Response to Reviewer 1

"compare their method to QSGD": Our ImageNet experiments are based on the code base of [4]. [4] already showed that (1) vanilla QSGD is worse than signum with majority vote; (2) when two-way compression is used, QSGD is still significantly worse. In this paper, we show that the proposed method outperforms signum, and thus outperforms QSGD.

"also appeared in [14]": Ours is based on more advanced error feedback mechanism [9,16], which is more general and any δ-approximate compressor can be used, while [14] is restricted to quantization methods. Moreover, [14] (with no theoretical guarantee) does not accumulate quantization errors as in MEM-SGD, EF-SGD and the proposed method.

"distinguish their work from this one": Our dist-EF-SGD is better than Wu et al. (2018) as: (1) [Wu]: uses multi-bit quantization of QSGD; Ours: allows arbitrary compressors, including the unbiased quantization of QSGD (see lines 130-132). Thus, Algorithm 2 reduces to dist-EF-QSGD with two-way compression when QSGD's unbiased quantization is used; (2) [Wu]: past quantization errors are decayed exponentially. Thus, error feedback is limited to a small number of iterations; Ours: past quantization errors are decayed by a time-varying factor depending on stepsize. It can be shown from Corollary 2 that our stepsize choice ensures this factor to converge to one (so error neither explodes nor decays rapidly); (3) [Wu]: theoretical analysis only on quadratic objectives, with convergence to a neighborhood of optimal solution; Ours: global convergence to a stationary point for general nonconvex objectives. (4) [Wu]: uses two more hyper-parameters than ours; (5) Ours: employs Nesterov's momentum for better performance; (6) [Wu]: uses all-to-all broadcast (which may involve large network traffic and idle time); Ours: uses parameter-server architecture.

Response to Reviewer 2

"**fairly original**": Ours is the first that studies gradient compression with Nesterov's momentum in parameter-server, and shows theoretical guarantees. [11,14,20] only heuristically consider momentum, and [4] uses exponential moving average momentum without convergence analysis. [1,2,9,16,17] only study gradient compression with vanilla SGD.

"**averaged over only 3 repetitions**": Standard deviation is already very small. We will add more in the final version.

"**binary tree allreduce and ring allreduce**": Compression at server can be implemented between reduce and broadcast steps in tree allreduce, or between reduce-scatter and allgather steps in ring allreduce. However, tree allreduce and ring allreduce require repeated gradient aggregations, and compressed gradients cannot be directly summed without first decompressing. Hence, heavy overheads may be incurred. Also, sparsity of aggregated sparse gradients may decrease rapidly during allreduce, and increases communication costs. Moreover, compression after summing up decompressed gradients at intermediate node in an allreduce step incurs further accuracy loss.

"**step sizes are also being rescaled to η_{t-1}/η_t**": Only stepsize for error $e_{t,i}$ is rescaled, not that for $g_{t,i}$.

Response to Reviewer 3

"**layerwise compression has been proposed by AdaComp**": Adacomp sparsifies each gradient block by heuristic thresholding and then performs ternary quantization, which requires additional pass to compute maximum gradient values and encode/decoding gradient indices (hefty compression overhead), Moreover, it uses all-to-all broadcast (with heavy network traffic and idle time). The proposed method is a sign-based quantization (easy parallelization and cheap compression) with block-level scaling. It also has provable theoretical guarantees.

"**novelty is limited**": Ours is the first work that studies distributed SGD with Nesterov's momentum and two-way compression. Strong convergence results are provided, showing a linear speedup of using M workers.

"**why [22] is cited?**": That sentence in the introduction is to demonstrate success of deep learning in various applications.

"**cite the following**" Some of them are not very related. Povey et al. (2014) and Chen & Huo (2016) do not consider gradient compression. Chen et al. (2016) is on synchronous SGD, and we have cited the classic papers [6,11,23].

"**all_reduce can be used instead** ": Please see our reply to Reviewer 2. Also, we use allreduce for SGDM (line 246).

"**"gradients in a deep network typically have similar magnitudes in each layer"... provide experimental evidence**": Experimental evidence is indeed provided in Figure 1(a) and discussed in lines 181-182.

"**Dist-EF-BlockSGD should be compared with Dist-EF-SGD**": Dist-EF-SGD is a general algorithm that allows arbitrary compressors satisfying Definition 1. Thus, Dist-EF-BlockSGD is a particular instantiation of Dist-EF-SGD. To see the effect of blockwise compressor, we run extra experiments using a ResNet110 on CIFAR-100 with mini-batch size 16 per worker. Dist-EF-BlockSGD improves average test accuracy of non-block version from 74.7% to 75.0%.

"**benefit little from momentum**": This depends on mini-batch size. A larger mini-batch means smaller variance σ^2, so $(F(x_0) - F_*)$ in the bound of dist-EF-SGDM (lines 211-212) is more dominant. Use of momentum ($\mu > 0$) is then beneficial. Figure 2 shows that with mini-batch size of 16 or 32 (per worker), momentum methods are faster, particularly before epoch 100. At epoch 100, the learning reate is reduced (line 347), and the difference is less obvious.