Due to space constraints we only address major concerns; all suggestions will be included in the final version.

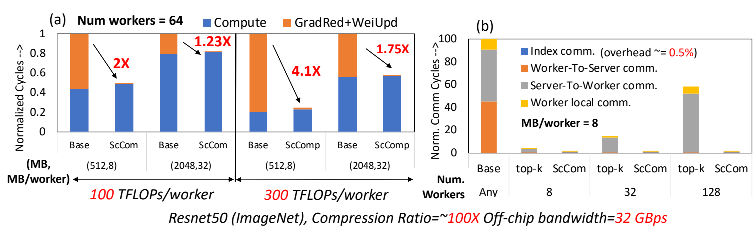
Q1(R1) novelty of low pass (LP) filter: The proposed LP filter is fundamentally different from previous weighted error-feedback work [arXiv:1806.08054] and [doi.org/10.1609/aaai.v34i04.5706]. Our method aims to mitigate the impact of increased gradient noise in large batch size training so it is necessary to apply a discounting factor β (ex: 0.1) in **new incoming residues**: $m_i^{t+1} = (1-\beta)m_i^t + \boldsymbol{\beta} e_t$. Previous work adds a forgetting factor β (<1) in **error feedback** to bound the variance of **previous** residues, but it does not apply any discounting factor to incoming residues: $m_i^{t+1} = (1-\beta)m_i^t + e_t$. Fig. 2(c) shows clearly that applying β to **new incoming residues** is critical for improved local memory correlation with high learning rates. Experimentally we've observed that when using previous weighted error-feedback, large MB ResNet18 (ImageNet) shows 3.7% degradation compared to proposed LP filter for 64 workers (proposed LP filter: 69.8% vs previous weighted feedback: 66.1%). In theory, compared to prior arts, the extra term resulted from the model compression is also shrinking in a rate of $\mathcal{O}(1/T)$. Please refer to eq.(A52) in appendix D.

Q2(R1) CLT-k and other top-k methods: Compared to previous top-k methods (ex:[arXiv:1901.04359]), CLT-k has two major differences: (i) CLT-k is a commutative operator so network convergence is guaranteed. As suggested in eq.8 of [arxiv.org/pdf/1809.10505.pdf], without explicit assumptions, non-commutative compressors do not guarantee convergence. (ii) CLT-k has $\mathcal{O}(1)$ in both scalability and compression overhead (due to local sparsity patterns). To approximate top elements, techniques such as gTop$k$ and powerSGD require merging local top$k$ elements, which incurs non-perfect scalability such as $\mathcal{O}(\log(n))$. We will compare and cite related work (gTop-k) in the final draft.

Q3(R1) Remark3 all-reduce ring: In ring all-reduce, we divide the gradient buffer into n (worker number) parts and assign each worker a part. In the 1st iteration of reduce-scatter phase, each worker selects top-k in its corresponding piece and sends selected indexes/gradients to the next worker. Then in the following iterations of reduce-scatter, each worker will just receive the incoming indexes/gradients, sum them with local gradients; then send results to the next one. In each mini-batch iteration, we re-assign the piece amongst workers. Additional top-k index exchange is not needed.

Q4(R1, R3) Large datasets/small batch size: In theory, large dataset/small batch size introduces more noise to gradients and deceases statistical similarity between workers and is thus tougher to deal with. In sec.3 we assume min. overlap of hamming dist. between workers to guarantee contraction < 1, which is a mild assumption in practice. Fig.3 shows that in per-worker MB=32; the hamming dist. is still above 0.32. In pilot experiments, we even tried per-worker MB=8 on CIFAR10 without noticeable degradation. In addition, Table 1,2 had broadly reported results on large datasets (ImageNet, WMT14). These empirical observations are consistent to [arxiv.org/pdf/1712.06559.pdf], which proved that SGD has a small critical batch size to approximate a full gradient descent iteration, no matter the size of dataset.

Q5(R2, R4) System performance: Appendix-F shows ScaleCom's scalability in system performance; more details here for practical applicability. The fraction of time expended in gradient/weight communication limits the overall end-to-end training time improvement achieved with ScaleCom. As shown in Figure a, when minibatch/worker is increased from 8 to 32, the communication time (as a fraction of total time) decreases from 56% to 20%. Consequently, for a 100 TFLOPs/worker peak compute capability, ScaleCom achieves total training speedup of 2× to 1.23× even with ∼100× compression. Fraction of communication time grows with increase in peak TOPs (100 to 300), resulting in speedup of 4.1× to 1.75×. The key trait of ScaleCom is its *performance scalability to larger number of workers* independent of minibatch/worker. As shown in Figure b, the communication cost of prior top-k approaches increase linearly with number of workers, whereas ScaleCom remains constant.



Q6(R3) LP filter and momentum SGD; sensitivity of β in LP filter: [momentum SGD]: Intuitively, momentum SGD can be viewed as a form of filtering (moving average) on current and past gradients, which smooths out noisy gradients to update weight more accurately. Analogously, we perform filtering on the residual gradients (see eq.(5)) to improve signal integrity in local memory. Connection will be discussed in the revised version. [β sensitivity]: We observed that β is robust to different networks' convergence in the range of 0.1-0.3. Thus, β 0.1 is used in Table2.

Q7(R4,R1) top-k index commun. and sync: (i) [commun.]: Since the index vector has the same degree of compression as the gradient vector, it occupies only 0.5% of baseline commun. time (see Figure(b) in Q5). Also, the cost remains constant with increased workers ($\mathcal{O}(1)$ scalability) (ii) [sync]: While ScaleComp incurs an additional sync step, it has negligible impact on performance. Similar to fully sync. SGD the slowest worker determines when the gradient commun. can begin. Once this point is reached by all workers, additional sync for handshaking cost little extra time.

Q8 (R4) Section 3 (theory) exposition and intuition: We provided the following table to explain section 3's main results and connected them to other parts of paper. For Remark 4, linear speedup refers to that when T is large enough, $1/\sqrt{nT}$ leads convergence rate. As worker number n increases, required iteration T linearly decreases to achieve the same convergence[arxiv.org/abs/1705.09056]. Our theorem 1 shows this; indicates its applicability in distributed training.

	Lemma1: contraction property	**Lemma2: contraction in distributed setting**	**Theorem1: ScaleCom's convergence rate same as SGD ($1/\sqrt{T}$)**
Intuition	Higher correlation between workers brings CLT-k closer to true top-k.	Require positive correlation between workers in distr. setting	Ideally ScaleCom's noise does not impact final conv. results
Connect to exp.	Fig.2 and 3 show high correlation so our contraction is close to true top-k.	Fig.2 and 3 show positive correlation between workers	Table 1,2 (Fig4,5) verified ScaleCom's convergence same as baseline