General comments: • We will make our implementation open source, if this work gets accepted. • $\mathbf{x}_0$ and $\widehat{\mathbf{x}}_0^{(r)}$ can be randomly set to the same point and our analysis does not require them to be $\mathbf{0}$; we can change this, if accepted. • We will define mini-batch size b in the main text of the final version, if accepted. • While the aggregate broadcast has not been our focus, as in [4, 19, 38], it can be inexpensive when the broadcast routine is implemented in a tree-structured manner as in many MPI implementations, or if the parameter server aggregates the sparse quantized updates and broadcasts it. We will include a discussion if accepted. • We have compared the communication budget required for achieving a given target accuracy across different techniques and demonstrate that *Qsparse-local-SGD* is the most efficient as it improves on the Pareto frontier. Therefore, we are not making an absolute statement, but a comparison between our scheme and the state-of-the-art, thereby, not contradicting the views of the speaker.

Description of technical challenges: As discussed in lines 87 to 90 of the paper, the analysis of *Qsparse-local-SGD* poses several technical challenges, including (i) the proof of controlled evolution of the error as well as the deviation of the local iterates, and (ii) asynchronous updates together with distributed compression using operators which satisfy Def 3, including our composed (*Qsparse*) operators. To the best of our knowledge, all the results in the paper are new and there are several technical challenges to prove each of them; we highlight a small selection of them below.

From literature [3, 29], we know that methods with error compensation work only when the evolution of the error is controlled. Unlike previous works, *Qsparse-local-SGD* stores the compression error of the net *local update*, which is a sum of at most H gradient steps and the historical error, in the local memory. Our analysis of the controlled evolution of memory while using any compression operator satisfying Def 3 required some work; see Appendix B.3. Furthermore, we also need to prove that the local iterates which evolve on their own do not arbitrarily deviate from each other; analysis of which is provided in Appendix B.4. Another useful technical observation is that the composition of a quantizer and a sparsifier results in a contraction operator (Def 3); see Appendix A for proofs on the same.

This gets more involved in the asynchronous setting as discussed in line 274. In such a scenario, the proof of the average true sequence being close to the virtual sequence is non trivial (Appendix C.5, C.6) and requires carefully chosen reference points on the global parameter sequence lying within bounded steps of the local parameters. This problem also arises in bounding the deviation of local iterates, the details of which is provided in Appendix C.3, C.4.

Experiments: We do have comparisons between the individual techniques of quantization, sparsification, local iterations, and their smaller combinations, with *Qsparse-local-SGD*. Our preliminary experiments demonstrated superior performance of $Sign$ operator composed with TopK-SGD, as compared to other quantizers and sparsifiers. Therefore, we use the $Sign$ operator as our quantizer and TopK as the sparsifier. Our main objective through numerics has been to demonstrate that our algorithm outperforms the cases when quantization, sparsification, and local iterations are being used individually or in pairs. In Fig. 1c, we observe that SGD with 4 local iterations requires $2.83\times$ less bits than vanilla SGD to achieve 70% top-1 accuracy, EF-signSGD and TopK-SGD require $16\times$ and $128\times$ less bits, respectively, for the same. Therefore, both quantization and sparsification, when individually used with error feedback, are superior to vanilla SGD with or without local iterations. When these techniques are all put together, we have SignTopK-SGD with 8 or 16 local iterations, which demonstrates more than $1024\times$ gain over vanilla SGD and is superior to all the above methods. We observe similar trends in the Appendix in Fig. 4, for top-5 accuracies, and in Fig. 2, which runs the synchronous operation, for a convex objective. We also have comparisons for the asynchronous operation in Fig. 3 in Appendix D.

We wish to clarify that Fig. 1-4 compare the performance of our composed (*Qsparse*) operator, namely SignTopK, with (i) Top_k SGD with error compensation (TopK), (ii) SignSGD with error compensation (EF-signSGD), and (iii) vanilla SGD (SGD). All of these are specializations of *Qsparse-local-SGD*. Furthermore, SignTopK uses a synchronization period of 1, whereas SignTopK_4L uses a synchronization period of 4, similarly for _8L, _16L.

Parameter tuning: *Qsparse-local-SGD* has a lot of flexibility and many different specializations, which can be realized by using local iterations with different combinations of quantizers and sparsifiers (in a piecewise manner). One way of tuning the number of quantization levels (s) and sparsification parameter (k) numerically is that, for a given γ and H, find an (s, k) from the family of operators corresponding to the γ that minimizes the total number of bits. H can be varied without disturbing the proposed asymptotic limits of local iterations, see Corollary 1, 2 and Theorem 3, 4.

Comparison with paper on decentralized optimization [15]: We had given a brief distinction of our work with [15] in footnote 2 of the paper, which we expand below. The main distinctions of *Qsparse-local-SGD* from [15] are (i) our algorithm employs local iterations with both synchronous and asynchronous operations, (ii) provides guarantees for both smooth (non-convex) and strongly convex objectives, (iii) combines quantization and sparsification. [15] uses operators satisfying Def 3, which could be either a quantizer or sparsifier (not both) with focus on decentralized SGD *without* local iterations and *only* for strongly convex objectives. In contrast, we propose a class of composed operators (*Qsparse*) satisfying Def 3, (which is directly useful in [15]), which is applied infrequently onto the net local updates before synchronization. Our analysis also results in characterization of asymptotic limits of local computations for general compression operators. Therefore, to the best of our understanding, the results in the paper are not covered in [15].