Thanks for all the valuable comments. Please check our responses below. We will address all minor comments.

**Reviewer 1:**
Q1: Please clarify the text and also state that $W^t$ means $W$ to the power of $t$ (if I am correct!).
A: Thanks for the suggestion. You are absolutely correct! We will make it more clear in the revised version.

**Reviewer 2:**
Q1: Results do not seem to recover the single-machine guarantees when $M=1$ (only one node).

A: In the bound in Theorem 1, there are 3 terms. The only term that prevents from recovering the single-machine case is the third term (i.e., $\frac{4kDL_\sigma+\sigma^2}{\sqrt{mM}}$). This term completely comes from the procedure of bounding the LHS of (15) in the supplement, and is loose when $M=1$. However, thanks to the Assumption 1(iv), when $M=1$, the LHS of (15) is negative and disappears in the proof of Theorem 1. In terms of learning rate, when $M=1$, we have $\rho=0$, so $t=0$ and hence $\eta=O(1/L)$. Hence we can exactly recover [Theorem 1, 15], if we replace $\frac{4kDL_\sigma+\sigma^2}{\sqrt{mM}}$ by $\frac{4kDL_\sigma+\sigma^2}{\sqrt{mM}}1_{M>1}$ with 1 being the indicator function in the statement of Theorem 1. We will change it in the revised version.

Q2: The claimed $O(\log(1/\epsilon))$ communication complexity at the busiest node. If I am correct, this is the per-iteration complexity.
A: You are absolutely correct. We adopt the notion ‘communication complexity on the busiest node’ from [5, Table 1].

**Reviewer 3:**
Q1: Is there any difficulty in proving consensus of local iterates?
A: We can indeed prove the $\epsilon$-consensus, since our algorithm allows $t$ rounds of decentralized communication in each iteration and $t=O(\log(1/\epsilon))$. We will mention it in revision.

Q2: Please explain the meanings of barrier and lock-step.
A: In each iteration, a learner does not proceed until it finishes exchanging and averaging its weights with its neighbors. This data dependency forms an implicit barrier (i.e., we do not need to enforce an explicit barrier in the program) so that all the learners process the same number of iterations (i.e., mini-batches) at any given time (i.e., lock-step).

**Reviewer 4:**
Q1: The proposed algorithm and theoretical analysis are proposed, but they both are direct extensions of existing results, which significantly weaken the contribution.
A: We respectfully disagree. To handle this challenging nonconvex-nonconcave min-max problem, we have to design a novel algorithm such that (1) it is simple and user-friendly to large-scale decentralized training system; (2) it can be proved to have polynomial time complexity; (3) it is able to deliver good empirical performance in large-scale GAN training. Satisfying these requirements simultaneously is difficult. It is *NOT* a direct extension of any existing results. Our algorithm is carefully designed and we conduct extensive and comprehensive empirical studies.

First, we use novel algorithm design: maintaining two update sequences, designing logarithmic communication rounds, and updating the discriminator and generator simultaneously. Second, in experiments, we consider both medium-scale (WGAN-GP on CIFAR10) and large-scale (SA-GAN on ImageNet) GAN training, with both high and low latency environment, and our proposed algorithms consistently deliver remarkable performance. It would be appreciated if the reviewer can provide us concrete references so that we can compare or argue against.