We would like to thank the four reviewers for their feedback. We first discuss a common shared by **R1**, **R5**, and **R6**:

– Numerical Experiments: Our experiments in the main paper aim at showing the advantages over DADAM, a decentralized variant of Adam method, developed in [Nazari et. al., 2019]. We recall that the purpose of this paper is to provide both an *algorithmic* and theoretical framework for decentralized variant of adaptive gradient methods. Hence, single-server Adam method does not constitute a baseline for our method, rather its decentralized version DADAM does. We highlight the advantages over SGD comparing Figure 2, 3 and Figure 4 in section F of the appendix where our proposed algorithm is less sensitive to the lr, which is one edge of adaptive methods. While DADAM shows divergence (Fig. 1), our decentralized framework, using AMSGrad as a prototype, and D-PSGD of [Lian et. al., 2017] are exhibiting great convergence. Figure 1 is convincing on the need for a convergent decentralized adaptive method, thus fixing the divergence issue of DADAM (shown both theoretically and empirically through Figure 1).

R1: We thank the reviewer for the remarks:

– Comparison with [Chen et. al, 2020] ([C20]): [C20] is one of a few recent attempts to use adaptive gradient methods with the periodic model averaging technique in federated learning. Essentially, the divergence issue in both [C20] and our paper are caused by asynchronous adaptive lr. [C20] use the parameter server to maintain a synchronized adaptive learning rate (lr) sequence to cope with the issue, leading to local AMSGrad (LAMS). Our setting is different since *a central server is not available*, thus we use average consensus mechanism to gradually synchronize adaptive lr. Since both decentralized AMSGrad (DAMS) and LAMS use AMSGrad as prototype, they reduce to similar ones if local iterations $k = 1$ in LAMS and the graph is fully connected in DAMS. With differences being the $\hat{v}_{t,i}$ is maintained by each worker and the extra ϵ in line 10 of DAMS. The key difference is that we study how to use adaptive gradient methods in decentralized optimization **without** a parameter server, rather than under federated learning settings. As asked by the reviewer, it is indeed possible to extend the periodical averaging technique used in [C20] to our decentralized setting. The resulting algorithm will execute line 7,8,11 every k iterations and $\tilde{u}_{t,i}$ will not be updated in local iterations. We expect our result to have similar dependency on k as in [C20], i.e., the big-O rate will not be affected for $k \leq O(T^{1/4})$ and applies to our framework.

– Bias of v, the estimate of the second order moment: The [mean of square of gradients] vs [square of mean of gradients] problem will not be a source of bias in most cases. Using the same AdaGrad example with $\hat{v}_{t,i} = \frac{1}{t}\sum_{k=1}^{t} g_{k,i}^2$, when t is large and ϵ is small, the adaptive lr $\tilde{u}_{t,i}$ is similar to its tracking target $\frac{1}{N}\sum_{i=1}^{N} \hat{v}_{t,i} = \frac{1}{Nt}\sum_{i=1}^{N}\sum_{k=1}^{t} g_{k,i}^2$, which is still the mean of square of stochastic gradients. If we want to estimate second moment of the gradient estimator over the optimization trajectory, it is unbiased. However, this could indeed be biased if we want to estimate the second moment at early iterations, because the distribution of stochastic gradients could change across iterations. In the second case, the average consensus mechanism will induce a small bias due to the time lag in consensus of $\tilde{u}_{t,i}$. The effect of such a bias on the training is usually problem dependent. It is possible to kill the bias by using fresh stochastic gradients to estimate the adaptive lr, but this will introduce extra computation cost and is usually not used in practice.

R5: We thank you for the valuable comments.

– Comparison with [Chen et. al, 2019] ([C19]) and [Zhou et al., 2018]] ([Z18]): We compare Th. 3.1 of [C19] with our Th. 2. The term multiplied by C_1 in our Th. 2 have similar source as the terms multiplied by C_1 and C_4 in Th. 3.1 of [C19]. The terms multiplied by C_4 and C_5 in our Th. 2 have similar source as the terms multiplied by C_2 and C_3 in [C19]. The other terms in our Th. 2 are caused by *consensus errors* of variable and adaptive lr. We also compare Th. 5.1 in [Z18] with our Th. 3. The C_1' terms in Th. 3 have similar sources as M_1 and M_3 terms in [Z18], the C_4' term corresponds to the M_2 term in [Z18]. Note that [Z18] can show an improved rate assuming sparse gradients while we do not consider such assumptions.

R6: Thank you for the thorough analysis.

– Explanations on the assumptions: As rightly mentioned by the reviewer, the stepsize is in order $\alpha_t = 1/\sqrt{T}$. The dependence in d leads to a small lr in the presence of large networks but our Th. states that the rate would then be as fast as we present it. Hence, our bound prevails over the intuition that the convergence will be slow due to a small lr.

R8: We thank the reviewer for his/her interest in our paper. Below we address your concerns:

– Discussion on the matrix W: The way to set W is not unique, one common choice for undirected graph is the maximum-degree method in [Boyd et. al. "Fastest mixing Markov chain on a graph.", 2004] (denote d_i as degree of vertex i and $d_{\max} = \max_i d_i$, this method sets $W_{i,j} = 1/d_{\max}$ if $i \neq j$ and (i,j) is an edge, $W_{i,i} = 1 - d_i/d_{\max}$, and $W_{i,j} = 0$ other wise, a variant is $\gamma I + (1-\gamma)W$ for some $\gamma \in [0,1)$), this choice can be viewed as transition matrix for a random walk, it ensures assumption A4 for many common connected graph types. A more refined choice of W coupled with a comprehensive discussion on λ in our Th. 2 can be found in [Boyd et. al. "Fastest mixing Markov chain on graphs with symmetries.", 2009], e.g., $1-\lambda = O(1/N^2)$ for cycle graphs, $1-\lambda = O(1/\log(N))$ for hybercube graphs, $\lambda = 0$ for fully connected graph. Intuitively, λ can be close to 1 for sparse graphs and to 0 for highly connected graphs. This is consistent with the bound in Th. 2, which is large for λ close to 1 and small for λ close to 0 since average consensus on sparse graphs takes longer time.