

# Communication-Efficient Federated Learning with Accelerated Client Gradient

Geeho Kim<sup>\* 1</sup>   Jinkyu Kim<sup>\* 1</sup>   Bohyung Han<sup>1,2</sup>  
<sup>1</sup>ECE & <sup>2</sup>IPAI, Seoul National University  
 {snow1234, jinkyu, bhhan}@snu.ac.kr

## Abstract

*Federated learning often suffers from slow and unstable convergence due to the heterogeneous characteristics of participating client datasets. Such a tendency is aggravated when the client participation ratio is low since the information collected from the clients has large variations. To address this challenge, we propose a simple but effective federated learning framework, which improves the consistency across clients and facilitates the convergence of the server model. This is achieved by making the server broadcast a global model with a lookahead gradient. This strategy enables the proposed approach to convey the projected global update information to participants effectively without additional client memory and extra communication costs. We also regularize local updates by aligning each client with the overshoot global model to reduce bias and improve the stability of our algorithm. We provide the theoretical convergence rate of our algorithm and demonstrate remarkable performance gains in terms of accuracy and communication efficiency compared to the state-of-the-art methods, especially with low client participation rates. The source code is available at our project page<sup>1</sup>.*

## 1. Introduction

Federated learning (FL) [27] is a large-scale machine learning framework that learns a shared model in a central server through collaboration with a large number of remote clients with separate datasets. Such a decentralized learning framework achieves a basic level of data privacy because the data stored in local clients are unobservable by the server and other clients. On the other hand, federated learning algorithms are particularly sensitive to communication and computational costs due to limited resources on many clients, such as mobile or IoT devices.

A baseline algorithm of federated learning, FedAvg [27], updates a subset of client models using a gradient descent method based on their local data. The resulting models are

then uploaded to the server for estimating the global model parameters via model averaging. As extensively discussed in the convergence of FedAvg [3, 37–39, 46], multiple local updates conducted before server-side aggregation provide theoretical support and practical benefits of federated learning by reducing communication costs significantly.

Despite its initial success, federated learning faces two key challenges: high heterogeneity in training data distributed over clients and limited client participation rates. Several studies [15, 49] have shown that multiple local updates in the clients with non-*i.i.d.* (independent and identically distributed) data lead to client model drift, in other words, diverging updates in individual clients. Such a phenomenon introduces a high variance issue in FedAvg steps for global model updates, which hampers convergence to the optimal average loss over all clients [13, 17, 25, 26, 41, 42]. The challenge related to client model drift is exacerbated when the client participation rate per communication round is low.

To properly address the issue of client heterogeneity, we propose a novel federated learning algorithm, Federated averaging with Accelerated Client Gradient (FedACG), which conveys the momentum of the global gradient to clients and enables the momentum to be incorporated into the local updates in the individual clients. Specifically, FedACG transmits the global model integrated with the global momentum in the form of a single message, which allows each client to perform its local gradient update step along the landscape of the global loss function. This approach is effective in reducing the gap between global and local losses. In addition, FedACG adds a regularization term in the objective function of clients to make the local gradients more consistent across clients. We show that subtle differences in federated learning algorithms can have a significant impact on the final results and discuss the behavior of FedACG together with related methods.

The main contributions of this paper are summarized as follows.

- We propose a simple yet effective federated optimization algorithm, called FedACG, that proactively adjusts the initial local model using a lookahead gradient and aligns

<sup>\*</sup>indicates equal contribution.

<sup>1</sup> <https://github.com/geehokim/FedACG>

the gradients of individual clients with that of the server.

- FedACG is free from additional communication costs, extra computation in the server, and memory overhead of clients; these properties are desirable for the real-world settings of federated learning.
- FedACG demonstrates its outstanding performance in terms of communication efficiency and robustness to client heterogeneity, especially with the low client participation rates.
- We introduce a federated learning benchmark<sup>1</sup> to facilitate the evaluation of federated learning algorithms. The benchmark contains the implementations of various algorithms including FedACG.

The rest of the paper is organized as follows. We first review the prior works in Section 2. Section 3 discusses the technical details of the proposed approach. Section 4 validates its effectiveness. We conclude our paper in Section 5.

## 2. Related Work

Federated learning is first introduced by McMahan et al. [27]. They formulate the problem and propose FedAvg as a solution to address main challenges in federated learning, such as massively distributed clients and partial client participation. Subsequent works attempt to address the challenge of non-*i.i.d.* client data in federated learning empirically [49] and derive convergence rates depending on the level of heterogeneity [13, 17, 25, 26, 41, 42].

There exists a long line of research on client-side optimization aimed at reducing the divergence of clients from the global model. FedProx [25] penalizes the difference between the server and client parameters, while FedDyn [1] and FedPD [48] adopt cumulative gradients of each client for dynamic regularization of local updates. FedDC [10] introduces the auxiliary drift variables of each client to reduce the impact of the local drift on the global objective. Another line of work adopts variance reduction techniques in client updates to eliminate inconsistent updates across local models. SCAFFOLD [15] and Mime [16] use control variates for local updates, while FedDANE [24] adds a gradient correction term based on the server gradient. FedPA [2] reduces the bias in client updates by estimating the global posterior on the client side. On the other hand, some approaches adopt a contrastive loss [23, 28, 35], knowledge distillation [19, 22], logit calibration [47], feature decorrelation [36], or a generative model [50] to ensure the similarity between the representations in the global and local models. FedSAM [32] and FedASAM [4] apply SAM [9] as a client-side optimizer to reduce the gap between global and local losses. However, most of these methods require full participation [18, 28, 48], additional communication costs [7, 10, 15, 16, 24, 45, 50], or extra client storage [1, 10, 15, 23], which are problematic in real-

istic federated learning scenarios.

Momentum-based optimization techniques have also been explored for the stability and speed-up of convergence. These approaches incorporate a momentum SGD [14, 40] or an adaptive gradient-descent method [4, 33] into server model updates while FedCM [45] and FedADC [30] employs global momentum to correct gradients in local updates. STEM [18] and FedGLOMO [7] apply the STORM algorithm [6] to both server- and client-level SGD procedures to reduce variance in server model updates. Although these methods require additional communication overhead to transmit the global momentum for local updates, FedACG saves the costs by broadcasting the momentum-integrated model as a single message.

Meanwhile, there is another set of works that aims to reduce the communication costs per round by compressing the transmitted model. FedPAQ [34], FedCAMS [43], and FedCOMGATE [11] use low-bit precision to quantize the communicated messages, while FedPara [29] reparameterizes the model parameters using a low-rank Hadamard product. These works are orthogonal to ours but can be integrated into our algorithm.

## 3. Proposed Approach

### 3.1. Preliminaries

**Problem setup** Let  $\mathcal{L}_i(\theta) := \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_i} [\ell_i((\mathbf{x}, y); \theta)]$  be the empirical loss function of the client  $C_i \in \{C_1, \dots, C_N\}$  with a local dataset denoted by  $\mathcal{D}_i$ . Then, our goal is to train a model that minimizes the average loss of all clients as follows:

$$\min_{\theta} \left\{ \mathcal{L}(\theta) := \sum_{i=1}^N \omega_i \mathcal{L}_i(\theta) \right\}, \quad (1)$$

where  $\theta$  is the parameter of the global model and  $\omega_i$  is the normalized weight of the  $i^{\text{th}}$  client, which is proportional to the size of the local dataset  $|\mathcal{D}_i|$ . We focus on the non-*i.i.d.* data setting, where local datasets have heterogeneous distributions. Note that the communication of raw data between clients and the server is strictly prohibited in principle due to privacy concerns.

**FedAvg algorithm** FedAvg [27] is a standard solution of federated learning, where the server simply aggregates all the participating client models to obtain the global model. Specifically, in the  $t^{\text{th}}$  communication round, the server broadcasts the latest global model, represented as  $\theta^{t-1}$  to the active clients in  $\mathcal{S}_t \subseteq \{C_1, \dots, C_N\}$ . Each participating client optimizes its local model by using the global model as its initial point, *i.e.*,  $\theta_{i,0}^t := \theta^{t-1}$ . After  $K$  iterations of the local training, each client uploads its local updates  $\Delta_i^t := \theta_{i,K}^t - \theta_{i,0}^t$  to the server, and then the server aggregates them as  $\Delta^t := \sum_{i \in \mathcal{S}_t} \omega_i \Delta_i^t$ . The server constructs

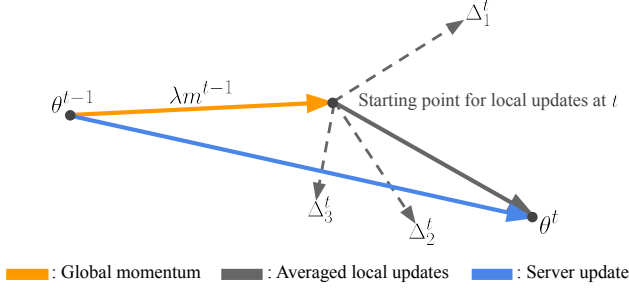


Figure 1. **An illustration of the proposed accelerated client gradient method.** We first partially update the global model in the direction of the global momentum (orange) and then aggregate local updates (gray), resulting in the server model in the next round (blue). This anticipatory update aligns individual local updates with the global gradient, achieving speed-up of convergence.

the next server model  $\theta^t := \theta^{t-1} + \Delta^t$  for broadcasting in the next round. Due to non-*i.i.d.* data and limited client participation rate in each round of training, FedAvg suffers from client drift [15]. Such a phenomenon results in the inconsistent updates of client models caused by overfitting to local data of individual clients, which consequently leads to the high variance of the global model. This tendency is aggravated over multiple communication rounds in federated learning because each client initializes its parameters using the global model.

### 3.2. Federated averaging with Accelerated Client Gradient (FedACG)

To reduce the inconsistency between the local models and the divergence of the resulting global model, we incorporate global momentum into the local models to guide local updates.

**Accelerated client gradient** The main idea of FedACG is to revise the initialization of client models using the global model integrated with a global gradient, allowing more effective and stable updates of local models. Since the direct computation of the global gradient is impractical in FL, we utilize the global momentum  $m^{t-1}$  as a viable approximation, which is updated as  $m^{t-1} := \lambda m^{t-2} + \Delta^{t-1}$ , at each round. Specifically, in the  $t^{\text{th}}$  communication round, the server augments the recent global model  $\theta^{t-1}$  with the global momentum  $m^{t-1}$ . As illustrated in Figure 1, The server then broadcasts this accelerated global model, represented as  $\theta^{t-1} + \lambda m^{t-1}$ , as a single message to the active clients in  $S_t \subseteq \{1, \dots, N\}$ . Note that  $\lambda \in [0, 1)$  controls the importance of the global momentum. Each participating client optimizes its local model from the momentum-integrated initialization. This proactive initialization allows each client to find its local optimal solution along the trajectory of the global gradient, which improves the consistency

---

#### Algorithm 1 FedACG

---

**Input:**  $\beta, \lambda$ , initial server model  $\theta^0$ , number of clients  $N$ , number of communication rounds  $T$ , number of local iterations  $K$ , local learning rate  $\eta$

Initialize global momentum  $m^0 = 0$

**for each round**  $t = 1, 2, \dots, T$  **do**

    Sample subset of clients  $S_t \subseteq \{C_1, \dots, C_N\}$

    Server sends  $\theta^{t-1} + \lambda m^{t-1}$  for all clients in  $S_t$

**for each client**  $C_i \in S_t$ , **in parallel do**

        Initialize local model  $\theta_{i,0}^t \leftarrow \theta^{t-1} + \lambda m^{t-1}$

**for each local iteration**  $k = 1, 2, \dots, K$  **do**

            Compute mini-batch loss

$f_i(\theta_{i,k-1}^t) \leftarrow \ell_i(\theta_{i,k-1}^t)$

$+ \frac{\beta}{2} \|\theta_{i,k-1}^t - (\theta^{t-1} + \lambda m^{t-1})\|^2$

$\theta_{i,k}^t \leftarrow \theta_{i,k-1}^t - \eta \nabla f_i(\theta_{i,k-1}^t)$

**end**

$\Delta_i^t \leftarrow \theta_{i,K}^t - \theta_{i,0}^t$

        Client sends  $\Delta_i^t$  back to the server

**end**

**In server:**

$\Delta^t = \sum_{i \in S_t} \omega_i \Delta_i^t$

$m^t = \lambda m^{t-1} + \Delta^t$

$\theta^t = \theta^{t-1} + m^t$

**end**

**Return**  $\theta^t$

---

tency of local updates in FedACG. Our approach has a similar motivation with meta-learning [8], where a meta-learner identifies the optimal point to facilitate the optimization in all target tasks. After  $K$  iterations of local training, the server updates its momentum and constructs the next server model, denoted as  $\theta^t := \theta^{t-1} + m^t$ , in preparation for the next round. Algorithm 1 outlines the procedure of FedACG.

**Regularization with momentum-integrated model** In addition to the initial acceleration for local training, we augment the client's loss function with the quadratic term  $\frac{\beta}{2} \|\theta_{i,k}^t - (\theta^{t-1} + \lambda m^{t-1})\|^2$  which penalizes the difference between the local online model  $\theta_{i,k}^t$  and the accelerated global model  $\theta^{t-1} + \lambda m^{t-1}$ . Note that,  $\beta$  controls the intensity of the penalty. The penalized term takes advantage of the global gradient information  $\lambda m^t$  to reduce the variations of client-specific gradients,  $\Delta_i^t$ . This regularization term further enforces the local model not to deviate from the accelerated point, preventing each client from falling into biased local minima.

### 3.3. Discussion

While our formulation has something in common with the existing works that also address client heterogeneity using global gradient information for local updates, FedACG has major advantages. First, unlike [10, 15, 45], the server and

clients in FedACG communicate only model parameters without imposing additional network overhead for transmitting gradients and other information; the server broadcasts  $(\theta^{t-1} + \lambda m^{t-1})$  as a single message and each client sends  $\Delta_i^t$  to the server. This is a critical benefit because the rise in communication cost challenges many practical federated learning applications involving clients with limited network bandwidths. Second, FedACG is robust to the low participation rate of clients and allows new-arriving clients to join the training process without a warm-up period because, unlike [1, 10, 15, 23], clients are supposed to neither store their local states nor use them for model updates.

**Comparison with FedAvgM** Although FedACG apparently looks similar to FedAvgM in the sense that both methods employ the global momentum for optimization, they have critical differences. To analyze the difference between the two algorithms, we decompose the process of updating the global model in FedACG into two steps: 1) updating the previous global model,  $\theta^{t-1}$ , with the global momentum term,  $\lambda m^t$ , and computing the interim model  $\Phi^{t-1} = \theta^{t-1} + \lambda m^{t-1}$ , and 2) updating the interim model with the aggregated local gradients,  $\Delta^t|_{\Phi^{t-1}}$ , to derive the new global model,  $\theta^t = \Phi^{t-1} + \Delta^t$ . While FedAvgM performs the local updates from the previous global model, *i.e.*,  $\Delta^t := \Delta^t|_{\theta^{t-1}}$ , FedACG computes the local updates from the interim model parameter, *i.e.*,  $\Delta^t := \Delta^t|_{\Phi^{t-1}}$ . In other words, the two algorithms have different initializations,  $\theta^{t-1}$  vs.  $\Phi^{t-1}$ , for training local models. As will be discussed in Section 4.3, FedACG's lookahead initialization results in more robust models compared to FedAvgM. Such a difference accumulates over multiple rounds communication and eventually leads to significant performance gaps.

### 3.4. Convergence analysis of FedACG

We now discuss the convergence of the FedACG algorithm in general non-convex FL scenarios. To establish the convergence analysis for FedACG, we make three assumptions; (1) the local loss function  $\mathcal{F}_i(\cdot)$  is  $L$ -smooth, (2) its stochastic gradient  $\nabla f_i(\mathbf{x}) := \nabla \mathcal{F}_i(\mathbf{x}; \mathcal{D}_i)$  is unbiased and possesses a bounded variance, *i.e.*,  $\mathbb{E}_{\mathcal{D}_i} \|\nabla f_i(\mathbf{x}) - \nabla \mathcal{F}_i(\mathbf{x})\| < \sigma^2$ , and (3) the average norm of local gradients is bounded by a function of the global gradient magnitude as  $\frac{1}{N} \sum_{i=1}^N \|\nabla \mathcal{F}_i(\mathbf{x})\|^2 \leq \sigma_g^2 + B^2 \|\nabla \mathcal{F}(\mathbf{x})\|^2$ , where  $\sigma_g \geq 0$  and  $B \geq 1$ . These assumptions are widely used for analyzing the non-convex loss functions in FL in the previous works [1, 15, 16, 32, 33, 45]. Note that our convergence proof is free from the bounded gradient assumption of the global or local loss while it is commonly used for the proofs in momentum-based or adaptive optimization methods [33, 44, 45].

We now state the convergence result of FedACG; the de-

tailed proof is provided in Section D in the supplementary document.

**Theorem 1. (Convergence for non-convex functions)** Suppose that local functions  $\{\mathcal{F}_i\}_{i=1}^N$  are non-convex and  $L$ -smooth. By setting  $\eta \leq \frac{(1-\lambda)^2}{64KL(B^2+1)}$ , FedACG satisfies

$$\begin{aligned} & \min_{t=1, \dots, T} \mathbb{E} \|\nabla \mathcal{F}(\theta^{t-1} + \lambda m^{t-1})\|^2 \\ & \leq \mathcal{O} \left( \frac{M_1 \sqrt{LD}}{\sqrt{TK|S_t|}} + \frac{(LD(1-\lambda)^2)^{\frac{2}{3}} M_2^{\frac{1}{3}}}{(T+1)^{\frac{2}{3}}} + \frac{B^2 LD}{T} \right), \end{aligned}$$

where  $M_1^2 := \sigma^2 + K \left(1 - \frac{|S_t|}{N}\right) \sigma_g^2$ ,  $M_2 := \frac{\sigma^2}{K} + \sigma_g^2$ , and  $D := \frac{\mathcal{F}(\theta^0) - \mathcal{F}(\theta^*)}{1-\lambda}$ .

Theorem 1 provides the convergence rate of FedACG, which matches the best convergence rate of existing FL methods [15, 16].

## 4. Experiments

We present empirical evaluation results of FedACG and competing federated learning methods. Refer to the supplementary document for more details about our implementation including hyperparameter setting and results from ablation studies.

### 4.1. Experimental setup

**Datasets** We conduct a set of experiments on three datasets, CIFAR-10 [20], CIFAR-100 [20], and Tiny-ImageNet [21], with different levels of data heterogeneity and participation rates. We generate *i.i.d.* data splits by randomly assigning training examples to individual clients without replacement. For non-*i.i.d.* datasets, we simulate the data heterogeneity by sampling the label ratios from a Dirichlet distribution with a symmetric parameter, 0.3 or 0.6, following the strategies in Hsu et al. [14]. In both *i.i.d.* and non-*i.i.d.* cases, each client holds the same number of examples as in other works [19, 45]. We extend our experiments to the widely adopted FL benchmark, LEAF [5], known for its realistic settings. LEAF introduces heterogeneity in class distribution, data quantity, and feature alignment. For FEMNIST, CelebA, and ShakeSpeare, we use the non-*i.i.d.* data splits provided by LEAF.

**Baselines** We compare our method, FedACG, with state-of-the-art federated learning techniques, which include FedAvg [27], FedProx [25], FedAvgM [14], FedADAM [33], FedDyn [1], FedCM [45], MOON [23], FedMLB [19], FedNTD [22], FedLC [47], FedDC [10], and FedDecorr [36]. The standard ResNet-18 [12] is employed as our backbone network for all experiments after replacing the batch normalization with the group normalization, following Hsieh et al. [13].



Table 1. Results on three benchmarks with two different federated learning settings. For (a) a moderate-scale experiment, the number of clients and the participation rate, are set to 100, and 5%, respectively, while (b) a large-scale setting has 500 clients with a 2% participation rate. The Dirichlet parameter is commonly set to 0.3. Accuracies at the target round and the communication round to reach target test accuracy are based on exponential moving averages with parameter 0.9. The arrows indicate whether higher ( $\uparrow$ ) or lower ( $\downarrow$ ) is better. FedCM<sup>†</sup> and FedDC<sup>‡</sup> require 50% and 100% additional communication costs at each communication round, respectively.

(a) Moderate-scale: 100 clients, 5% participation												
Method	CIFAR-10				CIFAR-100				Tiny-ImageNet			
	Acc. (% , $\uparrow$ )		Rounds ( $\downarrow$ )		Acc. (% , $\uparrow$ )		Rounds ( $\downarrow$ )		Acc. (% , $\uparrow$ )		Rounds ( $\downarrow$ )	
	500R	1000R	81%	85%	500R	1000R	47%	55%	500R	1000R	35%	38%
FedAvg [27]	74.36	82.53	840	1000+	41.88	47.83	924	1000+	33.94	35.37	645	1000+
FedProx [25]	73.70	82.68	826	1000+	42.43	48.32	881	1000+	34.14	35.53	613	1000+
FedAvgM [14]	80.56	85.48	519	828	46.98	53.29	515	1000+	36.32	38.51	416	829
FedADAM [33]	72.33	81.73	908	1000+	44.80	52.48	691	1000+	33.22	38.91	658	945
FedDyn [1]	84.82	88.10	392	646	48.38	55.79	424	883	37.35	41.18	344	573
MOON [23]	83.32	86.30	371	686	53.15	58.37	284	640	36.62	40.33	410	627
FedCM <sup>†</sup> [45]	78.92	83.71	624	1000+	52.44	58.06	293	747	31.61	37.87	694	1000+
FedMLB [19]	74.98	84.04	714	1000+	47.39	54.58	490	1000+	37.20	40.16	415	539
FedLC [47]	78.37	84.79	680	1000+	42.74	47.23	980	1000+	35.03	35.95	500	1000+
FedNTD [22]	76.05	83.78	685	1000+	43.22	49.29	721	1000+	33.91	37.33	547	1000+
FedDC <sup>‡</sup> [10]	<b>86.52</b>	87.47	323	519	54.25	59.01	333	553	40.32	45.51	340	403
FedDecorr [36]	76.62	83.40	728	1000+	43.52	49.17	767	1000+	33.40	34.86	1000+	1000+
<b>FedACG (ours)</b>	85.13	<b>89.10</b>	<b>319</b>	<b>450</b>	<b>55.79</b>	<b>62.51</b>	<b>260</b>	<b>409</b>	<b>42.26</b>	<b>46.31</b>	<b>226</b>	<b>331</b>

(b) Large-scale: 500 clients, 2% participation												
Method	CIFAR-10				CIFAR-100				Tiny-ImageNet			
	Acc. (% , $\uparrow$ )		Rounds ( $\downarrow$ )		Acc. (% , $\uparrow$ )		Rounds ( $\downarrow$ )		Acc. (% , $\uparrow$ )		Rounds ( $\downarrow$ )	
	500R	1000R	73%	77%	500R	1000R	36%	40%	500R	1000R	24%	30%
FedAvg [27]	58.74	71.45	1000+	1000+	30.16	38.11	842	1000+	23.63	29.48	523	1000+
FedProx [25]	57.88	70.75	1000+	1000+	29.28	36.16	966	1000+	25.45	31.71	445	799
FedAvgM [14]	65.85	77.49	753	959	31.80	40.54	724	955	26.75	33.26	386	687
FedADAM [33]	61.53	69.94	1000+	1000+	24.56	34.36	1000+	1000+	21.88	28.08	648	1000+
FedDyn [1]	65.49	77.92	732	936	31.58	41.01	691	927	24.35	29.54	483	1000+
MOON [23]	69.15	78.06	617	872	33.51	42.41	601	828	26.69	31.81	382	741
FedCM <sup>†</sup> [45]	69.27	76.57	742	1000+	27.23	38.79	872	1000+	19.41	24.09	975	1000+
FedMLB [19]	58.68	71.38	1000+	1000+	32.30	42.61	643	803	28.39	33.67	382	579
FedLC [47]	60.16	70.10	1000+	1000+	29.58	36.78	936	1000+	22.14	26.83	676	1000+
FedNTD [22]	60.65	73.20	991	1000+	28.95	36.31	995	1000+	24.67	32.16	475	800
FedDC <sup>‡</sup> [10]	71.86	<b>83.49</b>	518	686	34.64	45.93	569	741	25.72	28.92	420	1000+
FedDecorr [36]	60.01	72.83	1000+	1000+	30.56	38.20	850	1000+	24.34	30.28	499	959
<b>FedACG (ours)</b>	<b>73.61</b>	82.80	<b>484</b>	<b>605</b>	<b>35.68</b>	<b>48.40</b>	<b>505</b>	<b>616</b>	<b>31.47</b>	<b>38.48</b>	<b>246</b>	<b>447</b>

**Evaluation metrics** To evaluate the generalization performance of the methods, we use the entire test set of CIFAR-10, CIFAR-100, and Tiny-ImageNet. Since both the training speed as well as the final accuracy are important factors in federated learning, we measure: (i) the performance achieved at a specified number of rounds and (ii) the number of rounds required for an algorithm to attain the desired level of target accuracy, following Al-Shedivat *et al.* [2]. For the methods that fail to achieve the target accuracies within the maximum communication round, we append a + sign to the communication round number.

## 4.2. Main results

We first present the performance of the proposed approach, FedACG, on CIFAR-10, CIFAR-100, and Tiny-ImageNet

by varying the number of clients, data heterogeneity, and participation rate. Our experiments have been performed in two different settings; one is a moderate scale, which involves 100 clients with a 5% participation rate per round, and the other is with a large number of clients, 500 with a participation rate of 2%. Because the number of clients in the large-scale setting is five times higher than that in the moderate-scale experiment, the number of examples per client is reduced by 80%.

Table 1a demonstrates that FedACG improves accuracy and convergence speed significantly and consistently compared with other federated learning methods in most cases. This is partly because FedACG allows each client to look ahead in the direction of the potential global update and aligns the local model updates with the global gradient tra-

Table 2. Results from reduced participation rates (2% for 100 clients, 1% for 500 clients) on CIFAR-10 and CIFAR-100 with the Dirichlet parameter 0.3. FedCM<sup>†</sup> and FedDC<sup>‡</sup> require 50% and 100% additional communication costs for each communication round, respectively.

Method	CIFAR-10						CIFAR-100					
	100 clients			500 clients			100 clients			500 clients		
	Acc. (% , $\uparrow$ )		Rounds ( $\downarrow$ )	Acc. (% , $\uparrow$ )		Rounds ( $\downarrow$ )	Acc. (% , $\uparrow$ )		Rounds ( $\downarrow$ )	Acc. (% , $\uparrow$ )		Rounds ( $\downarrow$ )
	500R	1000R	78%	500R	1000R	68%	500R	1000R	44%	500R	1000R	35%
FedAvg [27]	65.92	78.13	977	54.71	68.96	949	38.19	44.62	924	26.94	35.69	950
FedProx [25]	65.78	75.82	1000+	55.18	69.80	919	36.69	45.16	921	26.92	35.41	963
FedAvgM [14]	68.09	79.91	748	57.82	71.12	812	39.24	53.47	504	29.29	39.36	755
FedADAM [33]	68.09	78.61	978	48.26	54.60	1000+	40.95	51.14	592	18.21	23.70	1000+
FedDyn [1]	74.27	80.20	660	54.86	70.78	858	38.94	48.88	716	27.86	36.31	896
MOON [23]	71.52	75.42	1000+	<b>64.55</b>	73.89	645	39.91	46.51	730	28.29	36.37	886
FedCM <sup>†</sup> [45]	52.45	64.50	1000+	49.21	60.38	1000+	14.52	23.06	1000+	16.32	22.59	1000+
FedMLB [19]	65.85	79.45	899	52.81	66.86	1000+	40.09	53.34	565	29.78	39.64	724
FedLC [47]	72.90	80.90	736	54.89	68.31	967	39.70	42.10	1000+	27.73	35.24	918
FedNTD [22]	69.11	80.43	797	54.53	68.69	961	38.13	48.03	708	27.56	35.86	932
FedDC <sup>‡</sup> [10]	<b>77.76</b>	82.86	<b>473</b>	60.56	75.06	681	41.50	51.37	670	29.14	38.84	789
FedDecorr [36]	71.29	78.99	817	56.62	70.24	845	39.42	48.45	718	31.03	38.70	705
<b>FedACG (ours)</b>	76.36	<b>84.73</b>	543	63.70	<b>76.45</b>	<b>618</b>	<b>49.56</b>	<b>56.89</b>	<b>358</b>	<b>31.74</b>	<b>45.18</b>	<b>581</b>

Table 3. Results on the Dirichlet (0.3) split of CIFAR-100 with dynamic client updates during training: we maintain 250 clients but each client is replaced with a probability of 0.5 at every 100 rounds. The experiment runs for 10 stages and the client participation ratio is 4%.

Method	Acc. (% , $\uparrow$ )		Rounds ( $\downarrow$ )	
	500R	1000R	30%	38%
FedAvg [27]	28.61	35.87	577	1000+
FedProx [25]	28.17	35.89	602	1000+
FedDyn [1]	29.45	38.47	517	941
MOON [23]	30.88	39.57	430	852
FedNTD [22]	28.45	36.26	578	1000+
FedDC <sup>‡</sup> [10]	31.35	36.82	469	1000+
<b>FedACG (ours)</b>	<b>32.70</b>	<b>41.51</b>	<b>376</b>	<b>769</b>

jectory. Note that FedCM and FedDC respectively require  $1.5\times$  and  $2\times$  network costs for each communication round since they communicate the current model and the associated gradient information per round, while the rest of the algorithms only need to transmit model parameters.

For the large-scale setting, Table 1b illustrates the outstanding performance in the three benchmarks, except for the accuracy at 1K rounds on CIFAR-10. A noticeable observation is that the overall performance is lower than the case with a moderate number of clients. This is because the number of training data for each client decreases and each client suffers more from heterogeneous data distributions. Nevertheless, we observe that FedACG outperforms other methods consistently in most cases; the accuracy gap between FedACG and its strongest competitor becomes larger in these more challenging scenarios. The results from the large-scale experiments exhibit the robustness of FedACG to the data heterogeneity and low client participation rates. We present more comprehensive results for the convergence

Table 4. Contribution of individual components in FedACG. The results are measured after 1K rounds on CIFAR-10 and CIFAR-100 with 2% participation rate over 500 clients.

Server update w/ momentum	Accelerated gradient	Local reg.	CIFAR-10	CIFAR-100
			71.45	38.11
		✓	70.75	36.16
✓			77.49	40.54
✓		✓	76.16	44.64
✓	✓		82.20	46.80
✓	✓	✓	<b>82.80</b>	<b>48.40</b>

of FedACG in the supplementary document.

### 4.3. Analysis

**Effect of low participation rate** One of the critical challenges in federated learning is the partial participation of clients, which can slow down the convergence of the global model. To verify the robustness of FedACG to low client participation rates, we conduct experiments with 500 clients and a participation rate as low as 1%. Since the numbers of local epochs and iterations are set to 5 and 50, respectively, each client has little training opportunity with few training examples and client heterogeneity increases significantly. As shown in Table 2, FedACG outperforms the other methods in most cases, with the performance gap between FedACG and the second best method, FedDC, being even more significant than when the participation rate is 2% with 500 clients, from -0.69%p to 1.39%p on CIFAR-10 and from 2.47%p to 6.34%p on CIFAR-100 at round 1000. This is partly because the local states in FedDC become stale quickly in this scenario, requiring extra iterations for convergence, whereas FedACG is not affected by this issue.

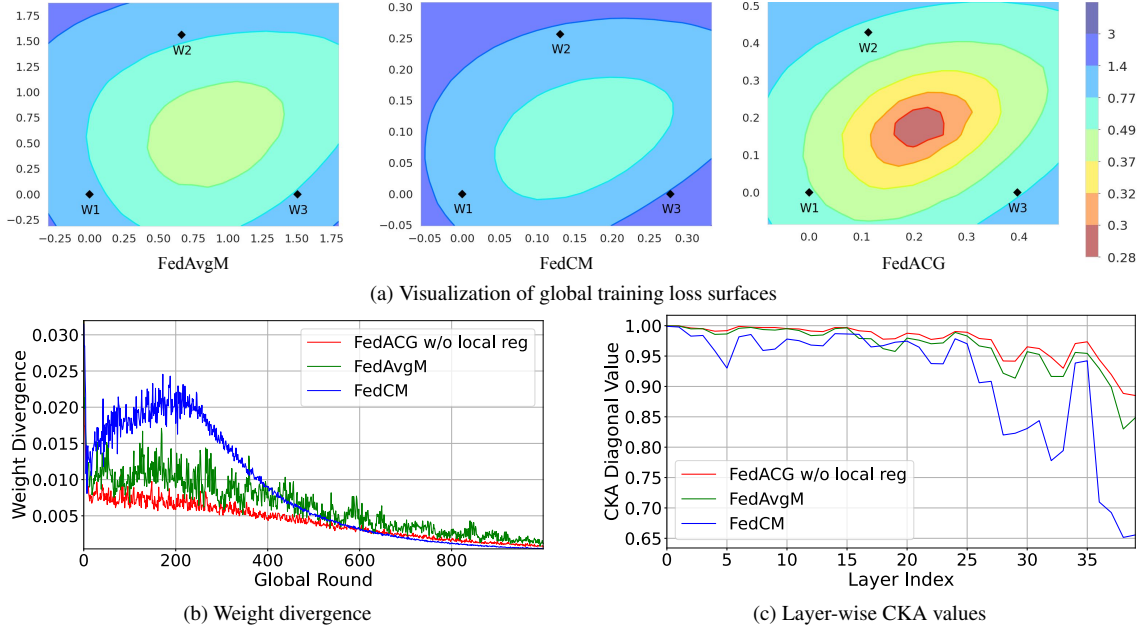


Figure 2. **Benefit of accelerated client gradient.** For FedAvgM, FedCM, and FedACG (without local regularization for fair comparisons) on CIFAR10, we visualize (a) global training loss surfaces with three local models as black circles in the parameter space, (b) weight divergence, and (c) layer-wise CKA values. In (c), the  $x$ -axis denotes the layer index of ResNet-18 while the  $y$ -axis corresponds to CKA values measured on the global validation set.

**Evaluation on dynamic client set** Since FedACG does not require storing local model history for local updates, it is conceptually better suited for scenarios with newly participating clients. To validate this property, we conduct an experiment, where we maintain 250 clients in each round but replace half of the clients on average every 100 rounds by setting the replacement probability of each client to 0.5. The experiment has been performed on CIFAR-100 with Dirichlet (0.3) splits, assuming a participation ratio of 4% for each communication round. Table 3 shows that FedACG outperforms FedAvg and FedDyn. Note that FedDyn performs worse than FedAvg since the client models suffer from heterogeneity and divergence when new clients have no informative local states.

**Ablation study** Table 4 presents the contributions of individual components in the experiment on CIFAR-10 for the large-scale setting. We observe that the accelerated client gradient for local training makes a more critical impact on accuracy after 1,000 rounds. It is worth noting that the proposed regularization term in the local loss function shows a larger performance gain when used with the accelerated client gradient while employing the regularization term alone does not necessarily lead to a beneficial outcome on CIFAR-10 and CIFAR-100.

**Comparison with FedAvgM and FedCM** To better understand the effectiveness of the accelerated client gradient,

we compare two momentum-based algorithms, FedAvgM and FedCM, by visualizing global loss surfaces, weight divergence, and layer-wise CKA values during training. Figure 2a highlights a better generalization of FedACG’s local models to global loss compared to other methods. Figures 2b and 2c reveal that the local models of FedACG exhibit less divergence in the parameter space and more consistent feature representations, respectively. These findings demonstrate that the accelerated client gradient in FedACG effectively mitigates client drift stemming from data heterogeneity.

**Hyperparameters** We test the accuracy of our algorithm for the Dirichlet (0.3) and *i.i.d.* splits by varying the values of  $\lambda$  and  $\beta$ , which control the momentum integration of the server model and the weight of the proximal term, respectively. As shown in Table 5a, the performance of FedACG remains stable in a range of  $\lambda$  values from 0.75 to 0.9. Despite minor fluctuations, the accuracy remains high, peaking at  $\lambda = 0.85$ . Table 5a also shows that the accuracy is stable with respect to  $\beta$ .

**Integration into quantization approaches** Our framework is orthogonal to quantization-based FL algorithms, which enables seamless integration with such approaches. Table 6 presents consistent and remarkable improvements when combining FedACG with quantization-based algorithms, such as FedPAQ [34].

Table 5. Ablation study for  $\lambda$  (a) and  $\beta$  (b), w.r.t the accuracy at  $1K^{\text{th}}$  round on CIFAR-10 with 2% participation and 500 clients.

(a) Sensitivity of FedACG with respect to $\lambda$				
$\lambda$	0.75	0.8	0.85	0.9
Dir(0.3)	81.32	82.52	82.80	82.64
<i>i.i.d.</i>	85.52	86.82	86.83	86.16

(b) Sensitivity of FedACG with respect to $\beta$				
$\beta$	0.001	0.01	0.1	1
Dir(0.3)	82.10	82.80	82.32	82.44
<i>i.i.d.</i>	86.54	86.83	86.72	85.92

Table 6. Integration of FedACG into quantization-based federated learning approach under non-*i.i.d.* settings on CIFAR-100; 5% participation rate for 100 clients and 2% for 500 clients. The Dirichlet parameter is commonly set to 0.3.

Method	100 clients		500 clients	
	500R	1000R	500R	1000R
FedAvg [27]	41.88	47.83	30.16	38.11
FedPAQ [34]	36.57	41.99	23.80	30.21
FedACG + FedPAQ	<b>43.04</b>	<b>50.53</b>	<b>31.94</b>	<b>39.55</b>

#### 4.4. Experiments on realistic datasets

We conduct experiments on additional realistic datasets, FEMNIST and CelebA in LEAF [5], which include other kinds of non-*i.i.d.* scenarios such as feature skewness and data imbalance between clients. For these experiments, we set the number of clients to 2,000, with data splits following [5], and randomly select 5 clients to participate in training during each communication round. We employ a two-layer CNN for FEMNIST and a four-layer CNN for CelebA. Table 7 illustrates that FedACG also outperforms other baselines on both datasets for most cases, highlighting its robustness to heterogeneity with data quantity and feature alignment. Note that, while FedACG requires 20 more communication rounds than FedDC to reach the target accuracy on FEMNIST, it sends 56.8% fewer parameters than FedDC.

We also evaluate FedACG in a different domain, next word prediction task, on the ShakeSpeare in LEAF, which also involves a significant data imbalance between clients. We adopt an LSTM as the backbone network, and the client participation rate per round is set to 5%. Table 8 presents that FedACG is also effective for the language domain, while FedCM exhibits poor performance even with extensive hyperparameter tuning.

## 5. Conclusion

This paper addresses a realistic federated learning scenario, where a large number of clients with heterogeneous data

Table 7. Results on the realistic datasets involving feature skewness and data imbalance between clients. FedCM<sup>†</sup> and FedDC<sup>‡</sup> require 50% and 100% additional communication costs per communication round, respectively.

Method	FEMNIST		CelebA	
	Acc. (% , $\uparrow$ )	Rounds ( $\downarrow$ )	Acc. (% , $\uparrow$ )	Rounds ( $\downarrow$ )
	500R	78%	500R	88%
FedAvg [27]	78.38	328	89.92	134
FedProx [25]	78.34	328	89.90	132
FedAvgM [14]	78.37	256	89.85	113
FedADAM [33]	75.96	500+	87.00	500+
FedDyn [1]	79.80	227	89.74	126
MOON [23]	78.33	336	87.95	500+
FedCM <sup>†</sup> [45]	72.79	500+	88.89	222
FedNTD [22]	78.42	330	89.31	122
FedDC <sup>‡</sup> [10]	80.11	<b>149</b>	88.97	126
<b>FedACG (ours)</b>	<b>80.61</b>	169	<b>90.09</b>	<b>108</b>

Table 8. Results in the language domain on the next word prediction task under non-*i.i.d.* setting using the ShakeSpeare dataset.

Method	Acc. (% , $\uparrow$ )		Rounds ( $\downarrow$ )	
	500R	1000R	42%	45%
FedAvg [27]	45.01	46.55	94	500
FedProx [25]	45.09	46.29	99	477
FedAvgM [14]	44.63	45.91	63	690
FedADAM [33]	44.89	44.30	68	1000+
FedDyn [1]	39.23	44.10	749	1000+
MOON [23]	42.02	42.65	499	1000+
FedCM <sup>†</sup> [45]	—	—	—	—
FedNTD [22]	45.01	46.5	94	513
FedDC <sup>‡</sup> [10]	30.62	44.27	926	1000+
<b>FedACG (ours)</b>	<b>46.36</b>	<b>48.23</b>	<b>57</b>	<b>290</b>

and limited participation constraints hinder the convergence and performance of trained models. To tackle these issues, we proposed a novel federated learning framework that aggregates past global gradient information for guiding client updates and regularizes the local update directions aligned with the global information. The proposed algorithm provides global gradient information to individual clients without incurring additional communication or memory overhead. We made a formal proof of the convergence rate of the proposed approach. FedACG demonstrates the effectiveness in terms of robustness and communication efficiency in the presence of client heterogeneity through extensive evaluation on multiple benchmarks.

**Acknowledgments** This work was partly supported by Samsung Advanced Institute of Technology (SAIT), and by the National Research Foundation of Korea grant [No.2022R1A2C3012210] and the Institute for Information & communications Technology Planning & Evaluation (IITP) grants [2022-0-00959; 2021-0-02068; 2021-0-01343] funded by the Korea government (MSIT).



## References

- [1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *ICLR*, 2021. [2](#), [4](#), [5](#), [6](#), [8](#), [11](#), [13](#), [14](#)
- [2] Maruan Al-Shedivat, Jennifer Gillenwater, Eric Xing, and Afshin Rostamizadeh. Federated learning via posterior averaging: A new perspective and practical algorithms. In *ICLR*, 2021. [2](#), [5](#)
- [3] Debraj Basu, Deepesh Data, Can Karakus, and Suhas N Digavi. Qsparse-local-sgd: Distributed sgd with quantization, sparsification, and local computations. *IEEE Journal on Selected Areas in Information Theory*, 1(1):217–226, 2020. [1](#)
- [4] Debora Caldarola, Barbara Caputo, and Marco Ciccone. Improving generalization in federated learning by seeking flat minima. In *ECCV*, 2022. [2](#)
- [5] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečný, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. In *NeurIPS*, 2019. [4](#), [8](#)
- [6] Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. In *NeurIPS*, 2019. [2](#)
- [7] Rudrajit Das, Anish Acharya, Abolfazl Hashemi, Sujay Sanghavi, Inderjit S Dhillon, and Ufuk Topcu. Faster non-convex federated learning via global and local momentum. *arXiv preprint arXiv:2012.04061*, 2020. [2](#)
- [8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. [3](#)
- [9] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *ICLR*, 2021. [2](#)
- [10] Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In *CVPR*, 2022. [2](#), [3](#), [4](#), [5](#), [6](#), [8](#), [13](#), [14](#)
- [11] Farzin Haddadpour, Mohammad Mahdi Kamani, Aryan Mokhtari, and Mehrdad Mahdavi. Federated learning with compression: Unified analysis and sharp guarantees. In *AISTATS*, 2021. [2](#)
- [12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. [4](#)
- [13] Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. The non-iid data quagmire of decentralized machine learning. In *ICML*, 2020. [1](#), [2](#), [4](#)
- [14] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019. [2](#), [4](#), [5](#), [6](#), [8](#), [13](#), [14](#)
- [15] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for on-device federated learning. In *ICML*, 2020. [1](#), [2](#), [3](#), [4](#), [17](#)
- [16] Sai Praneeth Karimireddy, Martin Jaggi, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Breaking the centralized barrier for cross-device federated learning. In *NeurIPS*, 2021. [2](#), [4](#)
- [17] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. First analysis of local gd on heterogeneous data. *arXiv preprint arXiv:1909.04715*, 2019. [1](#), [2](#)
- [18] Prashant Khanduri, Pranay Sharma, Haibo Yang, Mingyi Hong, Jia Liu, Ketan Rajawat, and Pramod K Varshney. Stem: A stochastic two-sided momentum algorithm achieving near-optimal sample and communication complexities for federated learning. In *NeurIPS*, 2021. [2](#)
- [19] Jinkyu Kim, Geeho Kim, and Bohyung Han. Multi-level branched regularization for federated learning. In *ICML*, 2022. [2](#), [4](#), [5](#), [6](#), [11](#), [12](#), [13](#), [14](#)
- [20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. [4](#)
- [21] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. [4](#)
- [22] Gihun Lee, Minchan Jeong, Yongjin Shin, Sangmin Bae, and Se-Young Yun. Preservation of the global knowledge by not-true distillation in federated learning. In *NeurIPS*, 2022. [2](#), [4](#), [5](#), [6](#), [8](#), [13](#), [14](#)
- [23] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *CVPR*, 2021. [2](#), [4](#), [5](#), [6](#), [8](#), [13](#), [14](#)
- [24] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Feddane: A federated newton-type method. In *ACSCC*, 2019. [2](#)
- [25] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *MLSys*, 2020. [1](#), [2](#), [4](#), [5](#), [6](#), [8](#), [13](#), [14](#)
- [26] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *ICLR*, 2020. [1](#), [2](#)
- [27] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2017. [1](#), [2](#), [4](#), [5](#), [6](#), [8](#), [13](#), [14](#)
- [28] Xutong Mu, Yulong Shen, Ke Cheng, Xueli Geng, Jiaxuan Fu, Tao Zhang, and Zhiwei Zhang. Fedproc: Prototypical contrastive federated learning on non-iid data. *arXiv preprint arXiv:2109.12273*, 2021. [2](#)
- [29] Hyeon-Woo Nam, Moon Ye-Bin, and Tae-Hyun Oh. Fedpara: Low-rank hadamard product for communication-efficient federated learning. In *ICLR*, 2022. [2](#)
- [30] Emre Ozfatura, Kerem Ozfatura, and Deniz Gündüz. Fedadc: Accelerated federated learning with drift control. In *ISIT*, 2021. [2](#)
- [31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. [11](#)
- [32] Zhe Qu, Xingyu Li, Rui Duan, Yao Liu, Bo Tang, and Zhuo Lu. Generalized federated learning via sharpness aware minimization. In *ICML*, 2022. [2](#), [4](#)

- [33] Sashank J Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *ICLR*, 2021. 2, 4, 5, 6, 8, 13, 14, 17
- [34] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization. In *AISTATS*, 2020. 2, 7, 8
- [35] Seonguk Seo, Jinkyu Kim, Geeho Kim, and Bohyung Han. Relaxed contrastive learning for federated learning. In *CVPR*, 2024. 2
- [36] Yujun Shi, Jian Liang, Wenqing Zhang, Vincent YF Tan, and Song Bai. Towards understanding and mitigating dimensional collapse in heterogeneous federated learning. In *ICLR*, 2023. 2, 4, 5, 6, 11, 12, 13, 14
- [37] Sebastian U Stich. Local sgd converges fast and communicates little. In *ICLR*, 2019. 1
- [38] Sebastian U Stich and Sai Praneeth Karimireddy. The error-feedback framework: Better rates for sgd with delayed gradients and compressed communication. *arXiv preprint arXiv:1909.05350*, 2019.
- [39] Jianyu Wang and Gauri Joshi. Cooperative sgd: A unified framework for the design and analysis of local-update sgd algorithms. *Journal of Machine Learning Research*, 22(213): 1–50, 2021. 1
- [40] Jianyu Wang, Vinayak Tantia, Nicolas Ballas, and Michael Rabbat. Slowmo: Improving communication-efficient distributed sgd with slow momentum. In *ICLR*, 2019. 2
- [41] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *NeurIPS*, 2020. 1, 2
- [42] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 37(6):1205–1221, 2019. 1, 2
- [43] Yujia Wang, Lu Lin, and Jinghui Chen. Communication-efficient adaptive federated learning. In *ICML*, 2022. 2
- [44] Xidong Wu, Feihu Huang, Zhengmian Hu, and Heng Huang. Faster adaptive federated learning. In *AAAI*, 2023. 4
- [45] Jing Xu, Sen Wang, Liwei Wang, and Andrew Chi-Chih Yao. Fedcm: Federated learning with client-level momentum. *arXiv preprint arXiv:2106.10874*, 2021. 2, 3, 4, 5, 6, 8, 11, 13, 14, 17
- [46] Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *AAAI*, 2019. 1
- [47] Jie Zhang, Zhiqi Li, Bo Li, Jianghe Xu, Shuang Wu, Shouhong Ding, and Chao Wu. Federated learning with label distribution skew via logits calibration. In *ICML*, 2022. 2, 4, 5, 6, 11, 12, 13, 14
- [48] Xinwei Zhang, Mingyi Hong, Sairaj Dhople, Wotao Yin, and Yang Liu. Fedpd: A federated learning framework with optimal rates and adaptivity to non-iid data. In *arXiv preprint arXiv:2005.11418*, 2020. 2
- [49] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018. 1, 2
- [50] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *ICML*, 2021. 2