# CONVERGENT ADAPTIVE GRADIENT METHODS IN DECENTRALIZED OPTIMIZATION

**Anonymous authors**
Paper under double-blind review

## 1 REVIEWER 1

This paper studied the decentralized adaptive gradient methods and provided convergence guarantees. Experiment on MNIST is conducted to show the effectiveness of the proposed approach.

The theoretical result is weak. The linear speedup result is not proved as in (Lian et al. 2017), the benefits of adaptive gradient methods are also not illustrated in the bound in Theorem 2 and Theorem 3.

The learning rate scheme is not practical and does not hold in practice. As illustrated in Theorem 2 and 3, the learning rate is set to be less than .

The LHS of Theorem 2 and Theorem 3 are not the standard gradient squared norm but the scaled version. It is unclear what is the bound if the LHS is the standard gradient squared norm as in (Lian et al. 2017). It is important to use the same measure as in the previous literature for fair comparison.

The experiment is weak. Doing distributed training only on a tiny dataset on MNIST is not sufficient. I would like to see results on larger datasets such as CIFAR and ImageNet.

**Our Reply.**

To COMPLETE

## 2 REVIEWER 2

Cons:

In section 3.2, the paper claims AdaGrad and AMSGrad satisfy the condition to guarantee the convergence of Algorithm 2 while Adam does not. It seems not to be an obvious conclusion from the reference Chen et al. (2019). Is there more explanation or proof about why AdaGrad and AMSGrad satisfy the condition? And does it mean Adam still diverges even after using the algorithmic approach proposed in this paper? In Theorem 2, the convergence analysis result of Algorithm 2 is given. However, the convergence of common adaptive gradient methods such as AdaGrad and Adam is still not clear. Therefore, the 'convergent adaptive gradient method' in the title is very misleading. Do most of the adaptive gradient methods have the same theoretical guarantees as AMSGrad? $\mathbb{E}[\sum_{t=1}^{T}\|(-\hat{V}t-2+\hat{V}t-1)\|_{abs}] = o(T)$ is the key condition to ensure the convergence. But when the above equation is $O(\sqrt{T})$, the convergence rate is worse than the centralized counterpart. Is that case possible? In section 3.4, the experiment is divided into homogeneous and heterogeneous data, which is very confusing. What is the reason for doing this and what will happen if we just deal with the dataset normally? The heterogeneous data is treated very intentionally. Is there any discussion about when the treatment of heterogeneous data is important? In the homogeneous data experiment, the performance of DADAM and decentralized AMSGrad are similar. What is the reason that the learning rates on different node tend to be similar? Is that a common case? Maybe the experiment on more dataset is needed to address this concern. Besides, how will such similarity among data impact the theoretical convergence?

**Our Reply.**

To COMPLETE

## 3 REVIEWER 3

However, I have some minor comments to improve the manuscript.

The experimental evaluation of the work is quite limited. I understand the space limit but it would have been nice to see more experiments instead of showcasing Algorithm 2 with an extra example in Algorithm 3. It is important to see the convergence behavior of the method (on the training data) with respect to the DGD on various datasets/networks in practice, rather than observing how the testing accuracy behaves. Note that your method does not guarantee any specific generalization behavior and therefore I believe it is more suited to report the experiments only in terms of training performance when you are out of space.

What are the drawbacks of this method? I can see more memory requirements for the agents due to the new variable $\tilde{u}$ for instance. Do you have any quantified evaluation in this respect? I suspect it can be significant especially if the trained model is large and the agents have limited memory/computational resources

In Section 3.2, the author says "Algorithm 2 can become different adaptive gradient methods by specifying $r_t$ as different functions. E.g., when we choose ..., Algorithm 2 becomes a decentralized version of AdaGrad." This sentence is not accurate as algorithms like AdaGrad and Adadelta do not use momentum on the past gradients. They only use the squared values of the past gradients. I believe your method, as I mentioned above, is a general framework for momentum-based techniques including ADAM, AdaMax, NADAM, etc, which brings me to the next question.

Is it possible to generalized your method for an adaptive gradient descent algorithm that does not use the momentum of the gradients? For example, take AdaGrad with a fixed learning rate of $\eta$ instead of $m_t$. How does your convergence behavior change?

**Our Reply.**

To COMPLETE

## 4 REVIEWER 4

This paper discusses the problem of adaptive acceleration in the decentralized setting. The premise is that decentralized versions, while successful in the simple SGD setting, do not extend well in the acceleration settings, e.g. Adam and Adagrad. They first present a counterexample where a simple decentralized scheme, applying Adam, converges to a nonstationary point. (Suggestion: I would maybe add a cleaner, more flushed out version of this proof in the appendix, maybe with illustrations.)

Overall, everything the paper presented seemed reasonable. The motivation and counterexample in DADAM case are solid, and the following theorems seem to suggest gradient error norm at rate which is reasonable in nonconvex optimization. The intuition in the adjusted merging scheme is also reasonable, and makes sense that it would work better than vanilla merging schemes. However, I did not have a chance to carefully check the proofs, which are clearly the main contribution of the paper.

One thing I would suggest is a more thorough set of numerical experiments. The two examples shown, in fact all the methods converge, and while the proposed method converges faster, it isn't really verifying the paper's main point, which is that the standard distributed methods diverge and the proposed method converges. Showing this on a standard machine learning task would improve motivation.

**Our Reply.**

To COMPLETE

## 5 REVIEWER 5

This paper consider the decentralized adaptive algorithms. At the first glance, I am really happy to that the adaptive methods are used for the decentralized optimization. However, after I read the main document, I do not think paper actually analyzes the decentralized adaptive algorithms.

In line 9 of Algorithm 1, the denominator is $\sqrt{\hat{v}_{t,i}}$

However, in Algorithms 2 and 3, it is changed as $\sqrt{u_{t,i}}$. In the proofs, the authors proved the convergence based on $u_{t,i} \geq \epsilon$. This is actually the DSGD. The proofs can be quite simple. And the restriction $\alpha = O(\sqrt{\epsilon})$ can be easily removed. This paper does not present any insights for the decentralized adaptive methods. It only depends on $u_{t,i}$. The numerical results show that the

proposed method is similar as DSGD. As mentioned before, it is actually DSGD but with slightly modification.

**Our Reply.**

To COMPLETE