the proposed algorithm, COMP-AMS with error feedback (Alg. 2), is a straightforward extension of the classical distributed SGD that uses the AMSGrad step-sizes and compression in the update rule. The results are not surprising, and the assumptions used for the theoretical results are the same with several papers analyzing methods in the distributed non-convex setting. However, to the best of my knowledge and as the authors also mentioned, there is no analysis of adaptive method with error feedback under the same assumptions. The presentation of Theorem 1 and Corollary 1 is clear, and the discussion of what they mean is very informative and easy to follow. To the best of my knowledge, Corollary 1 is the first result showing that compressed adaptive methods with EF converge as fast as their standard counterpart.

Even if the assumptions and proof techniques are very similar to previous papers, the proposed scheme could be a step towards designing and analyzing more efficient adaptive and distributed algorithms. I ve spot-checked the proofs of the supplementary material, and the results seem correct.

Limitations:

Figures 2 and some plots of Figure 3 should be improved to be accessible for a color-blind audience. At the moment, there is no clear distinction between the lines.

This is not a limitation of the paper rather a suggestion for extension. I would like to see the convergence of the method for the strongly convex case. In this setting, there are several analyses of other distributed compressed algorithms for which one can guarantee faster convergence.

Our reply:

Reviewer D7iz (3;5):

The authors claimed in Lines 37-39 that 'Burdensome gradient transmission would slow down the whole training system, or even be impossible because of the limited bandwidth in some applications. 'I do not necessarily agree with that. The authors assume a data center setting, Lines 26-29, and in theory, what they claim in Lines 37-39 is correct! In the phase 2016-2019, people were claiming this and got away. Unfortunately, many recent dedicated works on gradient compression under various settings (with varying network speeds) have shown in practice; compression/decompression has high computational overhead, so high that we barely see a benefit when the bandwidth approaches 1Gbps. See [GRACE: A compressed communication framework for distributed machine learning, in IEEE ICDCS 2021] and [Agarwal et al. in On the Utility of Gradient Compression in Distributed Training Systems, 2021, ArXiv]. For bandwidths above 10 Gbps (even I will argue at 1 Gbps bandwidth, which is not uncommon in data centers, new Nvidia A100 servers come with 200 Gbps NICs), the network overhead is negligible, so compression does not pay off. Therefore, compression would not be beneficial in any setting of distributed training but could be in federated learning where clients are geographically remote and network bandwidths are low.

What do you mean by 'In general, larger bias and variance of the compressed

gradients usually bring more significant performance downgrade in terms of convergence [53, 2]?' This statement is too general and might be misleading. Therefore, I do not essentially agree with this statement as well. If you mean to say 'compression downgrade in terms of convergence,' I will humbly mention it does not. In the non-convex setting, compressed (irrespective of quantization, sparsification, error feedback) SGD has the same asymptotic convergence rate as uncompressed baseline SGD. Moreover, in terms of test accuracy of DNN training (which is more important than the loss function convergence), compressed SGD outperforms its uncompressed counterpart in many cases. Please see again [GRACE: A compressed communication framework for distributed machine learning, in IEEE ICDCS 2021].

Please check the paper's Notation [On the Convergence of Adam and Beyond, by Reddi et al. in ICLR 2018]. The authors strictly mention the abuse of notation in presenting a non-standard way of writing a 'vector dividing by another vector.' However, you never mentioned that anywhere in your manuscript, and suddenly, out of the blue, Algorithm 1 appears, making me think how could one divide a vector by another vector. This is a bad practice.

I disagree with 'This is also the first result in the literature regarding the linear speedup property of distributed adaptive learning under gradient compression.' Moreover, the authors claim, 'Furthermore, adopting gradient compression in adaptive methods has also been rarely studied in the literature,' is imprecise. Quickly refer to the paper: [13] Quantized Adam with Error Feedback by Chen et al. in 2020. Although there is some difference (desired) of this manuscript with [13], these strong claims are misleading. However, I thank the authors for discussing the paper later. Also, can you clarify your claim in Lines 203-204? You cite Theorem 1 (although wrongly) from [13]? After quick checking, I also think this is not true.

Please see [1-bit Adam: Communication Efficient Large-Scale Training with Adam s Convergence Speed, by Tang et al. 2021, ArXiv] that claims ADAM does not converge with error feedback. Therefore, I am a little curious about how your claims stand with the error feedback?

Uniform bound of Stochastic gradients as in Assumption 3 is too strong. In practice, it should hold when f(x) is bounded, which is never the case for DNN training. Therefore, I will encourage you to use Assumption 3 in [54], see Page 7.

Why do you need to use the complex form of Youngs inequality (To be precise, it is Peter-Paul's inequality) in arriving equation (9) in Appendix? The inequality you used with is a fancy version of the inequality with . The philosophy behind using the one with is that if one quantity has an extremely large norm, then can balance that. Otherwise, it is wiser to use the simpler version of the inequality with . However, I note that you used the simple one in Lines 696+3.

The inequality just above line 707 is not correct. There should be a factor multiplied with . I also do not understand what 707 mean.

What is upper case N in Line 674?

You chose in line 699. This essentially, makes blow up. Does not it? Now, you also chose

, and is a function of and as well. Although I did not carefully check the

polynomial order, I anticipate there is a discrepancy. Please correct me if I am wrong. Also, kindly justify these choices.

The numerical experiments in this paper are sub-standard. You may think of using the GRACE framework and deploy your COMP-AMS algorithm. Another suggestion is to compare baseline SGD with and without compression to highlight why one should use the COMP-AMS algorithm. In the present numerical experiment, this motivation is totally missing.

How do you determine if one compressor outperforms the other? If they send different data volumes, the claim 'one compressor outperforms the other' is meaningless. For example, please see Figure 3. If I understand correctly, in CIFAR-10 n=16, CP-AMS TopK-0.01 is sending the least number of bits, and the accuracy suffers the most; similar observations can be made for other figures. Therefore, one suggestion is to set the parameters to send the same average data volumes and then compare their performance.

What does this line mean? Line 158: 'Note that, larger q indicates important an compression while smaller q implies better approximation of the true gradient.' I did not understand.

Our reply:

Reviewer uKmS (6;3):

In general, the paper is clear and well-written. The ingredients of the proposed method (AMSGrad and gradient compression) have already appeared elsewhere and have been thoroughly studied, however their combination is – at the best of my knowledge – novel. Thus, the paper has some novelty. The main weakness of this contribution is that QAdam still provides superior generalization performance, for both CIFAR-10 and IMDB (MNIST is probably just too easy so QAdam offers no improvement). The superior generalization of QAdam comes at the cost of a larger communication complexity (1-2 orders of magnitudes).

A few more detailed comments are below:

(1) Have the authors experimented also on other datasets and found out that QAdam again performs better?

(2) As for compression methods, the authors discuss Top-K and Block-Sign. How would quantisation-based methods (e.g., QSGD) perform here?

(3) Is the dependence on tight in Theorem 1? Does the scaling in of the theoretical results match experimental evidence?

(4) Typos/imprecisions: (i) L. 157, 'important an compression' –> 'an important compression'; (ii) L. 163, mention that the sign function is applied component wise; (iii) L. 215, is the bound uniform in ?; (iv) L. 259, should be

Our reply: