# Trustworthy Efficient Communication for Distributed Learning using LQ-SGD Algorithm

Hongyang Li, Lincen Bai*, Caesar Wu, Mohammed Chadli, Said Mammar, Pascal Bouvry

***Abstract*—We propose LQ-SGD (Low-Rank Quantized Stochastic Gradient Descent), an efficient communication gradient compression algorithm designed for distributed training. LQ-SGD further develops on the basis of PowerSGD by incorporating the low-rank approximation and log-quantization techniques, which drastically reduce the communication overhead, while still ensuring the convergence speed of training and model accuracy. In addition, LQ-SGD and other compression-based methods show stronger resistance to gradient inversion than traditional SGD, providing a more robust and efficient optimization path for distributed learning systems.**

***Index Terms*—distributed Learning, gradient compression, low-rank approximation, logarithmic quantization, communication efficiency, gradient inversion, trustworthiness, stochastic gradient descent**

## I. Introduction

With the rapid development of learning models, distributed training has become a fundamental approach to improving model performance and scalability. However, these distributed training systems typically rely on numerous compute nodes working collaboratively, where the synchronization of model parameters and gradients introduces significant communication overhead. As model sizes continue to increase and the number of nodes expands, communication overhead has become the primary bottleneck. In large-scale models, communication time even surpasses computation time, severely degrading the overall efficiency of the training process [3], [4], [6], [14], [17]. In addition to efficiency concerns, the trustworthiness and data privacy risks associated with distributed learning are also becoming increasingly critical [5], [11], [12]. Frequent exchanges of gradient information may inadvertently expose sensitive training data, making distributed learning systems vulnerable to privacy attacks.

To reduce communication bottlenecks, a variety of gradient compression techniques have been proposed [2], [13], [14], [16], [19], [20]. Among them, PowerSGD leverages low-rank approximation to significantly reduce communication overhead while maintaining competitive model performance, and has become one of the state-of-the-art solutions for distributed optimization. However, existing methods still face limitations in terms of compression efficiency.

To address these issues, we propose a novel gradient compression algorithm for distributed learning systems, namely **LQ-SGD**. LQ-SGD builds upon the PowerSGD [20] framework by introducing a logarithmic gradient quantization mechanism, combining low-rank approximation with quantization strategies. This design further reduces communication overhead while effectively maintaining convergence speed and model accuracy. Here:

1) We propose LQ-SGD, an innovative algorithm to further reduce the communication cost by incorporating log-quantization techniques into PowerSGD.
2) We evaluate the convergence performance and communication efficiency of LQ-SGD on multiple standard datasets (e.g., MNIST, CIFAR-10/100).
3) We explore the impact of gradient compression on model trustworthiness and that possesses stronger robustness in defending against gradient inversion attacks.

## II. Related Work

### A. Communication Bottleneck in Distributed Training

In large-scale systems, communication time often surpasses computation time, becoming the primary bottleneck that limits overall training efficiency [6]. For example, in training a 1.3-billion parameter model, communication accounts for 50% to 90% of the total training time as the system scales [3]. This severe imbalance between computation and communication makes it difficult to fully utilize distributed computational resources. Reducing communication costs is therefore crucial for enabling efficient and scalable distributed learning. This motivates the development of communication-efficient techniques.

### B. Gradient Compression Techniques

To reduce communication costs in distributed learning, three mainstream gradient compression techniques have been widely adopted: quantization, sparsification, and low-rank approximation. Quantization methods, such as 1-bit SGD [16] and QSGD [2], reduce the precision of gradients to lower the communication bandwidth. Sparsification techniques, including Top-K selection [1] and Deep Gradient Compression (DGC) [14], transmit only the some significant gradients. Low-rank approximation methods, such as PowerSGD [20], compress gradients by approximating them with low-rank matrices, achieving substantial communication reduction in structured layers.

## III. Preliminaries

### A. Distributed SGD Basics

Distributed Stochastic Gradient Descent (Distributed SGD) is a widely used framework for accelerating large-scale model

[1]University of Luxembourg, Luxembourg. {hongyang.li, caesar.wu, pascal.bouvry}@uni.lu
[2]University of Paris-Saclay, France. {lincen.bai, said.mammar, mohammed.chadli}@univ-evry.fr
*Corresponding author.

training. In this paradigm, the global training dataset is partitioned into subsets, which are distributed across $N$ compute nodes (or workers). At each iteration $t$, each worker $i$ independently computes its local gradient $\mathbf{g}_t^{(i)}$ on its data shard. These local gradients are then aggregated across all workers—typically via an *All-Reduce* operation—to obtain the global gradient $\mathbf{g}_t$:

$$\mathbf{g}_t = \frac{1}{N}\sum_{i=1}^{N} \mathbf{g}_t^{(i)}. \tag{1}$$

After aggregation, the model parameters $\mathbf{w}_t$ are updated synchronously using the global gradient:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \mathbf{g}_t, \tag{2}$$

where $\eta$ is the learning rate.

## B. PowerSGD

Unlike quantization or sparsification methods that compress gradients element-wise, PowerSGD [20] exploits the low-rank structure of gradient matrices—especially in large neural networks—to achieve significant communication reduction.

In typical deep learning models, such as convolutional and fully-connected layers, the gradients of weight matrices often have a high degree of redundancy. PowerSGD approximates these high-dimensional gradients using low-rank matrix factorization before communication. Specifically, given a gradient matrix $G \in \mathbb{R}^{m\times n}$, PowerSGD approximates $G$ as a product of two low-rank matrices:

$$G \approx PQ^{\top} \tag{3}$$

where $P \in \mathbb{R}^{m\times r}$ and $Q \in \mathbb{R}^{n\times r}$, and $r \ll \min(m, n)$ is the rank of the approximation.

After the step of aggregation in distributed learning , $P$ and $Q$ are used to reconstruct an approximation of $G$. Since $P$ and $Q$ are much smaller than $G$ (when $r$ is small), this approach drastically reduces the amount of data transmitted during synchronization. PowerSGD achieves compression rates proportional to $r(m+n)/(mn)$, making it highly scalable for layers with large weight matrices.

## C. Model Trustworthiness

In distributed and federated learning, the frequent exchange of gradients exposes models to significant privacy risks. One of the most critical threats is the **Gradient Inversion Attack** (GIA), which aims to reconstruct the original training data from shared gradients.

Given the gradient $\mathbf{g}_t$ received by the server or other participants, the attacker reconstructs dummy inputs $\hat{\mathbf{x}}$ by minimizing the cosine similarity loss between the gradient $\mathbf{g}_t$, along with a regularization term [5]:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left[1 - \frac{\langle \nabla_{\mathbf{w}}\mathcal{L}(f(\mathbf{x};\mathbf{w}), y), \mathbf{g}_t\rangle}{\|\nabla_{\mathbf{w}}\mathcal{L}(f(\mathbf{x};\mathbf{w}), y)\|_2 \, \|\mathbf{g}_t\|_2} + R(\mathbf{x})\right], \tag{4}$$

where $\mathcal{L}$ is the loss function, $f(\cdot;\mathbf{w})$ is the model, $y$ is an optional label (depending on the attack setting), and $R(\mathbf{x})$ is a regularization term, such as total variation (TV), to improve reconstruction quality.

## IV. Proposed Method

By incorporating log-quantization techniques, we present here the proposed LQ-SGD algorithm, which is an enhanced version of PowerSGD [20].

While PowerSGD effectively reduces the communication cost by compressing the gradient matrix $\mathbf{G}_t \in \mathbb{R}^{n\times m}$ into two low-rank factor matrices $\mathbf{P}_t \in \mathbb{R}^{n\times r}$ and $\mathbf{Q}_t \in \mathbb{R}^{m\times r}$, the size of these factor matrices remains substantial, especially when a higher approximation rank $r$ is needed to maintain convergence. To further reduce the communication overhead, we introduce a logarithmic quantization strategy. This method achieves efficient compression with little impact on the accuracy of numerical representations by assigning higher precision to smaller values. Meanwhile, LQ-SGD, like PowerSGD, incorporates an error feedback mechanism and a hot-start initialization strategy, thus guaranteeing stable convergence of the algorithm even when high-intensity quantization is employed. The algorithm procedure of LQ-SGD has be summarized in **Algorithm 1**.

## A. Logarithmic Quantization of $\mathbf{P}_t$ and $\mathbf{Q}_t$

Since heavy-tailed behavior is commonly observed in practical scenarios [8], it is essential to design quantization schemes that account for such characteristics. In the presence of heavy-tailed or power-law distributed gradients, nonuniform quantization methods are particularly effective. These schemes allocate higher precision to smaller gradient values—where the majority of informative signals are concentrated—while more aggressively compressing large outliers, thereby reducing redundancy and improving overall efficiency.

We adopt the following quantization function:

$$q(x) = \mathrm{sign}(x)\cdot\frac{\log(1+\alpha|x|)}{\log(1+\alpha)}, \quad \alpha > 0, \tag{5}$$

where $\alpha$ controls the curvature of the logarithmic mapping. This design prioritizes accuracy near the zero value, since these small amplitudes form a major part of the typical gradient distribution [2], [14].

The matrices after quantizing $\mathbf{P}_{\text{quant}}$ and $\mathbf{Q}_{\text{quant}}$ are sent across workers via the All-Reduce operation. Upon reception, each worker will dequantize to reconstruct the original information:

$$x = \mathrm{sign}(q(x))\cdot\frac{(1+\alpha)^{|q(x)|}-1}{\alpha}. \tag{6}$$

In practical deployments, we adopt a separable symbol encoding scheme. The normalized quantized values $|q(x)| \in [0,1]$ are mapped to a discrete set of $2^b$ uniformly spaced bins, determined by a $b$-bit representation. Under this design, each quantized scalar requires only $b$ bits for transmission; in our experiments, we typically set $b = 8$. To implement this efficiently, we discretize the continuous log-based function via precomputed quantization levels and nearest-neighbor matching. This strategy retains the precision structure of logarithmic mapping and supports importance-aware compression [15].

## B. Error Feedback Mechanism

Through the error feedback mechanism, LQ-SGD compensates the signal distortion caused by low-rank approximation and quantization to a certain extent. After the gradient approximation reconstruction is completed :

$$\hat{\mathbf{G}}_t = \mathbf{P}_t \mathbf{Q}_t^\top, \tag{7}$$

the residual error is computed:

$$\mathbf{E}_t = \mathbf{G}'_t - \hat{\mathbf{G}}_t. \tag{8}$$

This error is accumulated and added to the gradient in the next iteration:

$$\mathbf{G}'_{t+1} = \mathbf{G}_{t+1} + \mathbf{E}_t. \tag{9}$$

This technique is often referred to as an error feedback mechanism , such as EF-SGD [9], which establishes the corresponding theoretical framework. Some low-rank compression methods in recent years, such as PowerSGD [20], also introduce error feedback in their design to compensate for the errors caused by the low-rank approximation.

In LQ-SGD, we adopt this mechanism to ensure that the information lost due to low-rank factorization and logarithmic quantization is gradually recovered, preserving the convergence properties.

**Algorithm 1** Low-Rank Quantized Stochastic Gradient Descent (LQ-SGD)

```
Require: Gradient matrix $\mathbf{G}_t \in \mathbb{R}^{n\times m}$, rank $r$, quantization
    bits $b_p, b_q$, logarithmic scale $\alpha$, learning rate $\eta$
Ensure: Updated model parameters $\mathbf{w}_{t+1}$
 1: Initialize error matrix $\mathbf{E}_0 \leftarrow \mathbf{0}$
 2: Initialize $\mathbf{Q}_0 \sim \mathcal{N}(0,1)$
 3: for each iteration $t = 1, 2, \ldots$ do
 4:     $\mathbf{G}'_t \leftarrow \mathbf{G}_t + \mathbf{E}_{t-1}$        ▷ Error feedback compensation
 5:     if $t > 1$ then
 6:         $\mathbf{Q}_t \leftarrow \mathbf{Q}_{t-1}$        ▷ Warm-start from previous Q
 7:     else
 8:         $\mathbf{Q}_t \sim \mathcal{N}(0,1)$        ▷ Random initialization
 9:     end if
10:     $\mathbf{P}_t \leftarrow \mathbf{G}'_t \mathbf{Q}_t$
11:     $\mathbf{P}_t \leftarrow$ Orthonormalize($\mathbf{P}_t$)        ▷ Power iteration step
12:     $\mathbf{P}_{\text{quant}} \leftarrow$ LogQuantize($\mathbf{P}_t, b_p, \alpha$)
13:     All-Reduce $\mathbf{P}_{\text{quant}}$
14:     $\mathbf{P}_t \leftarrow$ LogDequantize($\mathbf{P}_{\text{quant}}, b_p, \alpha$)
15:     $\mathbf{Q}_t \leftarrow \mathbf{G}'^\top_t \mathbf{P}_t$
16:     $\mathbf{Q}_{\text{quant}} \leftarrow$ LogQuantize($\mathbf{Q}_t, b_q, \alpha$)
17:     All-Reduce $\mathbf{Q}_{\text{quant}}$
18:     $\mathbf{Q}_t \leftarrow$ LogDequantize($\mathbf{Q}_{\text{quant}}, b_q, \alpha$)
19:     $\hat{\mathbf{G}}_t \leftarrow \mathbf{P}_t \mathbf{Q}_t^\top$        ▷ Reconstruct gradient
20:     $\mathbf{E}_t \leftarrow \mathbf{G}'_t - \hat{\mathbf{G}}_t$        ▷ Update error feedback
21:     $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta \hat{\mathbf{G}}_t$        ▷ Model update
22: end for
```

## C. Communication and Computational Complexity

Compared to PowerSGD, which requires transmitting $r(n+m)$ full-precision floating-point values (typically 32 bits each) per iteration, LQ-SGD reduces the communication cost to $r(n+m) \times b$ bits per iteration, where $b$ is the number of quantization bits used in $\mathbf{P}_{\text{quant}}$ and $\mathbf{Q}_{\text{quant}}$. This achieves a compression ratio of $32/b$ relative to PowerSGD.

The extra cost of quantization / de-quantization in LQ-SGD is only $\mathcal{O}\big(r(n+m)\big)$, whereas the two matrix–matrix products in PowerSGD dominate at $\mathcal{O}(nmr)$, the added computation is practically negligible on modern GPUs.

## V. Experimental Results

In this section, we evaluate the performance and trustworthiness of our proposed LQ-SGD algorithm in distributed training tasks.

### A. Experimental Setup

The experiments are conducted on a distributed cluster comprising 5 worker nodes, each equipped with an NVIDIA RTX 4090 GPU, and 1 central node responsible for gradient aggregation. The system follows a parameter server-like architecture to simulate realistic distributed training conditions.

We evaluate the proposed **LQ-SGD** algorithm against three baselines: Original SGD, PowerSGD [20], and TopK-SGD [18]. For TopK-SGD, the sparsity ratio is adjusted to achieve compression rates comparable to those of the other methods. Experiments are performed on three widely used image classification datasets: CIFAR-10, CIFAR-100, and MNIST. For the model architecture, we employ ResNet-18 [7], which offers a good balance between model complexity and computational efficiency in distributed environments. All algorithms are implemented within the same distributed training pipeline to ensure a fair and consistent comparison.

The evaluation covers the following metrics:

- **Accuracy**: Top-1 test accuracy on the validation set.
- **Communication Cost**: The size (in MB) of gradient data transmitted per-epoch.
- **Computation Time**: The per-epoch computation time measured, excluding communication time (in seconds).
- **Structural Similarity Index Measure (SSIM)**: To evaluate privacy leakage resistance under GIA. Lower SSIM scores indicate better protection against data reconstruction from shared gradients.

### B. Performance Evaluation: Communication Cost, Computation Time, and Accuracy

TABLE I: Performance Comparison on CIFAR-10

| Method | Accuracy | Size (x1) | Computing Time |
|---|---|---|---|
| Original SGD | 0.9432 | 3325 (x1108.3) | 2.2937 |
| Power SGD (Rank 1) | 0.9451 | 14 (x4.7) | 2.3359 |
| TopK SGD[†] | 0.8821 | 14 (x4.7) | 3.6173 |
| **LQ-SGD (Rank 1)** | **0.9290** | **3 (x1.0)** | 2.5714 |

TABLE II: Performance Comparison on CIFAR-100

| Method | Accuracy | Size (x1) | Computing Time |
|---|---|---|---|
| Original SGD | 0.7445 | 3339 (x1113.0) | 2.2882 |
| Power SGD (Rank 1) | 0.7404 | 14 (x4.7) | 2.1588 |
| TopK SGD[†] | 0.6070 | 14 (x4.7) | 3.5946 |
| **LQ-SGD (Rank 1)** | **0.7181** | **3 (x1.0)** | 2.5631 |

TABLE III: Performance Comparison on MNIST

| Method | Accuracy | Size (x1) | Computing Time |
|---|---|---|---|
| Original SGD | 0.9940 | 3964 (x991.0) | 2.4909 |
| Power SGD (Rank 1) | 0.9929 | 16 (x4.0) | 2.3617 |
| TopK SGD$^\dagger$ | 0.9940 | 16 (x4.0) | 3.9826 |
| **LQ-SGD (Rank 1)** | **0.9939** | **4 (x1.0)** | 2.8442 |

$^\dagger$ TopK SGD achieves an effective compression ratio aligned with Power SGD (Rank 1) and LQ-SGD (Rank 1) in this experiment.

Table I, II and III summarizes the accuracy, communication cost, and computation time of different algorithms on CIFAR-10, CIFAR-100, and MNIST at compression rank 1. LQ-SGD achieves substantial communication cost reduction compared to both PowerSGD and TopK-SGD. For example, on CIFAR-10, LQ-SGD reduces the communication volume by over 75% relative to PowerSGD (3 MB vs. 14 MB), and by more than 99.9% compared to uncompressed SGD (3 MB vs. 3325 MB). Similar reductions are observed on CIFAR-100 and MNIST, demonstrating LQ-SGD's ability to significantly lower bandwidth requirements. For comparison, TopK-SGD uses a sparsification strategy by transmitting only a fixed percentage of the largest gradient entries, achieving an effective compression ratio similar to PowerSGD at rank 1 (approximately 4.7× compression in our experiments). Despite its aggressive compression, LQ-SGD maintains high accuracy. On CIFAR-10, it achieves 92.9% Top-1 accuracy, only slightly lower than PowerSGD (94.5%) and outperforming TopK-SGD (88.2%). On CIFAR-100 and MNIST, LQ-SGD shows similarly competitive accuracy while offering greater communication efficiency. LQ-SGD introduces only a marginal increase in computation time compared to PowerSGD, with the overhead from logarithmic quantization being minimal, and maintains reasonable per-epoch computation times across all datasets, remaining within 2.5 to 2.8 seconds per epoch on CIFAR-10, CIFAR-100, and MNIST.

**Overall Efficiency.** In large-scale distributed training, communication time often dominates, accounting for over 50% of the total training time [3]. By significantly reducing communication volume, LQ-SGD can substantially lower total training time with negligible impact from the slightly increased computation time. Figures 1, 2, and 3 further demonstrate LQ-SGD's convergence behavior. It achieves convergence rates comparable to PowerSGD and TopK-SGD, stabilizing within 150 epochs across all datasets at different compression level.

***Performance on ImageNet***: Figure 4 presents the Top-1 accuracy and cross-entropy loss of LQ-SGD on ImageNet. With Rank 7, LQ-SGD matches the accuracy and convergence of OriginalSGD, reaching 75% Top-1 accuracy after 300 epochs. Rank 2 maintains stable convergence with slightly reduced accuracy, while Rank 1 still achieves reasonable performance despite its aggressive compression.

These results confirm that LQ-SGD provides an excellent trade-off between communication cost and model accuracy, making it highly practical for bandwidth-constrained, large-scale distributed training.

### *C. Trustworthiness Evaluation: Resistance to Gradient Inversion Attacks*

We evaluate the privacy-preserving capabilities of LQ-SGD by conducting gradient GIA on CIFAR-10, CIFAR-100, and MNIST. To quantify the similarity between reconstructed and original images. Figures 5a–5c present the SSIM results across different compression ranks for LQ-SGD, PowerSGD, TopK-SGD, and vanilla SGD.

Across all datasets, compression-based methods, including LQ-SGD, consistently yield lower SSIM scores compared to original SGD, indicating enhanced resistance to gradient inversion attacks. Among them, LQ-SGD achieves a favorable balance between privacy protection and model accuracy. Notably, while TopK-SGD obtains the lowest SSIM at high compression ratios, it often does so at the cost of substantial accuracy degradation. In contrast, LQ-SGD maintains strong model performance while offering meaningful privacy benefits.

These results confirm that gradient compression can effectively mitigate privacy leakage in distributed learning, and highlight LQ-SGD as a trustworthy and efficient choice for privacy-sensitive applications. This observation is in line with our earlier findings that gradient compression naturally enhances robustness against GIA [10].

## VI. Discussion

LQ-SGD effectively balances communication efficiency with model performance and ensures trustworthiness. However, it has two main limitations: (1) our evaluation is limited to image classification tasks, and its scalability to diverse data has yet to be tested; (2) we have not evaluated LQ-SGD on very large architectures (e.g., transformer-based LLMs).

## VII. Conclusion

In this paper, we presented LQ-SGD and highlight it as a trustworthy and communication-efficient solution for distributed learning. In future work, we plan to extend LQ-SGD to diverse data sets and much larger network sizes.

## References

[1] Alham Fikri Aji and Kenneth Heafield. Sparse communication for distributed gradient descent. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 440–445, 2017.

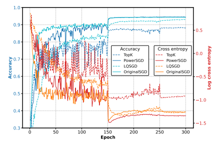
[2] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1709–1720, 2017.

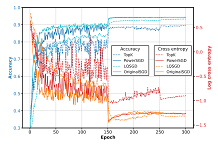
[3] Quentin Anthony, Benjamin Michalowicz, Jacob Hatef, Lang Xu, Mustafa Abduljabbai, Aamir Shafi, Hari Subramoni, and Dhabaleswar K Panda. Demystifying the communication characteristics for distributed transformer models. In *2024 IEEE Symposium on High-Performance Interconnects (HOTI)*, pages 57–65. IEEE, 2024.

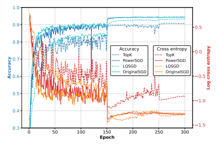
[4] Yang Chen, Min Chen, Yanzhi Zhang, Liang Yang, and Victor CM Leung. Communication-efficient distributed learning: A comprehensive survey. *IEEE Transactions on Parallel and Distributed Systems*, 2023.

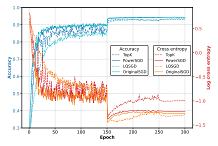
[5] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in neural information processing systems*, 33:16937–16947, 2020.

[6] Priya Goyal, Piotr Dollar, Ross Girshick, et al. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.


(a) CIFAR-10, Rank 1


(b) CIFAR-10, Rank 2


(c) CIFAR-10, Rank 4


(d) CIFAR-10, Rank 30

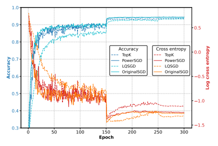
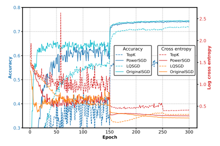
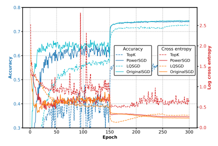
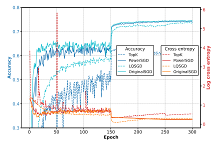
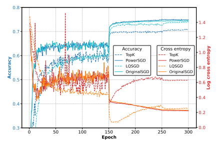

(e) CIFAR-10, Rank 50

Fig. 1: Results for CIFAR-10 dataset with different compression ranks.


(a) CIFAR-100, Rank 1


(b) CIFAR-100, Rank 2


(c) CIFAR-100, Rank 4


(d) CIFAR-100, Rank 30

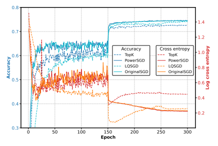
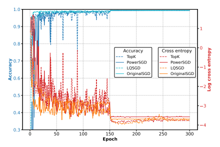
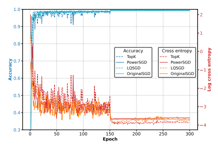
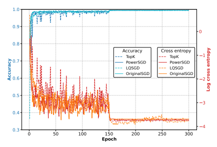
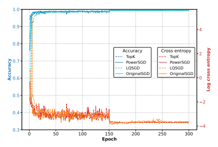

(e) CIFAR-100, Rank 50

Fig. 2: Results for CIFAR-100 dataset with different compression ranks.


(a) MNIST, Rank 1


(b) MNIST, Rank 2


(c) MNIST, Rank 4


(d) MNIST, Rank 30

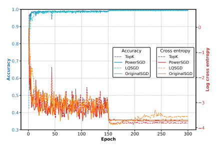
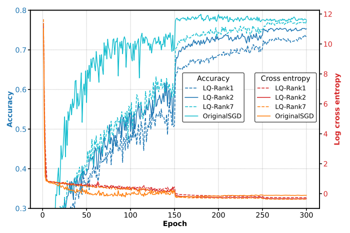

(e) MNIST, Rank 50

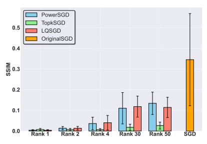
Fig. 3: Results for MNIST dataset with different compression ranks.

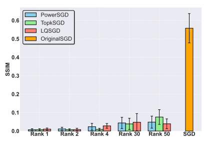
†Note: Top K-SGD does not have a specific rank; the 'Rank' values are provided for consistency in compression rate comparisons.



(a) ImageNet

Fig. 4: Demonstrates stable convergence up to 300 epochs.


(a) CIFAR-10


(b) CIFAR-100

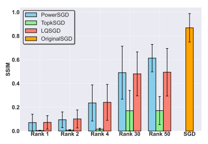

(c) MNIST

Fig. 5: SSIM scores under different compression ranks.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 770–778, 2016.

[8] Liam Hodgkinson, Zhichao Wang, and Michael W Mahoney. Models of heavy-tailed mechanistic universality. *arXiv preprint arXiv:2506.03470*, 2025.

[9] Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian U Stich, and Martin Jaggi. Error feedback fixes signsgd and other gradient compression schemes. In *International Conference on Machine Learning (ICML)*, pages 3252–3261, 2019.

[10] Hongyang Li, Caesar Wu, Mohammed Chadli, Said Mammar, and Pascal Bouvry. Trustworthiness of stochastic gradient descent in distributed learning. *arXiv preprint arXiv:2410.21491*, 2024.

[11] Hongyang Li, Caesar Wu, Mohammed Chadli, Said Mammar, and Pascal Bouvry. Lightweight trustworthy distributed clustering. *arXiv preprint arXiv:2504.10109*, 2025.

[12] Qiongxiu Li, Jaron Skovsted Gundersen, Milan Lopuhaä-Zwakenberg, and Richard Heusdens. Adaptive differentially quantized subspace perturbation (adqsp): A unified framework for privacy-preserving distributed average consensus. *IEEE Transactions on Information Forensics and Security*, 19:1780–1793, 2023.

[13] Qiongxiu Li, Richard Heusdens, and Mads Græsbøll Christensen. Communication efficient privacy-preserving distributed optimization using adaptive differential quantization. *Signal Processing*, 194:108456, 2022.

[14] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. In *International Conference on Learning Representations (ICLR)*, 2018.

[15] Ali Ramezani-Kebrya, Fartash Faghri, Ilya Markov, Vitalii Aksenov, Dan Alistarh, and Daniel M Roy. Provably communication-efficient data-parallel sgd via nonuniform quantization.

[16] Frank Seide, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns. In *INTERSPEECH*, pages 1058–1062, 2014.

[17] Christopher J Shallue, Jaehoon Lee, Joseph Antognini, Jascha Sohl-Dickstein, Roy Frostig, and George E Dahl. Measuring the effects of data parallelism on neural network training. *arXiv preprint arXiv:1811.03600*, 2018.

[18] Shaohuai Shi, Qiang Wang, Kaiyong Zhao, Zhenheng Tang, Yuxin Wang, Xiang Huang, and Xiaowen Chu. A distributed synchronous sgd algorithm with global top-k sparsification for low bandwidth networks. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 2238–2247. IEEE, 2019.

[19] Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified sgd with memory. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4447–4458, 2018.

[20] Thijs Vogels, Sai Praneeth Karimireddy, and Martin Jaggi. Powersgd: Practical low-rank gradient compression for distributed optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 14236–14246, 2019.