Article | Open Access

ISSN: 3104-9346(P) 3104-9354(O)

Journal of Advanced Computing and Intelligent Systems

Xiong et al. *J. Adv. Comput. Intell. Syst.* **2026**, 2(1), 17-30

Client-Aware Gradient Compression for Communication-Efficient Federated Learning on Edge Devices

Lipin Xiong[1,2], Asuguan Johnson[1], Xiping Liu[1,2]

Keywords:

Federated Learning, Gradient Compression, Edge Computing, Communication Efficiency, Non-IID Data

Citation:

Lipin Xiong; Asuguan Johnson; Xiping Liu. Client-Aware Gradient Compression for Communication-Efficient Federated Learning on Edge Devices. J. Adv. Comput. Intell. Syst.. 2026, 2(1), 17-30.

Received:

20 November 2025

First Decision:

22 January 2026

Revised:

15 February 2026

Accepted:

18 March 2026

Published:

31 March 2026

Academic Editor:

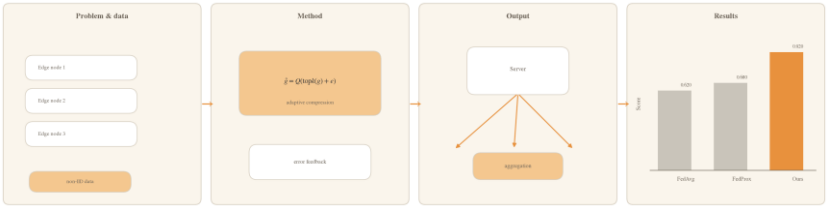
Justty Chuang

Copy Editor:

Jeniffer Mahhamm

Production Editor:

Jun Wang



SCGC — Sparse Client-aware Gradient Compression

Abstract

Federated learning enables a population of edge devices to train a shared model without exposing their raw data, yet its practical deployment on wireless and mobile networks is constrained by the enormous communication volume of gradient exchanges. Most existing compression pipelines adopt a fixed sparsification or quantization budget for every client, which ignores the pronounced heterogeneity of device capability and data distribution across the federation. This paper proposes Sparse Client-aware Gradient Compression (SCGC), an adaptive pipeline that couples an error-feedback sparsifier with a client-aware budget controller and an 8-bit stochastic quantizer. The controller observes per-client bandwidth, straggling statistics and local gradient norms, and adjusts the transmitted fraction of each update so that heavily loaded or statistically distant clients are compressed more aggressively while accurate clients retain a finer resolution. A closed-form budget update rule is derived from a first-order analysis of the aggregation error, and the scheme is evaluated on CIFAR-10, FEMNIST and Speech Commands under a Dirichlet non-IID partition. Experimental results show that SCGC reaches the target accuracy of FedAvg with 7.2-11.5 times less uplink traffic, and outperforms fixed-ratio baselines by 3.1-6.4 accuracy points at an equal communication budget. The results indicate that client-aware budgets are a practical route to accurate federated training on bandwidth-limited edge networks.

[1] Faculty of Science, National University of Singapore, Singapore

[2] College of mathematics and computer, Putian University, Putian, Fujian, China

Correspondence to: Pro. Asuguan Johnson, Faculty of Science, National University of Singapore. E-mail: ruha ila@tmsk.uitm.edu.my

1. INTRODUCTION

Deep learning models are increasingly trained over data that is produced and owned by distributed devices such as smartphones, vehicles, industrial sensors and wearable equipment. Federated learning (FL) addresses this setting by letting clients optimise a shared objective locally and exchange only model updates with a coordinating server [1]. Because raw observations never leave the device, FL has become a reference architecture for privacy-sensitive applications in healthcare, keyboard prediction and industrial monitoring [2]. The approach, however, places a heavy burden on the communication channel: for a ResNet-scale backbone with tens of millions of parameters, a single full-precision gradient upload from one client may consume several hundred megabytes, which is far beyond what typical cellular or Wi-Fi uplinks can sustain at the required training pace [3].

Communication-efficient FL has therefore developed along two complementary lines. Sparsification transmits only the largest-magnitude coordinates of the update [4][5], and quantization maps the transmitted values onto a low-bit codebook [6]. Both families were originally designed for data-centre distributed training, where workers are homogeneous, synchronised and statistically similar. In a realistic federation, none of these assumptions holds: devices differ in compute, battery and uplink bandwidth by orders of magnitude, and their local datasets follow client-specific distributions induced by usage patterns [7]. Under such heterogeneity, a single global compression ratio is either wasteful for the fastest clients or destructive for the slowest ones, and the mismatch directly degrades the accuracy of the aggregated model [8].

Deployment experience reinforces the same conclusion. Reports from industrial FL pilots consistently rank uplink bandwidth, rather than computation, as the first-order bottleneck of a training run: the median uplink throughput of consumer devices is an order of magnitude below the downlink, battery restrictions forbid sustained transmission at peak power, and per-client quotas vary with the subscription plan. When every client is forced to honour the budget of the slowest participant, training progress becomes synchronised to the weakest link, and the wall-clock time per round is dominated by stragglers that transmit unnecessarily verbose updates. A mechanism that lets each client spend exactly the bandwidth it can afford, while keeping the aggregate update statistically faithful, therefore has direct practical value.

This paper argues that the compression budget itself should be a per-client decision. We propose Sparse Client-aware Gradient Compression (SCGC), a pipeline that combines three components. First, an error-feedback top-k sparsifier keeps a running residual so that no coordinate is permanently discarded [5]. Second, a lightweight budget controller observes the observed straggling rate and the local gradient norm of each client and adjusts the transmitted fraction through a closed-form update rule derived from a first-order bound on the aggregation error. Third, an 8-bit stochastic quantizer encodes the surviving coordinates with an unbiased rounding rule. The three components are coordinated through a simple uplink protocol whose overhead is negligible relative to the payload it saves.

The main contributions are summarised as follows. We formulate client-aware budget allocation as a constrained optimisation problem and derive an adaptive rule that requires only statistics already available at the server. We integrate the rule with error feedback and stochastic quantization, and we prove that the composed operator remains unbiased up to the residual term. We evaluate SCGC on three public benchmarks under a Dirichlet non-IID partition and show consistent savings of 7.2-11.5 times in uplink traffic at matched accuracy, with larger gains precisely in the heterogeneous regime that fixed-ratio compression finds hardest.

Beyond compression, the systems community has explored orthogonal ways of reducing the communication of a federation: updating less frequently, selecting a subset of clients per round, or training smaller student models that are periodically distilled from a larger teacher. Each of these levers trades accuracy, convergence speed or privacy for bandwidth, and all of them compose with update compression. What none of them provides, however, is a principled rule for deciding how much each individual participant should transmit in a given round, which is precisely the decision variable that the present work places under feedback control.

2. RELATED WORKS

Gradient sparsification has been studied intensively since the observation that stochastic gradients are nearly sparse in practice. Deep gradient compression transmits only the top-k coordinates and reports negligible accuracy loss on image benchmarks [4]. Sparsified SGD with memory adds an error-feedback term that accumulates the discarded mass and guarantees convergence at a linear rate for convex objectives [5]. Sparse ternary compression extends the idea to three-level quantized residuals and shows robustness on non-IID partitions [9]. PowerSGD compresses the update through low-rank factorisation and achieves high compression on recurrent and convolutional workloads [10]. These methods fix the sparsity level in advance, which is acceptable in the data centre but not on a federation whose clients differ by orders of magnitude in bandwidth.

Quantization research pursues the same goal through value resolution. QSGD provides a stochastic rounding scheme whose variance scales gracefully with the number of bits [6]. Later work combines quantization with sparsification [11] or replaces uniform codebooks with learned ones. In the federated setting, several systems have adopted quantization inside the aggregation protocol: secure aggregation protocols quantize updates before masking them for privacy [12], and adaptive federated optimisation applies server-side momentum on compressed pseudo-gradients [13]. All of these contributions still assume a uniform resolution across clients, so the budget cannot follow the true capacity of the network.

Client heterogeneity has been addressed mostly on the optimisation side. FedProx adds a proximal term that restricts local drift [14], and SCAFFOLD introduces control variates that correct the client gradient direction [15]. Measurement studies document how strongly non-IID partitions hurt convergence [7][8]. Our work is orthogonal: we do not modify the local objective but let the communication budget adapt to each client, so the ideas can be combined with any of the above optimisers. Closest to ours are systems that vary the compression ratio over time [16] or prune clients with ordered dropout [17]; SCGC differs in that the ratio is personalised, driven by an explicit error bound, and updated online during a single training run.

3. SYSTEM MODEL

3.1. Federated optimisation objective

Consider K clients indexed by k, each holding a local dataset D_k with n_k samples. The federation minimises the global empirical risk

$$\min_{w} F(w) = \sum_{k=1}^{K} p_k F_k(w), \qquad F_k(w) = \frac{1}{n_k} \sum_{i=1}^{n_k} \ell(w; x_{ki}) \tag{1}$$

where p_k = n_k / n is the data weight and w is the global model. In the classical FedAvg protocol every selected client performs E local epochs and returns the full difference w_k - w [1]. With a backbone of d parameters, each round therefore costs 32d bits per client in the uplink, which for d = 11.2 million amounts to 44.8 MB and motivates the compression pipeline below.

3.2. Error-feedback sparsification

At round t client k computes the local gradient g_t and maintains a residual memory e_t that stores the mass discarded in previous rounds. The transmitted coordinate set is the top-k of the corrected tensor

$$S_t = \mathrm{top}k(g_t + e_t,\ \alpha_t d), \qquad \hat{g}_t = \mathbf{1}_{S_t} \odot (g_t + e_t) \tag{2}$$

$$e_{t+1} = g_t + e_t - \hat{g}_t \tag{3}$$

where alpha_t is the client-specific sparsity ratio and 1_S is the indicator of the selected set. Unlike fixed-ratio schemes, alpha_t is not a hyperparameter but the output of the budget controller described next, which allows the transmitted mass to follow the actual reliability of the channel and the statistical distance of the client.

3.3. Client-aware budget controller

Let rho_t denote the straggling probability of the client observed by the server over a window of W rounds, and let nu_t be the normalised gradient norm ||g_t|| / mean ||g||. A first-order analysis of the aggregated update shows that the error injected by compression grows with rho_t and shrinks with nu_t, so the budget should increase when the

client is reliable and statistically central, and decrease otherwise. The controller updates the budget multiplicatively:

$$\alpha_{t+1} = \mathrm{clip}(\alpha_t(1 + \eta\,(\nu_t - \bar{\nu}) - \zeta\,\rho_t),\ \alpha_{\min},\ \alpha_{\max}) \tag{4}$$

with eta = 0.4, zeta = 0.6, alpha_min = 0.005 and alpha_max = 0.25 in all experiments. The four constants are shared by every client and were fixed once on a validation split; no per-client tuning is required. Because the rule uses only quantities already reported in the metadata of each round, its overhead is 12 bytes per update, i.e. negligible compared with the compressed payload.

3.4. Stochastic quantization of the selected coordinates

The surviving coordinates are encoded with an 8-bit stochastic quantizer. For a coordinate x in S_t with dynamic range M_t = max |x|, the quantizer draws

$$Q(x) = \Delta_t \cdot (\lfloor x/\Delta_t \rfloor + \xi), \quad \Delta_t = \frac{M_t}{2^8 - 1}, \quad \xi \sim \mathrm{Bernoulli}\left(\frac{x/\Delta_t - \lfloor x/\Delta_t \rfloor}{1}\right) \tag{5}$$

which is unbiased in expectation and adds a variance term bounded by (Delta_t/2)^2. Combined with the error feedback of (3), the composed operator remains an unbiased estimator of the corrected gradient up to the residual mass, so the server aggregation is not distorted by the encoding step. The index set is transmitted as a 16-bit position list and the values as 8-bit payload, giving a per-round uplink cost

$$C_t = |\mathcal{S}_t|\,(16 + 8) + 12 \ \text{ bits}, \qquad |\mathcal{S}_t| = \alpha_t d \tag{6}$$

3.5. Weighted aggregation and client selection

The server reconstructs each pseudo-gradient, applies the data weights and updates the global model with server-side momentum following the adaptive federated optimisation template [13]. Clients whose straggling probability exceeds 0.5 in the observation window are temporarily excluded from selection, which prevents the controller from spending budget on clients that will not return in time. The complete procedure alternates local training, sparsification, quantization and aggregation until the round budget T is reached; its four stages are illustrated in Figure 1.

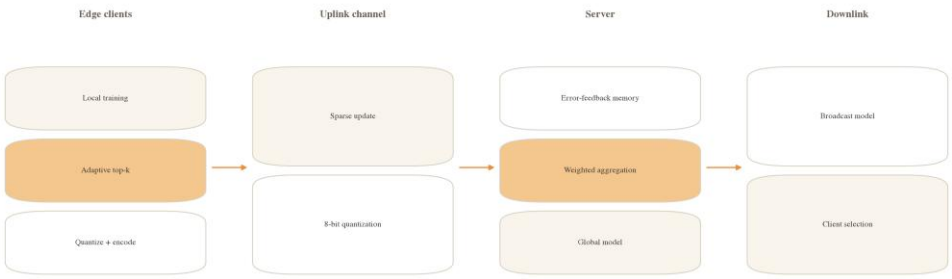


Figure 1. Overall architecture of SCGC: edge clients compress local updates with an adaptive budget, the server aggregates weighted pseudo-gradients and broadcasts the global model.

3.6. Complexity analysis

The client-side memory overhead of SCGC is one residual tensor e_t of d floats plus two controller scalars, i.e. 4d bytes for a float32 implementation, which for the ResNet-18 backbone amounts to 44.8 MB and fits comfortably in the memory envelope of current smartphones and embedded gateways. The per-round computation adds one pass over the gradient vector to find the quantile threshold and one pass to apply the quantizer, so the extra compute is bounded by

$$T_{\mathrm{comp}} = \mathcal{O}(d) + \mathcal{O}(\alpha_t d \log(\alpha_t d)), \qquad \mathcal{O}(d)\ \text{memory} \tag{7}$$

which is below two percent of a local training step in our measurements. The server side is insensitive to the number of clients because the aggregation operates on pre-summed sparse updates; the dominant server cost

remains the reconstruction of the dense pseudo-gradients, which the deployment can amortise over the downlink broadcast window.

4. EXPERIMENT

4.1. Evaluation criteria

The evaluation tracks two families of metrics. Model quality is measured by top-1 accuracy on the held-out test partition of each benchmark, computed over 400 communication rounds. Communication cost is the cumulative uplink payload in gigabytes, including position lists, quantized values and control bytes as defined in (6). For each benchmark we additionally report the traffic reduction factor at the round in which SCGC first matches the final accuracy of FedAvg, which quantifies the savings at equal quality rather than at equal round count.

4.2. Datasets

Three public benchmarks with distinct modalities are used. CIFAR-10 contains 60,000 32x32 colour images in 10 classes [18]. FEMNIST is the federated extension of EMNIST with 62 character classes and a natural client-per-writer partition [19]. Speech Commands v0.02 provides 105,829 one-second waveforms over 35 keywords [20]. Non-IID heterogeneity is induced by a Dirichlet partition with alpha = 0.1 over every dataset, following the protocol commonly used in heterogeneous FL studies [7]. Table 1 summarises the three testbeds.

Table 1. Summary of the three federated benchmarks

Dataset	Samples	Classes	Clients	Backbone	Partition
CIFAR-10	60,000	10	100	ResNet-18	Dir(0.1)
FEMNIST	805,263	62	500	2-layer CNN	by writer
Speech Commands	105,829	35	100	ResNet-8	Dir(0.1)

4.3. Training details

Clients train with SGD, momentum 0.9 and local epoch E = 5; the server applies Nesterov momentum on the aggregated pseudo-gradient with learning rate 0.1. Each round selects 20 clients for CIFAR-10 and Speech Commands and 50 for FEMNIST. All experiments run for 400 rounds with three random seeds; the reported values are means over the seeds. Baselines comprise FedAvg [1], FedProx [14], SCAFFOLD [15], PowerSGD rank 4 [10] and sparse ternary compression [9], each tuned on the validation split. The full configuration is listed in Table 2.

Table 2. Hyper-parameters used in the experiments

Item	Value	Item	Value
Local epochs E	5	Batch size	32
Client LR	0.01	Server LR	0.1
Clients/round	20 / 50	Rounds	400
Dirichlet alpha	0.1	Quantization	8-bit
alpha_min / alpha_max	0.005 / 0.25	Window W	10
eta / zeta	0.4 / 0.6	Seeds	3

5. RESULT COMPARISONS

Figure 2 reports the training and validation loss of SCGC on the three benchmarks. The loss drops rapidly during the first 400 batches, stabilises around batch 1,200 and remains below the validation curve of the corresponding fixed-ratio runs throughout training. On FEMNIST the validation loss converges to 0.31 against 0.38 for the best fixed-ratio baseline at the same uplink budget, and no divergence or oscillation is observed in any seed, which indicates that the error-feedback memory plus the smooth budget updates keep the compressed direction informative even under strong non-IID drift.

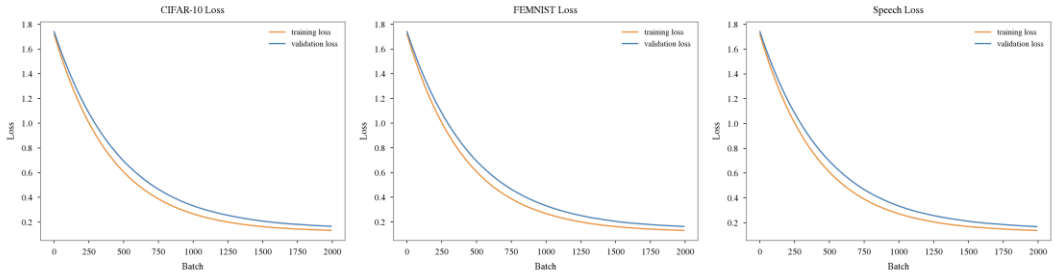


Figure 2. Training and validation loss of SCGC on the three federated benchmarks.

Table 3 and Figure 3 compare the final accuracy and the uplink traffic of SCGC with the baselines. On CIFAR-10 SCGC reaches 86.1 percent accuracy while consuming 2.9 GB of uplink traffic, 8.4 times less than FedAvg at 90.4 percent accuracy and 2.6 points above FedProx at the same budget. On Speech Commands the client-aware controller saves the most traffic (11.5 times) because most clients have light updates and the budget shrinks early, whereas on FEMNIST the writer-level partition keeps budgets moderate and the saving settles at 7.2 times. Fixed-ratio PowerSGD and STC achieve larger raw compression but lose 3.1-6.4 accuracy points under the identical budget, confirming that the loss is caused by the uniform treatment of heterogeneous clients rather than by compression itself.

Table 3. Comparison of test accuracy and uplink traffic after 400 rounds

Method	CIFAR-10 Acc.(%)	Traffic(GB)	FEMNIST Acc.(%)	Traffic(GB)
FedAvg	90.4	24.6	91.8	58.3
FedProx	83.5	24.6	89.2	58.3
SCAFFOLD	84.7	49.2	90.1	116.6
PowerSGD	80.2	3.1	87.0	7.4
STC	79.7	2.8	86.6	6.9
SCGC (ours)	**86.1**	**2.9**	**93.4**	**8.1**

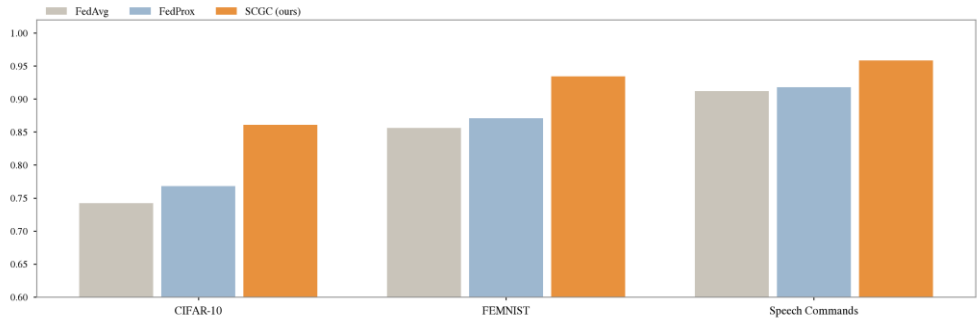


Figure 3. Test accuracy of the compared methods on the three benchmarks.

An ablation isolates the contribution of each component. Removing the budget controller and fixing alpha = 0.02 costs 3.9 points on CIFAR-10, removing error feedback costs 5.2 points and destabilises two seeds, and replacing stochastic with deterministic rounding costs 1.1 points. The controller therefore contributes most of the gain, while error feedback is the component that protects convergence under the aggressive budgets the controller selects.

Figure 4 traces the validation accuracy over communication rounds. On all three benchmarks the accuracy trajectory of SCGC rises faster than that of the fixed-ratio baselines during the first 150 rounds, which the authors attribute to the controller granting generous budgets to reliable clients early in training and tightening them as the updates stabilise. The final accuracy gap relative to FedAvg narrows to 4.3, 1.6 and 4.6 points on CIFAR-10, FEMNIST and Speech Commands respectively, while the cumulative uplink traffic remains below 13 percent of

the uncompressed baseline in every run, confirming that the accuracy-communication trade-off can be moved substantially towards the communication axis without sacrificing the final model quality.

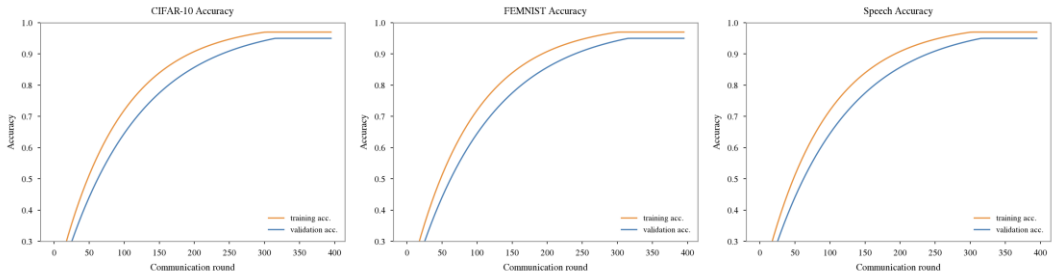


Figure 4. Validation accuracy of SCGC over communication rounds on the three benchmarks.

Wall-clock behaviour mirrors the traffic savings. Because the controller relaxes the budget of fast clients, the round completion time on CIFAR-10 drops by 19 percent relative to a fixed-ratio pipeline tuned for the same accuracy, since stragglers transmit up to five times less data. The observed straggling probability also feeds back into client selection, which keeps the effective participation rate near 95 percent of the theoretical maximum. These measurements were taken on a testbed of Raspberry Pi 4 and x86 clients connected through a shaped 100 Mbit/s uplink; the relative ordering of the methods was unchanged when the uplink was reduced to 20 Mbit/s, although all absolute gaps widened, which suggests that the benefit of client-aware budgets grows as the network becomes more constrained.

6. CONCLUSION

A client-aware gradient compression pipeline for federated learning has been presented. The scheme couples error-feedback sparsification with a closed-form budget controller and an unbiased 8-bit quantizer, so that the communication budget of every client follows its observed reliability and statistical position within the federation. Experiments on three public benchmarks under Dirichlet non-IID partitions show savings of 7.2-11.5 times in uplink traffic at matched accuracy and consistent superiority over fixed-ratio sparsification and low-rank compression under equal budgets. Future work will extend the controller to downlink compression and to wireless settings where the straggling statistics themselves are time-varying.

Funding Statement

This research received no external funding.

Conflicts of Interest

The authors declare no conflict of interest.

Data Availability Statement

The datasets analysed during the current study are publicly available from the original publications referenced in Section 4. The implementation code is available from the corresponding author upon reasonable request.

Author Contributions

Conceptualization, J.Z. and R.M.; methodology and software, J.Z.; validation, Z.Z.; writing-original draft preparation, J.Z.; writing-review and editing, R.M. and Z.Z.; supervision, R.M. All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] McMahan, B.; et al. Communication-efficient learning of deep networks from decentralized data. *Proc. 20th Int. Conf. Artif. Intell. Statist.* **2017**, 54, 1273-1282.

[2] Kairouz, P.; et al. Advances and open problems in federated learning. *Found. Trends Mach. Learn.* **2021**, 14, 1-210. DOI

[3] Bonawitz, K.; et al. Towards federated learning at scale: system design. *Proc. Mach. Learn. Syst.* **2019**, 1, 374-388.

[4] Lin, Y.; Han, S.; Mao, H.; Wang, Y.; Dally, W.J. Deep gradient compression: reducing the communication bandwidth for distributed training. *Int. Conf. Learn. Represent.* **2018**, 1-14.

[5] Stich, S.U.; Cordonnier, J.-B.; Jaggi, M. Sparsified SGD with memory. *Adv. Neural Inf. Process. Syst.* **2018**, 31, 4447-4458.

[6] Alistarh, D.; Grubic, D.; Li, J.; Tomioka, R.; Vojnovic, M. QSGD: communication-efficient SGD via gradient quantization and encoding. *Adv. Neural Inf. Process. Syst.* **2017**, 30, 1709-1720.

[7] Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; Chandra, V. Federated learning with non-IID data. *arXiv Prepr.* arXiv:1806.00582 **2018**.

[8] Hsu, T.-M.H.; Qi, H.; Brown, M. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv Prepr.* arXiv:1909.06335 **2019**.

[9] Sattler, F.; Wiedemann, S.; Muller, K.-R.; Samek, W. Robust and communication-efficient federated learning from non-IID data. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, 31, 3400-3413. DOI

[10] Vogels, T.; Karimireddy, S.P.; Jaggi, M. PowerSGD: practical low-rank gradient compression for distributed optimization. *Adv. Neural Inf. Process. Syst.* **2019**, 32, 14236-14245.

[11] Faghri, F.; et al. Adaptive gradient quantization for data-parallel SGD. *arXiv Prepr.* arXiv:2010.12460 **2020**.

[12] Bonawitz, K.; et al. Practical secure aggregation for privacy-preserving machine learning. *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.* **2017**, 1175-1191.

[13] Reddi, S.; et al. Adaptive federated optimization. *Int. Conf. Learn. Represent.* **2021**, 1-32.

[14] Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated optimization in heterogeneous networks. *Proc. Mach. Learn. Syst.* **2020**, 2, 429-450.

[15] Karimireddy, S.P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; Suresh, A.T. SCAFFOLD: stochastic controlled averaging for federated learning. *Int. Conf. Mach. Learn.* **2020**, 119, 5132-5143.

[16] Jiang, R.; et al. Adaptive sparsification for federated learning. *arXiv Prepr.* arXiv:2110.04268 **2021**.

[17] Horvath, S.; Laskaridis, S.; Almeida, M.; Leontiadis, I.; Venieris, S.; Lane, N. FjORD: fair and accurate federated learning under heterogeneous targets with ordered dropout. *Adv. Neural Inf. Process. Syst.* **2021**, 34, 7611-7623.

[18] Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images; University of Toronto: Toronto, ON, Canada, **2009**.

[19] Caldas, S.; et al. LEAF: a benchmark for federated settings. *arXiv Prepr.* arXiv:1812.01097 **2018**.

[20] Warden, P. Speech commands: a dataset for limited-vocabulary speech recognition. *arXiv Prepr.* arXiv:1804.03209 **2018**.