OS-FED: One Snapshot Is All You Need

Xuwei Qian[1] Jinghui Zhang[1*] Yuchuan Tan[1] Wenbo Huang[1] Zhen Wu[1]
Shen Zhou[1] LiSha Gao[2] Ding Ding[1] Fang Dong[1]
[1]Southeast University [2]State Grid Nanjing Power Supply Company
{xuwei.qian, jhzhang, tanyuchuan, zhen-wu, zhoushen, dingding-1, fdong}@seu.edu.cn,
wenbohuang1002@outlook.com, sun_gls@163.com

Abstract

Reducing communication overhead in federated learning (FL) is challenging but crucial for large-scale distributed privacy-preserving machine learning. Unfortunately, directly compressing model updates often leads to sub-optimal convergence due to information loss, while increasing local computation can cause model divergence. Hence, this paper proposes a drastically different approach that adheres to the maxim that "a picture is worth a thousand words". *We observe that the entire gradient information from local training can be effectively reconstructed from a compact, image-like representation. Based on this observation, we propose a novel approach,* **OS-FED**, *which performs **O**ne-**S**hot **FED**erated Learning by transmitting only a single, compact snapshot (comprising an image and a set of learnable labels) per round. To realize this approach,* OS-FED *presents new snapshot synthesis techniques to (1) target the accumulated update of a trajectory segment to tackle gradient noise, (2) design a multi-grid snapshot that decouples conflicting gradient directions, and (3) incorporate error compensation to maintain training stability under extreme compression. Extensive experiments on CV and NLP benchmarks show that* OS-FED *reduces communication costs by 1.5-16× compared to state-of-the-art algorithms , resulting in 18-45% faster convergence.*

1. Introduction

In recent years we have witnessed a remarkable leap in distributed machine learning [4, 11, 13], driven by breakthroughs in federated learning [22, 31] (FL) frameworks and the access to vast amounts of decentralized data. Particularly, pioneering algorithms, such as Federated Averaging [31] (FedAvg), have matured to the point where they can produce highly effective models based on data held on user devices, without compromising privacy.

However, one major drawback of existing FL frameworks is that they suffer from a severe communication bottleneck [26–28]. The generated model updates, based on local training, often reflect multi-million parameter gradients that must be transmitted to a central server. For instance, in our ImageNet-10 experiment (Table 1), training a mid-sized RegNetX [34] model for just 10 rounds incurred a staggering total communication traffic of 8,040 MB. This creates a significant overhead that scales with model size, which is particularly challenging as clients typically connect over slow and unreliable networks [22]. This problem is poised to worsen with the growing popularity of LLMs [7, 16, 23].

To advance communication-efficient FL, a straightforward way is to increase the local computation on each client to reduce communication rounds, using methods like FedAvg [10, 31]. Doing so, however, is insurmountably challenging as performing extensive local updates on non-IID

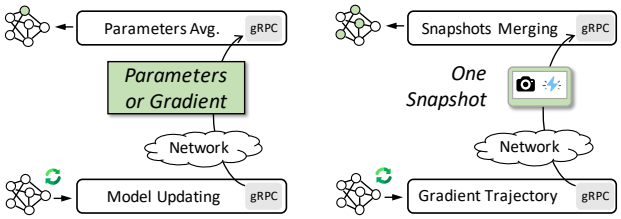

Figure 1. **(a)** Standard federated learning, such as FedAvg, requires transmitting large model updates, leading to a severe communication bottleneck. **(b)** We address this problem by synthesizing a single, compact gradient multi-grid snapshot for one-shot federated learning (OS-FED).

Technique	Traffic size (MB, lower is better)	Accuracy (higher is better)
FedAvg	8040	0.65
OS-FED (this paper)	57	0.64
Quantization (FedMPQ)	117	0.56
OS-FED w/ FedMPQ [8]	7	0.63
LoRA-FAIR [6]	149	0.62
OS-FED on LoRA-FAIR	57	0.62

Table 1. **Performance comparison of OS-FED and baselines on ImageNet-10 with the RegNetX-4.0GF model.** Traffic size is the total communication volume from 10 clients over 10 rounds. The default snapshot size for OS-FED is 224x224x3 with 32-bit precision. Full results are presented in Section 4.1.

*Corresponding author.

data can lead to client drift and model divergence [26], and training large models is highly compute-intensive for edge devices. Another principled approach towards communication efficiency is to directly compress the model updates (i.e., sparsification [49, 59] or quantization [3, 5]). However, many complex gradient structures are difficult to capture with these lossy compression schemes or cannot be well recovered by the server due to high information loss or the stateless nature of clients in FL [19, 22, 53].

In this paper, instead of directly compressing the high-dimensional gradient update, we are interested in whether we can learn a compact proxy to reconstruct it. Our key insight is that the gradient generated by a model can be effectively reconstructed from a simple image and a corresponding label. Consequently, we find that simply sending a standard 32-bit 224x224x3 image for server-side reconstruction requires only 0.57 MB, a reduction of over 140× compared to an 80.4 MB update from a mid-sized RegNetX model, with negligible label transmission cost.

Based on this insight, we present **OS-FED**, a **O**ne-**S**hot[1] **FED**erated learning framework where clients transmit only a single snapshot per communication round (see Figure 1). To achieve this goal, we overcome three primary challenges. First, rather than compressing a single gradient step, OS-FED explicitly tackles gradient noise by compressing the accumulated update over an entire trajectory segment. Second, we develop a novel multi-grid snapshot with a single image and a set of learnable labels. This design decouples conflicting gradient directions into different spatial regions of the snapshot. We also provide theoretical insights into the benefits of this multi-grid structure. Lastly, an error compensation mechanism mitigates information loss from this extreme compression by carrying the residual error over to the next round, vital for maintaining training stability.

We validate our framework with standard FL benchmarks (CV and NLP). Overall, OS-FED can leverage the local training trajectory to achieve communication-efficient FL in single or multiple local update scenarios. OS-FED needs neither complex gradient manipulation nor multi-step simulation, bypassing the problems of information loss and training instability. Moreover, OS-FED is compatible with existing LoRA-based frameworks and quantization mechanisms in a plug-and-play manner (see Table 1). To the best of our knowledge, this is the first method that enables one-shot federated learning on large, GPT-style language models [33] while achieving competitive results.

2. Related Work

Network Bottleneck in Federated Learning. Federated Learning (FL) has been widely studied as a privacy-preserving distributed learning paradigm, with numerous model architectures and learning strategies proposed [9, 24, 28, 35, 39, 46, 51]. A key factor in FL is the iterative process of clients downloading a global model, computing updates locally, and uploading them for aggregation. Recently, the overwhelming success of large-scale models, particularly Large Language Models (LLMs), has intensified the need for efficient training methods on decentralized data [7, 12, 43–45]. Thus, questions concerning communication efficiency while maintaining model performance remain a crucial open problem in FL [22, 26, 27].

Model Update Compression. A dominant approach to mitigating the communication bottleneck is to directly compress the model updates sent from clients to the server. Research in this area largely follows two main strategies: sparsification and quantization. Sparsification methods, such as Top-k, aim to transmit only a small subset of the most significant gradient values, thereby reducing the data volume [2, 29, 36, 39, 59]. Quantization methods, on the other hand, reduce the numerical precision of each value in the gradient vector, with aggressive variants like signSGD [5] using only a single bit per parameter [3, 8, 40]. In contrast, our approach avoids direct, lossy compression of the gradient space and instead learns a compact representation that can reconstruct the gradient trajectory segment.

Local Data Compression. An alternative paradigm is local data compression, which aims to condense a client's local dataset into a small, synthetic one that is then transmitted to the server for direct retraining of the global model [48, 55]. This approach[2], however, represents a significant departure from the decentralized learning principle of FL. By centralizing a data proxy for server-side training, it introduces potential privacy risks that FL was designed to mitigate [32, 38, 52, 60]. Moreover, generating this synthetic dataset relies on a complex bi-level optimization that is not only computationally intensive for resource-constrained clients, but also necessitates accessing private label information, which limits its practical applicability.

Our method differs from prior works in several key aspects: (1) OS-FED does not impose any restrictions on the label setup, as the labels are learnable and independent of the client's private data; (2) its snapshot generation is a simple single-level process, not requiring complex bi-level optimization (meta-learning [17]); and (3) the snapshot merging on the server is only for gradient recovery, not for retraining, making it extremely fast.

3. One-Shot Federated Learning

In this chapter, we elaborate on our proposed framework, OS-FED, designed to address the core communication bottleneck in federated learning. The central idea is to com-

[1] Unlike some works [18, 38, 41, 60], "*one-shot*" in our context refers to the transmission of just **one** snap**shot** per communication round.

[2] This paradigm is often termed dataset distillation [47, 54] or condensation in methods like DataDAM [37] and the work of Zhao et al. [57, 58].

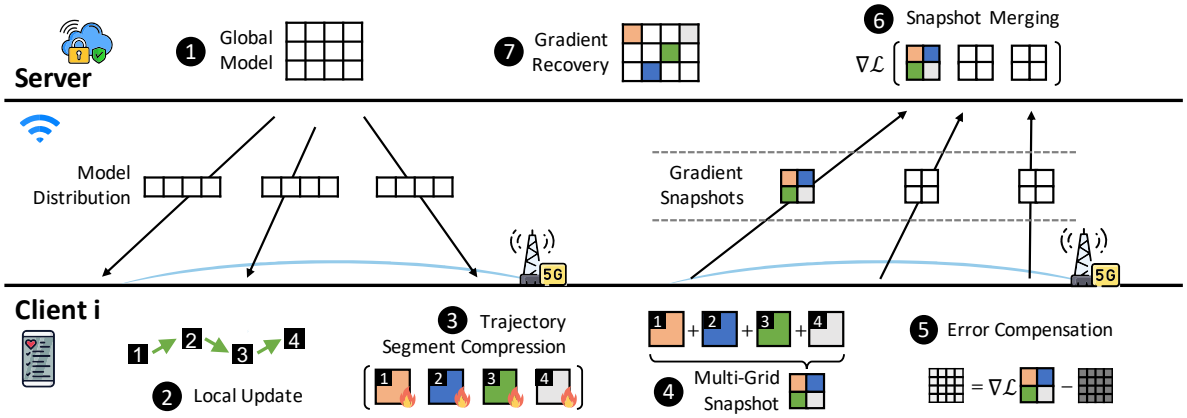

Figure 2. **Illustration of One-Shot Federated Learning (OS-FED).** A client i first **(1)** downloads the current global model from the server and performs **(2)** local updates on its private data. The gradient trajectory segment is then **(3)** compressed into a **(4)** single Multi-Grid snapshot. The compression residual is calculated and maintained via **(5)** error compensation for the next round. On the server side, snapshots from all clients are aggregated through **(6)** snapshot merging. Finally, the server performs **(7)** gradient recovery from the merged snapshot to update the global model.

press the complex local update process of a client into an information-dense representation, which we term a *gradient multi-grid snapshot*. A central server can then update the global model by performing an efficient, single-step aggregation over the snapshots collected from all clients.

Figure 2 illustrates the overall framework. In this section, we first introduce the framework of OS-FED in Section 3.1, then describe the details of the learning strategy for snapshots in Section 3.2, and finally discuss the key properties of OS-FED in Section 3.3.

3.1. Overview

Federated Learning Setup. We consider a federated learning (FL) system where C clients are selected to participate in each communication round t. The overarching goal of FL is to train a global model $\mathbf{w}$ that minimizes a global objective function, which is the average of the local loss functions across the participating clients. Formally,

$$\min_{\mathbf{w}\in\mathbb{R}^d} \mathcal{L}(\mathbf{w}) := \frac{1}{C}\sum_{i=1}^{C} \mathcal{L}_i(\mathbf{w}) \tag{1}$$

where d is the dimension of the model parameters and $L_i(\cdot)$ is the loss function for the i-th client on its local dataset. Standard algorithms like FedAvg iteratively solve this by having each client perform K epochs of local training starting from the current global model $\mathbf{w}_t$. The net result of this local training is the accumulated update $\Delta\mathbf{w}_{t,i}$, which represents the sum of all gradients computed along the local training trajectory. Formally, $\Delta\mathbf{w}_{t,i} = \eta_{\text{local}} \sum_{k=1}^{K} g_{t,i}^{(k)}$, where η_{local} is the client-side learning rate and $g_{t,i}^{(k)}$ is the gradient at the k-th local step. Transmitting these high-dimensional updates $\Delta\mathbf{w}_{t,i}$, which encapsulate the entire local training trajectory, constitutes the primary communication bottleneck in FL.

One Snapshot as Gradient Proxy. To tackle the communication bottleneck, we propose learning a compact Snapshot $\boldsymbol{S}_i$ to represent the accumulated update $\Delta\mathbf{w}_{t,i}$. The core idea is that the synthetic gradient generated from this single snapshot approximates the accumulated gradient of the entire local training trajectory. Generally, $\boldsymbol{S}_i$ comprises a synthetic image and learnable labels (detailed in Sec. 3.2).

For each client i, the objective is to find an optimal $\boldsymbol{S}_{t,i}^*$ that best reconstructs its target update $\Delta\mathbf{w}_{t,i}$. This is formulated as a constrained optimization problem, minimizing the distance between the snapshot's generated gradient and the target update under a strict communication budget B. We formally define this process as:

$$\begin{aligned} \boldsymbol{S}_{t,i}^* = \underset{\boldsymbol{S}_i}{\operatorname{argmin}}\ & D(\nabla\mathcal{L}(\boldsymbol{S}_i, \mathbf{w}_t), \Delta\mathbf{w}_{t,i}) \\ \text{s.t.}\quad & \|\boldsymbol{S}_i\|_0 \le B \end{aligned} \tag{2}$$

Here, $\boldsymbol{S}_{t,i}^*$ is the optimal snapshot for client i. The function $\nabla\mathcal{L}(\boldsymbol{S}_i, \mathbf{w}_t)$ generates a synthetic gradient from the snapshot $\boldsymbol{S}_i$ and the current model $\mathbf{w}_t$, where $\mathcal{L}$ is a loss function. $D(\cdot,\cdot)$ is a gradient distance metric that quantifies the discrepancy between the two vectors, while $\|\cdot\|_0$ denotes the $L0$ norm, enforcing this budget by counting the non-zero parameters in the snapshot.

Upon receiving the optimized snapshots from all C clients, the server performs a single-step aggregation to obtain an estimate of the average of all true accumulated updates. This is achieved by generating synthetic gradients from a batch of all snapshots and averaging them. This relationship is expressed as:

$$\frac{1}{C}\sum_{i=1}^{C} \Delta\mathbf{w}_{t,i} \approx \nabla\mathcal{L}_{\text{mean}}([\boldsymbol{S}_{t,i}^*]_{i=1}^{C}, \mathbf{w}_t) \tag{3}$$

where the left-hand side is the ideal global update in FedAvg, and our single-step aggregation on the right-hand

side serves as its efficient approximation. Here, $[\boldsymbol{S}^*_{t,i}]_{i=1}^C$ denotes the concatenation of all client snapshots along the batch dimension. The global model is then updated using this efficiently computed aggregate: $\mathbf{w}_{t+1} = \mathbf{w}_t - \nabla\mathcal{L}_{\text{mean}}([\boldsymbol{S}^*_{t,i}]_{i=1}^C, \mathbf{w}_t)$.

In contrast to conventional compression methods such as sparsification or quantization that operate directly on the gradient space, our approach optimizes the snapshot entirely in a compact parameter space. This decoupling provides significant flexibility. We elaborate on the learning process for the snapshots in the following section.

3.2. Learning Gradient Multi-Grid Snapshot

We now detail the client-side process for synthesizing and optimizing the gradient multi-grid snapshots. First, we define the full gradient trajectory which serves as the compression target. Second, we describe the parameterization of our snapshot and its optimization process. Finally, we detail the error compensation update mechanism.

Gradient Trajectory Compression. In federated learning with high-rate compression, the errors introduced by the compression operator in each round are non-trivial. If these errors are simply ignored, they can accumulate over communication rounds, potentially slowing down the convergence of the global model or even causing it to diverge from the optimal solution. A robust compression framework must therefore track and compensate for these errors.

To this end, our method maintains an error accumulation vector e locally on each client, which is initialized to a zero vector at the beginning of the training. In each round, the true compression target—termed the target gradient trajectory $\Delta\mathbf{g}$—is the sum of the current round's accumulated update $\Delta\mathbf{w}$ and the error e inherited from all previous rounds. This is formally expressed as:

$$\Delta\mathbf{g}_{t,i} = \Delta\mathbf{w}_{t,i} + \mathbf{e}_{t-1,i} \tag{4}$$

This design ensures that any information from the gradient trajectory that is not fully captured by the snapshot in the current round is not discarded. Instead, it is retained in the error vector e and will be reconsidered for compression in subsequent rounds.

Multi-Grid Snapshot. The parameterization of the snapshot $\boldsymbol{S}$ is a critical design choice, as it directly determines the trade-off between compression efficiency and the ability to accurately represent the gradient trajectory. A straightforward approach would be to parameterize the snapshot as a single, monolithic synthetic image paired with a single label. However, this monolithic design is often insufficient. The accumulated update, resulting from K epochs of local training, often encodes a rich set of features pointing in diverse directions, especially on non-IID data. A single synthetic sample struggles to capture this diversity, creating an expressive power bottleneck that can limit the accuracy of the gradient approximation (see Figure 3, Vanilla).

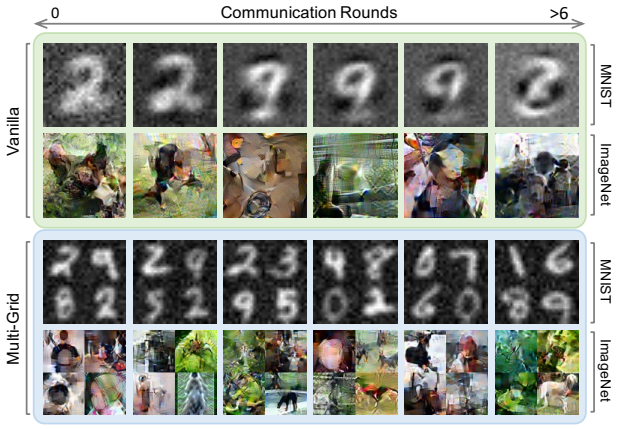

Figure 3. **Visualization of learned snapshots from early training rounds.** The Vanilla approach, which learns a single monolithic snapshot, struggles to represent conflicting gradient information, resulting in blurred patterns (e.g., features of '2' and '9' mixing). In contrast, our Multi-Grid approach (e.g., M=2 for this 2x2 grid) disentangles gradient information into different cells, maintaining a richer and clearer representation throughout training.

To address this challenge, we introduce the multi-grid Snapshot. We parameterize the snapshot $\boldsymbol{S}$ as a unified collection of learnable tensors, which encapsulates both a grid-like feature map and a corresponding set of M^2 independent label embeddings. We then design a deterministic and differentiable multi-grid function $\mathcal{F}(\cdot)$ to unfold the information packed within $\boldsymbol{S}$. Operationally, $\mathcal{F}(\cdot)$ takes $\boldsymbol{S}$ as input. It interprets the feature map as a grid of patches, upsamples each patch to the target input dimension via bilinear interpolation, and pairs them with their corresponding label embeddings also contained within $\boldsymbol{S}$. This process results in M^2 distinct synthetic samples. This "*Multi-Grid*" design allows different parts of a single snapshot to specialize in learning different aspects of the target gradient trajectory, thus significantly enhancing its expressive power.

Snapshot Optimization. The parameters of the snapshot $\boldsymbol{S}$ are learned by minimizing the discrepancy between the gradient generated from it and the target gradient trajectory $\Delta\mathbf{g}$. We use the normalized $L2$ distance as the optimization objective. To prevent numerical instability when the norm of $\Delta\mathbf{g}$ is close to zero, we add a small positive constant ε to the denominator. The loss function is defined as:

$$\mathcal{L}_{\text{match}} = \frac{||\nabla\mathcal{L}(\mathcal{F}(\boldsymbol{S}_i), \mathbf{w}_t) - \Delta\mathbf{g}_{t,i}||_2^2}{||\Delta\mathbf{g}_{t,i}||_2^2 + \varepsilon} \tag{5}$$

We normalize the $L2$ error by the squared norm of the target gradient trajectory $\Delta\mathbf{g}$. This normalization serves two purposes: first, it ensures that we still receive a strong optimization signal in later training stages where the magnitude of $\Delta\mathbf{g}$ may become small. Second, it helps to self-calibrate

the magnitude differences across different layers and parameters within the model. We have also experimented with other choices of loss functions, such as cosine distance, but found that our simple normalized $L2$ loss leads to more stable optimization and empirically better results.

Since the multi-grid function $\mathcal{F}(\cdot)$ is composed of differentiable operations, the entire computational graph from the snapshot $\boldsymbol{S}_i$ to the loss $\mathcal{L}_{\text{match}}$ is end-to-end differentiable. This allows us to employ efficient gradient-based optimizers. In our implementation, we use the L-BFGS [30] optimizer to iteratively update the parameters of $\boldsymbol{S}_i$ to find the optimal snapshot $\boldsymbol{S}^*_{t,i}$ that minimizes $\mathcal{L}_{\text{match}}$.

Error Compensation Update. After the optimal snapshot $\boldsymbol{S}^*_{t,i}$ is obtained, the compression residual for the current round is calculated to update the error state for the next round. This residual is defined as the difference between the target gradient trajectory $\Delta\mathbf{g}_{t,i}$ and the final synthetic gradient generated by $\boldsymbol{S}^*_{t,i}$. The new accumulated error $\mathbf{e}_{t,i}$ is then computed as:

$$\mathbf{e}_{t,i} = \Delta\mathbf{g}_{t,i} - \nabla\mathcal{L}(\mathcal{F}(\boldsymbol{S}^*_{t,i}), \mathbf{w}_t) \tag{6}$$

This updated error $\mathbf{e}_{t,i}$ captures the precise information that the snapshot $\boldsymbol{S}^*$ failed to represent in the current round. If $\mathbf{e}_{t,i}$ is a zero vector, it indicates a perfect, lossless compression for this round's trajectory. Otherwise, the non-zero residual $\mathbf{e}_{t,i}$ is not discarded; instead, it is carried over to the next round. This ensures that any information lost during one compression step has the opportunity to be compensated for in subsequent rounds, which is crucial for maintaining model performance under high compression rates.

3.3. Key Properties of OS-FED

Memory, and Computational Efficiency. The memory requirement is determined by both the main model's size and the computation on a small batch of M^2 synthetic samples. Since M is typically small (e.g., M=4), the memory footprint is comparable to a standard FedAvg training round. Computationally, client-side optimization takes merely $\sim$ 3.6 seconds, while server-side aggregation is reduced to a highly efficient $\mathcal{O}(1)$ single-step process, maximizing system throughput.

Privacy Guarantees. OS-FED natively mitigates privacy risks by fitting gradient dynamics rather than raw pixels. First, to quantify this, we analyzed the cosine similarity between snapshots (over 100 rounds) and original client data on ImageNet-1K. Over 75% of snapshots exhibit a similarity score below 0.25. Higher similarities are primarily observed during early training stages when gradient directions are simpler. Second, for applications requiring strict formal guarantees, OS-FED can be readily augmented by adding random noise during snapshot optimization $\mathcal{F}(\boldsymbol{S}_i + \boldsymbol{z})$. This end-to-end mechanism provides highly effective privacy protection with minimal computational overhead.

Theoretical Analysis. Here, we can assume a data point is m-dimensional. The natural data have regularity that makes difference from random noise [21]. We assume that data satisfying this regularity form a subspace $\mathcal{N} \subset \mathbb{R}^m$. That is, the client-side local dataset $\mathcal{T} = \{t_i\}_{i=1}^{n_t}$ satisfies $t_i \in \mathcal{N}$ for $i = 1, \ldots, n_t$. With abuse of notation, we denote the space of datasets with n data points as $\mathbb{R}^{n\times m} = \{\{d_i\}_{i=1}^n \mid d_i \in \mathbb{R}^m \text{ for } i = 1, \ldots, n\}$. We further define the space of all datasets $\mathcal{D} = \cup_{n\in\mathbb{N}}\mathbb{R}^{n\times m}$ and the space of synthetic samples generated by a multi-grid function $\mathcal{F} : \mathbb{R}^{1\times m} \to \mathbb{R}^{n'\times m}$, $\mathcal{M} = \{\mathcal{F}(\boldsymbol{S}) \mid \boldsymbol{S} \in \mathbb{R}^{1\times m}\}$.

Definition 1. *A function $D : \mathcal{D} \times \mathcal{D} \to [0, \infty)$ is a gradient $L2$ distance measure if, for any two datasets $\forall X, X' \in \mathcal{D}$ it is defined as the Euclidean ($L2$) distance between their corresponding loss gradients:*

$$D(X, X') = \left\|\nabla\mathcal{L}(X, \mathbf{w}) - \nabla\mathcal{L}(X', \mathbf{w})\right\|_2$$

where $\|\cdot\|_2$ denotes the $L2$ norm. This measure satisfies the following properties inherited from the $L2$ norm:

1. $D(X, X) = 0$ *and* $D(X, X') = D(X', X)$.
2. $\forall d \in \mathbb{R}^m$ *s.t.* d *is closer to* X' *than* d_i, $D(X \setminus \{d_i\} \cup \{d\}, X') \le D(X, X')$.
3. $D(X, X' \cup \{d_i\}) \le D(X, X')$.

The definition above states the reasonable conditions for gradient distance measurement. Based on this definition, we formally introduce the following proposition. We provide the proof in Appendix A.1.

Proposition 1. *If $\mathcal{N}^{n'} \subseteq \mathcal{M}$, then for the above-defined gradient $L2$ distance measure D,*

$$\min_{\boldsymbol{S}\in\mathbb{R}^{1\times m}} D(\mathcal{F}(\boldsymbol{S}), \mathcal{T}) \le \min_{\boldsymbol{S}\in\mathbb{R}^{1\times m}} D(\boldsymbol{S}, \mathcal{T}).$$

Proposition 1 states that our multi-grid function achieves the better optimum, *i.e.*, the gradient produced by the multi-grid snapshot is closer to the target gradient. Additionally, in Appendix A.2, we provide theoretical results under a more relaxed assumption.

4. Experiment

In this section, we empirically validate our method. We first present main results on CV tasks (Section 4.1), followed by NLP tasks (Section 4.2) to demonstrate versatility. Detailed ablation studies and analyses are provided in Section 4.3. Please see the Appendix for additional implementation details, ablations, and large-scale experiments.

Datasets & Partitioning. We evaluate OS-FED on standard CV benchmarks (MNIST [15], FMNIST [50], CIFAR-10/100 [25]) and an ImageNet subset [14, 42] for complex scenarios. For NLP, we use two widely-adopted text classification datasets: AGNews and Sogou News [56]. To simulate federated non-IID settings, all datasets are partitioned among clients using a Dirichlet distribution (Dir($\alpha = 0.5$)).

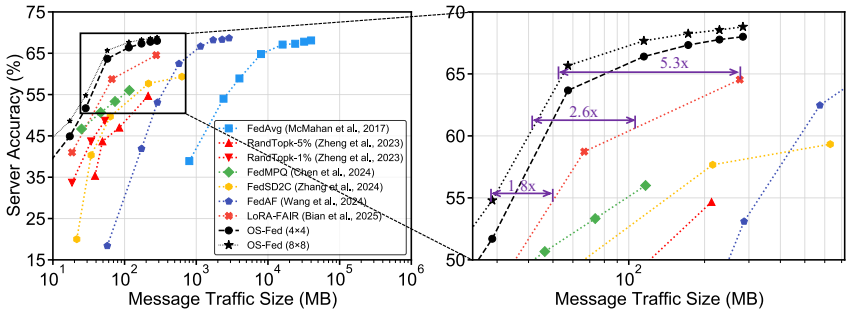

Figure 4. **Long-term performance trajectories on ImageNet-10 over 50 communication rounds.** Each curve represents the evolution of server accuracy versus the cumulative communication size for a given method. OS-FED (4x4) and OS-FED (8x8) refer to our method implemented with different multi-grid snapshot grid sizes.

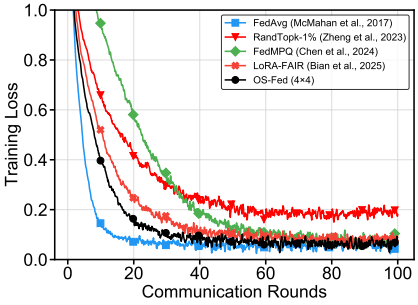

Figure 5. **Convergence speed comparison on ImageNet-10 (zoom in for details).** The plot shows the average training loss as a function of communication rounds.

Table 2. **Performance comparison on CIFAR-10 and CIFAR-100.** Size(MB) refers to the average message size per communication round. The best results are **bold-faced**.

Methods	Message Type	CIFAR-10		CIFAR-100	
		Acc.	Size(MB)	Acc.	Size(MB)
FedAvg [31]	Gradient	47.6%	447.2	27.4%	451.2
FedMPQ [8]	Gradient	31.5%	13.9	20.2%	14.1
FedSD2C [55]	Dataset	42.7%	35.2	27.2%	352.6
RandTopk [59]	Gradient	39.3%	22.4	25.0%	22.6
FedAF [48]	Dataset	41.9%	23.4	26.4%	234.4
LoRA-FAIR [6]	Gradient	41.8%	3.58	26.3%	3.61
OS-FED (2x2)	Snapshot	42.2%	**0.12**	26.9%	**0.12**
OS-FED (4x4)	Snapshot	**44.9%**	0.12	**30.3%**	0.12

Table 3. **Scalability and model-agnostic evaluation on MNIST and FMNIST.** We compare performance across different models (MLP and MnistNet) and client scales (10 and 60 clients).

Methods	Model+ Clients	MNIST		FMNIST	
		Acc.	Size(MB)	Acc.	Size(MB)
FedAF [48]	MLP+10	94.3%	5.98	**84.1%**	5.98
LoRA-FAIR [6]	MLP+10	92.8%	0.06	83.5%	0.06
OS-FED (2x2)	MLP+10	**94.6%**	**0.03**	83.9%	**0.03**
FedAF [48]	MN+10	95.2%	5.98	85.4%	5.98
LoRA-FAIR [6]	MN+10	93.9%	0.32	82.5%	0.32
OS-FED (2x2)	MN+10	**95.8%**	**0.03**	**86.0%**	**0.03**
FedAF [48]	MN+60	94.0%	35.9	84.8%	35.9
LoRA-FAIR [6]	MN+60	93.2%	1.92	82.3%	1.92
OS-FED (2x2)	MN+60	**94.7%**	**0.18**	**85.1%**	**0.18**

Models. For CV tasks, we use four models with varying complexity: a simple Multi-Layer Perceptron (MLP), a custom CNN for MNIST-like tasks (MnistNet), the widely-adopted ResNet-18 [20], and a efficient architecture, RegNetX-4.0GF [34]. For NLP tasks, we utilize GPT2-mini [33], a compact transformer-based language model, to validate OS-FED's applicability to NLP architectures.

Baselines. We compare OS-FED with several competitive baselines, covering three major categories. The first category is gradient compression, which directly reduces model update size. This includes methods based on sparsification (e.g., RandTopk [59]), quantization (e.g., FedMPQ [8]), and low-rank approaches (e.g., LoRA-FAIR [6]). The second category is dataset-based communication, where clients synthesize a small dataset for transmission. We compare against state-of-the-art data distillation methods like FedAF [48]. Lastly, we use the standard FedAvg without any compression as a performance reference.

4.1. Experiments on CV Tasks

Implementation details. For all experiments, we train the models using an SGD optimizer. The learning rate is initialized to 0.01 and annealed with a cosine schedule. The federated learning system consists of 10 clients, unless otherwise specified. We run a total of 100 communication rounds, with each client performing K=5 local epochs of training in each round. For experiments on the ImageNet subset, we use RegNetX-4.0GF, while for the CIFAR datasets, we use ResNet-18. For MNIST and FMNIST, we use MnistNet and MLP. The default grid size for OS-FED is set to 4x4 for the ImageNet subset and CIFAR experiments, and 2x2 for MNIST and FMNIST, unless otherwise specified.

Communication Efficiency. A primary goal of OS-FED is to achieve a superior trade-off between model accuracy and communication cost. Figure 4 plots the Top-1 accuracy against the cumulative communication size (in MB) for various methods on our ImageNet subset. The results clearly show that OS-FED establishes a new state-of-the-art efficiency frontier, with its curve positioned significantly to the top-left compared to all baselines. For instance, to reach 65% accuracy, OS-FED requires only 80.5MB of communication. In contrast, the best-performing gradient compression method, LoRA-FAIR, requires 280MB, while the uncompressed FedAvg baseline demands over 8100MB to achieve a similar result. This demonstrates a communication reduction of over 100x compared to FedAvg.

Convergence Speed. Beyond per-round efficiency, the total number of communication rounds required for convergence is another critical metric. Figure 5 illustrates the training loss as a function of communication rounds. OS-FED (solid black line) exhibits a significantly faster convergence rate than all competing methods. It reaches a low loss plateau within approximately 40 rounds, whereas gradient

Table 4. **Performance and practical efficiency on the AGNews and Sogou datasets.** FedAF+ refers to an augmented FedAF baseline that shares the same optimization objective as our method, except for the multi-grid snapshot design. The client-side compression time (Comp. Time) is tested on an Apple M1 Pro, while the server-side decompression time (Decomp. Time) is tested on an NVIDIA 3090 GPU. OS-FED features near-instantaneous decompression, making it highly suitable for practical deployments. The best results are **bold-faced**.

Method	Message Type	#Shape	AGNews		Sogou		Comp. Time	Decomp. Time
			Top-1 Acc.	Size(MB)	Top-1 Acc.	Size(MB)	(Client, Apple M1 Pro)	(Server, NVIDIA 3090)
FedAvg [31]	Gradient	Original	82.3%	3124	91.0%	3124	-	-
RandTopk-5%	Gradient	Topk grad.	76.1%	156	86.3%	156	1.83s	0.09s
RandTopk-1%	Gradient	-	56.4%	31.2	69.7%	31.2	-	-
FedMPQ [8]	Gradient	-	67.6%	97.6	75.4%	97.6	7.41s	0.13s
LoRA-FAIR [6]	Gradient	-	75.5%	25.0	85.8%	25.0	-	-
FedSD2C [55]	Dataset	-	77.0%	8.99	87.5%	95.9	-	-
FedAF [48]	Dataset	[BS, Seq. length]	75.4%	5.62	84.1%	59.9	42min	8min
FedAF+ [48]	Dataset	[BS, Seq. length]	80.0%	11.2	88.6%	119.9	56min	13min
OS-FED (2x2)	Snapshot	[1, 1, 448, 448][3]	80.0%	**7.66** (↓1.46x)	88.7%	**7.66** (↓15.7x)	**5.32s** (↓632x)	**0.68s** (↓1147x)
OS-FED (4x4)	Snapshot	[1, 1, 448, 448]	81.6%	7.66	90.2%	7.66	5.46s	0.82s
OS-FED (4x4)	Snapshot	[1, 1, 600, 600]	**82.1%**	13.7	**90.7%**	13.7	6.12s	1.24s

compression methods require over 60 rounds to approach a similar level of convergence.

Results on CIFAR. We present the detailed performance on CIFAR-10 and CIFAR-100 in Table 2. OS-FED consistently outperforms all baselines on both datasets while requiring substantially less communication. For example, on the challenging CIFAR-100 benchmark, OS-FED (4x4) achieves 30.3% accuracy with only 0.12MB of communication, surpassing the strongest baseline, LoRA-FAIR, by a 4.0% margin while using 30x less communication volume.

Results with Different Numbers of Clients. To evaluate the scalability of our method, we conduct experiments with different numbers of clients on MNIST and FMNIST, with results shown in Table 3. Although the accuracy of all methods degrades when scaling from 10 to 60 clients, OS-FED consistently maintains a performance margin over the baselines with over 10x less communication.

Results on Different Models. We verify that the benefits of OS-FED are not tied to a specific network architecture. As shown in Table 3, whether using a simple MLP or a more complex convolutional network like MnistNet, OS-FED achieves the highest accuracy with the lowest communication cost. For example, with the MLP on MNIST, OS-FED reduces the communication size by over 190x compared to FedAF while achieving better final accuracy.

4.2. Experiments on NLP Tasks

Implementation details. For the NLP experiments, we use GPT2-mini [33] as the backbone model for all methods. The model is trained using the AdamW optimizer with a learning rate of 1e-4, accompanied by a linear learning rate warm-up and decay schedule. For OS-FED, the snapshot, parameterized as a sequence of learnable embeddings.

Results on AGNews & Sogou. Table 4 presents the main results on the AGNews and Sogou News datasets. On the AGNews dataset, OS-FED (4x4) achieves a Top-1 accuracy of 81.6%, surpassing the best gradient compression baseline (LoRA-FAIR) by 6.1% and the data distillation method (FedAF+) by 1.6%. Crucially, this is achieved with a communication payload of only 7.66MB, representing a 3.3x reduction compared to LoRA-FAIR and a nearly 1.46x reduction compared to FedAF+. Similar trends are observed on the larger Sogou News dataset.

Practical Efficiency. Beyond theoretical communication costs, the practical latency of client-side compression and server-side decompression is critical for real-world FL systems. We report these timings in the final two columns of Table 4. While gradient compression methods are fast, dataset-based methods like FedAF incur a substantial overhead. The client-side synthesis (compression) for FedAF takes over 42 minutes, as it requires a bi-level optimization to generate a representative dataset. Critically, the server-side decompression—which involves retraining on the aggregated synthetic datasets—takes more than 8 minutes per round, rendering it impractical for many applications.

In contrast, OS-FED is highly efficient in practice. Its client-side snapshot optimization takes less than 6 seconds (5.32s). Most importantly, the server-side decompression is nearly instantaneous (0.68s) because it is not a retraining process but a single, efficient one-shot aggregation: a simple forward and backward pass on the received snapshots.

4.3. Analytical Results

Visualization. To provide an intuitive understanding of our method's mechanics, we visualize its learning process in Figure 6. The plot reveals a distinct "easy-to-hard" learning pattern. In the early stages of training, the model's gradients, captured by the snapshot, focus intensely on a few easier-to-distinguish classes. As training progresses and these simple concepts are mastered, the focus diversifies to

[3] A single 448x448 single-channel image.

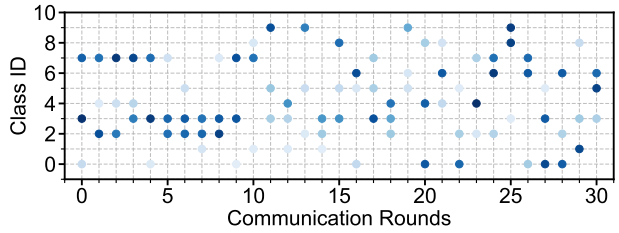

Figure 6. **Visualization of the learning dynamics of OS-FED.** Dot presence shows that a class's gradient was captured in the snapshot for a round, while color intensity indicates the gradient's magnitude (opaque is stronger). The model initially focuses on easier-to-distinguish classes (e.g., 2, 3, 7) and gradually shifts its attention to more difficult classes in later stages.

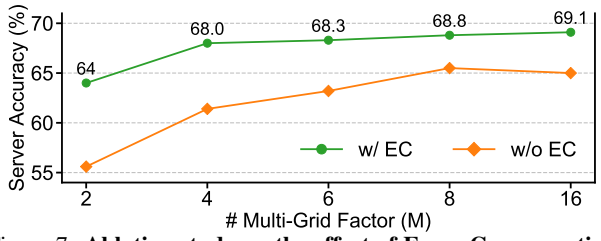

Figure 7. **Ablation study on the effect of Error Compensation (EC).** We plot the final server accuracy with and without EC across different snapshot complexities (Multi-Grid Factor).

Privacy-preserving Techniques	ImageNet-1K (on 60 clients) Top-1 Acc.↑	CosSim↓	SSIM↓	Overhead (per-client) Memory Cost	Wall-time
None	35.1%	0.122	0.07	4.55GB	3.6s
Optim. $\mathcal{F}(\boldsymbol{S}_i + \boldsymbol{z})$	33.5%	0.05	0.07	4.7GB (+0.15)	3.8s (+0.2)
DP-SGD [1]	33.4%	0.046	0.06	10.3GB	10.1s

Table 5. **Privacy-efficiency Trade-off on ImageNet-1K.** Our noise injection achieves superior privacy with lower overhead.

encompass more complex and confusable classes.

Privacy-Efficiency Trade-off. We empirically validate OS-FED's privacy-enhancing capabilities on ImageNet-1K. As detailed in Table 5, adding random noise during snapshot optimization $\mathcal{F}(\boldsymbol{S}_i + \boldsymbol{z})$ effectively suppresses visual similarity metrics (CosSim, SSIM) to levels comparable with traditional DP-SGD [1], while achieving a superior trade-off by maintaining higher model accuracy with significantly lower computational overhead.

Impact of Error Compensation. We study the importance of the error feedback mechanism, which is designed to compensate for compression residuals over time. Figure 7 plots the final test accuracy on ImageNet-10 with and without this mechanism, across different multi-grid factors (M). The results clearly demonstrate that including error compensation (**green line**) yields a consistent and significant performance gain over the variant without it (**yellow line**).

Impact of the Multi-Grid Function. To verify the effectiveness of our end-to-end snapshot design, we compare it against two alternatives in Table 6. Under the same communication constraints, OS-FED significantly outperforms the post-downsampling approach by a large margin (e.g., 63.5% vs. 48.8% on ResNet-18). Remarkably, OS-FED's performance closely approaches that of the Oracle, indicating that our multi-grid snapshot is a highly efficient and effective parameterization of the target gradient trajectory.

Table 6. **Ablation study on the snapshot design on ImageNet-10.** We compare our end-to-end OS-FED with OS-FED-post (a non-end-to-end approach of optimizing then downsampling) and an Oracle (an uncompressed upper bound).

Test Model	OS-FED (1×224×224)	OS-FED-post (16×56×56)	Oracle (16×224×224)
ResNet-18	63.5	48.8	65.6
RegNet-X	68.0	53.1	69.7

Table 7. **Ablation study of different optimization functions.** We compare our proposed single-level, normalized L2 loss ("*Single+Norm.+L2*") against alternatives, including bi-level optimization schemes ("*Bi*") common in dataset distillation.

Test Model	Bi+Cos (FedAF)	Bi +L2	Single +Cos	Single +L2	Single+Norm. +L2 (Ours)
ResNet-18	54.6	28.3	57.4	61.0	**63.5**
RegNet-X	49.8	28.7	60.1	64.6	**68.0**

Impact of the Optimization Objective. Table 7 presents an ablation study on the snapshot optimization loss function. We find that a single-level L2 distance metric generally outperforms cosine similarity for this task. More importantly, introducing the normalization term to the L2 distance, as proposed in our method, provides a substantial boost in performance. On the ResNet-18, our final objective ("*Single+Norm.+L2*") achieves 63.5% accuracy, outperforming the non-normalized L2 variant by 2.5%. This confirms that self-calibrating the magnitude of the loss signal is a crucial element for effective snapshot optimization.

5. Conclusion

In this paper, we presented OS-FED, a one-shot framework for highly communication-efficient federated learning. Whereas existing approaches focus on compressing the high-dimensional model update, which often leads to a sub-optimal solution, OS-FED demonstrates the possibility of a new paradigm: learning a single, compact snapshot to faithfully reconstruct the local training trajectory segment, which achieves superior performance while fundamentally decoupling communication cost from model size.

Acknowledgments. This work is supported by Jiangsu Science and Technology Major Special Program under Grant No. BG2024028; Jiangsu Provincial Frontier Technology Research and Development Program under Grant No. BF2024070; Fundamental Research Funds for the Central Universities; National Natural Science Foundation of China under Grant Nos. 62472094, 62572119, 62232004; Jiangsu Provincial Key Laboratory of Network and Information Security under Grant No. BM2003201; Key Laboratory of Computer Network and Information Integration (Ministry of Education, China) under Grant No. 93K-9.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016. 8

[2] Alham Fikri Aji and Kenneth Heafield. Sparse communication for distributed gradient descent. *arXiv preprint arXiv:1704.05021*, 2017. 2

[3] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. *Advances in neural information processing systems*, 30, 2017. 2

[4] Tal Ben-Nun and Torsten Hoefler. Demystifying parallel and distributed deep learning: An in-depth concurrency analysis. *ACM Computing Surveys (CSUR)*, 52(4):1–43, 2019. 1

[5] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pages 560–569. PMLR, 2018. 2

[6] Jieming Bian, Lei Wang, Letian Zhang, and Jie Xu. Lorafair: Federated lora fine-tuning with aggregation and initialization refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3737–3746, 2025. 1, 6, 7

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1, 2

[8] Huancheng Chen and Haris Vikalo. Mixed-precision quantization for federated learning on resource-constrained heterogeneous devices. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6138–6148, 2024. 1, 2, 6, 7

[9] Mingzhe Chen, Nir Shlezinger, H Vincent Poor, Yonina C Eldar, and Shuguang Cui. Communication-efficient federated learning. *Proceedings of the National Academy of Sciences*, 118(17), 2021. 2

[10] Yang Chen, Xiaoyan Sun, and Yaochu Jin. Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation. *IEEE transactions on neural networks and learning systems*, 31(10):4229–4238, 2019. 1

[11] Trishul Chilimbi, Yutaka Suzue, Johnson Apacible, and Karthik Kalyanaraman. Project adam: Building an efficient and scalable deep learning training system. In *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, pages 571–582, 2014. 1

[12] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022. 2

[13] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc'aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, et al. Large scale distributed deep networks. *Advances in neural information processing systems*, 25, 2012. 1

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[15] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 5

[16] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, 2021. 1

[17] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017. 2

[18] Neel Guha, Ameet Talwalkar, and Virginia Smith. One-shot federated learning. *arXiv preprint arXiv:1902.11175*, 2019. 2

[19] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Francoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018. 2

[20] 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*, pages 770–778, 2016. 6

[21] Jinggang Huang and David Mumford. Statistics of natural images and models. In *CVPR*, 1999. 5

[22] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021. 1, 2

[23] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 1

[24] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020. 2

[25] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[26] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *IEEE International Conference on Data Engineering*, 2022. 1, 2

[27] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020. 2

[28] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020. 1, 2

[29] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. *arXiv preprint arXiv:1712.01887*, 2017. 2

[30] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989. 5

[31] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 1, 6, 7

[32] Renjie Pi, Weizhong Zhang, Yueqi Xie, Jiahui Gao, Xiaoyu Wang, Sunghun Kim, and Qifeng Chen. Dynafed: Tackling client data heterogeneity with global dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12177–12186, 2023. 2

[33] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019. 2, 6, 7

[34] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10428–10436, 2020. 1, 6

[35] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics*, pages 2021–2031. PMLR, 2020. 2

[36] Atal Sahu, Aritra Dutta, Ahmed M Abdelmoniem, Trambak Banerjee, Marco Canini, and Panos Kalnis. Rethinking gradient sparsification as total error minimization. *Advances in Neural Information Processing Systems*, 34:8133–8146, 2021. 2

[37] Ahmad Sajedi, Samir Khaki, Ehsan Amjadian, Lucy Z. Liu, Yuri A. Lawryshyn, and Konstantinos N. Plataniotis. Datadam: Efficient dataset distillation with attention matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17097–17107, 2023. 2

[38] Saber Salehkaleybar, Arsalan Sharifnassab, and S Jamaloddin Golestani. One-shot federated learning: theoretical limits and algorithms to achieve them. *Journal of Machine Learning Research*, 22(189):1–47, 2021. 2

[39] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems*, 31(9):3400–3413, 2019. 2

[40] 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 *Fifteenth annual conference of the international speech communication association*, 2014. 2

[41] Arsalan Sharifnassab, Saber Salehkaleybar, and S Jamaloddin Golestani. Order optimal one-shot distributed learning. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[42] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, 2020. 5

[43] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. 2

[44] Szymon Tworkowski, Konrad Staniszewski, Mikołaj Pacek, Yuhuai Wu, Henryk Michalewski, and Piotr Miłoś. Focused transformer: Contrastive training for context scaling. *arXiv preprint arXiv:2307.03170*, 2023.

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 2

[46] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020. 2

[47] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018. 2

[48] Yuan Wang, Huazhu Fu, Renuga Kanagavelu, Qingsong Wei, Yong Liu, and Rick Siow Mong Goh. An aggregation-free federated learning for tackling data heterogeneity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26233–26242, 2024. 2, 6, 7

[49] Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang. Gradient sparsification for communication-efficient distributed optimization. *Advances in Neural Information Processing Systems*, 31, 2018. 2

[50] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 5

[51] Cong Xie, Sanmi Koyejo, and Indranil Gupta. Asynchronous federated optimization. *arXiv preprint arXiv:1903.03934*, 2019. 2

[52] Yuanhao Xiong, Ruochen Wang, Minhao Cheng, Felix Yu, and Cho-Jui Hsieh. Feddm: Iterative distribution matching for communication-efficient federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16323–16332, 2023. 2

[53] Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Francoise Beaufays. Applied federated learning: Improving google keyboard query suggestions. *arXiv preprint arXiv:1812.02903*, 2018. 2

[54] Ruonan Yu, Songhua Liu, and Xinchao Wang. Dataset distillation: A comprehensive review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1):150–170, 2024. 2

[55] Junyuan Zhang, Songhua Liu, and Xinchao Wang. One-shot federated learning via synthetic distiller-distillate communication. In *Advances in Neural Information Processing Systems*, pages 102611–102633, 2024. 2, 6, 7

[56] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, page 649–657, 2015. 5

[57] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. *arXiv preprint arXiv:2110.04181*, 2021. 2

[58] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, pages 12674–12685. PMLR, 2021. 2

[59] Fei Zheng, Chaochao Chen, Lingjuan Lyu, and Binhui Yao. Reducing communication for split learning by randomized top-k sparsification. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 2023. 2, 6

[60] Yanlin Zhou, George Pu, Xiyao Ma, Xiaolin Li, and Dapeng Wu. Distilled one-shot federated learning. *arXiv preprint arXiv:2009.07999*, 2020. 2