We thank the reviewers for their thoughtful comments. Their comments have helped us improve the experiments and clarity of the paper significantly as we discuss below.

[R1,R3] There is a large gap between SGD and SuperSGD. To clarify, we run both SGD and SuperSGD for the same number of steps and similar learning rate schedule but different number of GPUs with the same mini-batch size on each GPU. On ImageNet, SGD has lower accuracy than SuperSGD because the number of steps taken is not enough for convergence with mini-batch 64. On CIFAR-10, SGD has 1.5% higher accuracy that as noted in the paper is a known gap that can be reduced by extensive hyperparameter tuning [29]. We will clarify this in the paper.

[R1] Clarify that TRN uses three levels and it has competitive performance on AlexNet. Thank you for noting gradient clipping in TRN. In particular, Fig. 1 shows the improvement all methods gain from gradient clipping. We will also clarify in the revision that TRN uses three levels and it achieves 0.92% accuracy improvement for AlexNet with mini-batch size of 1024. We emphasize that our methods match the performance of SuperSGD in more training settings.

[R1] In Fig. 7(a), the validation accuracy should improve when the bucket size is very small. Thank you for noting this issue. We fixed a bug that only affected the runs with very small bucket sizes. As expected, all validation accuracies improve uniformly by reducing the bucket size. We will replace the figure in the revision.

[R1,R3] Provide experiments with more number of GPUs. We present new results in Table 1 similar to Table 1 in the paper but with 16 and 32 GPUs. Proposed methods ALQ and ALQ-N perform significantly better than prior works and reach the performance of SGD that has converged. As noted above, the SuperSGD is slightly worse than SGD.

[R3] Computation overhead is based on bucketsize=64, however all main results are based on bucketsize=8k/16k. On ImageNet, we save at least 60 hours from 95 hours of training and add only an additional cost of at most 10 minutes in total to adapt quantization. For bucket sizes 8192 and 16384 used in the paper and 3-8 bits, the per step cost relative to SuperSGD (32-bits) is between 21% to 25% for ResNet-18 on ImageNet and 32% to 36% for ResNet-50. That is the same as the cost of NUQSGD and QSGDinf without additional coding or pruning with the same number of bits and bucket sizes. The cost of the additional update specific to ALQ is between 0.4% and 0.5% of the total training time. We will provide a full table of timing results in the revision with varying bucket sizes and bits.

[R4] How about comparing different methods in terms of overall communication load? For fair comparison, none of the methods use an additional coding scheme or pruning to further reduce communication costs.

[R5] Do authors claim that ALQ/AMQ are robust to all values of momentum, batch size, etc? No. Our robustness claim refers only to the bucket size and number of bits. We have justified this claim by comparing the performance of methods across a wide range of bucket sizes and number of bits. We will clarify this in the revision.

[R3] How about shortening theoretical results? In the revision, we move the details of Theorem 4 to the appendix.

[R1] How are gradients/weights communicated through the network? We consider a synchronous setting for sharing gradients similar to [20] (we do not communicate weights).

[R5,R3] Elaborate on Related Work. How about LQ-Net? We will discuss expand related works in revision. The goal in LQ-Net is to quantize weights and activations such that the inner products of them can be computed efficiently using bitwise operations (single-processor setting). Compared to LQ-Net, our schemes are more efficient and do not need additional memory for encoding vectors. In training, LQ-Net updates levels for each vector to be quantized.

[R4]: Details of simulation setting. Appendix K describes the setting which we will expand on. We encourage reviewers to see the code.

[R3]: Can AMQ with p beyond $\frac{1}{2}$ be implemented efficiently? Selecting $p = \frac{1}{2}$ might simplify bitwise operations. However, as our experiments show, $p = \frac{1}{2}$ is far from optimal and the computational overhead of AMQ is negligible.

Method	16 GPUs	32 GPUs
SGD	**92.40%** ± **0.06**	**92.47%** ± **0.09**
SuperSGD	**92.17%** ± **0.08**	92.19% ± 0.04
NUQSGD	85.82% ± 0.03	86.36% ± 0.01
QSGDinf	89.61% ± 0.03	89.81% ± 0.05
TRN	88.68% ± 0.10	90.22% ± 0.05
ALQ	**91.91%** ± **0.06**	**91.89%** ± **0.07**
ALQ-N	**92.07%** ± **0.04**	**91.83%** ± **0.03**
AMQ	91.58% ± 0.05	91.38% ± 0.06
AMQ-N	91.41% ± 0.08	91.40% ± 0.02

Table 1: Validation accuracy of ResNet32 on CIFAR-10 using 3 quantization bits (except for SGD, SuperSGD, and TRN) and bucket size 16384.

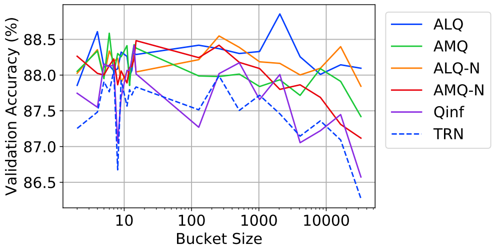

Figure 1: Effect of bucket size with gradient clipping from TRN on training ResNet8 on CIFAR-10.