Gradient-Aware Participation for Energy Reduction in Federated Learning with Extreme Label Skew

Gradient-Aware Participation for Energy Reduction in Federated Learning with Extreme Label Skew

Erich Malan, Valentino Peluso, Andrea Calimera, Enrico Macii*
Department of Control and Computer Engineering, Politecnico di Torino, Turin, Italy
*Interuniversity Department of Regional and Urban Studies and Planning, Politecnico di Torino, Turin, Italy
{erich.malan, valentino.peluso, andrea.calimera, enrico.macii}@polito.it

***Abstract*—Federated Learning (FL) enables distributed clients to train a global classification model collaboratively while preserving data privacy. A major challenge in FL is ensuring efficient training with limited computing and communication resources, especially when clients' datasets contain samples from a restricted subset of target classes, a problem known as *extreme label skew*. Under such a condition, model updates from clients are biased toward their local data distributions, resulting in slow convergence and increased energy consumption due to the need for additional training rounds. This paper introduces *FL with Gradient-Aware Participation* (FedGAP), a novel strategy aimed at reducing energy consumption while preserving model accuracy even with extreme label skew. FedGAP dynamically adjusts the cohort size, i.e., the number of participating clients per training round, based on the evolution of the global model's pseudo-gradient. By detecting stagnant phases where progress toward convergence stalls, FedGAP increases the cohort size to escape suboptimal regions and accelerate learning, thereby minimizing the waste of resources. Experiments on CIFAR-10 and CIFAR-100 demonstrate that FedGAP achieves up to 2.74× greater energy efficiency compared to state-of-the-art methods without compromising accuracy.**

***Index Terms*—Cross-device Federated Learning, Energy Optimization, Internet of Things, Extreme Label Skew**

I. Introduction and Motivations

Cross-device Federated Learning (FL) is a collaborative training protocol where distributed clients, such as mobile and IoT devices, implement machine learning (ML) without sharing their local data, ensuring compliance with data protection regulations [1]. In a typical FL architecture, a central server maintains a global model that is iteratively updated over multiple training rounds. Each round consists of three main stages: (i) the server randomly samples a subset of clients to form a cohort of workers; (ii) the selected workers download the global model, run a local training with private data, and upload the updates to the server; (iii) the server aggregates the collected updates improving the global model.

While FL has shown promise in a wide range of IoT applications [2], like image classification [3], fall detection [4], and intrusion detection systems [5], deployment in real-world scenarios poses several challenges. One of the main concerns is the energy budget associated with clients. Tiny and perhaps battery-powered devices might lack enough resources to keep on the required computing and transmission tasks over multiple rounds [6]. Moreover, each device usually collects data samples representing only a subset of the target classes.

TABLE I
Energy cost of FL on CIFAR-10 under homogeneous label distribution (L=10) and extreme label skew ($L\leq$4).

L	W	$A_{\text{th}} > 79.0\%$	$A_{\text{th}} > 79.5\%$	$A_{\text{th}} > 80.0\%$
=10	5	28.3	31.0	35.9
≤4	5	227.2	-	-
	10	298.7	317.3	455.2
	20	398.2	514.4	599.0

For instance, in intrusion detection services from network logs, only a few clients record events of illicit traffic, and none has examples of all types of attacks [5]. This kind of label distribution, known as *extreme label skew*, leads to an increase in the number of rounds required for training and substantial energy overhead for the clients. The problem is further exacerbated by partial client participation, as even the combined datasets of selected workers may still miss some target classes. Over successive rounds, the under-representation of certain classes can slow convergence and degrade accuracy.

The common ground in these issues is the cohort size, which becomes key in balancing the quality of training with energy consumption. Preliminary results reported in Table I demonstrate the existing relationship using as a benchmark a five-layer convolutional neural network trained on the CIFAR-10 dataset (more details about the setup are in Sec. IV-A). The table shows the average energy cost per client calculated as the average number of rounds each client must participate in to let the global model meet predefined accuracy thresholds A_{th}. The experiments, which involved 100 clients in a 5000-round training, explore configurations characterized by different numbers of classes each client holds L and increasing cohort sizes W. With an unskewed label distribution (L=10), the target accuracies are reached in a few tens of rounds, with energy costs ranging from 28.3 for $A_{\text{th}} > 79.0\%$ to 35.9 for $A_{\text{th}} > 80.0\%$. However, under extreme label skew ($L\leq$4), both accuracy and energy are affected. With a small cohort (W=5), the training succeeds in reaching the lowest target $A_{\text{th}} > 79.0\%$ (it fails for 79.5% and 80.0%), but the energy cost gets 8.03× larger than that observed in the unskewed label distribution (227.2 vs. 28.3). Interestingly, larger cohort sizes (W=10 and W=20) help mitigate the label skew effect providing the aggregation stages with local updates obtained from a more representative set of data and labels, but energy costs get drastically high as clients participate in more rounds, up to 16.69× higher

for $A_{\text{th}} > 80.0\%$ (599.0 vs. 35.9). This analysis reveals that a fixed cohort size either results in poor accuracy or high energy consumption and suggests the cohort size itself is a knob for quality-energy trade-off.

We thereby propose *FL with Gradient-Aware participation* (FedGAP), a novel resource management protocol that dynamically adjusts the cohort size based on the training progress. The assumption confirmed by our empirical observations is that during the first initial rounds, when the model is far from convergence, even small cohorts can provide effective updates to improving the global model; by contrast, engaging too many workers incurs a waste of energy, which should be avoided. As training moves further and the model approaches a stagnant phase where progress stalls, larger cohorts help mitigate the bias in local updates and boost convergence. FedGAP thus implements a cohort sizing mechanism triggered by the stagnant phases of learning inferred from the evolution of the global model pseudo-gradient. The dynamic scheme ensures efficient resource allocation and higher quality of training, even when labels are extremely skewed. Experiments on the CIFAR-10 and CIFAR-100 datasets validate the effectiveness of our proposal, showing that FedGAP reduces the energy consumption at the client side by 2.74× (best case) compared to state-of-the-art methods, still preserving, and in many cases improving, the attainable accuracy.

II. Related Work

Substantial efforts have recently been dedicated to enhancing FL energy efficiency. Research focused on optimization techniques designed to reduce both computational and communication energy costs. Notable examples include widely adopted model compression methods, such as pruning [7] and low-precision training [8], as well as strategies specifically developed for FL, such as partial synchronization via parameter [9] or tensor [10]–[12] freezing. While effective in reducing energy consumption, these techniques often incur accuracy degradation and lack adaptation to label-skewed datasets. Other studies directly addressed extreme label skew by focusing on accelerating model convergence. We classified these approaches into four groups based on the stage they operate: on-device training, label augmentation, model aggregation, and client sampling.

a) On-device Training: The optimization of local training has been extensively studied, with emphasis on the design of effective loss functions. Cross-entropy loss, commonly used for classification tasks, performs poorly on imbalanced datasets and is particularly ineffective under extreme label skew. Several works thus proposed tailored loss functions that can generate unbiased updates regardless of the local data distribution [13]–[16], as unbiased updates are supposed to accelerate convergence. For instance, FedRS [13] introduces a modified cross-entropy loss that suppresses gradients corresponding to missing classes. Other methods [17]–[20] employ knowledge distillation from the global model to retain information about missing classes but incur extra energy costs due to the required forward pass of the global model during each local training iteration.

b) Label Augmentation: An alternative approach to address label imbalance is to retrieve auxiliary data from public repositories [21] or generate synthetic training examples [22] for missing classes. However, these methods rely on the availability of appropriate data and incur additional energy costs for data retrieval or generation.

c) Model Aggregation: Server-side techniques modify the aggregation process to account for label skew. For instance, FedConcat [23] creates clusters of clients with similar label distributions and maintains separate model versions for each cluster; these versions are later merged into a unified model during the final stage of the learning process. FedConcat was built under the assumption of full client participation at each training round, which is unrealistic in cross-device settings, and adapting the underlying technique for partial participation might result problematic due to difficulties in learning representative clusters.

d) Clients' Sampling: Efficient client sampling policies can accelerate convergence, thereby reducing the clients' energy consumption. Two main approaches have garnered research interest: (i) prioritizing clients based on their effectiveness and (ii) dynamically adjusting cohort sizes to enhance data diversity. Several studies addressed the first approach [24], including advanced methods that integrate energy-aware policies based on selection metrics that consider both the update quality and the remaining battery charge of the clients [25]. Still, designing general metrics to assess client contributions remains a challenging task. The second approach, which is the focus of our work, remains largely under-explored. AdaFL [26] represents an initial attempt. It proposes an incremental strategy where cohort size increases at fixed intervals. Our proposed method, FedGAP, differentiates from AdaFL in two main aspects: (i) it dynamically adjusts the cohort size based on the global model's training progress rather than at predefined intervals, and (ii) it adapts to varying label distributions and dataset complexities. As demonstrated by our experiments (Sec. IV-B), these factors enable concurrent energy reduction and model accuracy improvement. More recently, FedDR [27] proposed a two-phase participation policy consisting of a longer phase with a small cohort followed by a shorter phase with an increased cohort. Its implementation requires knowledge of the clients' remaining data traffic to control the transition from low to high participation, which occurs when the remaining budget for all clients drops below a certain threshold. FedGAP differs as it requires no prior knowledge of the clients' resources.

III. Methodology

A. FL System

We consider a centralized, synchronous FL based on FedAvg [28], as outlined in Algorithm 1. Table II summarizes the notation used. The system architecture consists of a central server coordinating a fleet of end nodes $\mathcal{S}$. Each device $s \in \mathcal{S}$ maintains a local dataset $\mathcal{D}_s$. At the beginning, the server

Algorithm 1: FedAvg Workflow.

1 $\mathbf{w}^1 \leftarrow$ Random weights
2 **for** r = *1,...,*R **do**
3 Sample a subset $\mathcal{S}^r$ of clients
  /* On-device Training */
4 **for** $s \in \mathcal{S}^r$ **do in parallel**
5 $\mathbf{w}_s^{r,1} \leftarrow \mathbf{w}^r$
6 $\mathbf{w}_s^{r,K} \leftarrow$ TRAIN($\mathbf{w}_s^{r,1}$, $\mathcal{D}_s$, K)
7 $\boldsymbol{\delta}_s^r \leftarrow \mathbf{w}_s^{r,K} - \mathbf{w}^r$
8 Upload $\boldsymbol{\delta}_s^r$ to the server
  /* Server Aggregation */
9 $\boldsymbol{\delta}^r \leftarrow \frac{1}{|\mathcal{S}^r|} \sum_{s \in S^r} \boldsymbol{\delta}_s^r$
10 $\mathbf{w}^{r+1} \leftarrow \mathbf{w}^r + \boldsymbol{\delta}^r$

initializes the global model $\mathbf{w}^1$ with random weights (line 1) and then starts the training cycle that proceeds iteratively for R rounds (lines 2–10).

In each round r, the server selects a cohort of workers by randomly sampling a subset of clients $\mathcal{S}^r \subseteq \mathcal{S}$ (line 3). Each selected device s downloads the latest version of the global model $\mathbf{w}^r$ from the server (line 5), runs K iterations of local training using its local dataset $\mathcal{D}_s$ (line 6), computes the model updates $\boldsymbol{\delta}_s^r$ as the difference between the locally trained weights and $\mathbf{w}^r$ (line 7), and transmits these updates back to the server (line 8). The server aggregates the collected updates obtaining the "pseudo-gradient" $\boldsymbol{\delta}^r$ (line 9) and updates the global model accordingly obtaining $\mathbf{w}^{r+1}$ (line 10).

In standard FL systems, the cohort size is fixed across training rounds, that is, $|\mathcal{S}^r|$=W, $\forall r \in [1, R]$, with W the desired number of workers per round given as a predefined hyper-parameter. As shown in Sec. I, the choice of cohort size influences the energy consumed by the clients, as it indirectly defines the frequency of their involvement during the training flow. On the other hand, FedGAP promotes W as a direct control variable to reduce energy consumption.

B. Understanding Energy Model & Optimization Goal

We adopted the energy model outlined in [29]. The total energy consumption of a selected device s during a single training round r is denoted as E_s^r and consists of two contributions:

$$E_s^r = E_s^{\text{comp}} + E_s^{\text{comm}} . \tag{1}$$

E_s^{comp} is the energy consumed for computing the local update (lines 6 and 7 in Algorithm 1), which depends on factors such as the CPU specifications, clock frequency, model complexity, and the number of training iterations. E_s^{comm} is the energy consumed for communications with the server (lines 5 and 8 in Algorithm 1), influenced by the technology in use (e.g., 5G or WiFi) and the volume of the transmitted payloads.

The total energy consumption of client s during the whole training process is as follows:

$$E_s^{\text{tot}} = \sum_{r \in [1,R]} \mathbb{1}_{s \in \mathcal{S}^r} \cdot E_s^r , \tag{2}$$

TABLE II
SUMMARY OF NOTATION.

Notation	Description
R	Total number of training rounds
r	Current training round, $r \in [1, R]$
$\mathcal{S}$	Set of devices
$\mathcal{S}^r$	Training cohort sampled at round r, $\mathcal{S}^r \subset \mathcal{S}$
$\mathbf{w}^r$	Global model at round r
K	Number of on-device training steps
$\mathcal{D}_s$	Local dataset of device s
$\mathbf{w}_s^{r,K}$	Local model of device s at round r after K steps
$\boldsymbol{\delta}_s^r$	Difference between $\mathbf{w}_s^{r,K}$ and $\mathbf{w}^r$
$\boldsymbol{\delta}^r$	Pseudo-gradient at round r
$\mathbf{g}^{\mathbf{r}}$	Gradient alignment at round r
g^r	Gradient alignment score at round r

where $\mathbb{1}_{s \in \mathcal{S}^r}$ is an indicator function equal to 1 if client s is selected for training in round r and 0 otherwise.

In cross-device FL, clients show a variety of hardware configurations and network specifications, and hence different latency and power consumption profiles that make energy estimation difficult and highly inaccurate. Abstracting such device-specific features is paramount in federated contexts. We thus opted for a more representative proxy metric called *energy cost*, which measures the average number of rounds a client participates in during training. Formally:

$$\text{Energy Cost} = \frac{\sum_{s \in \mathcal{S}} \sum_{r \in [1,R]} \mathbb{1}_{s \in \mathcal{S}^r}}{|\mathcal{S}|} . \tag{3}$$

This metric offers an intuitive measure of the average effort required from a client, enabling a direct comparison of energy demands across different cohort sizes and setups. In other words, the energy cost in Eq. (3) serves as a hardware-neutral proxy for fair and systemic assessments.

The objective of the proposed FedGAP strategy is to minimize the energy cost required to achieve a target accuracy, introducing a gradient-aware policy that dynamically adjusts the cohort size based on the progress of the global model training.

C. Gradient-Aware Participation

Model training is an optimization problem that searches for the optimal set of weights that minimizes a loss function. As demonstrated in the mathematical analysis of [9], in a convex setting, the model weights $\mathbf{w}^r$ undergo a transient phase, during which their values change substantially, followed by a stationary phase, where updates are marginal. During the transient phase, the pseudo-gradients $\boldsymbol{\delta}^r$ of consecutive rounds are aligned, pointing in the same direction and driving the model toward convergence. As the model approaches the optimum, the pseudo-gradients oscillate around zero, leading to the stationary phase where changes to the weight values are minimal. However, real-world cross-device FL scenarios introduce nonidealities, like extreme label skew and partial device participation, which make the problem non-convex. As a result, the model might stuck in a sub-optimal region and

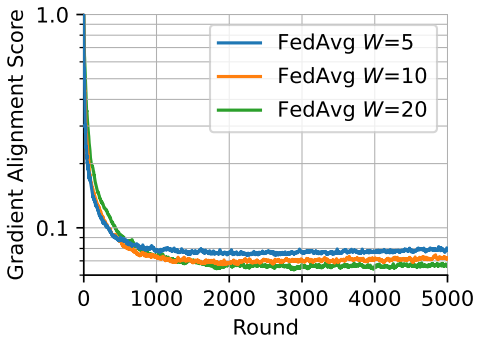

Fig. 1. Gradient alignment score during training on CIFAR-10 under extreme label skew ($L \leq 4$) using FedAvg with different cohort sizes (W).

Algorithm 2: FedGAP Control Logic.

1 $g_{\min} \leftarrow 1$
2 $t \leftarrow 0$
3 **for** r = *1,...,*R **do**
4 **if** $g^r < g_{min} - \epsilon$ **then**
5 $g_{\min} \leftarrow g^r$
6 $t \leftarrow 0$
7 **else**
8 $t \leftarrow t + 1$
9 **if** $t > \lambda$ **then**
10 Increment the cohort size by one.
11 $g_{\min} \leftarrow 1$
12 $t \leftarrow 0$

enter a stagnant phase, where the pseudo-gradients remain far from zero but fail to push the model toward the global optima.

The core idea of FedGAP lies in monitoring the evolution of the global model's pseudo-gradient over consecutive rounds to detect stagnant phases and trigger cohort resizing. To this end, we introduce the gradient alignment score as a training status probe suitable for FL. For an observation window of λ rounds, the gradient alignment $\mathbf{g^r}$ is defined as:

$$\mathbf{g^r} = \frac{||\sum_{i=r-\lambda+1}^{r} \boldsymbol{\delta}^i||_{\text{abs}}}{\sum_{i=r-\lambda+1}^{r} ||\boldsymbol{\delta}^i||_{\text{abs}}}, \tag{4}$$

where $\boldsymbol{\delta}^i$ is the pseudo-gradient vector at round i and $||\cdot||_{\text{abs}}$ represents the element-wise absolute value operation. To avoid the memory overhead of storing pseudo-gradients for λ rounds, we approximate the moving average using an exponential weighted moving average. Such approximation modifies the numerator ($\mathbf{m}^r$) and denominator ($\mathbf{p}^r$) in Eq. (4) as follows:

$$\mathbf{m}^r = \frac{2}{\lambda+1}\boldsymbol{\delta}^r + \left(1 - \frac{2}{\lambda+1}\right)\mathbf{m}^{r-1}, \tag{5}$$

$$\mathbf{p}^r = \frac{2}{\lambda+1}||\boldsymbol{\delta}^r||_{\text{abs}} + \left(1 - \frac{2}{\lambda+1}\right)\mathbf{p}^{r-1}, \tag{6}$$

where λ defines the virtual observation window of the values that contribute most to the final result. The resulting gradient alignment at round r is expressed as:

$$\mathbf{g^r} = \frac{||\mathbf{m}^r||_{\text{abs}}}{\mathbf{p}^r}. \tag{7}$$

Averaging the elements in $\mathbf{g^r}$ gives the scalar gradient alignment score g^r.

The gradient alignment score serves as an aggregated metric to assess the convergence status, with values $\in [0, 1]$. In general, values close to 1 indicate that consecutive pseudo-gradients are aligned, suggesting that the model is in a transient learning phase; values close to 0 imply that pseudo-gradients oscillate around zero, suggesting that the model is approaching convergence. Due to non-convexity, the gradient alignment score may plateau without reaching zero. The occurrence of this condition suggests the model is stuck and unable to progress further. FedGAP makes use of this information to implement a cohort resizing policy, which will be discussed later.

We first provide empirical evidence of our assumptions. The plot in Fig. 1 shows the evolution of the gradient alignment score during a 5000-round training on CIFAR-10 of a five-layer convolutional neural network, assuming extreme label skew ($L \leq 4$), and using the FedAvg algorithm with varying cohort sizes W, which is the experimental setup introduced for Table I. g^r starts from one and decreases sharply during the first few rounds. During this early stage, the cohort size has a marginal role, as indicated by the overlapping curves for W=5, W=10, and W=20. After a certain number of rounds, g^r tends to stabilize, eventually reaching a final value that changes with W. Larger cohort sizes result in lower scores, suggesting that more concurrent participants facilitate the learning progress. This analysis reinforces our idea of employing a dynamic participation policy. It suggests that smaller cohorts should be favored during the early phase to conserve energy without compromising training quality. Once the model gets stuck in a suboptimal point, the cohort size should increase to escape suboptimal regions and accelerate learning toward better solutions.

D. FedGAP Implementation

FedGAP implements a thresholding policy using the gradient alignment score, as detailed in Algorithm 2. The variable $g_{\min}$ keeps track of the lowest gradient alignment score observed during the training process, while the counter t tracks the number of consecutive rounds the gradient alignment shows negligible reduction. $g_{\min}$ and t are initialized to 1 and 0 respectively (lines 1–2). If g^r did not reduce by at least ϵ over the last λ rounds, the cohort size is incremented by one (lines 3–10). Once adjusted, the cohort size remains constant for at least λ rounds to allow the score to stabilize under the new settings; this is done by resetting $g_{\min}$ and t to their initial values (lines 11–12). The sensitivity of ϵ and λ is discussed in Sec. IV-C. It is also worth noting that the computation of the gradient alignment score and the execution of the FedGAP control logic are hosted by the server, relieving the training workers of any extra computation.

TABLE III
COMPARISONS ON CIFAR-10.

L	Method	$A_{\text{th}} > 79.0\%$	$A_{\text{th}} > 79.5\%$	$A_{\text{th}} > 80.0\%$
$\leq$**4**	**FedGAP** (Our)	**159.6**	**202.7**	**235.2**
	FedAvg W=5	227.2 (1.42×)	-	-
	FedAvg W=10	298.7 (1.87×)	317.3 (1.57×)	455.2 (1.94×)
	FedAvg W=20	398.2 (2.49×)	514.4 (2.54×)	599.0 (2.55×)
	FedRS W=5	-	-	-
	FedRS W=10	269.2 (1.69×)	-	-
	FedRS W=20	352.6 (2.21×)	-	-
	AdaFL	226.3 (1.42×)	235.7 (1.16×)	306.0 (1.30×)
$\leq$**3**	**FedGAP** (Our)	**240.5**	**281.2**	**351.8**
	FedAvg W=5	-	-	-
	FedAvg W=10	374.1 (1.56×)	479.1 (1.70×)	-
	FedAvg W=20	547.4 (2.28×)	615.4 (2.19×)	965.6 (2.74×)
	FedRS W=5	-	-	-
	FedRS W=10	-	-	-
	FedRS W=20	-	-	-
	AdaFL	328.8 (1.37×)	363.4 (1.29×)	417.9 (1.19×)

Best results in bold. A dash (-) indicates the target accuracy is not met.

TABLE IV
COMPARISONS ON CIFAR-100.

L	Method	$A_{\text{th}} > 42.0\%$	$A_{\text{th}} > 44.0\%$	$A_{\text{th}} > 46.0\%$
$\leq$**5**	**FedGAP** (Our)	**154.9**	**211.2**	**360.4**
	FedAvg W=5	-	-	-
	FedAvg W=10	261.7 (1.69×)	431.9 (2.04×)	-
	FedAvg W=20	340.2 (2.20×)	504.8 (2.39×)	875.0 (2.43×)
	FedRS W=5	178.8 (1.15×)	-	-
	FedRS W=10	204.1 (1.32×)	-	-
	FedRS W=20	403.4 (2.60×)	-	-
	AdaFL	217.4 (1.40×)	286.9 (1.36×)	506.7 (1.41×)
$\leq$**3**	**FedGAP** (Our)	**213.5**	**293.8**	**494.8**
	FedAvg W=5	-	-	-
	FedAvg W=10	349.4 (1.64×)	-	-
	FedAvg W=20	417.4 (1.96×)	587.6 (2.00×)	928.4 (1.88×)
	FedRS W=5	-	-	-
	FedRS W=10	-	-	-
	FedRS W=20	-	-	-
	AdaFL	261.5 (1.23×)	385.8 (1.31×)	764.3 (1.54×)

Best results in bold. A dash (-) indicates the target accuracy is not met.

IV. RESULTS

A. Experimental Setup

Datasets. We conducted experiments on two image classification tasks trained on the CIFAR-10 and CIFAR-100 datasets and using the standard splits for training and testing. Each training dataset was partitioned into non-overlapping subsets distributed among 100 clients. Following the partitioning procedure outlined in [28], each client was assigned at most L labels. That was achieved by creating $100 \times L$ data shards, with each shard containing samples of the same label, which were then randomly assigned to the clients.

Model and Training Settings. As a benchmark, we adopted the five-layer convolutional neural network from [28]. The FL process was iterated over 5000 training rounds. During each round, the on-device training consisted of K=20 steps of Stochastic Gradient Descent (SGD) optimization with the following hyper-parameters: momentum 0.9, weight decay $1\text{e}{-3}$, learning rate $1\text{e}{-2}$, batch size 50.

Baselines and Implementation Details. We compared our approach against three representative state-of-the-art baselines: FedAvg, [28], FedRS [13], AdaFL [26].

- **FedAvg** is the standard FL algorithm with a fixed and static cohort size consisting of W workers and the cross-entropy loss for on-device training. We tested FedAvg with varying cohort sizes ($W \in \{5, 10, 20\}$) to explore the energy-accuracy trade-off.
- **FedRS** adjusts the local training loss function to account for missing labels. Like FedAvg, we conducted a parametric analysis with $W \in \{5, 10, 20\}$, and, as suggested by the authors, we fine-tuned the hyper-parameter $\alpha \in \{0.5, 0.9\}$ picking the best-performing setting. It is worth mentioning that we considered FedRS in our comparisons to investigate the actual weakness of on-device training optimization (currently the prevailing approach), both in

terms of the quality of training and end energy efficiency.

- **AdaFL** implements a dynamic policy that gradually increases the cohort size, directly competing with our FedGAP. Specifically, AdaFL increases the cohort size W by one unit every ΔR rounds and up to a predefined maximum. We set an initial size of 5, a maximum size of 30, and tuned $\Delta R \in \{100, 200\}$.
- **FedGAP** is our proposal. Unless specified otherwise, we used λ=100 and ϵ=5e$-$4, with 5 initial workers and up to 30 workers.

Metrics. To assess performance, we tracked the global model's classification accuracy on the test set, computing a 30-round moving average to smooth inherent oscillations during training—a well-established evaluation protocol for reliable assessment [30]. Furthermore, we measured the average client's energy cost consumed to achieve predefined accuracy targets using Eq. (3).

B. Results and Discussion

Tables III and IV show a comparison of the energy cost of FedGAP against the baselines under consideration on the CIFAR-10 and CIFAR-100 datasets, respectively. These tables report the energy cost required to reach predefined accuracy thresholds A_{th}. Numbers in parentheses indicate the energy overhead of each baseline compared to FedGAP. We evaluated the performance under different degrees of extreme label skew: $L\leq$4 and $L\leq$3 for CIFAR-10; $L\leq$5 and $L\leq$3 for CIFAR-100. For FedRS and AdaFL, we reported the results achieved with the best hyper-parameter configuration.

At a glance, FedGAP always reaches the highest accuracy threshold and outperforms the other methods, which show energy overheads ranging from 1.16× to 2.74× on CIFAR-10, and from 1.15× to 2.60× on CIFAR-100.

The performance and efficiency of FedAvg are affected by the cohort size W. Using a fixed cohort size forces a static trade-off between accuracy and energy consumption. A larger cohort size is needed for high accuracy, but it results in increased energy consumption. Only configurations with W=20 achieve the highest accuracy level in all benchmarks, yet at the cost of substantial energy overhead. For instance, on CIFAR-10 with $L\leq$4, achieving $A_{\text{th}} > 79.0\%$ almost doubles the energy cost from 227.2 with W=5 to 398.2 with W=20. Smaller cohorts reduce energy but compromise the achievable accuracy. In particular, FedAvg with W=5 achieves only one target accuracy for one benchmark (CIFAR-10 with $L\leq$ 4) and performs poorly under higher degrees of label skews (CIFAR-10 with $L\leq$3) or when trained on the more complex dataset (CIFAR-100). FedAvg with W=10 gets below 80% for CIAFR-10 with $L\leq$3 and 46% for CIFAR-100 with $L\leq$5, and never exceeds 44% accuracy for CIFAR-100 with $L\leq$3.

FedRS shows higher energy efficiency for lower accuracy targets but struggles to achieve high accuracy. On CIFAR-100 with $L\leq$5, FedRS with W=5 reaches 42.0% accuracy with an energy cost of 178.8, much lower than FedAvg with W=10 (261.7) and W=20 (340.2). However, it fails to achieve accuracy beyond higher thresholds, even with larger cohorts.

TABLE V
SENSITIVITY ANALYSIS ON FEDGAP HYPER-PARAMETERS. RESULTS ON CIFAR-10 WITH $L\leq$3.

λ	ϵ	$A_{\text{th}} > 79.0$	$A_{\text{th}} > 79.5$	$A_{\text{th}} > 80.0$
100	**5e−4**	240.5	281.2	351.8
	1e−4	229.0	251.0	383.1
200	**5e−4**	197.5	208.3	242.4
	1e−4	215.8	252.1	273.6

TABLE VI
SENSITIVITY ANALYSIS ON FEDGAP HYPER-PARAMETERS. RESULTS ON CIFAR-100 WITH $L\leq$3.

λ	ϵ	$A_{\text{th}} > 42.0$	$A_{\text{th}} > 44.0$	$A_{\text{th}} > 46.0$
100	**5e−4**	213.5	293.8	494.8
	1e−4	197.8	297.1	494.9
200	**5e−4**	183.5	240.2	365.6
	1e−4	191.5	233.5	319.6

Indeed, while FedAvg with larger W can reach higher accuracy levels, e.g., A_{th}>44.0% with W=10 and A_{th}>46.0% with W=20, FedRS playing with larger W brings higher costs without substantial improvements in the maximum attainable accuracy. Furthermore, FedRS worsens under higher degrees of label skews, i.e., $L\leq$3, failing to meet any of the accuracy targets, both for CIFAR-10 and CIFAR-100. As per our understanding, the modified loss function implemented in FedRS accelerates convergence for the initial training phases but slows it down in the later stages, limiting the overall training effectiveness.

The dynamic protocol of AdaFL shows better accuracy-energy trade-offs compared to FedAvg and FedRS. It achieves all the accuracy targets while consuming less energy than the static baselines. We reported only one exception in CIFAR-100 with $L\leq$5 and accuracy target A_{th}>42.0%. In this specific case, AdaFL incurs an energy cost of 217.4, while FedRS achieves the same accuracy with a lower cost of 178.8.

Although AdaFL shows promising results, FedGAP, thanks to its gradient-aware policy, consistently achieves higher efficiency without sacrificing accuracy. On CIFAR-10, FedAvg, FedRS, and AdaFL consume up to 2.74×, 2.21×, and 1.42× more energy than FedGAP, respectively. On CIFAR-100, their energy overheads are up to 2.43×, 2.60×, and 1.54×, respectively. Remarkably, on CIFAR-100 with $L\leq$5, FedGAP attains 2% higher accuracy (44.0% vs. 42.0%) than AdaFL with a lower energy cost (211.2 vs. 217.4).

These findings showcase the advantages of the FedGAP's gradient-aware participation scaling policy. Unlike AdaFL, which increases the cohort size at fixed intervals irrespective of training dynamics, FedGAP adapts to the training dynamics more precisely, increasing the cohort size when actually needed. Overall, the analysis of results proves the superior flexibility and adaptability of FedGAP in optimizing the trade-off between energy efficiency and training performance across diverse datasets and label distributions.

C. FedGAP Sensitivity Analysis

In the previous comparisons, we fixed the hyper-parameter values of FedGAP to λ=100 and ϵ=5e$-$4. This subsection presents a sensitivity analysis for these hyper-parameters. Specifically, we conducted a grid-search exploration with $\lambda \in \{100, 200\}$ and $\epsilon \in \{5\text{e}{-4}, 1\text{e}{-4}\}$. The analysis focused on the two benchmarks: CIFAR-10 and CIFAR-100, both with $L \leq 3$. The results in Tables V and VI demonstrate the stability of FedGAP across different configurations, with all the accuracy targets consistently achieved.

Fine-tuning the hyper-parameters can yield additional energy savings. We observed that ϵ has a lower impact, while λ leads to substantial gains. For instance, on CIFAR-10, when targeting A_{th}>80.0, increasing λ from 100 to 200 reduces energy consumption by 1.45$\times$ (351.8 to 242.4) using the same value of ϵ=5e$-$4; similarly, on CIFAR-100 for A_{th}>46.0, the same adjustment improves efficiency by 1.35$\times$ (from 494.8 to 365.6). In summary, the results indicate that FedGAP benefits from larger λ values when energy efficiency is a priority.

V. Conclusion

We introduced FedGAP, a new energy minimization strategy for the federated training of classifiers across end devices. FedGAP addresses a key limitation of standard federated algorithms: the substantial increase in energy consumption of the participating clients when the local datasets contain samples from a subset of the target classes. By dynamically adjusting the number of training workers based on the global model progress, FedGAP offers an effective solution to mitigate the overhead, achieving up to 2.74$\times$ lower energy cost. Such results are driven by the introduction of a novel performance metric, the gradient alignment score, designed specifically to monitor training dynamics in federated contexts.

References

[1] H. Woisetschläger, A. Erben, B. Marino, S. Wang, N. D. Lane, R. Mayer, and H.-A. Jacobsen, "Federated learning priorities under the european union artificial intelligence act," *arXiv:2402.05968*, 2024.

[2] P. Qi, D. Chiaro, and F. Piccialli, "Small models, big impact: A review on the power of lightweight federated learning," *Future Gener. Comput. Syst.*, 2025.

[3] T. H. Hsu, H. Qi, and M. Brown, "Federated visual classification with real-world data distribution," in *ECCV*, 2020.

[4] P. Qi, D. Chiaro, and F. Piccialli, "FL-FD: federated learning-based fall detection with multimodal data fusion," *Inf. Fusion*, 2023.

[5] M. Bornstein, N. Nazir, J. Drgona, S. Kundu, and V. Adetola, "Finding MIDDLE ground: Scalable and secure distributed learning," in *CIKM*, 2024.

[6] A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, "A survey on federated learning for resource-constrained iot devices," *IEEE Internet Things J.*, 2022.

[7] L. Grativol, M. Léonardon, G. Muller, V. Fresse, and M. Arzel, "Federated learning compression designed for lightweight communications," in *ICECS*, 2023.

[8] J. Yuan, S. Wang, H. Li, D. Xu, Y. Li, M. Xu, and X. Liu, "Towards energy-efficient federated learning via int8-based training on mobile dsps," in *WWW*, 2024.

[9] C. Chen, H. Xu, W. Wang, B. Li, B. Li, L. Chen, and G. Zhang, "Synchronize only the immature parameters: Communication-efficient federated learning by freezing parameters adaptively," *IEEE Trans. Parallel Distributed Syst.*, 2024.

[10] E. Malan, V. Peluso, A. Calimera, and E. Macii, "Communication-efficient federated learning with gradual layer freezing," *IEEE Embed. Syst. Lett.*, 2023.

[11] E. Malan, V. Peluso, A. Calimera, E. Macii, and P. Montuschi, "Automatic layer freezing for communication efficiency in cross-device federated learning," *IEEE Internet Things J.*, 2024.

[12] V. Peluso, E. Malan, A. Calimera, and E. Macii, "Private tensor freezing for an efficient federated learning with homomorphic encryption," in *ICCD*, 2024.

[13] X. Li and D. Zhan, "Fedrs: Federated learning with restricted softmax for label distribution non-iid data," in *KDD*, 2021.

[14] J. Zhang, Z. Li, B. Li, J. Xu, S. Wu, S. Ding, and C. Wu, "Federated learning with label distribution skew via logits calibration," in *ICML*, 2022.

[15] Z. Fan, R. Zhang, J. Yao, B. Han, Y. Zhang, and Y. Wang, "Federated learning with bilateral curation for partially class-disjoint data," in *NeurIPS*, 2023.

[16] Y. Shi, J. Liang, W. Zhang, V. Y. F. Tan, and S. Bai, "Towards understanding and mitigating dimensional collapse in heterogeneous federated learning," in *ICLR*, 2023.

[17] G. Lee, M. Jeong, Y. Shin, S. Bae, and S. Yun, "Preservation of the global knowledge by not-true distillation in federated learning," in *NeurIPS*, 2022.

[18] J. Lu, S. Li, K. Bao, P. Wang, Z. Qian, and S. Ge, "Federated learning with label-masking distillation," in *MM*. ACM, 2023.

[19] D. Yu, X. Du, L. Jiang, S. Bai, W. Tong, and S. Deng, "Fedlec: Effective federated learning algorithm with spiking neural networks under label skews," *arXiv:2412.17305*, 2024.

[20] K. Guo, Y. Ding, J. Liang, Z. Wang, R. He, and T. Tan, "Exploring vacant classes in label-skewed federated learning," in *AAAI*, 2025.

[21] S. A. Tijani, X. Ma, R. Zhang, F. Jiang, and R. Doss, "Federated learning with extreme label skew: A data extension approach," in *IJCNN*, 2021.

[22] K. Sang, T. Rabbani, and F. Huang, "Balancing label imbalance in federated environments using only mixup and artificially-labeled noise," *arXiv:2409.13235*, 2024.

[23] Y. Diao, Q. Li, and B. He, "Exploiting label skews in federated learning with model concatenation," in *AAAI*, 2024.

[24] L. Fu, H. Zhang, G. Gao, M. Zhang, and X. Liu, "Client selection in federated learning: Principles, challenges, and opportunities," *IEEE Internet Things J.*, 2023.

[25] M. Savoia, E. Prezioso, V. Mele, and F. Piccialli, "Eco-fl: Enhancing federated learning sustainability in edge computing through energy-efficient client selection," *Comput. Commun.*, 2024.

[26] Q. Li, X. Li, L. Zhou, and X. Yan, "Adafl: Adaptive client selection and dynamic contribution evaluation for efficient federated learning," in *ICASSP*, 2024.

[27] E. Malan, V. Peluso, A. Calimera, and E. Macii, "Draft & refine: Efficient resource management in federated learning under pathological labels skew," in *ICECS*, 2024.

[28] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*, 2017.

[29] X. Qiu, T. Parcollet, J. Fernández-Marqués, P. P. B. de Gusmao, Y. Gao, D. J. Beutel, T. Topal, A. Mathur, and N. D. Lane, "A first look into the carbon footprint of federated learning," *J. Mach. Learn. Res.*, vol. 24, pp. 129:1–129:23, 2023.

[30] S. J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," in *ICLR*, 2021.