We thank all reviewers for their overwhelmingly positive feedback on our work. Each reviewer provided helpful suggestions to improve our manuscript that we address below, while providing extra experiments as requested.

Reviewer 1

- *“Can or should the adversarial cases listed in the paper [...] be modeled as *worst* case attacks?”*

Our work complements a recent, growing body of work on Byzantine ML, where worst-case failures capture a range of things that can go wrong during training: power outages, software bugs, bit-flips at the storage/network/app level, and adversarial nodes that corrupt the trained model by sending erroneous gradients. Due to the wide range of failures, modeling them as worst-case allows for universal robustness guarantees.

- *“Can the authors show simulations practical cases failures [...]?"*

Simulating many different types of failures is interesting but challenging from a system and cost-of-experiments perspective. Still, in our experiments on real distributed systems, we simulate the strongest known type of node failures/adversarial gradients, in order to showcase our performance even under the most challenging setups. Under all these setups, DETOX consistently improves robustness and speed by orders of magnitude.

- *“[...] how their approach is exactly affecting the communication and computation cost [...]?”*

Our communication cost is identical to the vanilla parameter server aggregation cost, as each node sends to the PS a single gradient. In terms of the cost of computation, we discuss in the paragraph “Improved speed” ln. 160 - 170, how DETOX improves the aggregation runtime to nearly linear per iteration, cutting down the quadratic runtimes of state-of-the-art robust aggregators. This improvement naturally varies with different aggregators used, as we discuss in the same section.

Reviewer 2

- *Typos and clarifying variable names.*

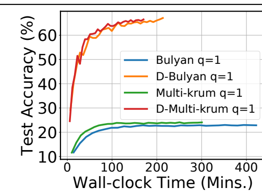
Typos fixed. We will restate variable names when it is not clear from context.

- *“The framework is [...] substantially more complex and may make adoption [...] more difficult.”*

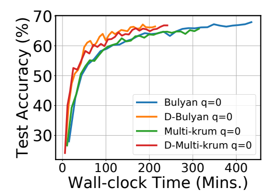
This is a valid concern. We want to note that DETOX is modular and hardcoded to the training process. From a user’s point-of-view, the only choice required is what the local aggregators $\mathcal{A}_0$ and $\mathcal{A}_1$ will be. In our implementation (anonymously available at: `http://bit.ly/2SRyvcS`) this can be done by changing one line of the code. Since this is a relatively minor code change, we hope that this will make adoption easier.

- *“Provide [...] results [...] for more values of q, including q=0.”*

We will provide a thorough study on the effect of varying q in the camera-ready version, including the ones shown in Figure 1. Due to the space limit, we show here the experimental results of $q = 0$ and $q = 1$ (under ALIE Byzantine attack). We observe that DETOX versions of robust aggregators consistently beat their standard versions. Different values of q do not seem to affect the robustness and scalability of DETOX.



(a) $q = 1$, VGG13-BN, ALIE attack



(b) $q = 0$, VGG13-BN

Figure 1: Comparison of DETOX paired with BULYAN, MULTI-KRUM versus their vanilla variants for (a) the ALIE attack on VGG13-BN and CIFAR-100 and (b) $q = 0$ (no failures

---

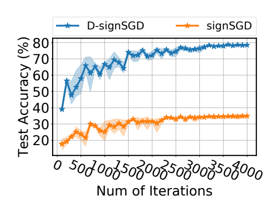
Reviewer 3

- *“[...] majority vote [...] might lead to a big loss in terms of variance reduction.”*

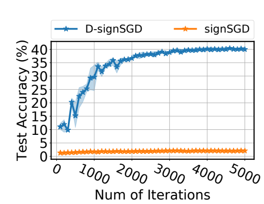
This is a subtle point that can cause confusion. DETOX makes nodes evaluate redundant gradients, so that there is no increase in variance. Notice that DETOX first assigns a set of br/p data points to each node group. The nodes in each group are assigned the same set of br/p points. The nodes then compute the *mean* of gradients of these points. All “honest” workers in a group return the same averaged gradient, while averaging leads to variance reduction by a factor of br/p. If the majority is won by the “honest” nodes in the group, this reduced variance gradient is propagated to the second phase of hierarchical aggregation. We clarify this in lines 172-176, and this fact is used in the proof of Theorem 3.

- *“[...] I would highly encourage the authors to try incorporating something like signSGD [...] in the base layer.”*

Thank you for the suggestion! We agree that incorporating DETOX with SIGNSGD is valuable. We conducted experiments on DETOX paired with SIGNSGD versus vanilla SIGNSGD under a *constant* Byzantine attack, where Byzantine nodes send a constant gradient matrix where all elements equal to -1. The experimental setup is $p = 45, q = 5$. The results are shown in Figure 2. We will include a longer version of this experiment in any camera-ready version.



(a) ResNet-18 on CIFAR-10



(b) VGG13-BN on CIFAR-100

Figure 2: Convergence of SIGNSGD with and without DETOX under *constant* gradient attack for: (a) ResNet-18 on CIFAR-10; (b) VGG13-BN on CIFAR-100