We thank the reviewers for their detailed and constructive comments, especially during these unprecedented times.

Our Motivation (R1,R3,R4,R5): is to enable on-device learning on power and bandwidth limited devices, where the communication reduction offered by current state-of-the-art compressed D-SGD is a good start, but is simply not enough for large DNNs with millions of parameters. It is therefore necessary to develop a strictly **complementary** source of communication reduction. Our paper provides a proof of concept for such a new source of communication reduction, establishing the theory and validating it on DNN experiments. Our algorithm isn't designed to compete (or be compared) with **other decentralized algorithms (R1, R4)** but instead to be combined with them to substantially reduce the communication overhead further. Our algorithm can be combined with: communicating only every τ gradient updates (**R1** [3]), adaptively communicating only updates of significant magnitude (**R1** [4]), and compression of the communication (**R1** [5,6]). We have expanded our literature review to discuss these algorithms (**R4**) and our interplay with them. Our new experiment in Fig. A below shows that by combining (simple) communication reduction schemes with our method one can operate under significantly stricter bandwidth constraints than was previously possible. For the revised paper, we will test additional communication reduction schemes on D-Dist.

Comparison to D-SGD (R1,R3,R4):Since on-device is a constrained setting there might be no choice but to lose some performance compared to the ideal D-SGD. However, in our new experiment in Fig. B we achieve close to D-SGD performance on MNIST with **16 nodes (R1, R4)**. Our new experiment in Fig. C demonstrates the gain from **different local models (R1, R4)** our method obtains, which can outperform D-SGD in practice by allowing more devices into the training process. Even by just joining two subsets with different models, our method matches D-SGD which is forced to run on each subset of 4 devices separately. We will add to the paper an experiment with 4 different models.

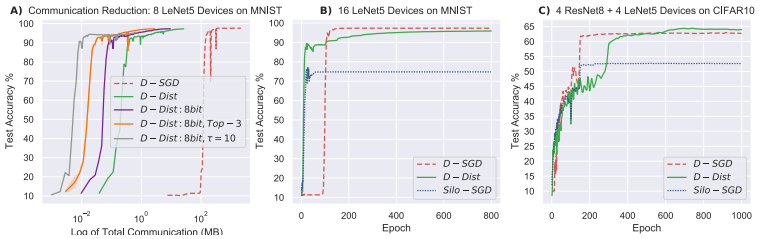

Figure: A) We apply the schemes of only communicating every τ gradient steps ($\tau = 10$) and compression via quantization (8 bit) and TopK (keep the top $K = 3$ elements in each soft decision) included in (R1 [3,6]) to D-Dist. In the revised paper, we will add the CIFAR10 counterparts for experiments A) and B), and both the epoch and communication plots for all experiments.

Communication (R1): Our algorithm communicates soft-decisions on the reference dataset (network soft-decisions) *but only* communicates a "network batch size" number of soft-decisions (even as low as 16) at each iteration. Analogous to how mini-batch SGD operates, we can tune the "network batch size". Our experiments show that we *save 100-600×* ***(improved to 10,000× in Fig A)*** *in communicated bytes* versus D-SGD when *comparing at the same accuracy level*.

The Reference Dataset (R4, R5): should as large as possible, up to device constraints similar to how we would select the training dataset size. While the gradient bounds in Lemma 5 increase, performance improves since the set of distillation stationary points gets more selective as devices have to agree on more data points (see experiment in Section 12.2.2). Reference data can be synthetic and then it is easy to obtain (as in co-regularization, see R1's comment).

Lipschitz Continuous Gradients (LCG)(R1): This holds since $\mathbf{s}, \mathbf{y}$ are bounded (line 133). If $\mathcal{L}_n(\mathbf{s}, \mathbf{y})$ has LCG (for every $\mathbf{y}$) then $\mathcal{L}_n(\mathbf{s}, \mathbf{y})$ has bounded gradients since $\|\nabla_s \mathcal{L}_n(\mathbf{s}, \mathbf{y}) - \nabla_s \mathcal{L}_n(\mathbf{s}_0, \mathbf{y})\| \leq \|\mathbf{s} - \mathbf{s}_0\| \leq M$ ($\nabla \mathcal{L}_n(\mathbf{s}_0, \mathbf{y})$ is a constant). Then $\mathcal{L}_n(\theta^n)$ is LC as a sum of compositions of LCs: $\mathcal{L}_n(\mathbf{s}, \mathbf{y})$ and $\mathbf{s}(\theta, x)$. $\nabla \mathcal{L}_n(\theta^n)$ has LCG from the chain rule, since $\mathbf{s}(\theta, x)$ has LCG. We now explain that in detail. Simply assuming that $\mathcal{L}_n(\theta^n)$ is LC is also fine.
Graphs (R4, R5): Our results apply to undirected graphs, a special case of a directed graph where strong connectivity coincides with connectivity. The graph is pre-defined. It can be generalized to a time-varying graph with a more cumbersome analysis. The graphs in this work were randomly drawn for a given maximum number of degrees per node.
Euclidean Distance & Smoothness (R1, R3): Our analysis needs the loss function to be Lipschitz smooth in both variables; KL-divergence is not. A smooth model is a common assumption in theoretical analyses ([4,7,9] in our paper).
First-order Necessary Condition (R4): We prove that devices not only converge, each to a local stationary point, but that they also agree on the reference soft-decisions. A device that accidentally converged to a local maximum or saddle point will suffer from poor performance and wouldn't produce the correct soft-decisions to agree with all other devices.
Lines 228, 233-234 (R4): If $\mathbf{z}_t^n$ is a probability vector $\forall n$ then so is $\tilde{W}\mathbf{z}_t$ (stochastic matrix) and the sum over $\mathbf{z}_t^n(x) - \mathbf{s}(\theta_t^n, x)$ is zero, making $\mathbf{z}_{t+1}^n$ a probability vector for all n. The averaging step is a matrix multiplication that if operating "in a vacuum", has a geometrical convergence rate (Lemma 7). The norm of the consensus error is $O(\eta_t)$ (Lemma 4). In comparison, even centralized SGD has an error of $O(\frac{1}{\sqrt{t}})$ at best (non-convex), slower than $O(\eta_t)$.
The "second moment" bound of Lemma 5 (R4): holds with probability 1, so it does not need an expectation.

We now refer to the highly relevant line of work of **Co-Regularization (R1)**. Our theory applies to **Regression (R1)** with minor modifications, and that's a very nice insight that we now discuss in the paper. Thanks!
Minor Comments: We have fixed all minor issues. Section 12.2.2 starts at line 589 and Section 12.2.3 at line 617.